Insight Security

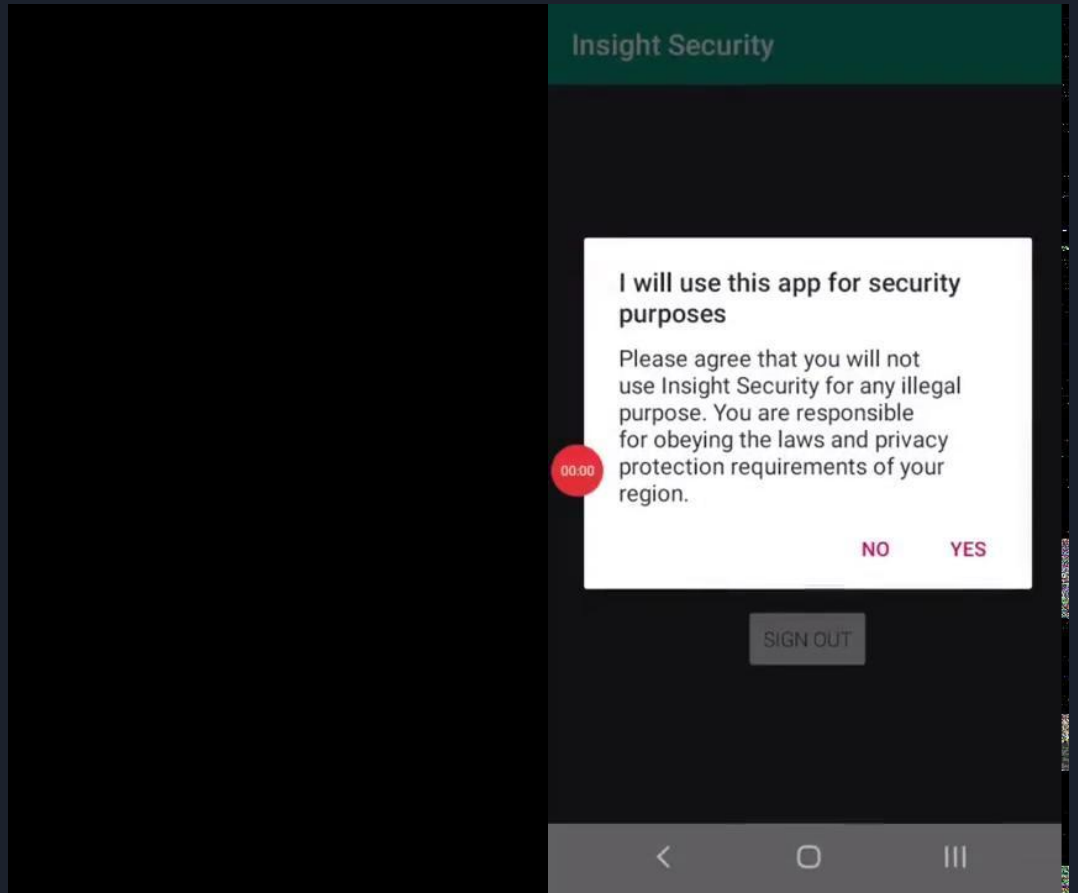
CS 499
May 18, 2020

Richard Basdeo
Eric Benjamin
Merna Eldesouki

What is Insight Security?

- Solves lower-income individuals' problem of not being able to afford a brand home security system by letting them turn their old phones into security systems
- Android application that connects user's phones to the hardware of old junk phones.
- We know if our product works when it detects excess sound and takes action (image capture, notifications between phones and more)

Camera + Viewer Perspective Demo



New Features Part 1

- *Noise Event Algorithm:* The algorithm now compares the current noise level to previous noise level data. If it is a certain percentage higher AND is over a certain base amount, a noise event is triggered
- *Notification Protection:* Prevents users from receiving more than one notification every five minutes, using a scheduled cloud function
- *Integrity Agreement:* The app shows a simple license agreement upon logging in, to ensure ethical use. If the user declines to agree, they are logged out
- *Battery Saver:* Once the app goes into camera mode, the screen turns off to conserve battery and it starts recording noise levels

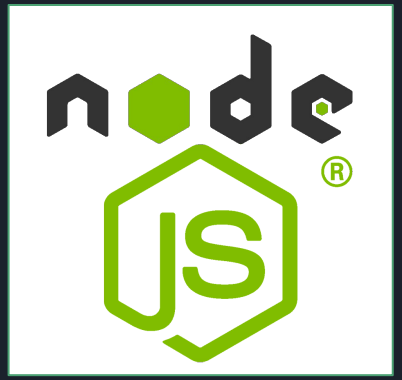
New Feature: Image Capture

- Once a noise spike is detected, an image is captured
- Additionally, user can choose to start a timer so that the app takes a picture once every 30 minutes automatically
- These images are viewable in-app from any phone logged into the same email address

New Feature: File Path Creation

- Used a timestamp to create unique paths for pictures
- Pictures are named “Hour_Min_Second_.jpeg”
- This allows me to:
  - Automatically upload photos
  - Delete photos

Tools



Work Distribution

Eric

Notifications

Database management

Merna

UI

Noise detection

Richard

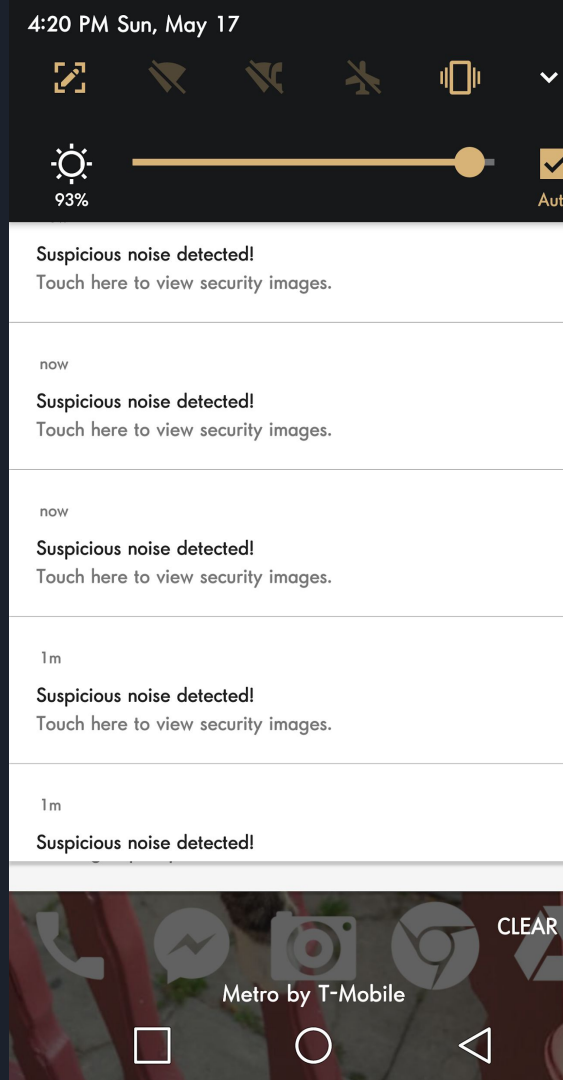
Image capturing

Google sign-in

Database management

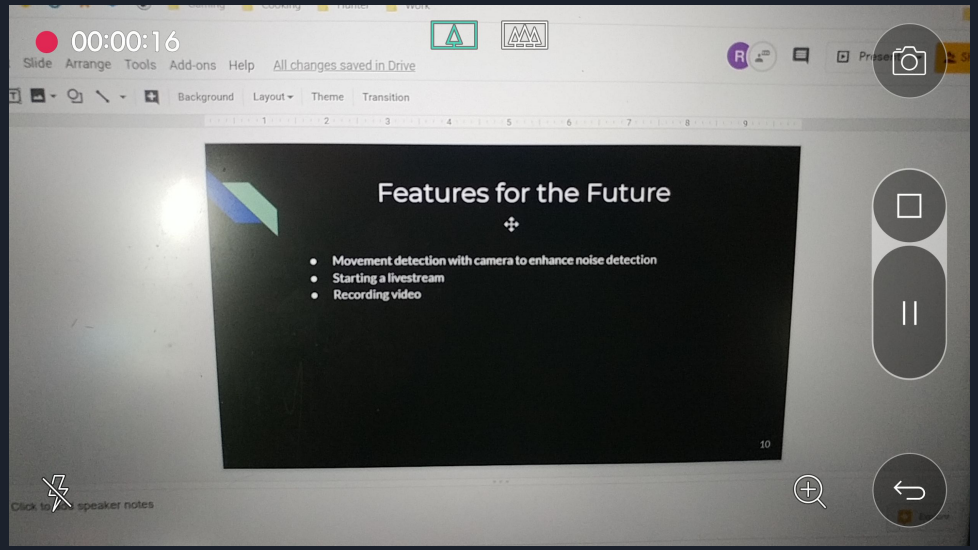
Challenges

- Sending out too many notifications during a period of constant noise
- Using the phone's camera without a user physically present
- Perfecting the noise event algorithm to not be too lax or too careful



Features for the Future

- Movement detection with camera to enhance noise detection
- Recording video when detecting a noise event
- Starting a livestream



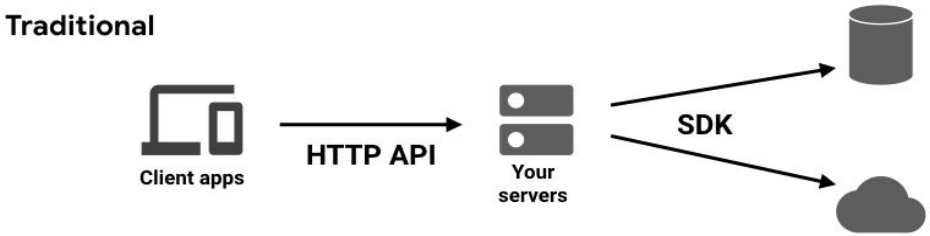
Questions?

Firebase Cloud Functions

Presented by Eric Benjamin and Merna Eldesouki

What is Firebase?

- Firebase acts as your server, API and Datastore
- Firebase is a BaaS



What are Firebase Cloud Functions?

Cloud Functions automatically run your code in response to events in your Firebase database. Firebase handles:

- Storing and running your code in a managed environment
- Scaling to support more or less usage
- Securing your code away from the client

Using Firebase Cloud Functions to Notify Users

- Our project utilized Firebase Cloud Functions to deliver push notifications to users whenever a noise event was detected on one of their devices
- We developed a Cloud Function to trigger when a user's noiseEvent value was updated to true, and then utilize Firebase Cloud Messaging to send the notifications


```
-M7JYin4slUwK_sFzorq
    noiseEvent: false
    notificationToken: "evycjOcbT5WgGQIlKqXd01:APA91bFDc_kKP8OpdLzH3Mtt
    notified: true
    userEmail: "eric.benjamin2010@gmail.cc
    userName: "Eric Benjamir
```

Limitations of Firebase Cloud Functions

- For mobile apps the main alternative to Google Cloud Functions is App Engine
- App Engine allows you to control individual instances and how they scale
- Firebase Cloud Functions are limited to nine minutes running time, while App Engine does not have a running time limit
- Firebase Cloud Functions only supports NodeJS, while App Engine allows for a variety of common programming languages

Costs Comparison (United States)

	Firebase Cloud Functions	Google App Engine
Free	2,000,000 invocations, 400,000 GB-sec, 200,000 CPU-sec, and 5 GB of Internet egress traffic each **month**	28 hours per day of "F" instances, 9 hours per day of "B" instances and 1 GB of egress each **day**
Paid (after free quotas are used)	$0.40/million invocations, $0.0025/thousand GB-seconds, $0.01/thousand CPU-seconds	$0.05 - $0.48 per hour per instance, depending on location and instance class
Outbound networking	$0.12/GB	$0.12/GB

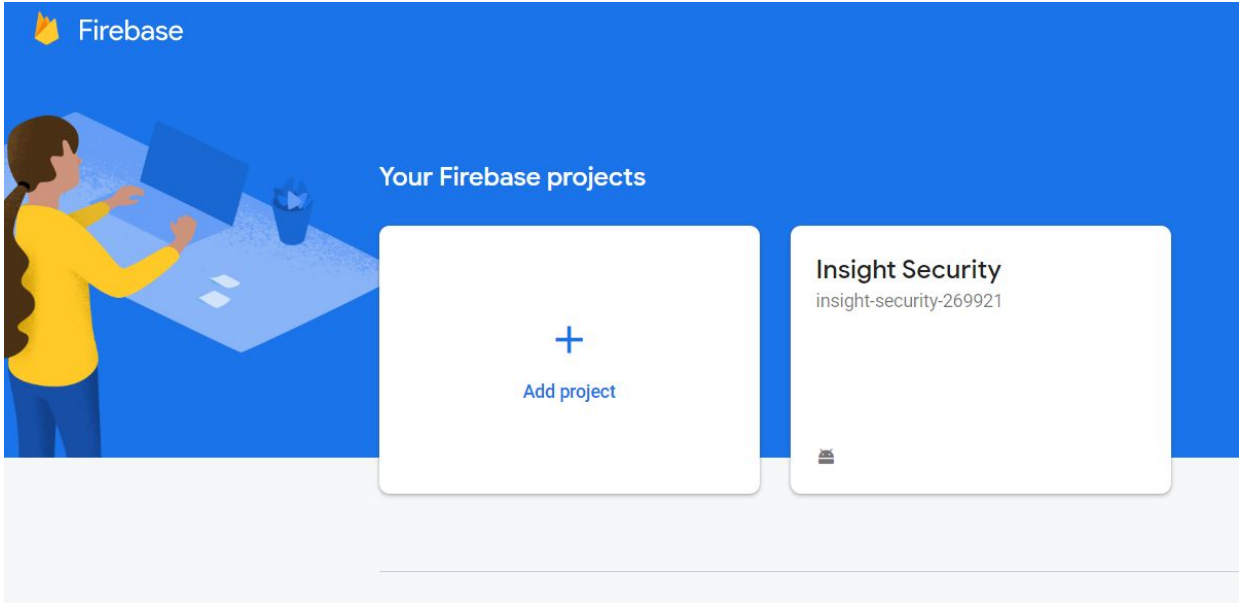
Reasons to use Firebase Cloud Functions

- Ease of setup and use: Deploy your function with one line
- Great documentation and a variety of code samples
- Ridiculously flexible, easily integrating with other Firebase Services, Google Cloud Platform services and more

```
patio@Patricios-MacBook-Pro ~/Desktop/react-test/build (master)
$ firebase deploy

=== Deploying to 'app-deployment-72fc1'...

i  deploying hosting
i  hosting[app-deployment-72fc1]: beginning deploy...
i  hosting[app-deployment-72fc1]: found 1 files in dist
✔  hosting[app-deployment-72fc1]: file upload complete
i  hosting[app-deployment-72fc1]: finalizing version...
✔  hosting[app-deployment-72fc1]: version finalized
i  hosting[app-deployment-72fc1]: releasing new version...
✔  hosting[app-deployment-72fc1]: release complete

✔  Deploy complete!

Project Console: https://console.firebase.google.com/project/app-deployment-72fc1/overview
Hosting URL: https://app-deployment-72fc1.firebaseapp.com
```

Getting Started with Firebase Cloud Functions

Step one: Create a Firebase project, or use an existing GCP project!



Getting Started with Firebase Cloud Functions

Step two: Installations

- Install NodeJS
- Install the Firebase Command Line Interface via npm, using:

```
npm install -g firebase-tools
```

Getting Started with Firebase Cloud Functions

Step three: Initialize your project

- Run ‘firebase login’ in your command line
- Run ‘firebase init functions’ in the directory where your Cloud Function code lives
- You can now write and deploy your Cloud Function!

Getting Started with Firebase Cloud Functions

Final step: Deploy your cloud function!

```
pato@Patricios-MacBook-Pro */Desktop/react-test/build (master)
$ firebase deploy

=== Deploying to 'app-deployment-72fc1'...

i  deploying hosting
i  hosting[app-deployment-72fc1]: beginning deploy...
i  hosting[app-deployment-72fc1]: found 1 files in dist
✔  hosting[app-deployment-72fc1]: file upload complete
i  hosting[app-deployment-72fc1]: finalizing version...
✔  hosting[app-deployment-72fc1]: version finalized
i  hosting[app-deployment-72fc1]: releasing new version...
✔  hosting[app-deployment-72fc1]: release complete

✔  Deploy complete!

Project Console: https://console.firebase.google.com/project/app-deployment-72fc1/overview
Hosting URL: https://app-deployment-72fc1.firebaseapp.com
```

Thank you!

We hope you'll consider Firebase and Firebase Cloud Functions for your own projects. They were a pleasure to work with!

Image Capture

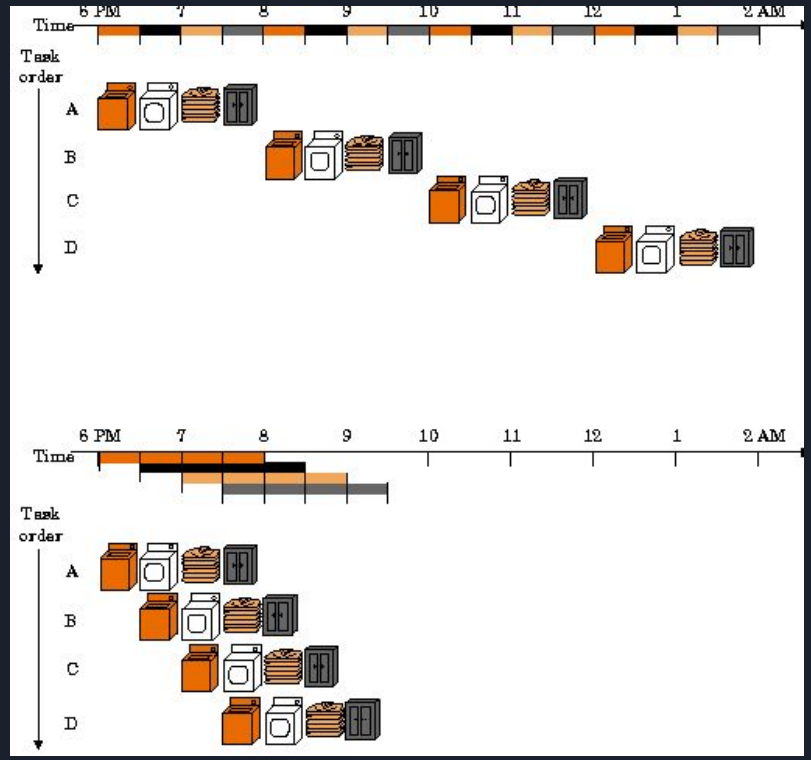
Presented by Richard Basdeo

Image Capture

Camera API

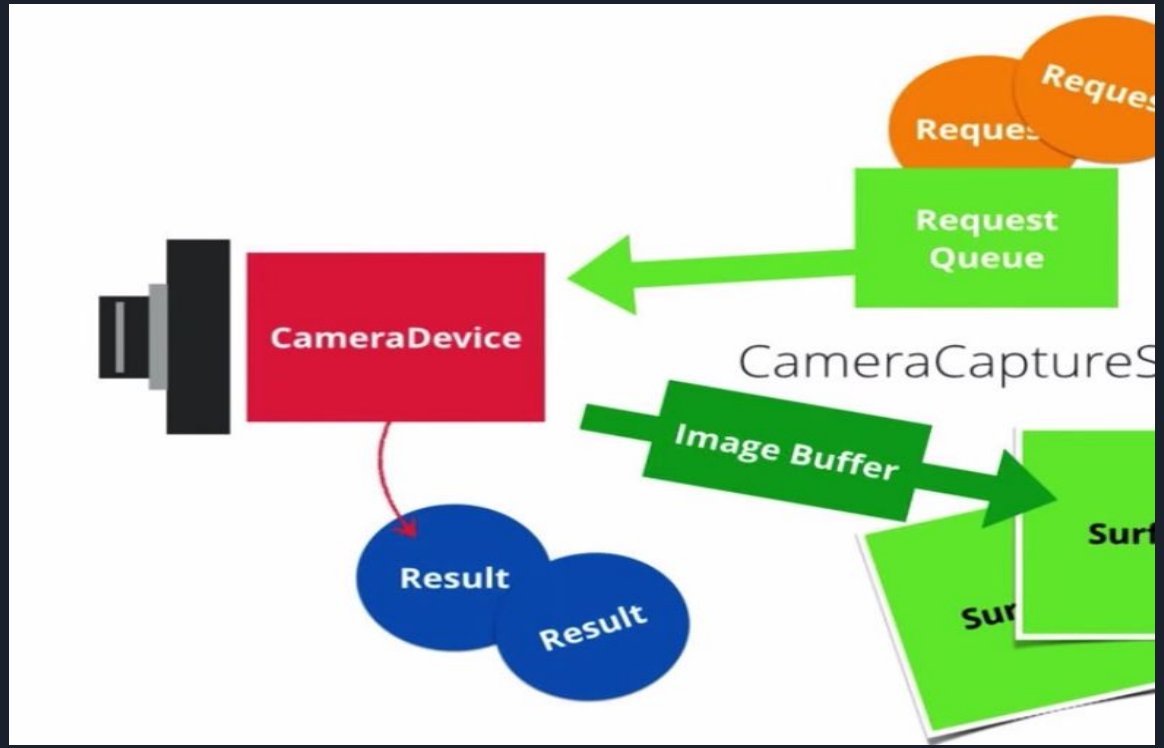
Camera2 API

Pipeline



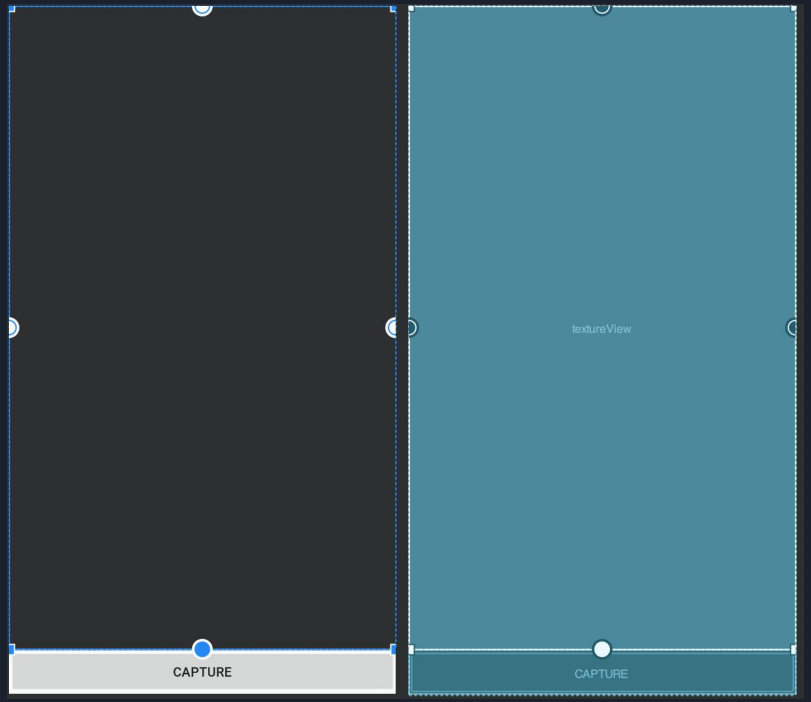
- Storing and executing instructions in a timely manner.
- Keep all parts of the camera busy to fully use its resources.

Pipelining In Camera2 API



How to take a picture without a user?

Create a Texture View:



Open The Camera

```
private void openCamera() {
    CameraManager manager = (CameraManager) getSystemService(Context.CAMERA_SERVICE);
    Log.e(TAG, msg: "is camera open");
    try {
        cameraId = manager.getCameraIdList()[0];
```

Create the Preview And Take Picture

```java
    protected void createCameraPreview() {
        try {
            SurfaceTexture texture = textureView.getSurfaceTexture();
            assert texture != null;
            texture.setDefaultBufferSize(imageDimension.getWidth(), imageDimension.getHeight());
            Surface surface = new Surface(texture);
            captureRequestBuilder = cameraDevice.createCaptureRequest(CameraDevice.TEMPLATE_PREVIEW);
            captureRequestBuilder.addTarget(surface);
            cameraDevice.createCaptureSession(Arrays.asList(surface), new CameraCaptureSession.StateCallback(){
                @Override
                public void onConfigured(@NonNull CameraCaptureSession cameraCaptureSession) {
                    //The camera is already closed
                    if (null == cameraDevice) {
                        return;
                    }
                    // When the session is ready, we start displaying the preview.
                    cameraCaptureSessions = cameraCaptureSession;
                    updatePreview();
                    takePicture();
```

Thank you!