CS 528 Mobile and Ubiquitous Computing

Lecture 6b: Mobile and Location-Aware Computing

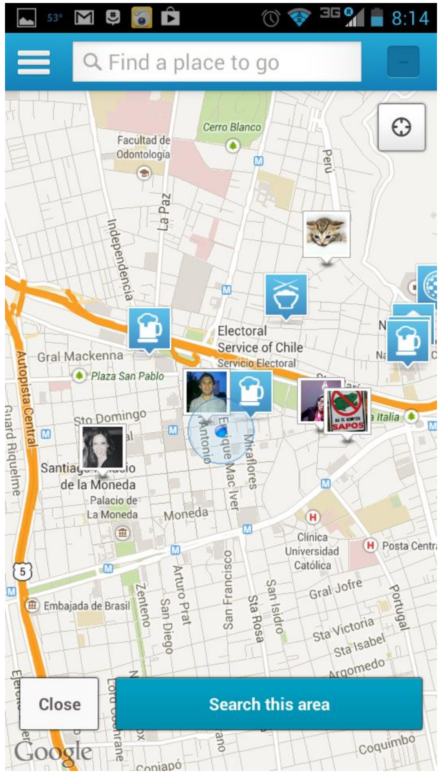
Emmanuel Agu

Location-Aware Computing

- **Definition:** Location-aware applications generate outputs/behaviors that depend on a user's location

- Examples:
  - Map of user's "current location"
  - Print to "closest" printer
  - Apps that find user's friends "closeby"
  - Reviews of "closeby" restaurants

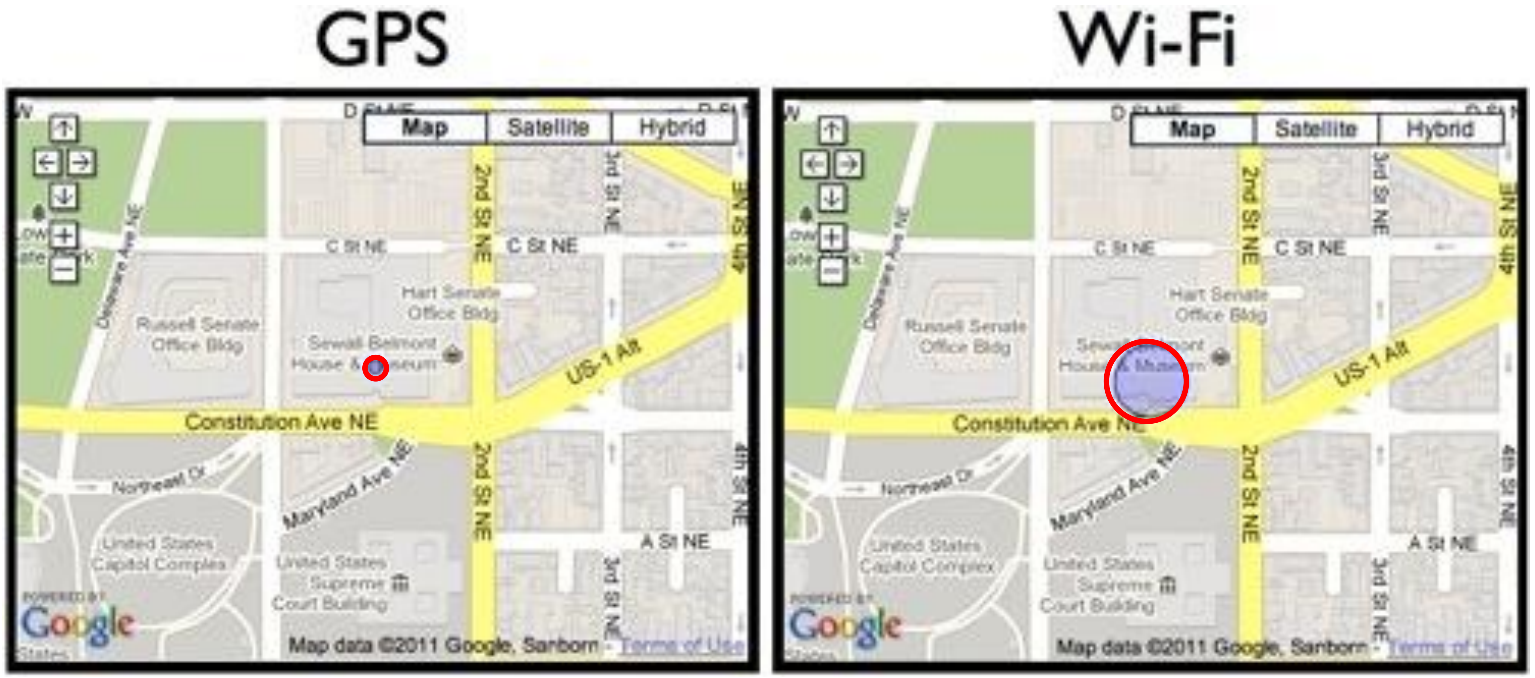
- Apps above require first determining user's location



Determining User Location on Smartphones

Location Tracking on Smartphones

- **Outdoors:** Uses GPS (More accurate but requires line of sight to satellites)
- **Indoors:** WiFi or cell tower signals (Location fingerprinting, less accurate)



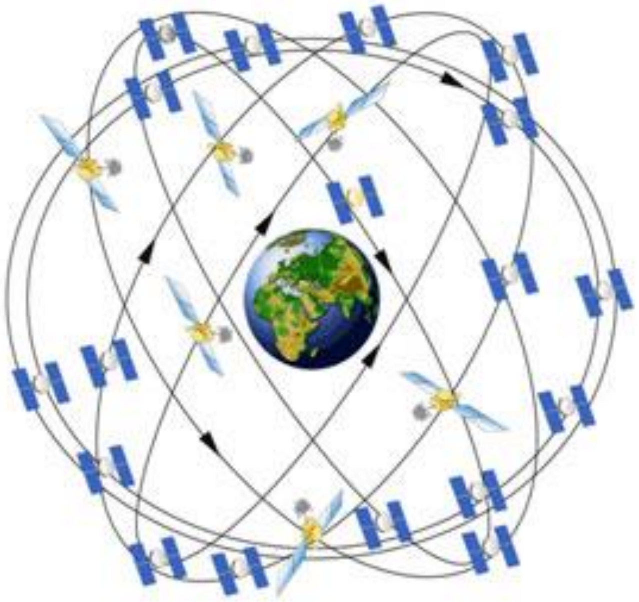
GPS

Wi-Fi

Map
Satellite
Hybrid
C St NE
2nd St NE
3rd St NE
4th St NE
Hart Senate Office Bldg
Russell Senate Office Bldg
Sewall Belmont House & Museum
US-1 Alt
Constitution Ave NE
Delaware Ave NE
Maryland Ave NE
Northeast Dr
United States Capitol Complex
United States Supreme Court Building
A St NE
Google
Map data ©2011 Google, Sanborn - Terms of Use

Global Positioning System (GPS)

- Originally 24 satellites orbiting earth, now 31
- **20,000 km above earth** (Medium earth orbit)
- 6 orbital planes with 4 satellites each
- 4 satellites visible from any spot on earth
- Location of any location on earth specified as <longitude,latitude>
- E.g. Worcester MA has **Latitude:** 42.2625,
  Longitude: -71.8027778



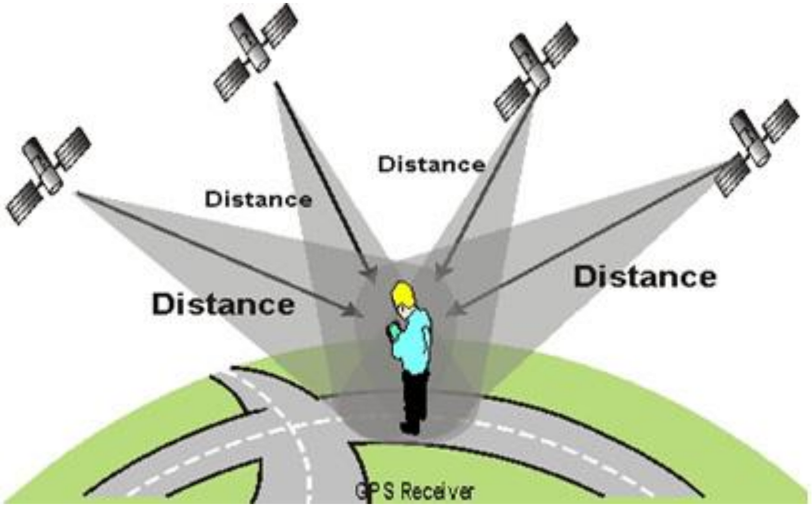
GPS User Segment

- **GPS satellites** broadcast accurate time-stamped packets

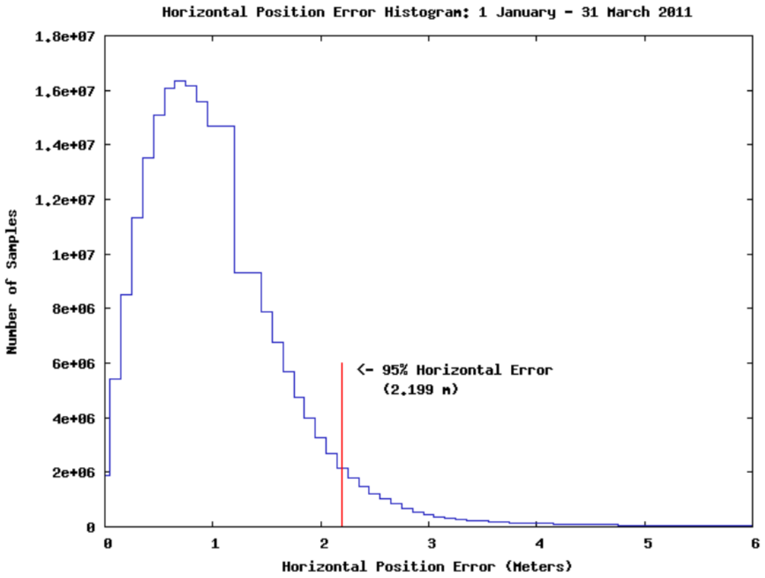
- GPS receiver calculates packet travel time/delay by comparing
  - Timestamps broadcast by satellite vs. time packet received

- **Trilateration:** GPS receiver compares delay from multiple satellites at known positions

- Accuracy within 16-32 feet (5 - 10 meters)

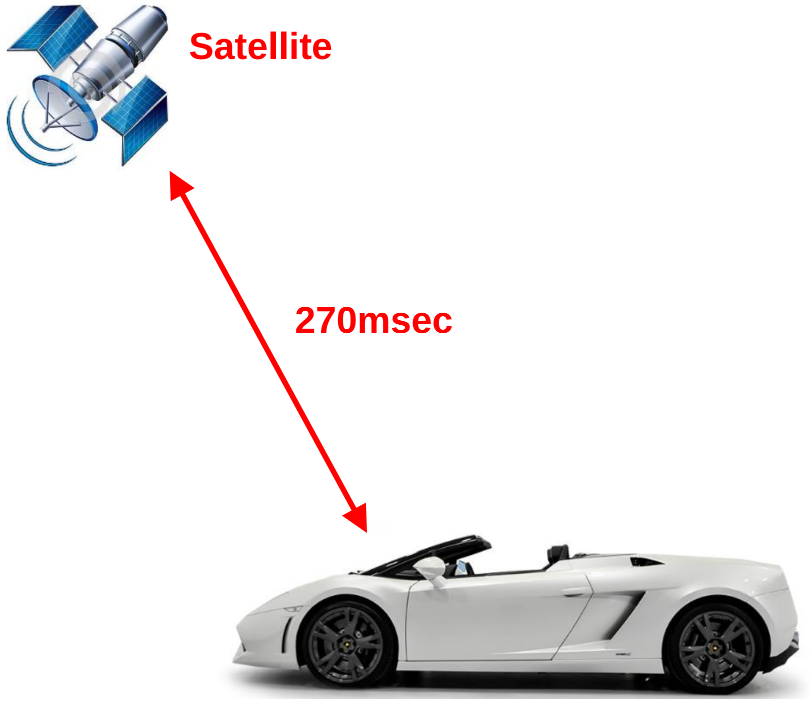


http://adamswalk.com/gpx-2/



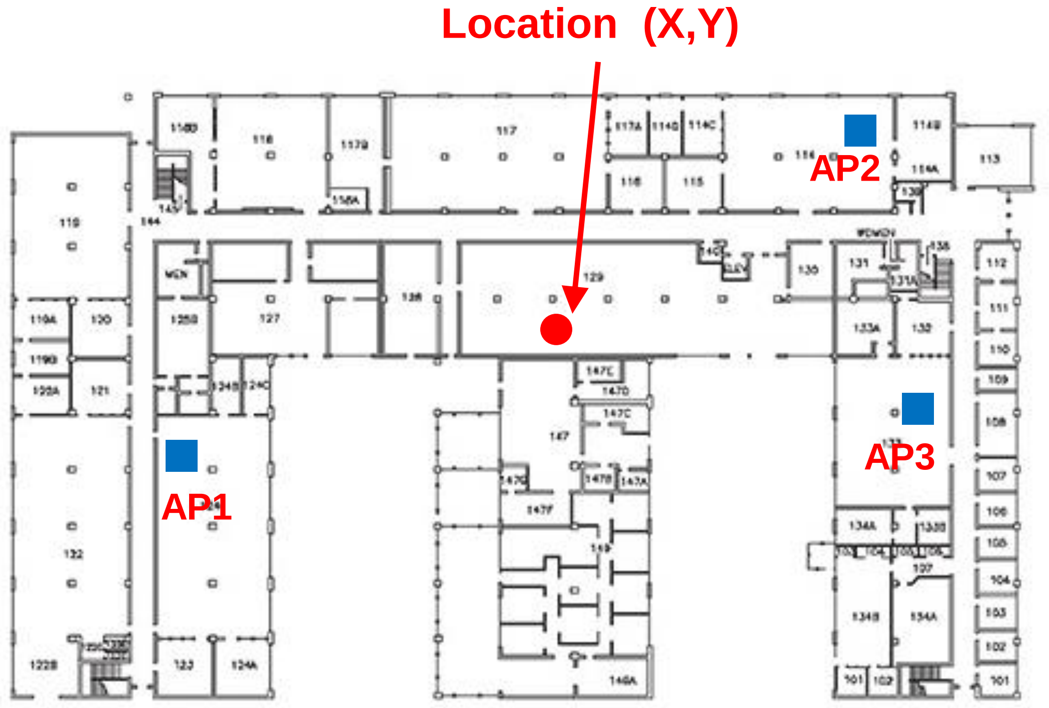
Determining User Location

- GPS reasonably accurate but
  - Requires line-of-sight between satellite and car receiver
  - Only works OUTDOORS (signals don’t penetrate buildings)
  - ~270 msec **Lag/delay** in acquiring satellite signal, or re- acquiring if lost
  - Drains battery power
- **Alternative:** Use Wi-Fi location sensing indoors

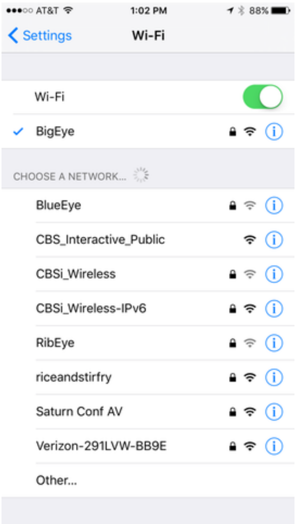


WiFi Location Fingerprinting

- **Key insight:** At each (X,Y) location, WiFi APs observed + their signal strengths, is unique



OBSERVED AP SIGNAL STRENGTH			
	AP1	AP2	AP3
(X,Y)	24	36	45



●●●○○ AT&T 1:02 PM 88%

< Settings Wi-Fi

Wi-Fi

✓ BigEye

CHOOSE A NETWORK...

BlueEye

CBS_Interactive_Public

CBSi_Wireless

CBSi_Wireless-IPv6

RibEye

riceandstirfry

Saturn Conf AV

Verizon-291LVW-BB9E

Other...

- **WiFi Location fingerprinting:** Estimate device's location based on combination of Wi-Fi access points seen + Signal Strengths

Location Estimation using Wi-Fi Fingerprinting

PRE-RECORDED TUPLES					
LOCATION		SIGNAL STRENGTH			
X	Y	AP1	AP2	AP3	AP4
:::	:::	:::	:::	:::	:::
80	145	32	28	12	8
40	145	36	20	10	6
:::	:::	:::	:::	:::	:::
220	355	-	25	36	44
260	355	4	21	39	42
:::	:::	:::	:::	:::	:::
350	210	16	-	28	36
:::	:::	:::	:::	:::	:::
380	145	22	12	-	44
:::	:::	:::	:::	:::	:::

OBSERVED SIGNAL STRENGTH			
AP1	AP2	AP3	AP4
-	24	36	45

Location (X,Y)??

- Inference Algorithms
  - Min. Threshold
  - Euclidean Dist.
  - Joint Probability
  - Bayesian Filters

Google builds and stores this database (APs + Signal Strength)
at each X,Y location)

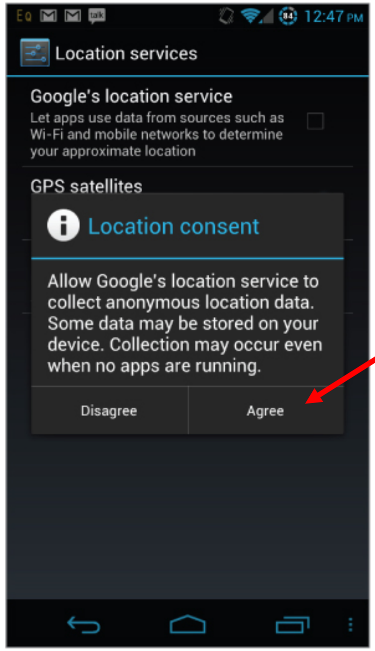
How to Build table of APs observed at (X,Y) Locations?

- Devices (e.g. smartphone) with GPS and WiFi turned on simultaneously build table
- Send data to third party repositories (e.g. Google or Wigle.net)
- Also called **war driving**
- Can record cell tower signal strength instead of AP

PRE-RECORDED TUPLES					
LOCATION		SIGNAL STRENGTH			
X	Y	AP1	AP2	AP3	AP4
:::	:::	:::	:::	:::	:::
80	145	32	28	12	8
40	145	36	20	10	6
:::	:::	:::	:::	:::	:::
220	355	-	25	36	44
260	355	4	21	39	42

GPS gathers Location (X,Y)

WiFi card gathers APs seen + Signal Strengths



Location services

Google's location service

Let apps use data from sources such as Wi-Fi and mobile networks to determine your approximate location □

GPS satellites

Location consent

Allow Google's location service to collect anonymous location data. Some data may be stored on your device. Collection may occur even when no apps are running.

Google gathers Location, AP seen Data if you consent

Location Sensing in Android Apps

Google Location APIs

https://developer.android.com/training/location

- Android now has 2 location APIs (older vs newer)

- Older Android framework location APIs (**android.location**)
  - Now phased out, still used in some older books, online sources.
  - Be careful what code you use!

- Newer location API is part of Google Play Services
- Need to set up Google Play services.
  - Download and install Google Play services component via SDK Manager
  - Set up Google Play service: https://developers.google.com/android/guides/setup

Google Location APIs: Get Last Known Location

https://developer.android.com/training/location/retrieve-current

- **Fused location provider:** location API in Google Play services
  - Intelligently combines different signals (GPS, WiFi) to provide location information app needs
  - High level, allows specification of parameters (e.g., accuracy and power consumption)

- Location object retrieved from **fused location provider**, contains extensive location information including:
  - Geographical location (longitude, latitude)
  - Direction of horizontal travel (bearing)
  - Altitude (height above sea level)
  - Velocity of device

Google Location APIs: Location Methods

https://developer.android.com/training/location/retrieve-current

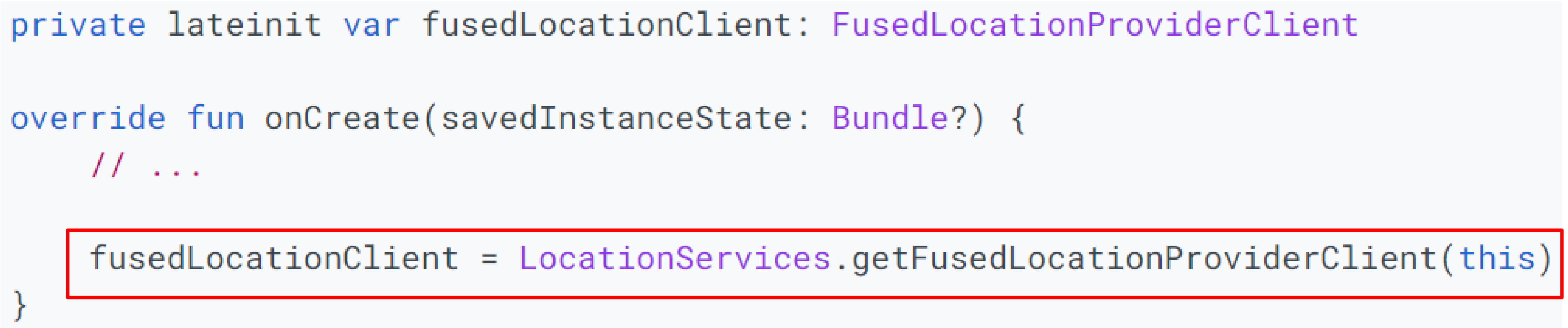
- Frequently just want device’s current location (or last received location update, cached)
  - Multiple apps can subscribe for location updates
  - Each time request made, cached on device
  - **Get Last Location:** Your app gets last received location (for another app?)

- **Fused location provider:** has 2 methods to get location estimate
  - **getLastLocation( ):** Fast location estimate, cached,
    - Minimizes battery consumption
    - But may be out of date if no other subscribed app used location. Multiple apps can subscribe for location updates.
  - **getCurrentLocation( ):** Gets fresh, more accurate location
    - More power hungry

Google Location APIs: Get Last Known Location

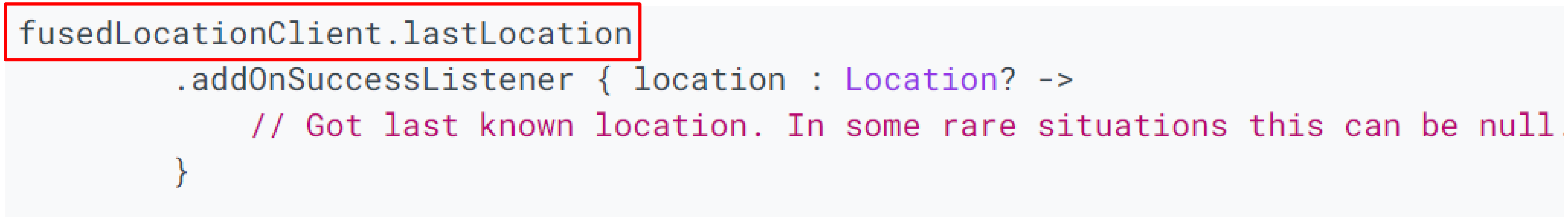
https://developer.android.com/training/location/retrieve-current

- **Step 1:** Create location services client in **onCreate( )** method:



```
private lateinit var fusedLocationClient: FusedLocationProviderClient

override fun onCreate(savedInstanceState: Bundle?) {
    // ...

    fusedLocationClient = LocationServices.getFusedLocationProviderClient(this)
}
```

- **Step 2:** Get last known location using **getLastLocation( )** method



```
fusedLocationClient.lastLocation
        .addOnSuccessListener { location : Location? ->
            // Got last known location. In some rare situations this can be null.
        }
```

Google Location APIs: Change Location Settings

https://developer.android.com/training/location/change-location-settings

- Settings are defined by **LocationRequest** data object

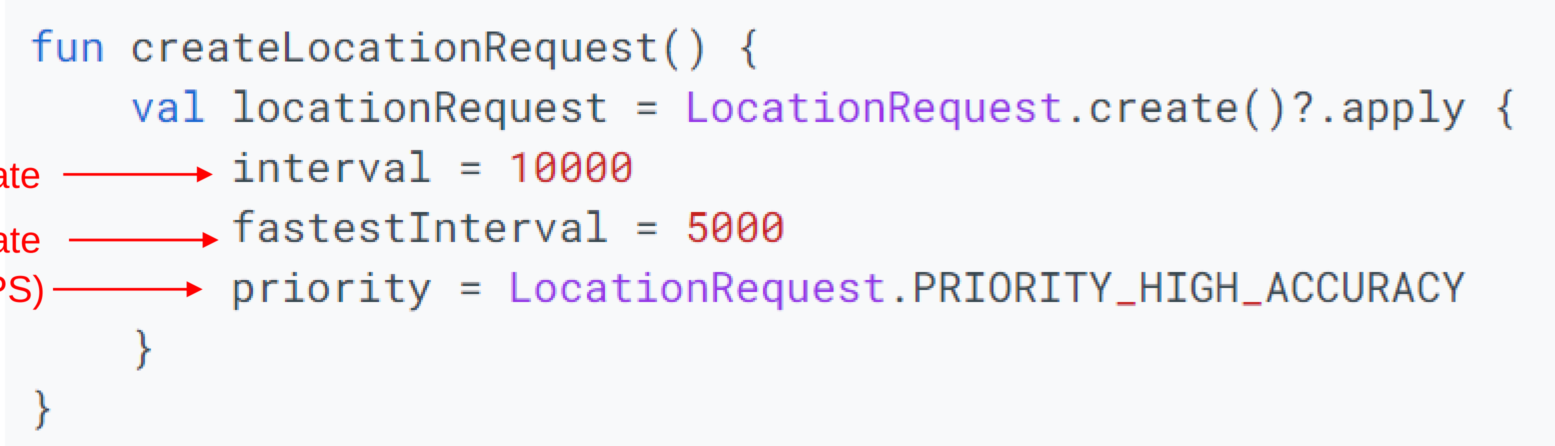
- Can set/change location-related settings. E.g.
  - Level of accuracy (WiFi vs GPS)
  - Power consumption
  - Update interval (how frequent)

- Example settings:
  - **Update interval:** Use **setInterval( )** method to set app’s preferred location update rate (in milliseconds)
  - **Fastest update interval:** Use **setFastestInterval( )** to set fastest rate in milliseconds at which app can handle location updates
  - **Priority:** Use **setPriority( )** to set priority. E.g.
    - **PRIORITY_BALANCED_POWER_ACCURACY:** consume less power, coarse accuracy (~100 meters), likely to use WiFi and Cell tower NOT GPS
    - **PRIORITY_HIGH_ACCURACY:** Use GPS or most precise location possible
    - **PRIORITY_LOW_POWER:** City-level precision (~10 kilometers)
    - **PRIORITY_NO_POWER:** Use negligible power. Receive updates trigged by OTHER apps. Does not request location updates

Google Location APIs: Change Location Settings

https://developer.android.com/training/location/change-location-settings

- Example code to create location request, set parameters:



```
fun createLocationRequest() {
    val locationRequest = LocationRequest.create()?.apply {
        interval = 10000
        fastestInterval = 5000
        priority = LocationRequest.PRIORITY_HIGH_ACCURACY
    }
}
```

Preferred update rate →

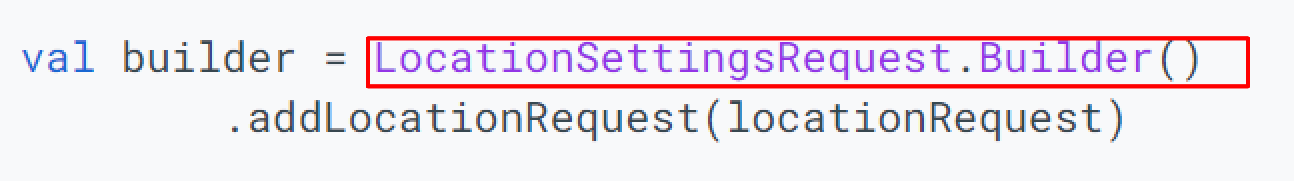
Fastest update rate →

Use high accuracy (GPS) →

Google Location APIs: Getting Current Location Settings

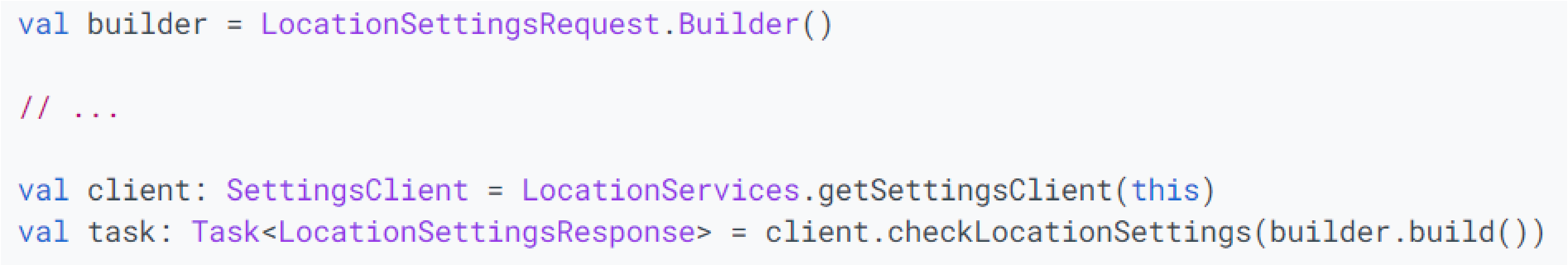
https://developer.android.com/training/location/change-location-settings

- Can check location settings using the **settings client**
- First create **LocationSettingsRequest.Builder**
- Then add one or more location requests



```
val builder = LocationSettingsRequest.Builder()
        .addLocationRequest(locationRequest)
```

- Can then check current location settings

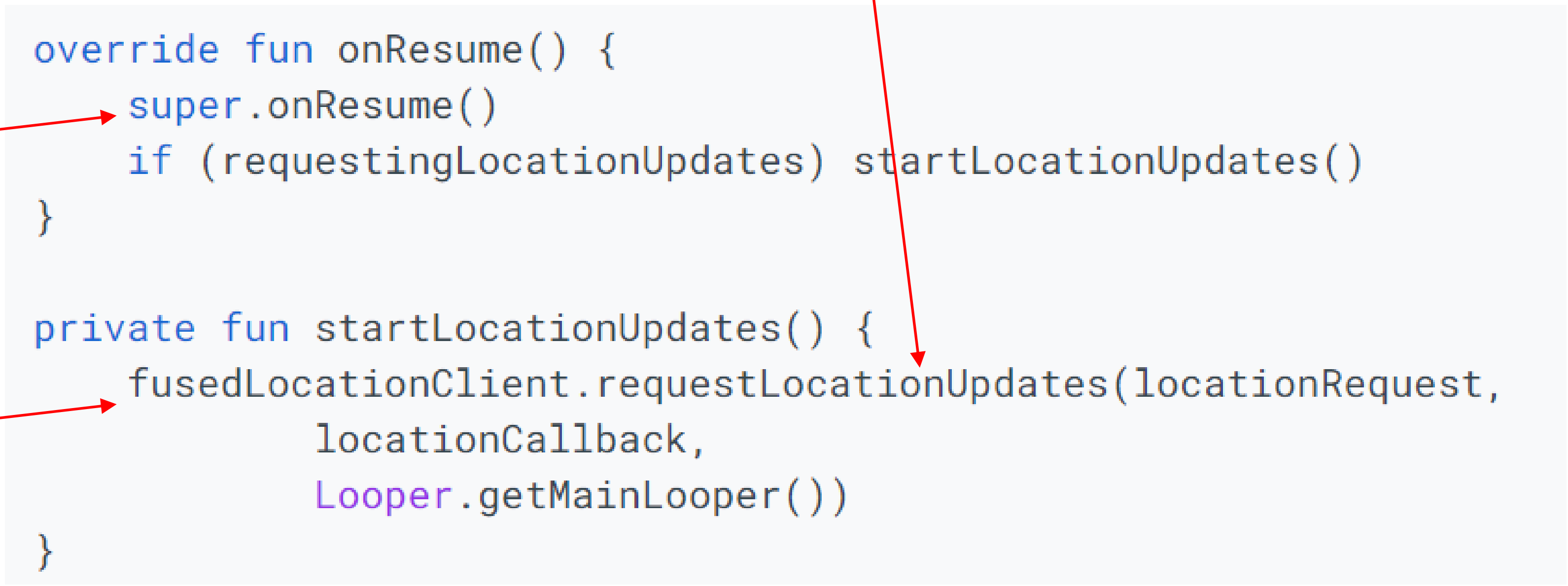


```
val builder = LocationSettingsRequest.Builder()

// ...

val client: SettingsClient = LocationServices.getSettingsClient(this)
val task: Task<LocationSettingsResponse> = client.checkLocationSettings(builder.build())
```

Google Location APIs: Getting location updates

https://developer.android.com/training/location/request-updates

- Getting location updates:
  - Example use cases: user walking or driving (needs regular updates as location continuously changing)
- Can request regular updates on device’s location using **requestLocationUpdates( )** method in fused location provider



```
override fun onResume() {
    super.onResume()
    if (requestingLocationUpdates) startLocationUpdates()
}

private fun startLocationUpdates() {
    fusedLocationClient.requestLocationUpdates(locationRequest,
            locationCallback,
            Looper.getMainLooper())
}
```

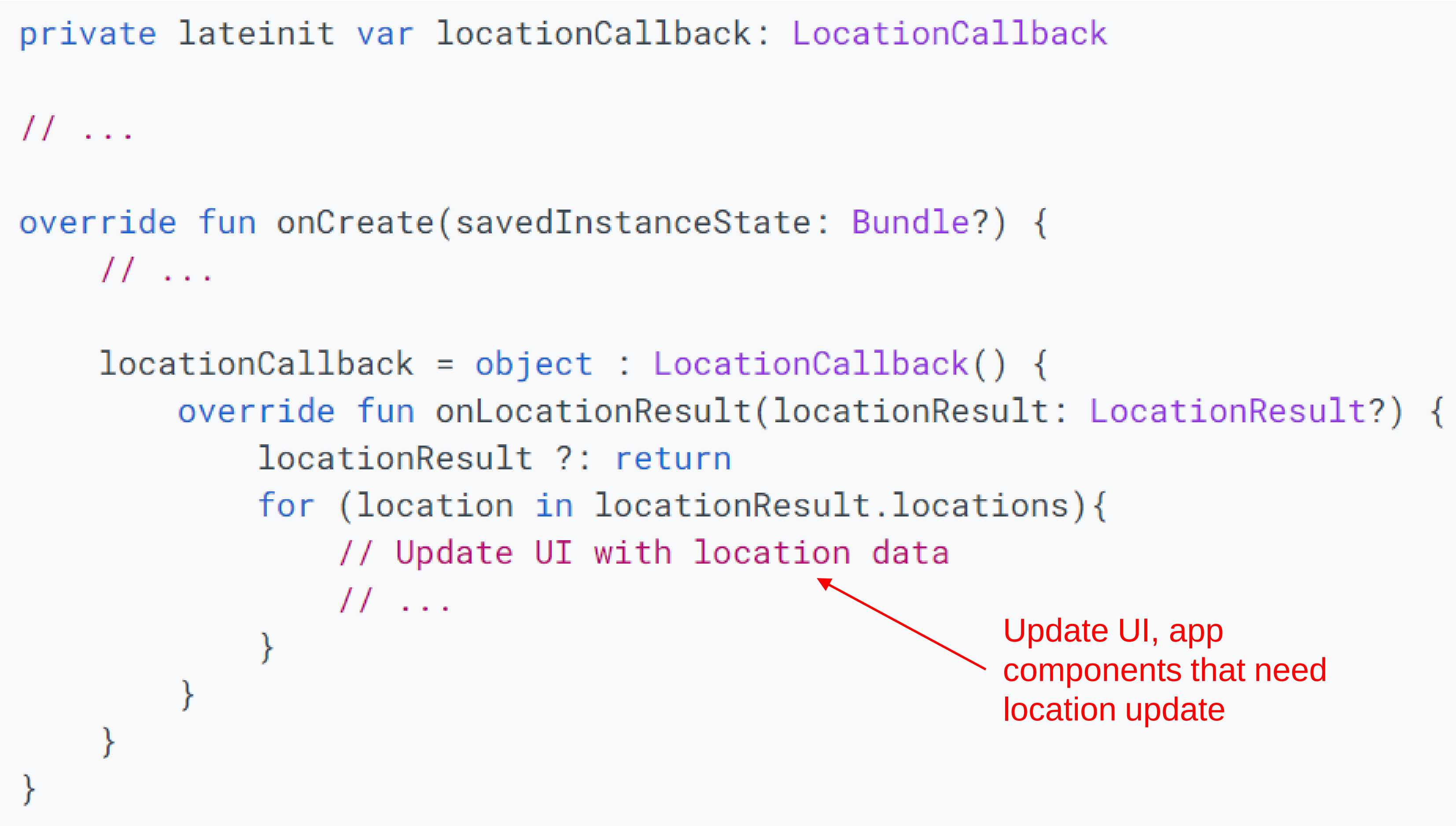
Restart continuous GPS updates in onResume( )

Function to request regular location updates

Google Location APIs: Getting location updates

https://developer.android.com/training/location/request-updates

- To define the location update callback
- Called by Android to send our app location updates



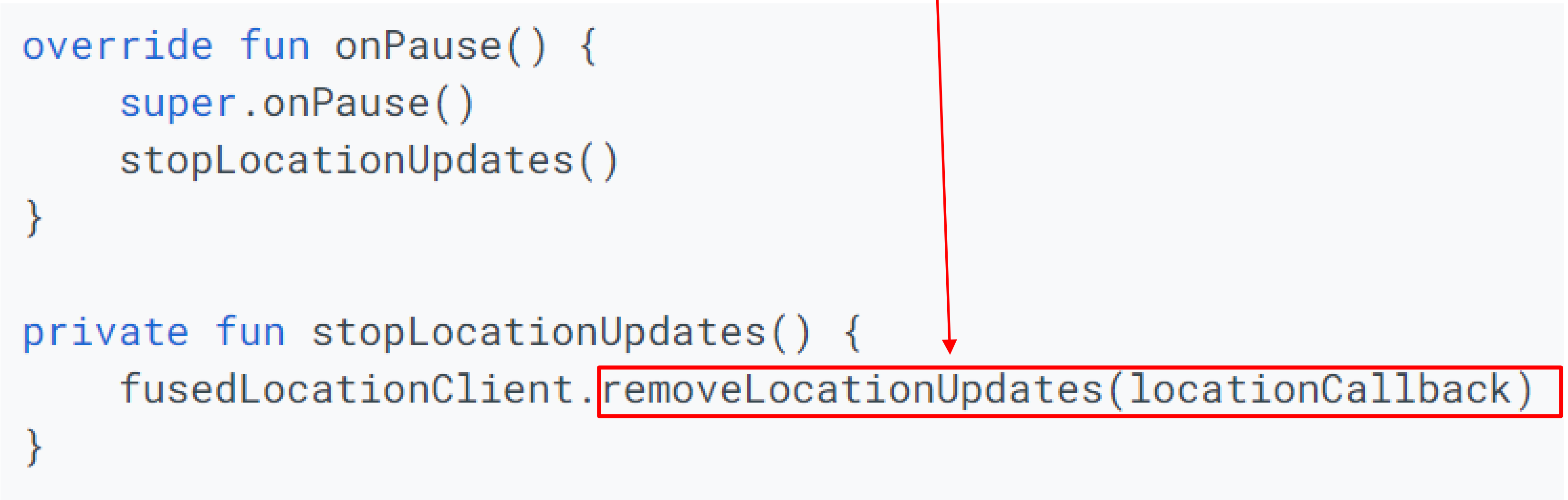
```
private lateinit var locationCallback: LocationCallback

// ...

override fun onCreate(savedInstanceState: Bundle?) {
    // ...

    locationCallback = object : LocationCallback() {
        override fun onLocationResult(locationResult: LocationResult?) {
            locationResult ?: return
            for (location in locationResult.locations){
                // Update UI with location data
                // ...
            }
        }
    }
}
```

Update UI, app components that need location update

Google Location APIs: Getting location updates

https://developer.android.com/training/location/request-updates

- To stop location updates, call **removeLocationUpdates( )** in **onPause( )** method

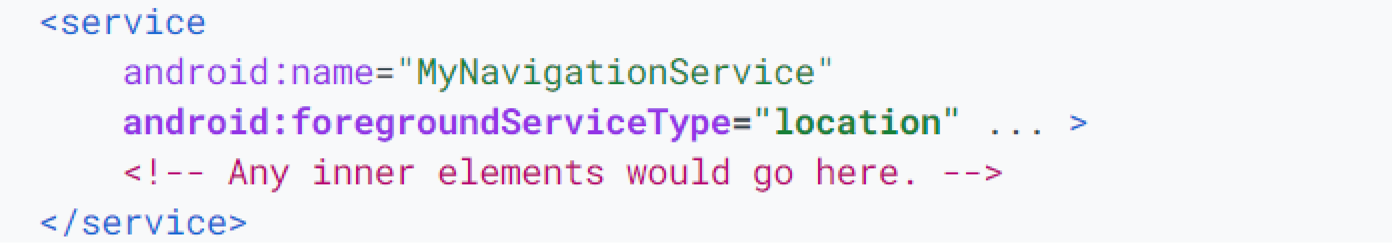


```
override fun onPause() {
    super.onPause()
    stopLocationUpdates()
}

private fun stopLocationUpdates() {
    fusedLocationClient.removeLocationUpdates(locationCallback)
}
```

Requesting User Permissions

https://developer.android.com/guide/topics/location/strategies.html

- Apps that use location services must request smartphone owner’s permission
- Types of location access:
  - **Category:** foreground or background
  - **Accuracy:** precise or approximate location
  - **Foreground:**
    - Uses location information once or for defined time period. E.g. messaging app shares location once
    - Activity that belongs to app is visible
    - Declare **foreground service type** of **location**

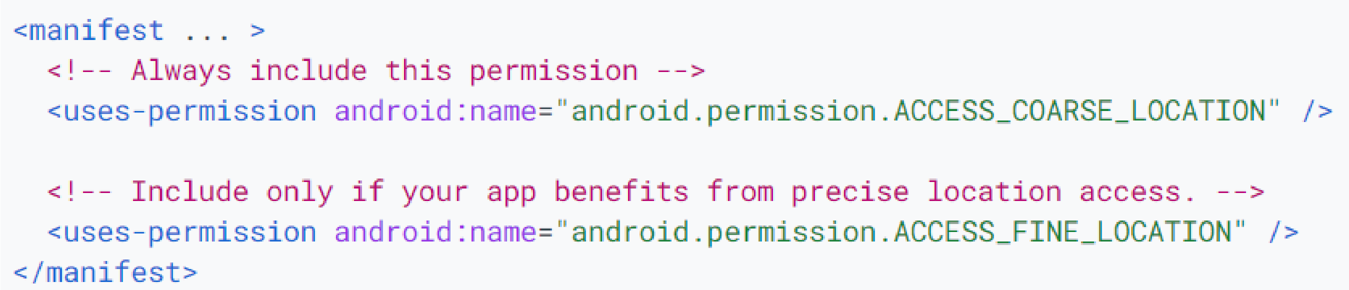


```
<service
    android:name="MyNavigationService"
    android:foregroundServiceType="location" ... >
    <!-- Any inner elements would go here. -->
</service>
```

Requesting User Permissions

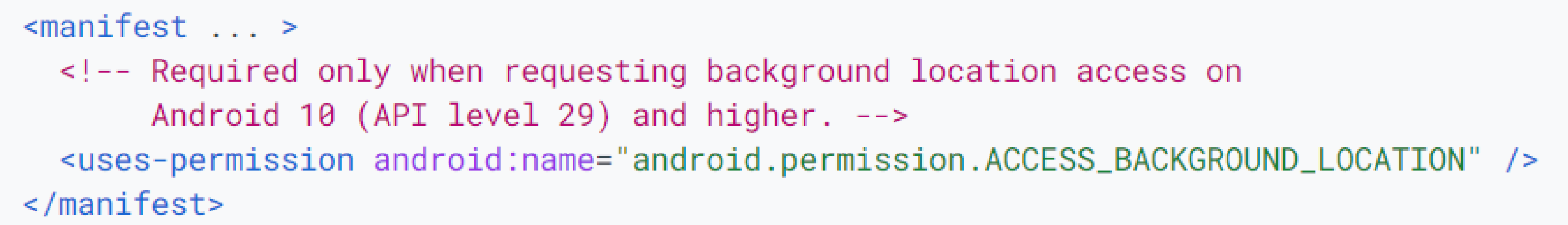
https://developer.android.com/guide/topics/location/strategies.html

- Declare foreground location when app requests either:
  - **ACCESS_FINE_LOCATION:** GPS
  - **ACCESS_COARSE_LOCATION:** WiFi or cell towers



```
<manifest ... >
  <!-- Always include this permission -->
  <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />

  <!-- Include only if your app benefits from precise location access. -->
  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
</manifest>
```

- Declare **background location** if app constantly shares location or uses **GeoFencing** API. E.g.
  - Example: App that turns on certain app features at specific locations (e.g. work vs. home)

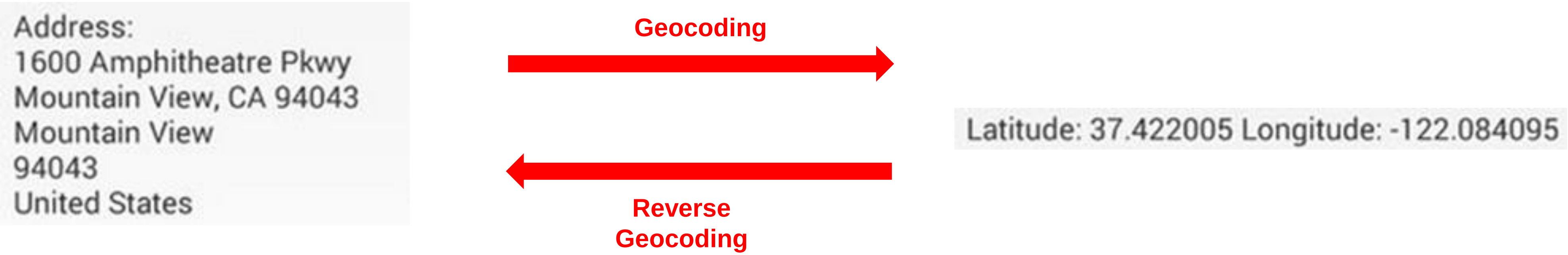


```
<manifest ... >
  <!-- Required only when requesting background location access on
       Android 10 (API level 29) and higher. -->
  <uses-permission android:name="android.permission.ACCESS_BACKGROUND_LOCATION" />
</manifest>
```

Location Representation

Semantic Location

- GPS represents location as <longitude,latitude>
- **Semantic location** is better for reasoning about locations
- **E.g.** Street address (140 Park Avenue, Worcester, MA) or (building, floor, room)
- **Android supports:**
  - **Geocoding:** Convert addresses into longitude/latitude coordinates
  - **Reverse geocoding:** convert longitude/latitude coordinates into human readable address



Address:
1600 Amphitheatre Pkwy
Mountain View, CA 94043
Mountain View
94043
United States

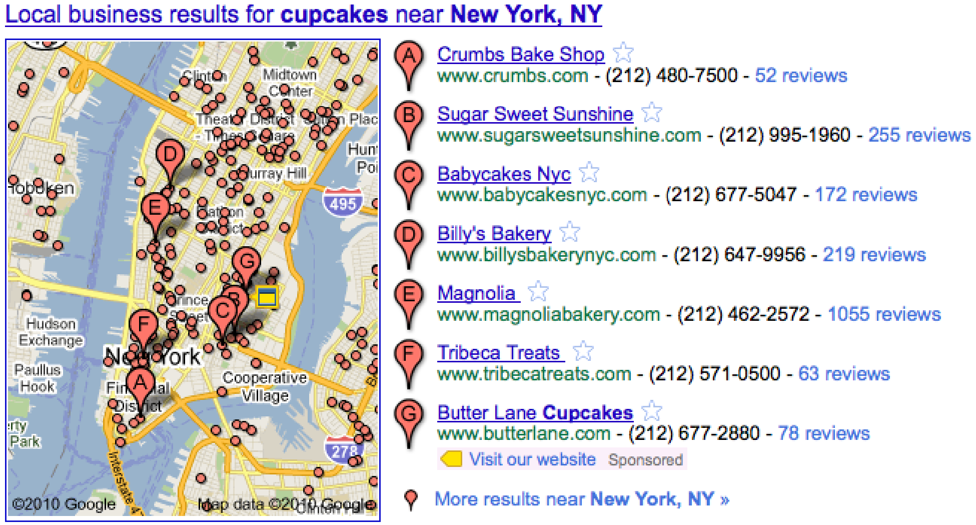
Latitude: 37.422005 Longitude: -122.084095

- **Android Geocoding API:** access to **geocoding** and **reverse geocoding** services using HTTP requests
  - https://developers.google.com/maps/documentation/geocoding/start

Google Places API Overview

https://developers.google.com/maps/documentation/places/web-service/overview

- Access **high-quality photos** of a place
- Users can also add place information to the database
  - E.g. business owners can add their business as a place in Places database
  - Other apps can then retrieve info after moderation



Local business results for **cupcakes** near **New York, NY**

- A Crumbs Bake Shop ☆
  www.crumbs.com - (212) 480-7500 - 52 reviews
- B Sugar Sweet Sunshine ☆
  www.sugarsweetsunshine.com - (212) 995-1960 - 255 reviews
- C Babycakes Nyc ☆
  www.babycakesnyc.com - (212) 677-5047 - 172 reviews
- D Billy's Bakery ☆
  www.billysbakerynyc.com - (212) 647-9956 - 219 reviews
- E Magnolia ☆
  www.magnoliabakery.com - (212) 462-2572 - 1055 reviews
- F Tribeca Treats ☆
  www.tribecatreats.com - (212) 571-0500 - 63 reviews
- G Butter Lane **Cupcakes** ☆
  www.butterlane.com - (212) 677-2880 - 78 reviews
  Visit our website Sponsored

More results near **New York, NY** »

Unlocking Location Context with t...
52
#GoogleDev100
#Android
100 Days of Google Developers

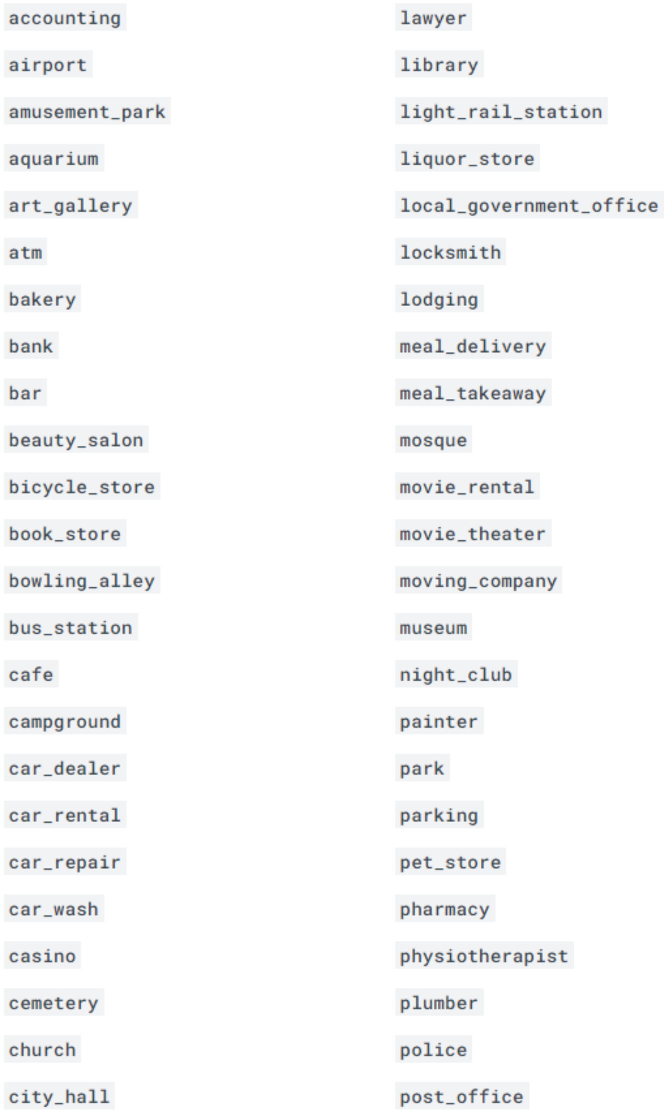
Google Places

https://developers.google.com/maps/documentation/places/web-service/overview

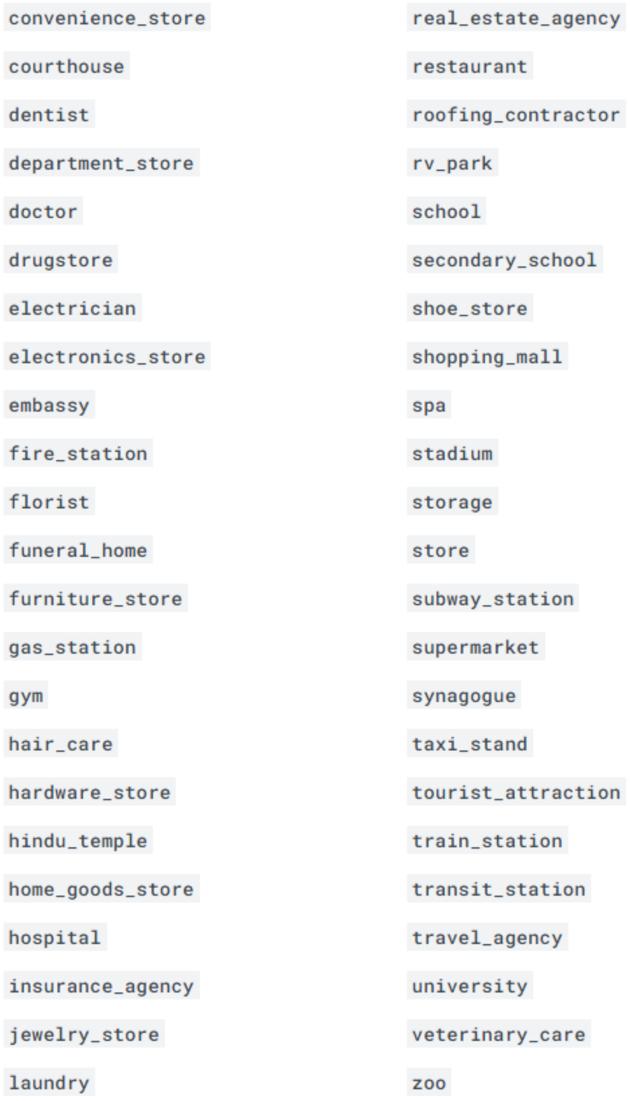
- **Place:** physical space that has a name (e.g. local businesses, points of interest, geographic locations)
  - E.g Logan airport, place type is **airport**
- **API:** Provides Contextual information about places near device, uses HTTP
- **Contextual information:**
  - Name of place
  - Address, geographical location
  - Place ID
  - Phone number
  - Place type,
  - Website URL,
  - etc.

Sample Place Types

https://developers.google.com/maps/documentation/places/web-service/supported_types



accounting
airport
amusement_park
aquarium
art_gallery
atm
bakery
bank
bar
beauty_salon
bicycle_store
book_store
bowling_alley
bus_station
cafe
campground
car_dealer
car_rental
car_repair
car_wash
casino
cemetery
church
city_hall
lawyer
library
light_rail_station
liquor_store
local_government_office
locksmith
lodging
meal_delivery
meal_takeaway
mosque
movie_rental
movie_theater
moving_company
museum
night_club
painter
park
parking
pet_store
pharmacy
physiotherapist
plumber
police
post_office

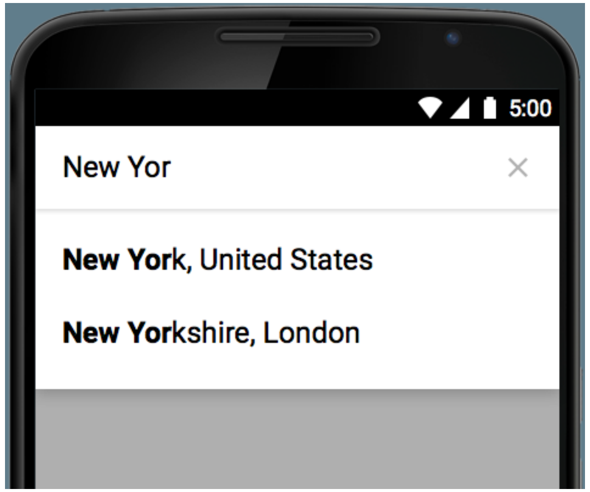


convenience_store
courthouse
dentist
department_store
doctor
drugstore
electrician
electronics_store
embassy
fire_station
florist
funeral_home
furniture_store
gas_station
gym
hair_care
hardware_store
hindu_temple
home_goods_store
hospital
insurance_agency
jewelry_store
laundry
real_estate_agency
restaurant
roofing_contractor
rv_park
school
secondary_school
shoe_store
shopping_mall
spa
stadium
storage
store
subway_station
supermarket
synagogue
taxi_stand
tourist_attraction
train_station
transit_station
travel_agency
university
veterinary_care
zoo

Google Places API Overview

https://developers.google.com/maps/documentation/places/web-service/overview

- **Place requests** available:
  - **Place Search** returns a list of places based on a user's location or search string.
  - **Place Details** returns more detailed information about a specific place, including user reviews.
  - **Place Photos** provides access to the millions of place-related photos stored in Google's Place database.
  - **Place Autocomplete** automatically fills in the name and/or address of a place as users type.
  - **Query Autocomplete** provides a query prediction service for text-based geographic searches, returning suggested queries as users type.
- **Autocomplete:** queries the location database as users type, suggests nearby places matching letters typed in



New York, United States

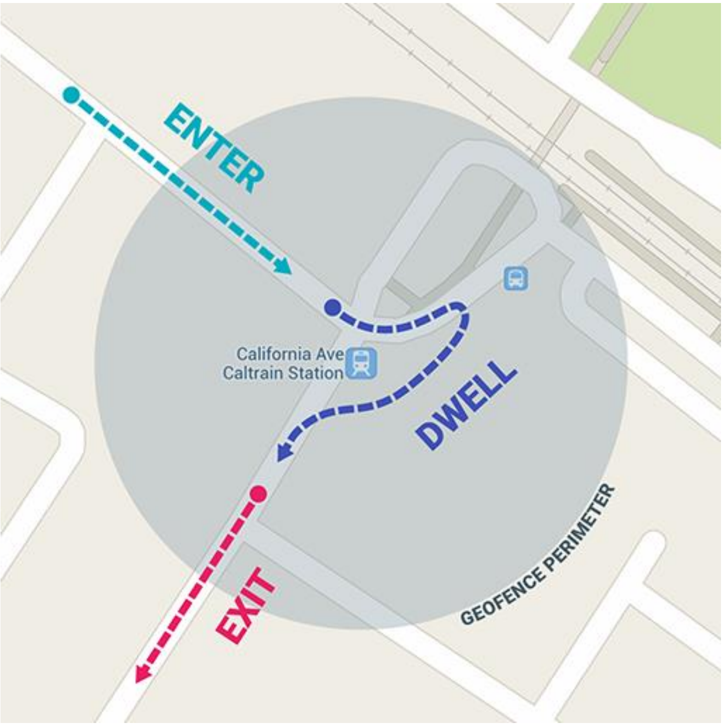
New Yorkshire, London

Other Useful Google Maps/Location APIs

GeoFencing

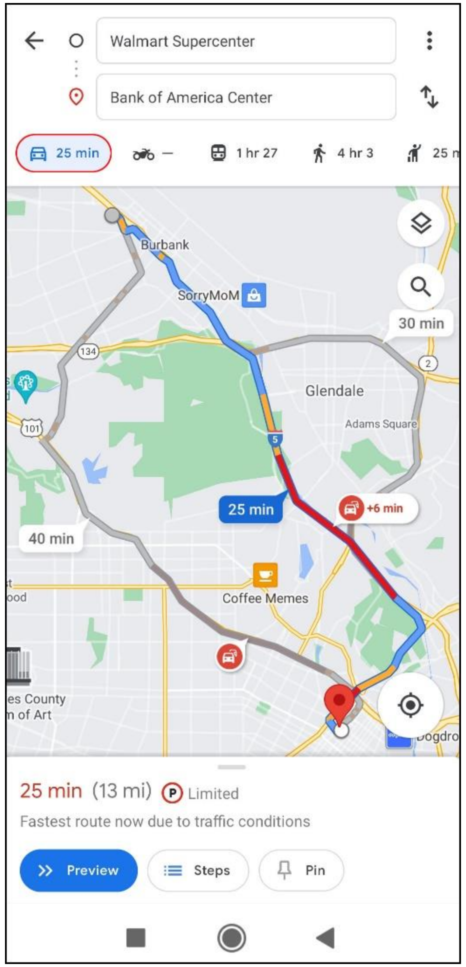
https://developer.android.com/training/location/geofencing

- **Geofence:** Sends alerts when user is within a certain radius to a location of interest (proximity)
- To specify geofence, indicating
  - Longitude, latitude of Geofence center
  - Radius of circle to monitor
- An app can specify up to 100 GeoFences
- Once geoFence configured, app receives:
  - **ENTER** event when user enters circle
  - **EXIT** event when user exits circle
- Can also specify a duration or **DWELL** user must be in circle before triggering event
- See Google tutorials on setting up GeoFence



Other Maps/Useful Location APIs

- **Directions API:** calculates directions between locations (walking, driving) as well as public transport directions
  - https://developers.google.com/maps/documentation/directions/overview
- **Distance Matrix API:** Calculate travel time and distance for multiple origin-destination pairs
  - Returns duration, distance for each origin-destination pair
  - https://developers.google.com/maps/documentation/distance-matrix/overview



Walmart Supercenter

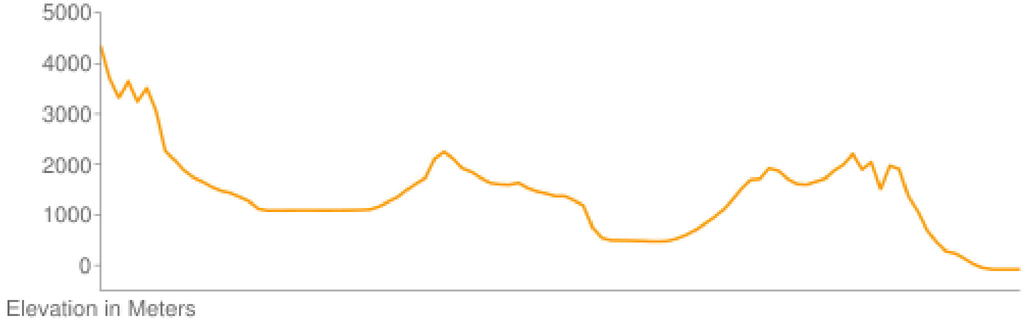
Bank of America Center

25 min (13 mi) Limited

Fastest route now due to traffic conditions

Other Useful Maps/Location APIs

- **Elevation API:** Returns elevation data for a location on earth
  - https://developers.google.com/maps/documentation/elevation/overview

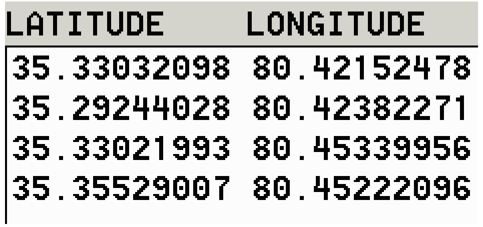


- **Roads API:**
  - https://developers.google.com/maps/documentation/roads/overview
  - snaps set of GPS coordinates to road user was likely travelling on (best fit)
  - Returns posted speed limits for any road segment (premium plan)
- **Time Zone API:** request time zone for location on earth
  - https://developers.google.com/maps/documentation/timezone/get-started

GPS Clustering & Analytics

Determining Points of Interest from GPS Location Sequences

- **Points of Interest:** Places where a person spends lots of time (e.g. home, work, café, etc)
- **Given a sequence GPS <longitude, latitude>** points, how to infer points of interest (PoI)
- **General steps:**
  - **Pre-process sequence of GPS points** (remove outliers, etc)
  - **Cluster points**
  - **Convert to semantic location**



LATITUDE	LONGITUDE
35.33032098	80.42152478
35.29244028	80.42382271
35.33021993	80.45339956
35.35529007	80.45222096

Step 1: Pre-Processing GPS Points (Remove Noise and Outliers)

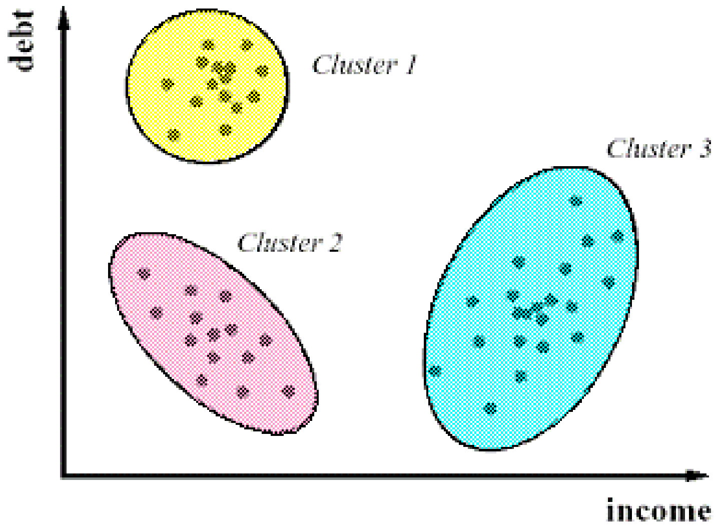
- **Remove low density points (few neighbors):**
  - i.e. places where little time was spent
  - E.g. radius of 20 meters, keep only clusters with at least 50 points
  - If GPS coordinates retrieved every minute, only considering places where you spent at least 50 minutes

- **Remove points with movement:**
  - GPS returns speed as well as <longitude, latitude> coordinates
  - If speed user is moving, discard that GPS point (Cannot be PoI)

- **Reduce data for stationary locations:**
  - When user is stationary at same location for long time, too many points generated (e.g. sitting at at chair)
  - Remove some points to speed up processing

Step 2: Cluster GPS Points

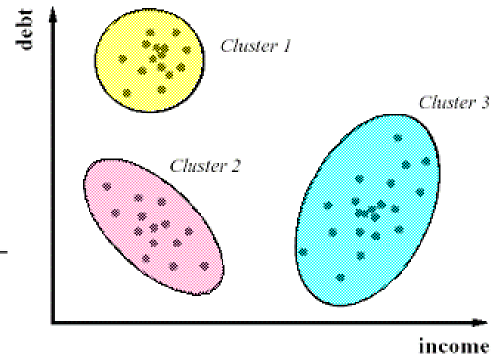
- **Cluster Analysis:** Group points



- Two main clustering approaches
  - K-means clustering
  - DBSCAN

K-Means Clustering

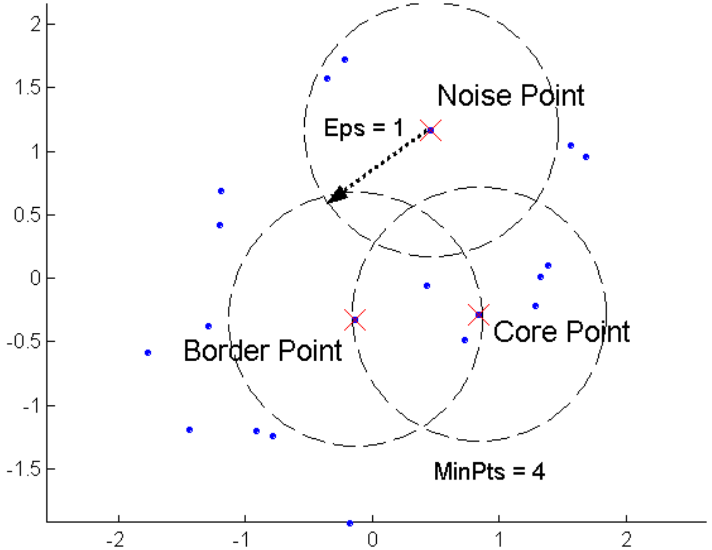
- Each cluster has a center point (centroid)
- Each point associated to cluster with closest centroid
- Number of clusters, *K*, must be specified
- Algorithm:



```
Select $K$ points as the initial centroids.
repeat
    Form $K$ clusters by assigning all points to the closest centroid.
    Recompute the centroid of each cluster.
until The centroids don't change
```

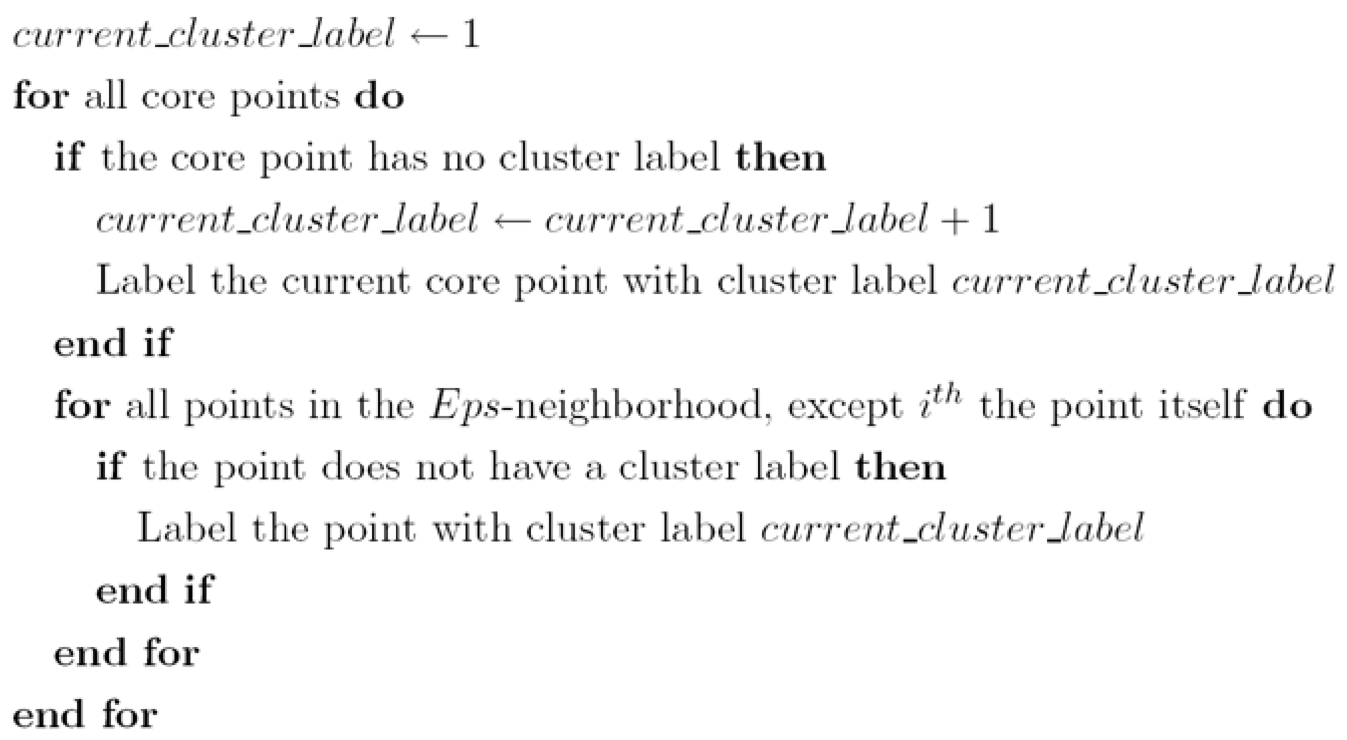
DBSCAN Clustering

- Density-based clustering
- **Density:** Number of points within specified radius (Eps)
- **Core points:** has > minPoints density
- **Border point:** has < minPoints density but within neighborhood of core point
- **Noise point:** not core point or border point



DBSCAN Algorithm

- Eliminate noise points
- **Cluster remaining points**



```
$current\_cluster\_label \leftarrow 1$
for all core points do
  if the core point has no cluster label then
    $current\_cluster\_label \leftarrow current\_cluster\_label + 1$
    Label the current core point with cluster label $current\_cluster\_label$
  end if
  for all points in the $Eps$-neighborhood, except $i^{th}$ the point itself do
    if the point does not have a cluster label then
      Label the point with cluster label $current\_cluster\_label$
    end if
  end for
end for
```

Number 1 cluster per cluster point 1.. N

Assign border points to closest cluster

Converting Clusters to Semantic Locations

- Can simply call reverse geocoding or Google Places on the centroid of the clusters

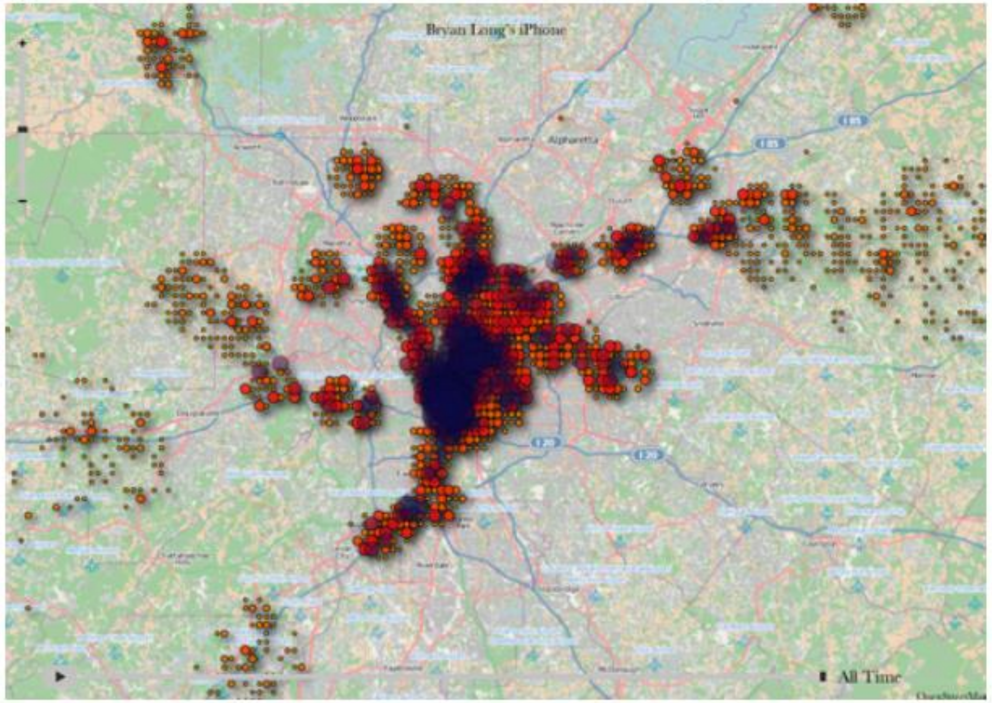
- Determining work? Cluster where user spends longest time most of the time (9-5pm)

- Determining home? Cluster where user spends most time 6pm – 6am

Visualizing Points of Interests visited

Visualizing Points of Interest (PoIs)

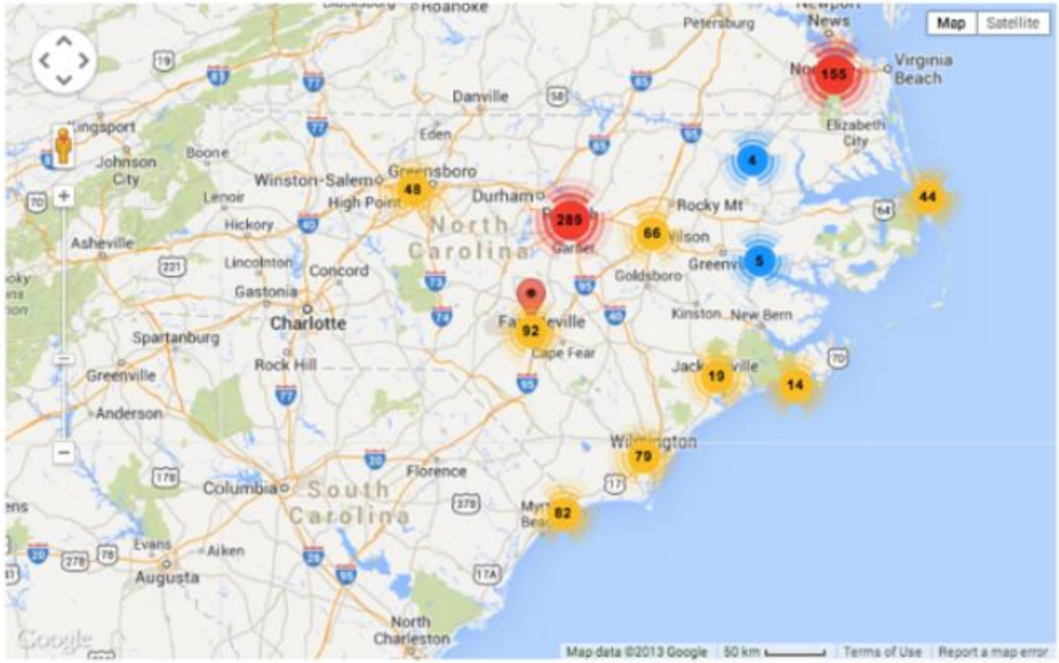
- Option 1:
  - Show a point for each location you visited?



Credit: Deepak Ganesan

Visualizing Points of Interest

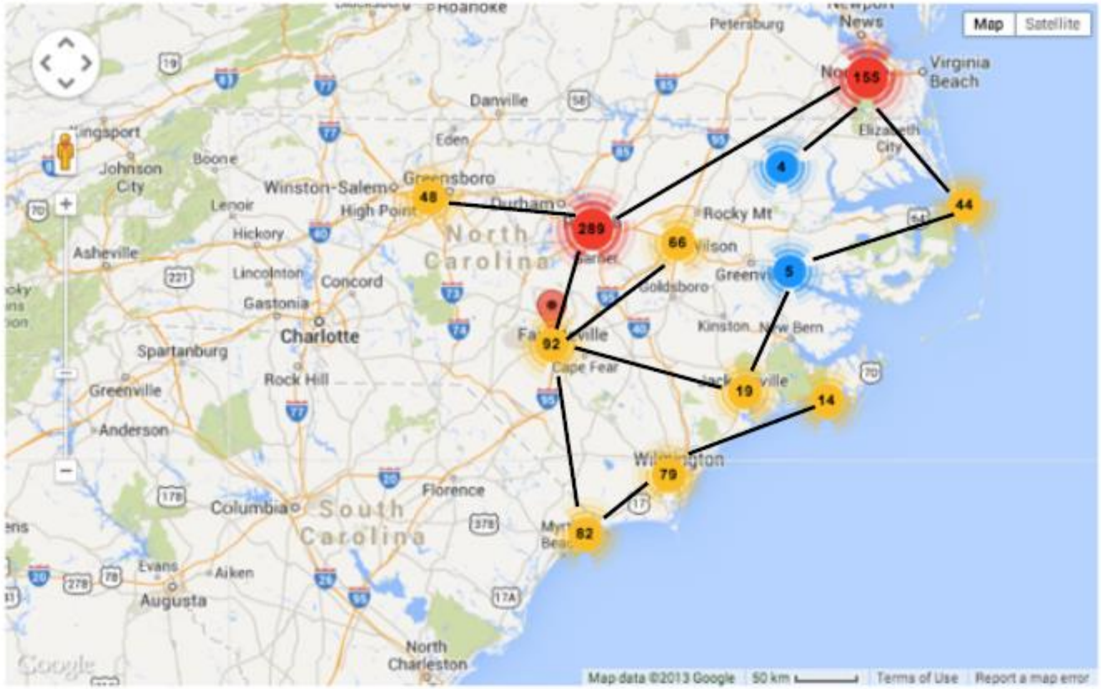
- Option 2:
  - Show a cluster for significant locations.



Credit: Deepak Ganesan

Visualizing Points of Interest

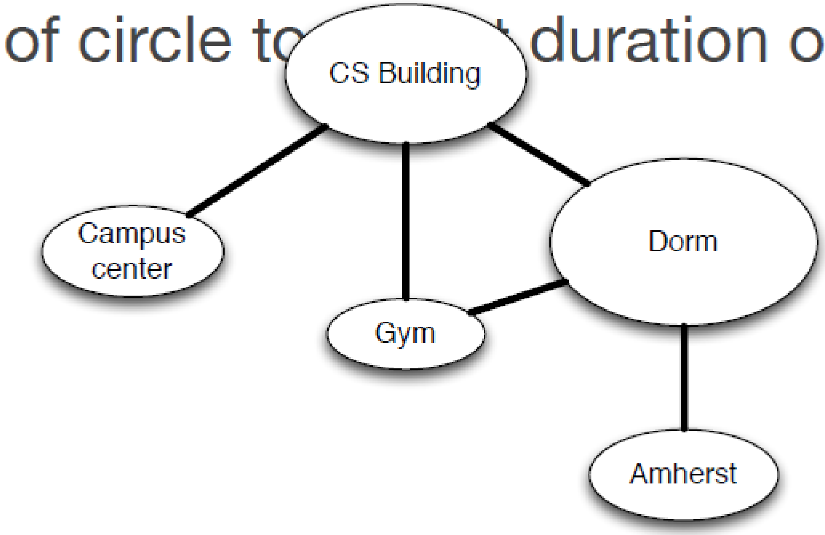
- Option 3:
  - Connect the clusters with lines



Credit: Deepak Ganesan

Visualizing Points of Interest

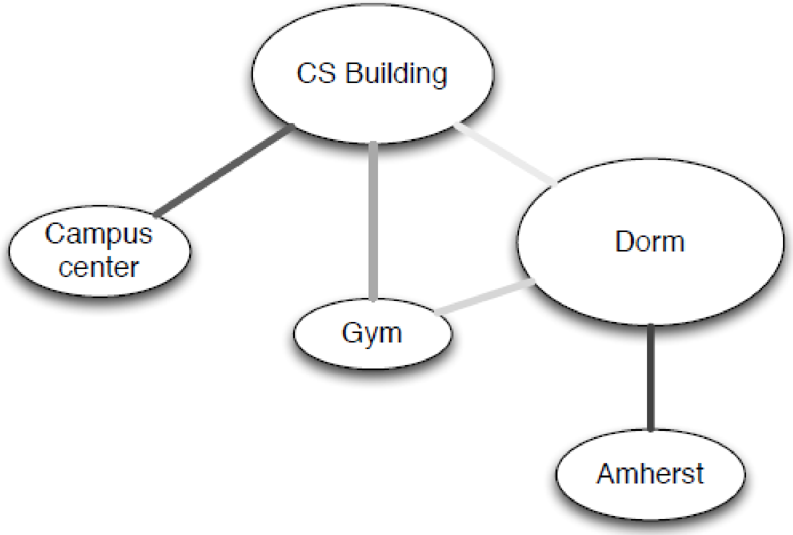
- Option 4
  - Show “semantic locations” instead of co-ordinates
  - Use size of circle to [illegible] duration of stay



Credit: Deepak Ganesan

Visualizing Points of Interest

- Option 5
  - Show semantic locations with time-of-day encoded in line opacity/saturation.



Credit: Deepak Ganesan

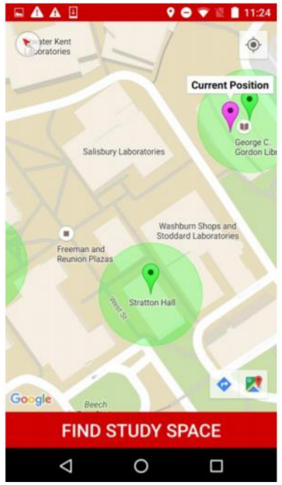
Some Interesting Location-Aware Apps

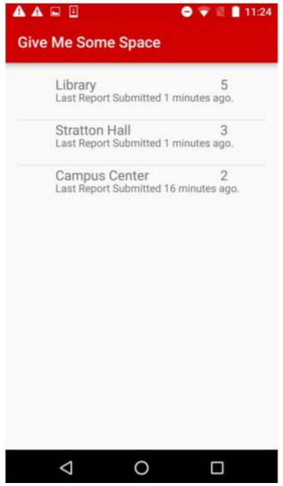
Location-Aware Final Projects from CS 4518 (Undergraduate offering)

- **Ground rules:**
  - Apps must use mobile, location or sensors
  - Try to solve problems of benefit to WPI community

- More than half of apps used location.

- **Give me some space:** Bianchi, Chow, Martinez ’16
  - Find available study spaces on campus during exam week
  - Set up geoFences at study locations, count users in/out





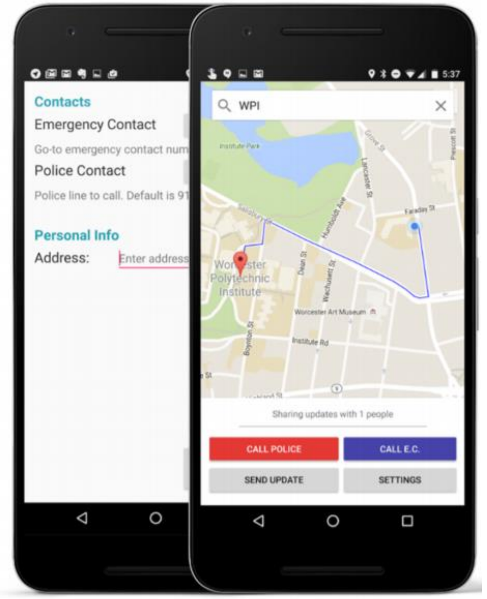
Library 5
Last Report Submitted 1 minutes ago.

Stratton Hall 3
Last Report Submitted 1 minutes ago.

Campus Center 2
Last Report Submitted 16 minutes ago.

Location-Aware Ideas from Previous Offerings

- **HomeSafe:** Nickerson, Feeley, Faust ’16
  - Safety app
  - Automatically sends message to users’ subscribers when they get home safely



Contacts

Emergency Contact

Go-to emergency contact num

Police Contact

Police line to call. Default is 91

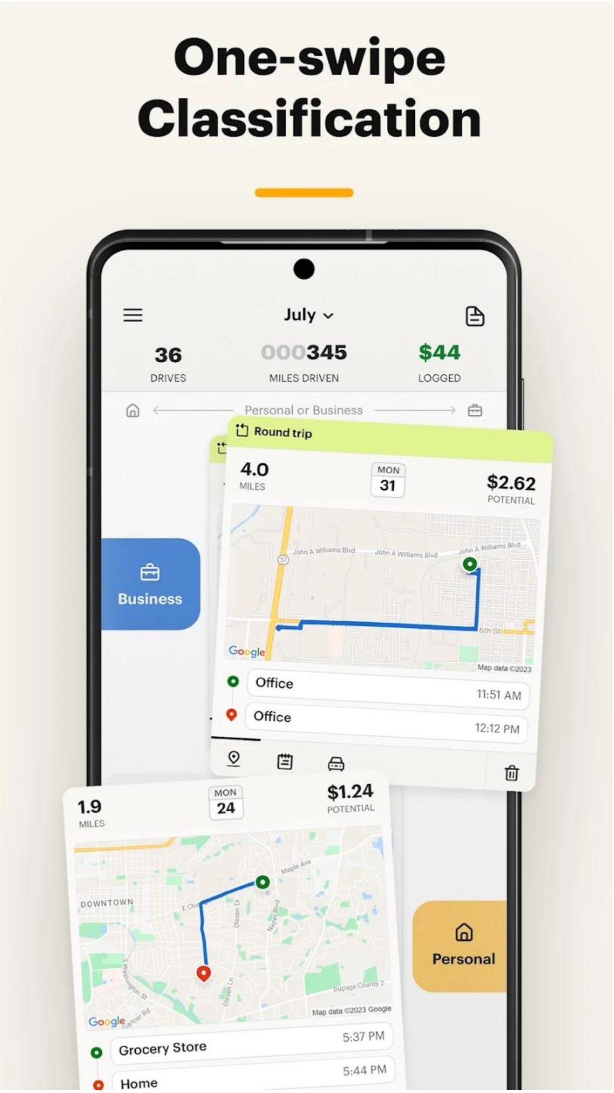
Personal Info

Address: Enter address

MileIQ

- **The Problem:** Mileage tracking is useful but a burden.
    - IRS deductions on taxes
    - Some companies reimburse employees for mileage,
- Passively, automatically tracks business mileage, IRS compliant
- Swipe right after drive to indicate it was a business trip
- A real existing business (https://mileiq.com)
- Project idea? Implement some of this functionality

- **What Android modules utilized? For what?**
- **What stats to decide if this is tackling important problem?**



References

- Android Nerd Ranch, 5th edition
- Google Android Tutorials