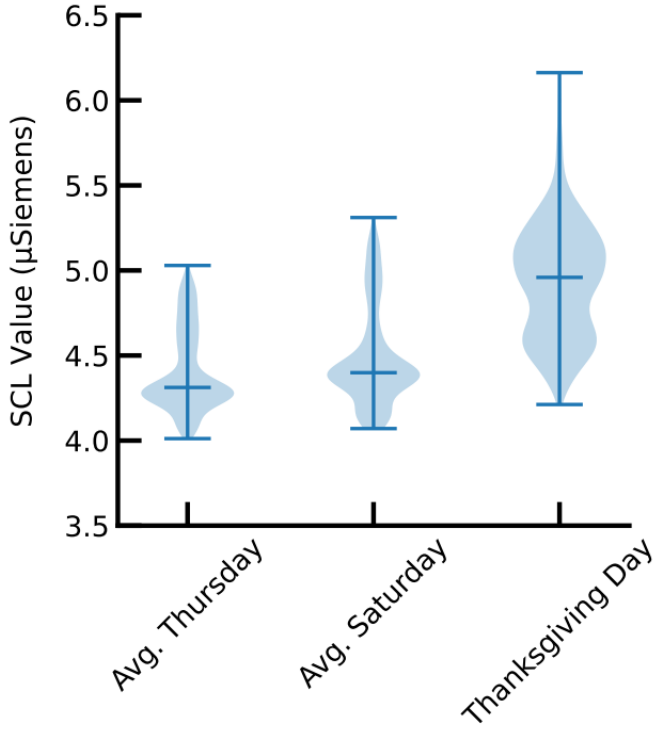
Other sensors &
Some interesting libraries

Electro Dermal Activity (EDA)

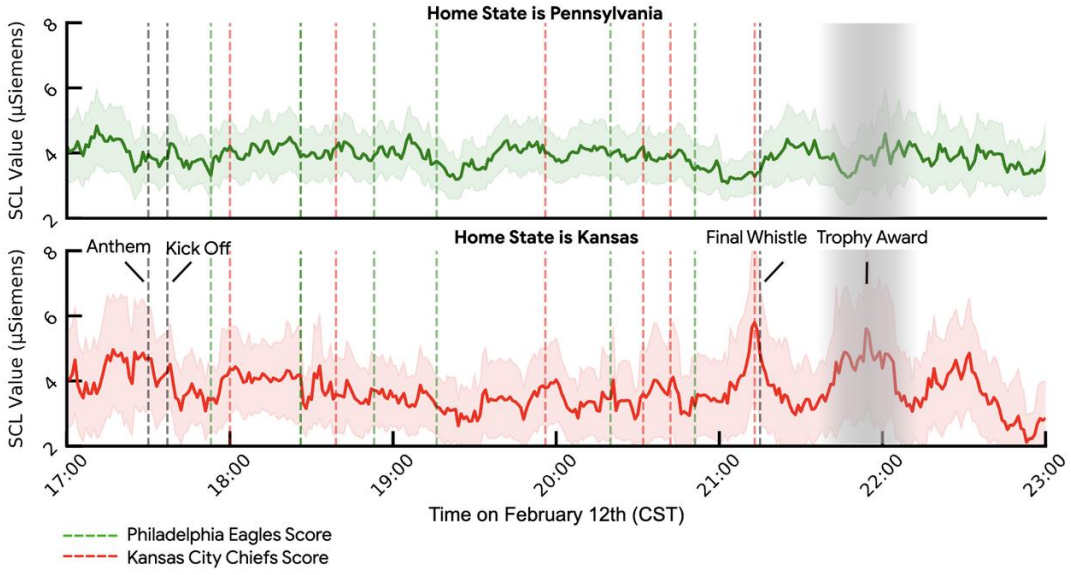
- Many smartwatches are equipped with an EDA sensor
- EDA, aka Galvanic Skin Response or Skin Conductance
  - enhanced electrical conductance that occurs during physiological/emotional arousal due to the activation of sweat glands
  - the Autonomic Nervous System (ANS) is a component of the peripheral nervous system that regulates involuntary physiologic processes including heart rate, blood pressure, respiration, digestion
  - the Sympathetic Nervous System is the part of ANS responsible for "fight-or-flight" response (HR increase, sweat stimulation, pupil dilation)
  - measured in microSiemens

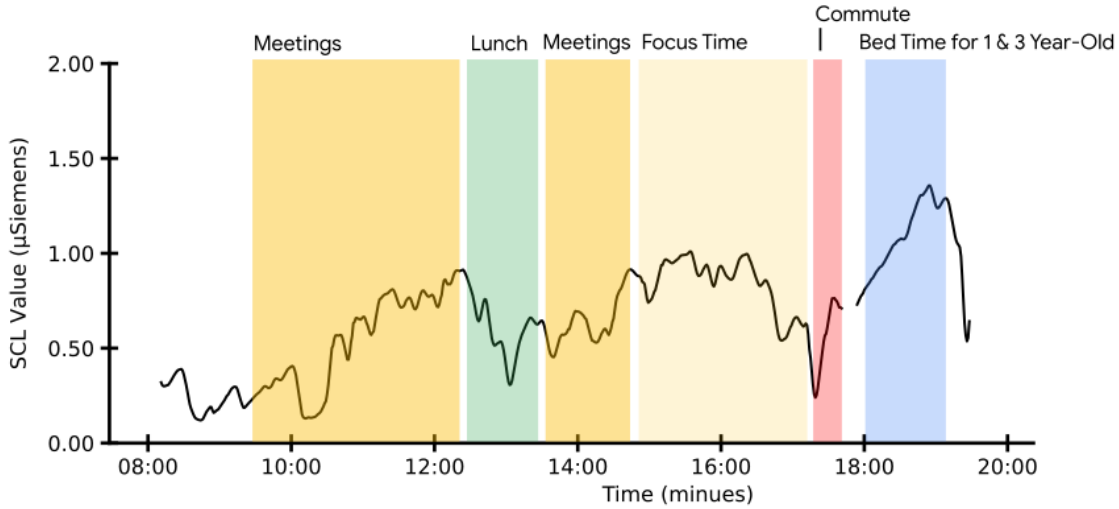
EDA

- Skin Conductance Level changes throughout the day

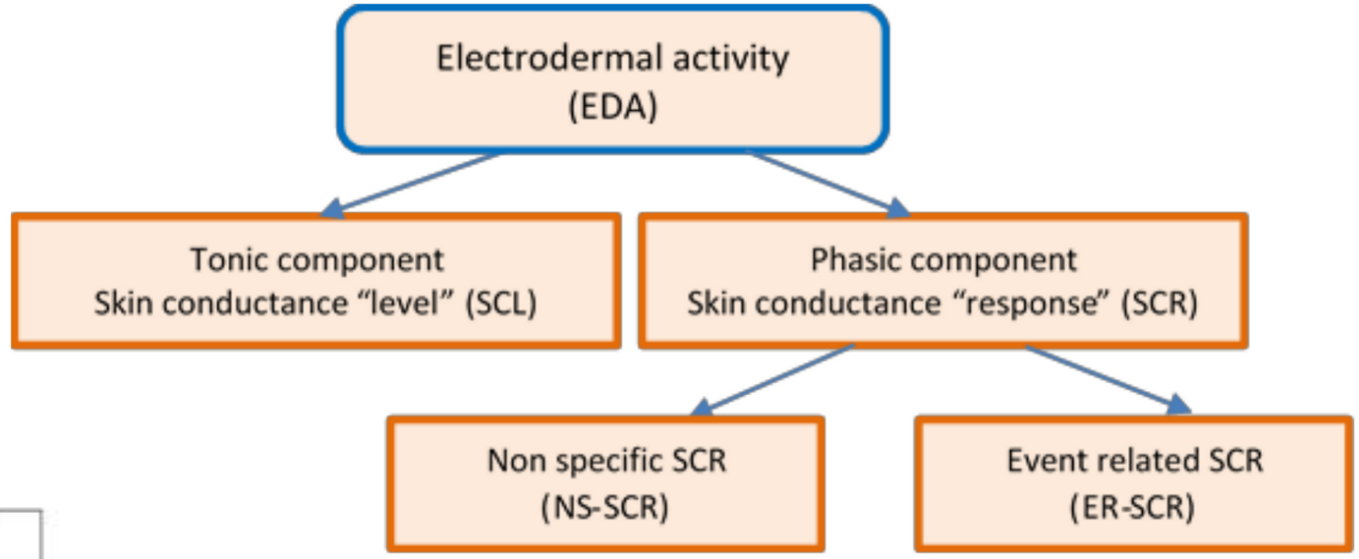


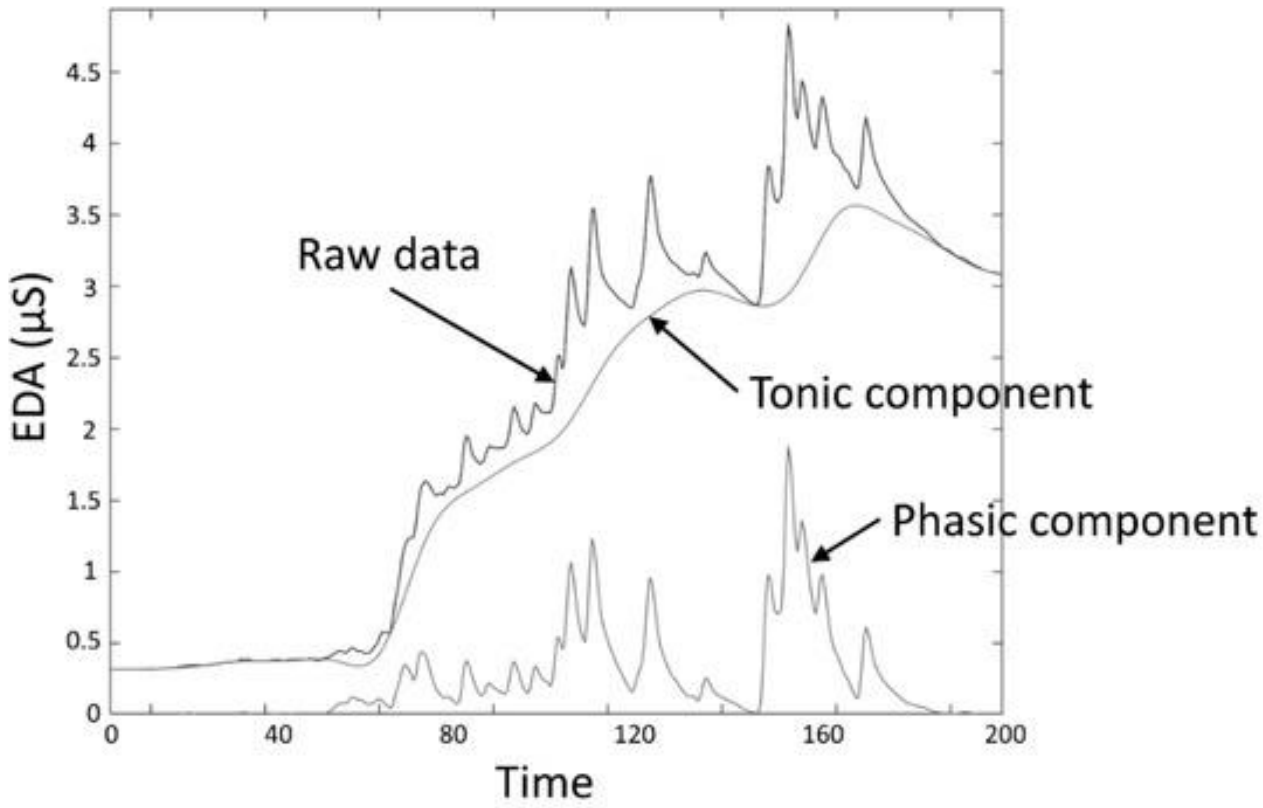
@Google research





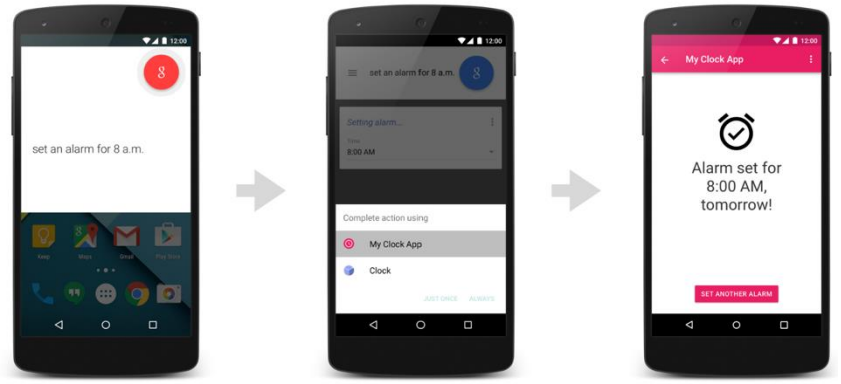
EDA


Electrodermal activity
(EDA)
Tonic component
Skin conductance "level" (SCL)
Phasic component
Skin conductance "response" (SCR)
Non specific SCR
(NS-SCR)
Event related SCR
(ER-SCR)


EDA (µS)
Time
Raw data
Tonic component
Phasic component
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
0
40
80
120
160
200

Voice based interaction

- Two forms
  - **Speech-to-text and Text-to-speech**: Convert user's speech to text and vice-versa
  - **Voice Actions**: Voice commands to smartphone (e.g. «set an alarm for 8 am» or «show heart rate»)



- Voice actions are delivered as intents
- There is a list of pre-defined actions
- The activity may also require confirmation from the user via voice
- Guides:
  - http://developer.android.com/reference/android/speech/SpeechRecognizer.html
  - https://developers.google.com/voice-actions/

```
<activity ...>
    <intent-filter>
        <action
android:name="android.intent.action.SET_ALARM"/>
        <category
android:name="android.intent.category.DEFAULT"/>
    </intent-filter>
</activity>
```

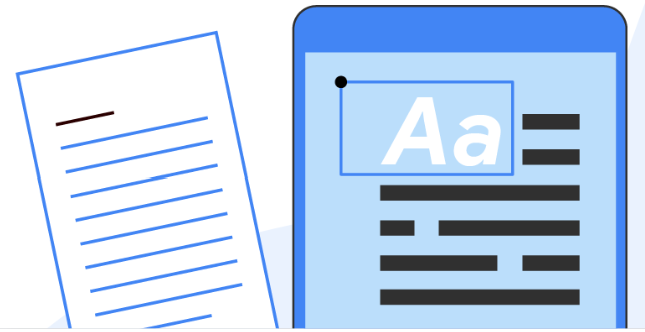
Google ML kit

- Machine learning for mobile devices
- Lots of capabilities
  - Vision
    - Barcode scanning
    - Face detection
    - Pose detection
    - Image labeling
    - Object detection and tracking
    - Text recognition
  - Language
    - Language identification
    - Translation
    - Smart reply
  - Custom
    - Programmer defined models

Barcode scanning and text recognition

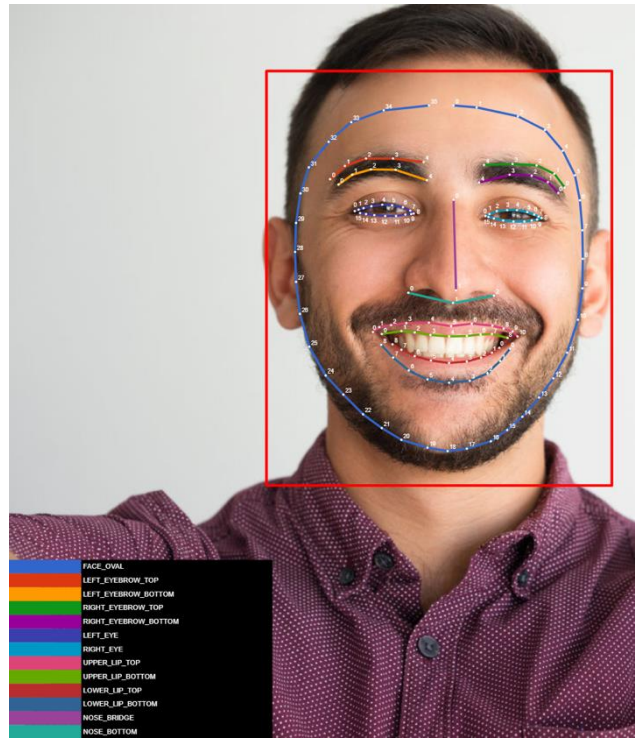
- Barcodes are a convenient way to pass information from the physical world to your app
  - Can encode structured data such as contact information or Wi-Fi network credentials

- Text of documents, business cards, driving license, etc can be automatically extracted
  - It may also return the position of text blocks in image

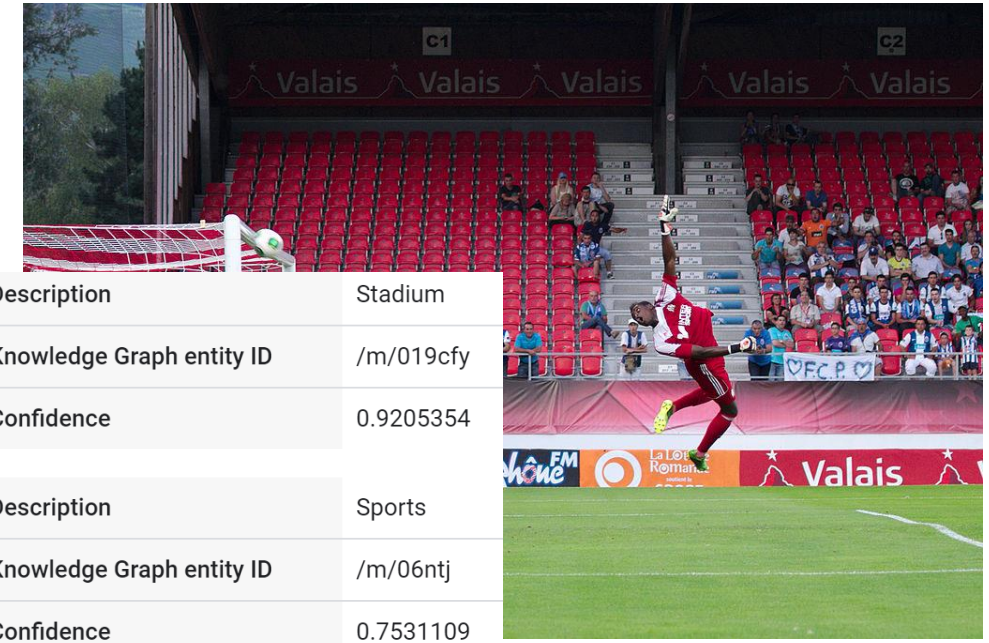


Face detection and image labeling

- Face detection API
  - detect faces in an image or video
  - identify key facial features
  - get the contours of detected faces



- Image labeling
  - list of the entities that were recognized: people, things, places, activities, etc
  - labels come with a score that indicates their confidence
  - On-device: 400+ labels, cloud: 10000+ labels

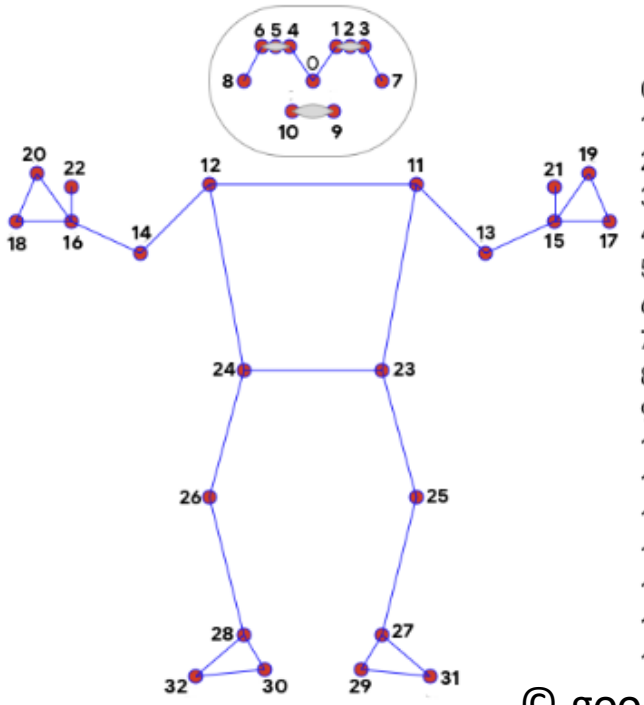


Description	Stadium
Knowledge Graph entity ID	/m/019cfy
Confidence	0.9205354

Description	Sports
Knowledge Graph entity ID	/m/06ntj
Confidence	0.7531109

Pose detection

- Recognises 33 skeletal points
- Face must be visible
- Returns x,y (and z) of points within image
- Stream or single image
- Low/high accuracy



Landmarks

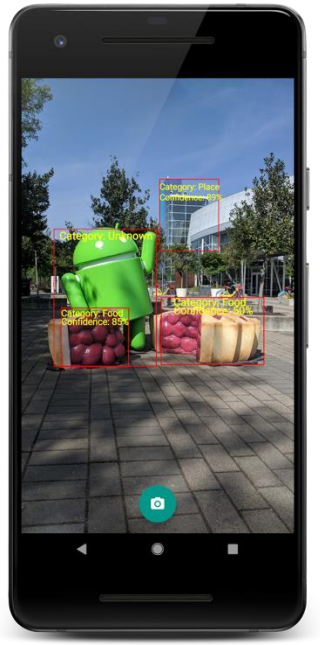
0. Nose
1. Left_eye_inner
2. Left_eye
3. Left_eye_outer
4. Right_eye_inner
5. Right_eye
6. Right_eye_outer
7. Left_ear
8. Right_ear
9. Left_mouth
10. Right_mouth
11. Left_shoulder
12. Right_shoulder
13. Left_elbow
14. Right_elbow
15. Left_wrist
16. Right_wrist
17. Left_pinky
18. Right_pinky
19. Left_index
20. Right_index
21. Left_thumb
22. Right_thumb
23. Left_hip
24. Right_hip
25. Left_knee
26. Right_knee
27. Left_ankle
28. Right_ankle
29. Left_heel
30. Right_heel
31. Left_foot_index
32. Right_foot_index

Landmark recognition and object detection/tracking

- Landmark recognition API:
  - each landmark's geographic coordinates and the region of the image the landmark was found.
  - automatically generate image metadata

Result	
Description	Brugge
Geographic Coordinates	51.207367, 3.226933
Knowledge Graph entity ID	/m/0drjd2
Bounding Polygon	(20, 342), (651, 342), (651, 798
Confidence Score	0.77150935

- Object tracking
  - Detect and track objects in images/videos
  - Coarse grained categorization: *fashion good*, *food*, *home good*, *place*, *plant*
  - First stage of multi-level detection systems



Source: google.com

Example: object detection

- Activity starts Image Gallery to let the user select an image
- Image Gallery returns the selected image *uri* in an intent
- Image is processed using ML kit

```
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    Uri uri = data.getData();
    FirebaseVisionImage image = null;
    try {
        image = FirebaseVisionImage.fromFilePath(this, uri);
    } catch (IOException e) {
        e.printStackTrace();
    }
    …
```

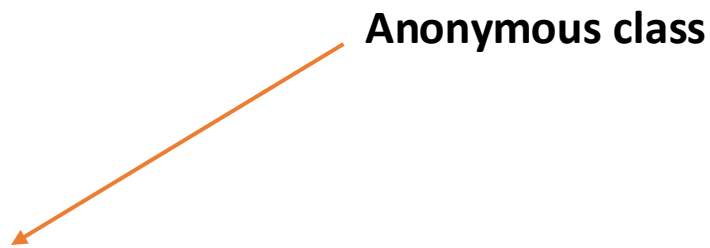
Example: object detection

- Set the options and create the detector

```
FirebaseVisionObjectDetectorOptions options =
        new FirebaseVisionObjectDetectorOptions.Builder()
                .setDetectorMode(FirebaseVisionObjectDetectorOptions.SINGLE_IMAGE_MODE)
                .enableMultipleObjects()
                .enableClassification()
                .build();

FirebaseVisionObjectDetector objectDetector =
        FirebaseVision.getInstance().getOnDeviceObjectDetector(options);
```

Example: object detection



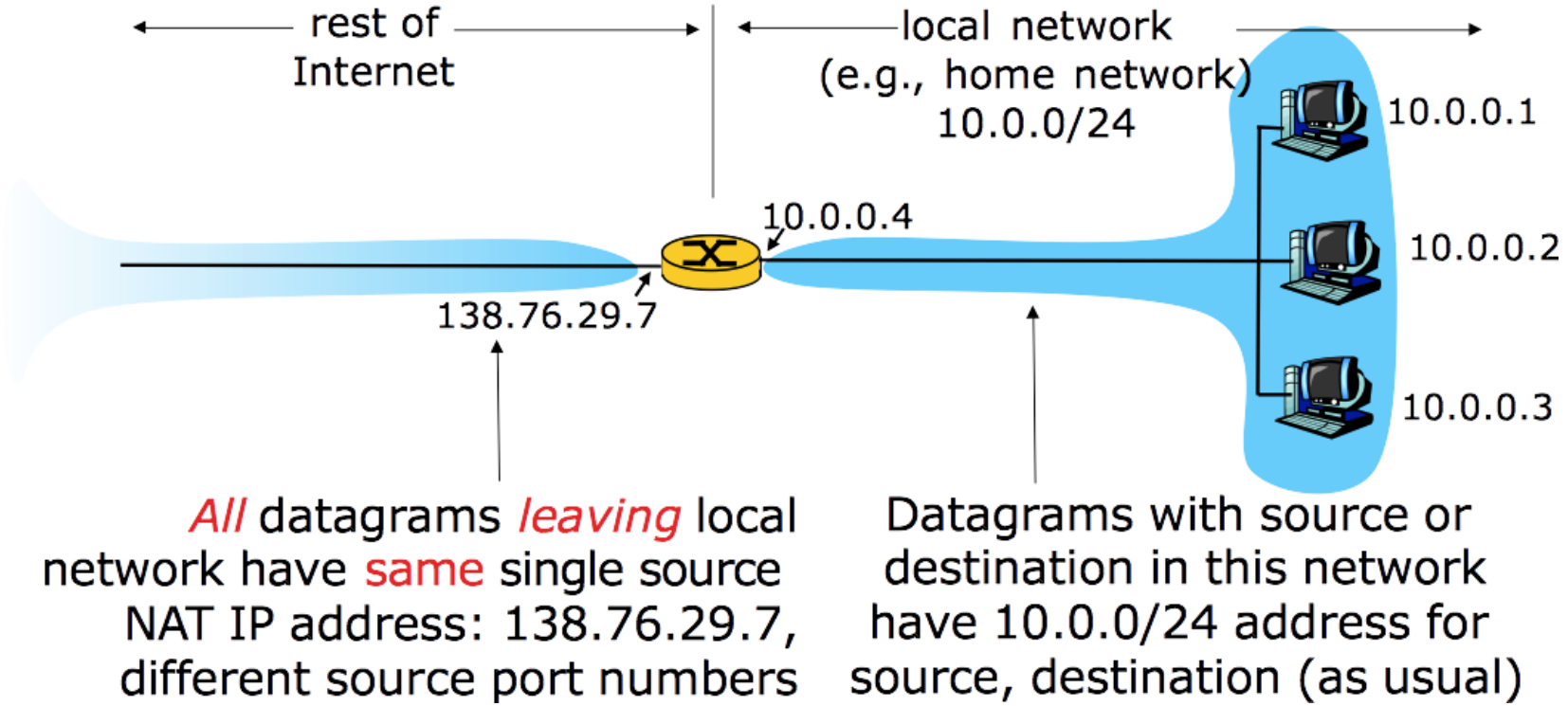

```
objectDetector.processImage(image)
        .addOnSuccessListener(
                new OnSuccessListener<List<FirebaseVisionObject>>() {
                    @Override
                    public void onSuccess(List<FirebaseVisionObject> detectedObjects) {
                        for (FirebaseVisionObject obj : detectedObjects) {
                            Integer id = obj.getTrackingId();
                            Rect bounds = obj.getBoundingBox();
                            int category = obj.getClassificationCategory();
                            Float confidence = obj.getClassificationConfidence();
                            Log.i(TAG, "OBJECT FOUND: category=" + category);
                        }
                    }
                })
        .addOnFailureListener(
          …
```

Firebase

- ML kit libraries are provided as part of Firebase
- Firebase is a mobile application development platform, acquired by Google in 2014
- First products were
  - Firebase Cloud Messaging
  - Firebase Realtime Database
- Now also
  - Authentication
  - Crash and performance analytics
  - Cloud-based content hosting
  - Remote configuration
  - Advertisement

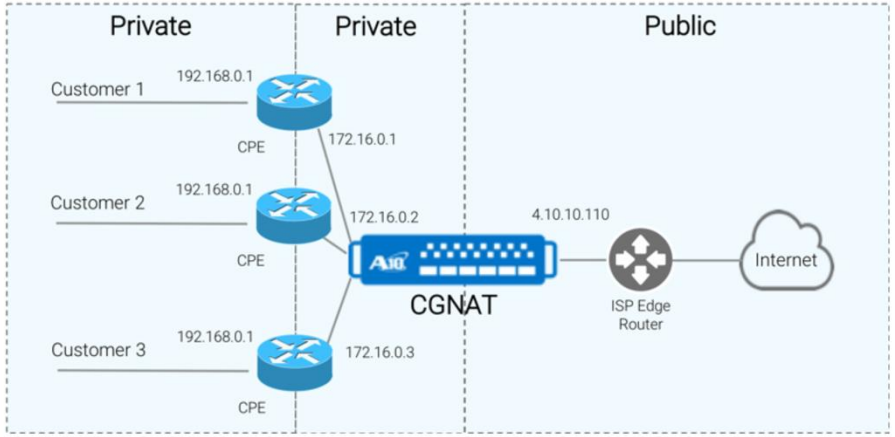
Firebase Cloud Messaging

- Problem: smartphones are generally behind a Carrier-grade NAT
- NAT?

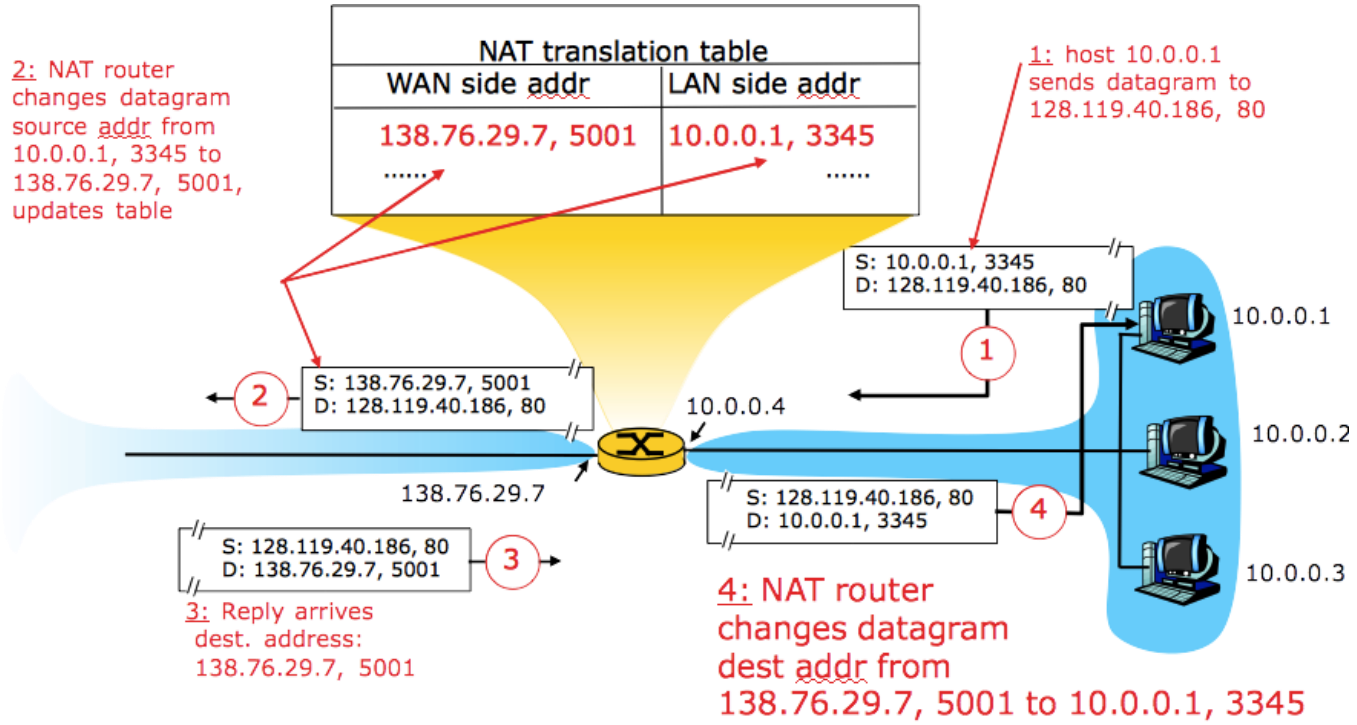


NAT: Network Address Translation

- Carrier-grade NATs are used by MNO to cope with shortage of IPv4 addresses
- Problem: being behind a NAT, smartphones cannot be reached if not initiators of communication



Source: A10 networks

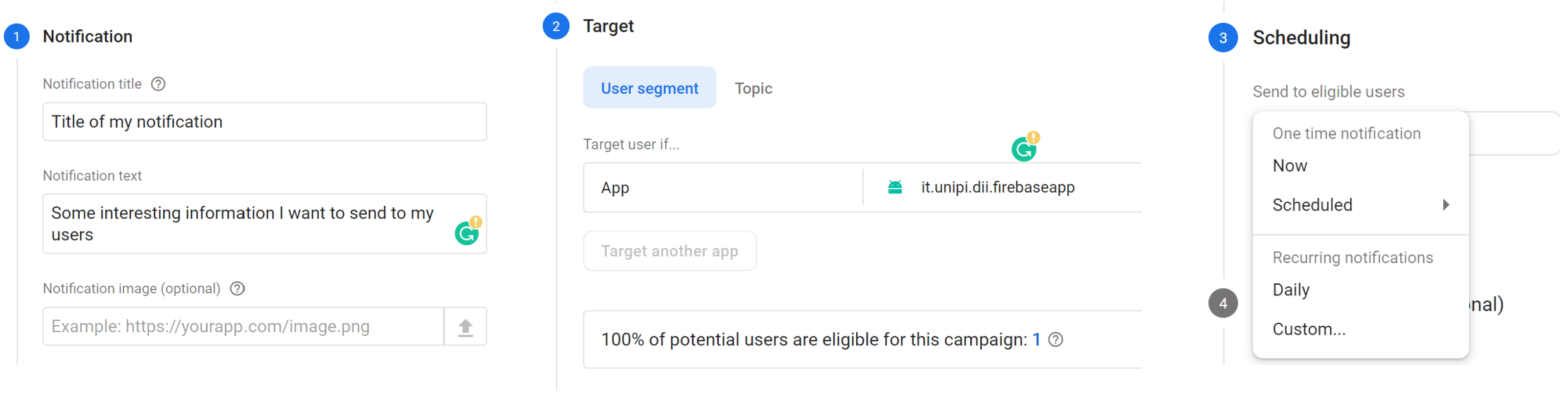


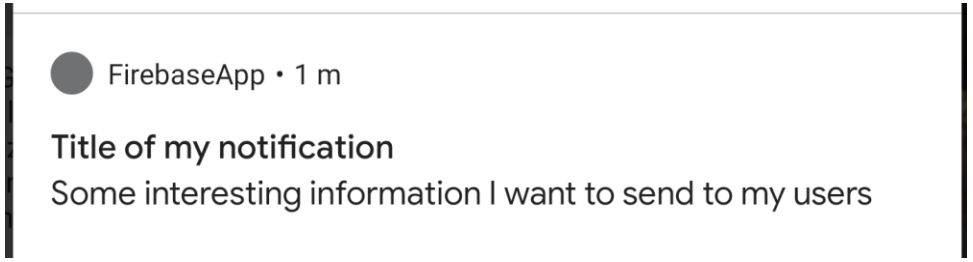
Firebase Cloud Messaging

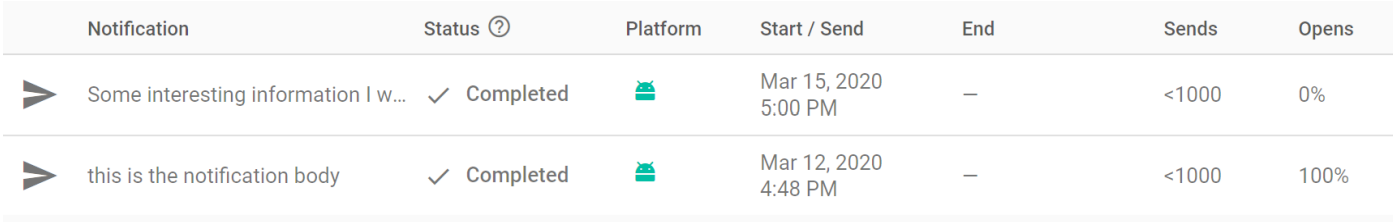
- Firebase Cloud Messaging
  - a cross-platform messaging solution that lets you send messages to smartphones
  - can notify a client app that new email or other data is available to sync
- Two types of messages *data messages* and *notification messages*
- *Data messages*:
  - you have to implement *FirebaseMessagingService* subclass
  - delivered to *onMessageReceived()* method
  - information is provided as a set of *<key, value>* pairs
- *Notification messages*:
  - if app is in background a notification is raised, then app started when notification selected
    - if message contains also *<key, value>* pairs delivered as extra of starting intent
  - if app is in foreground, everything delivered to *onMessageReceived()*

Console for sending notification messages

- Console for sending notification messages in Firebase

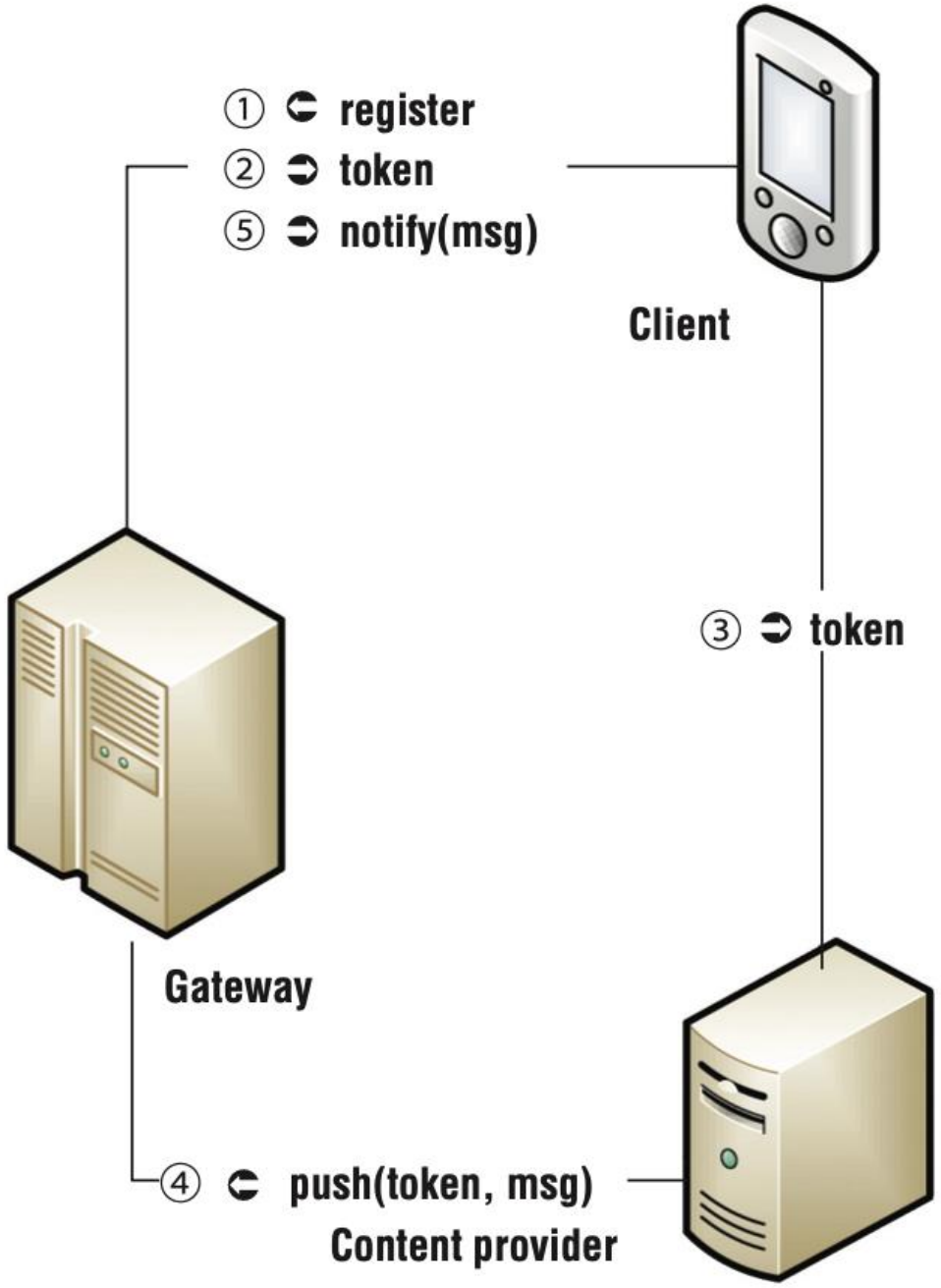






FCM

- Connecting clients and content providers



TensorFlow lite support

- TensorFlow is an ML platform developed by google
- TensorFlow models can be exported as TensorFlow Lite, which can be executed on Mobile devices
- The .tflite model is delivered as an asset of the mobile app
- The model is provided to an interpreter and then used for inference locally (no need to use the network, better latency)
- Library: There are wrapping classes for some common tasks (object detection, audio classification)
- Training not on mobile devices

References

- https://firebase.google.com/docs/ml-kit