

# CS 4518 Mobile and Ubiquitous Computing

## Lecture 10: Location-Aware Computing

**Emmanuel Agu**



# Reminder: Final Project

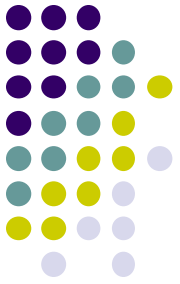


- 1-slide from group next Monday (2/6):
  - 2/40 of final project grade
- Slide should contain 3 bullets
  - 1. Problem you intend to work on**
    - Solve WPI/societal problem (e.g. walking safe at night)
    - Use at least location, 1 sensor or camera
    - If games, must gamify solution to real world problem
  - 2. Why this problem is important**
    - E.g. 37% of WPI students feel unsafe walking home
  - 3. Summary of envisioned mobile app (?) solution**
    1. E.g. Mobile app automatically texts users friends when they get home at night
- Can bounce ideas of me (email, or in person)
- Can change idea any time

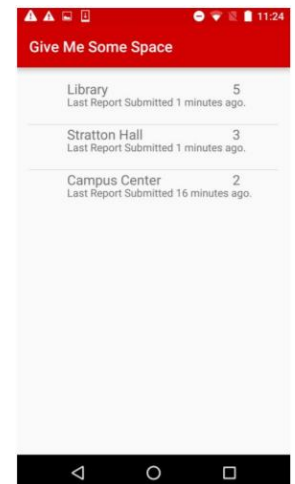


# Location-Aware Apps from Past Offerings

# Location-Aware Ideas from Previous Offerings



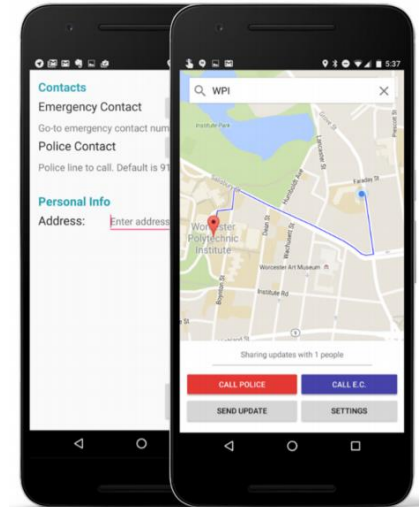
- **Ground rules:**
  - Apps must use mobile, location or sensors
  - Try to solve problems of benefit to WPI community
- More than half of apps used location.
- **Give me some space:** Bianchi, Chow, Martinez '16
  - Find available study spaces on campus during exam week
  - Set up geoFences at study locations, count users in/out



# Location-Aware Ideas from Previous Offerings



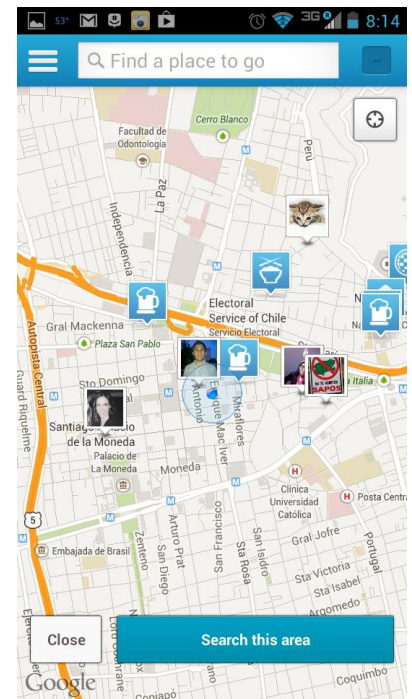
- **HomeSafe:** Nickerson, Feeley, Faust '16
  - Safety app
  - Automatically sends message to users' subscribers when they get home safely



# 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
- Above apps require first determining user's location





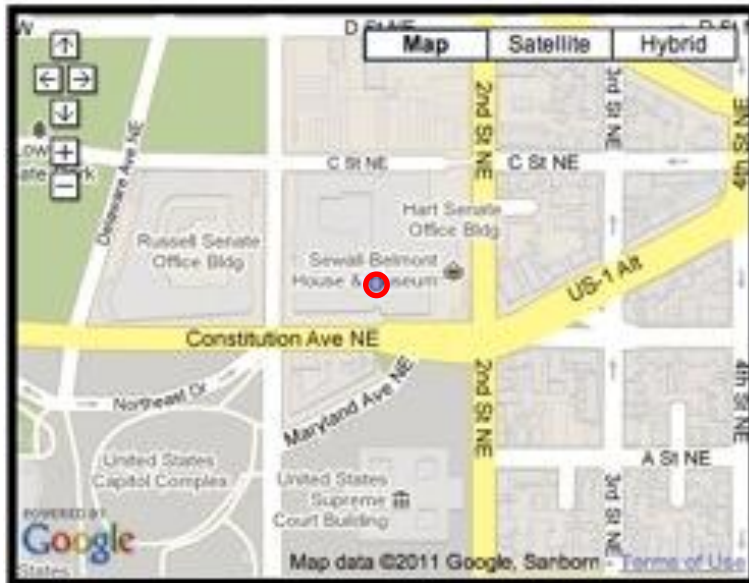
# Determining User Location on Smartphones



# Location Tracking on Smartphones

- **Outdoors:** Uses GPS (More accurate)
- **Indoors:** WiFi or cell tower signals (Location fingerprinting, less accurate)

GPS



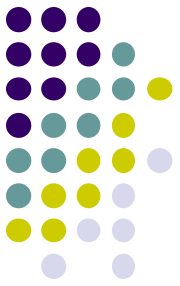
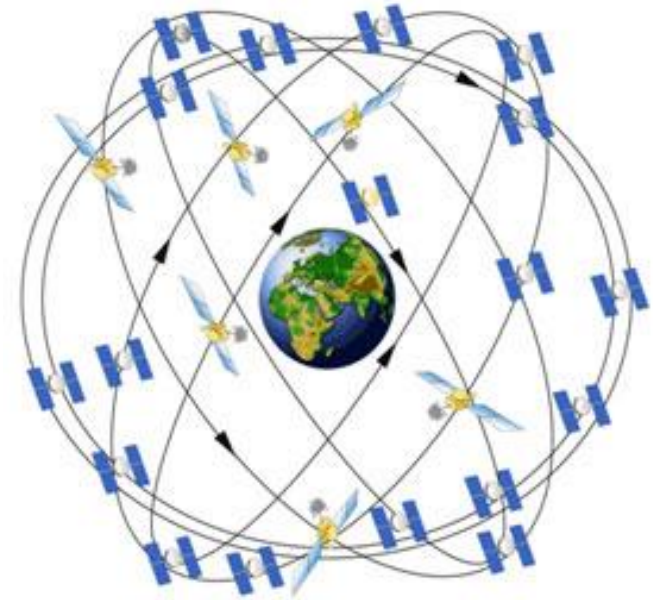
Wi-Fi





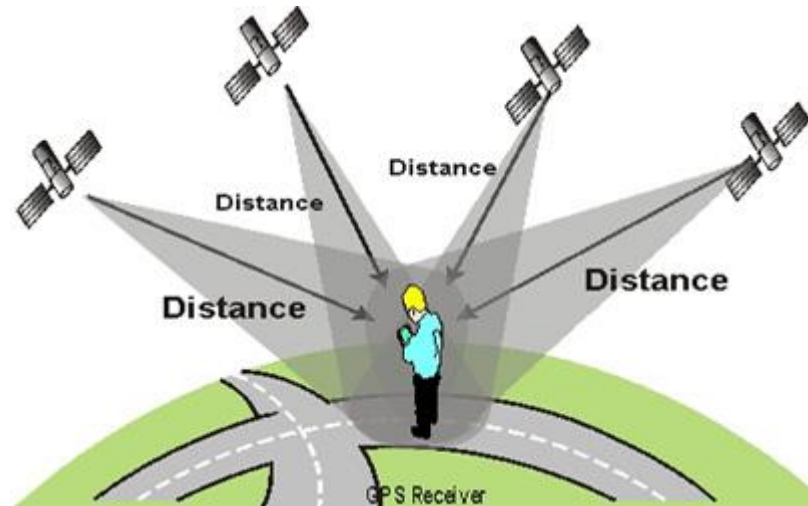
# Global Positioning System (GPS)

- 27 satellites orbiting earth
- **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

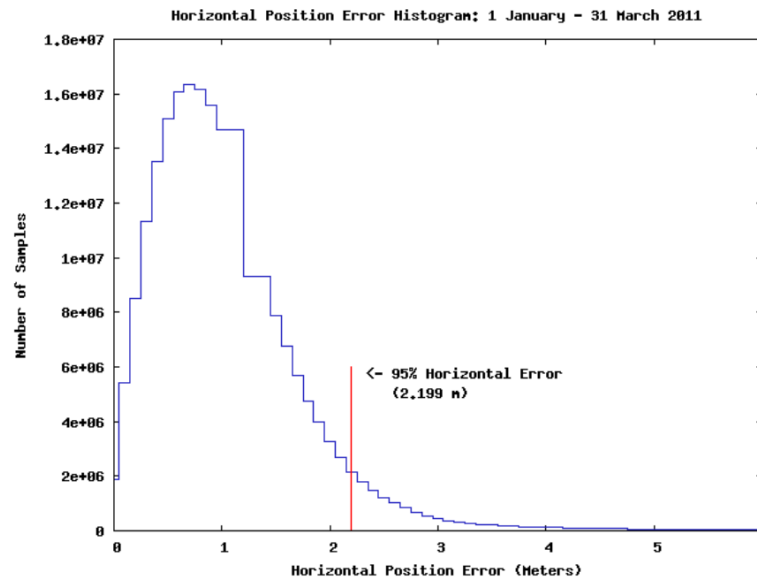


# GPS User Segment

- **Triangulation:** GPS receiver calculates user's position by comparing roundtrip delay of signals to multiple satellites at known positions
- Accuracy within 5 - 10 meters (16-32 feet)



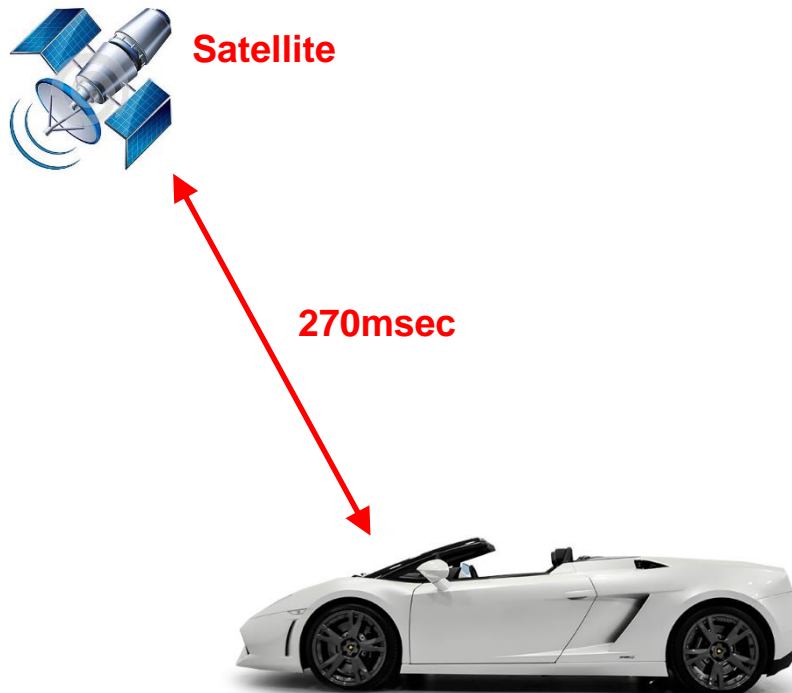
<http://adamswalk.com/gpx-2/>



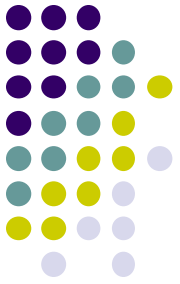
# Determining User Location



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

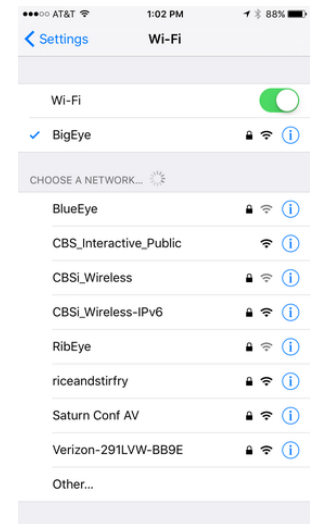
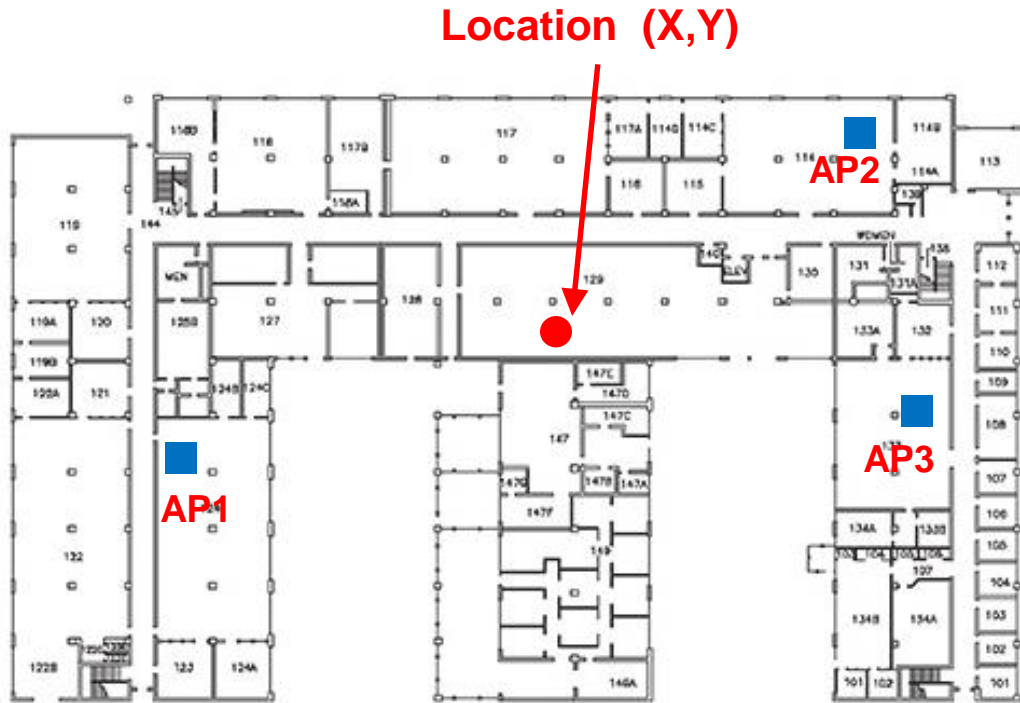


# WiFi Location Fingerprinting



- **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



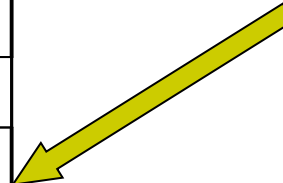
- **WiFi Location fingerprinting:** Infer device's location based on combination of Wi-Fi access points seen + Signal Strengths

# 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
...	...	...	...	...	...
<b>220</b>	<b>355</b>	-	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)??**

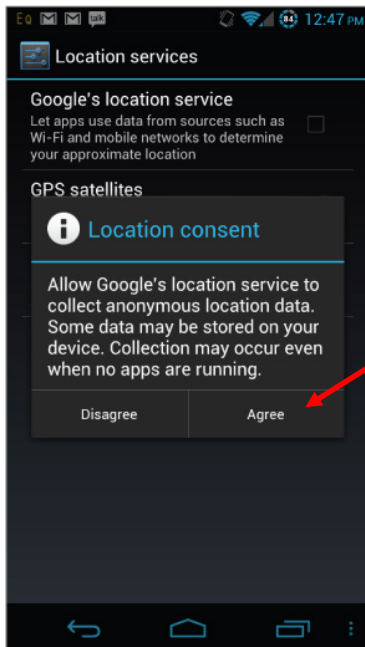
- ◆ Inference Algorithms
  - Min. Threshold
  - Euclidean Dist.
  - Joint Probability
  - Bayesian Filters

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 to third party repositories (e.g. Wigle.net) or Google
- Also called **war driving**
- Can record cell tower signal strength instead of AP



**Google gathers Location, AP seen Data if you consent**

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 Strength**

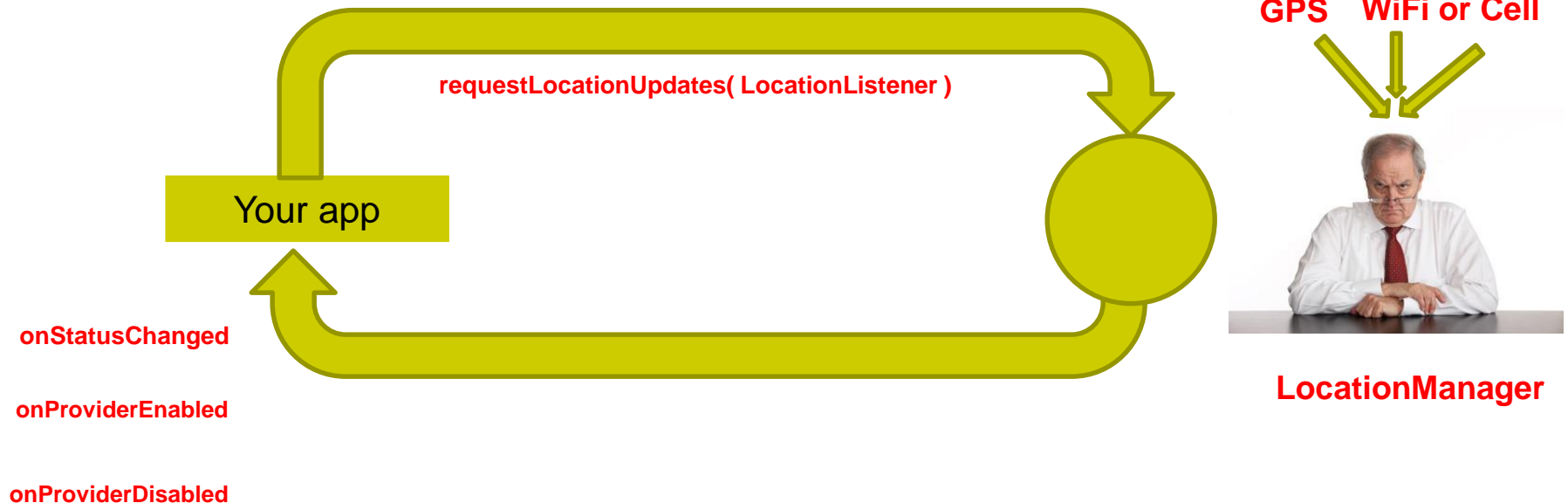
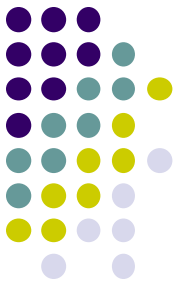


# Location Sensing in Android Apps

# Google Location APIs

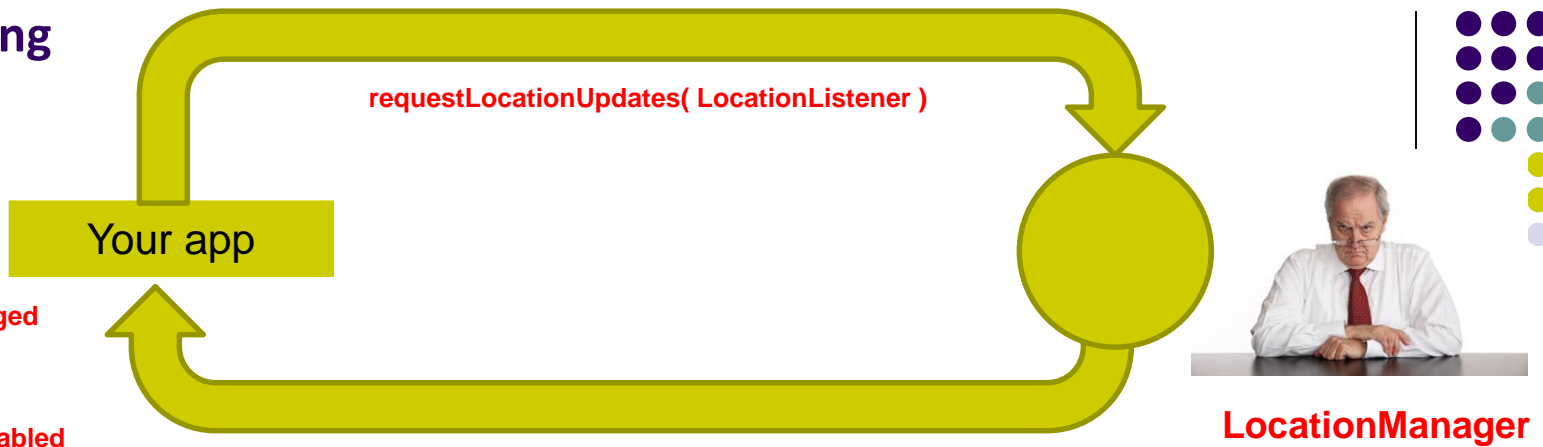
<https://developer.android.com/guide/topics/location/strategies.html>

- Location API is now part of Google Play Services (newer!)
- Older Android framework location APIs (**android.location**)
  - Used by most books, online sources. We will use that
  - <http://developer.android.com/guide/topics/location/strategies.html>
- **LocationManager:**
  - Android module receives location updates from GPS, WiFi, etc
  - App registers/requests location updates from LocationManager





# Requesting Location Updates



```
// Acquire a reference to the system Location Manager
```

```
LocationManager locationManager = (LocationManager) this.getSystemService(Context.LOCATION_SERVICE);
```

```
// Define a listener that responds to location updates
```

```
LocationListener locationListener = new LocationListener() {  
    public void onLocationChanged(Location location) {  
        // Called when a new location is found by the network location provider.  
        makeUseOfNewLocation(location);  
    }  
};
```

Create listener for Location info

```
    public void onStatusChanged(String provider, int status, Bundle extras) {}  
    public void onProviderEnabled(String provider) {}  
    public void onProviderDisabled(String provider) {}  
};
```

Callback methods called by Location manager (e.g. when location changes)

```
// Register the listener with the Location Manager to receive location updates
```

```
locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0, 0, locationListener);
```

Type of location Provider (e.g. cell tower and Wi-Fi based)

Listener that receives callbacks



# Requesting User Permissions

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

- Need smartphone owner's permission to use their GPS

```
<manifest ... >
  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
  ...
  <!-- Needed only if your app targets Android 5.0 (API level 21) or higher. -->
  <uses-feature android:name="android.hardware.location.gps" />
  ...
</manifest>
```

- **ACCESS\_FINE\_LOCATION:** GPS
- **ACCESS\_COARSE\_LOCATION:** WiFi or cell towers



# Getting Cached Copy of Location (Fast)

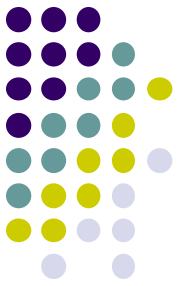
<https://developer.android.com/guide/topics/location/strategies.html>

- Getting current location may take a while
- Can choose to use location cached (possibly stale) from Location Manager

```
String locationProvider = LocationManager.NETWORK_PROVIDER;  
// Or use LocationManager.GPS_PROVIDER  
  
Location lastKnownLocation = locationManager.getLastKnownLocation(locationProvider);
```

# Stopping Listening for Location Updates

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



- Location updates consume battery power
- Stop listening for location updates whenever you no longer need

```
// Remove the listener you previously added  
locationManager.removeUpdates(locationListener);
```



# **Distance Travelled Updates using Services**

## **Example from Head First Android**

# Example: Odometer (Distance Travelled) updates as a Services

(Ref: Head First Android pg 541)



- **Services:** long running background processes, no UI
- May want background service (a module in our app) to continuously retrieve location updates from LocationManager, forward our Activity updates
- Ref: Head First Android pg 541
  - Example of using a Service
  - Nice Example app using Odometer Service
  - Tracks distance travelled
  - Gets, displays distance travelled every 10 secs

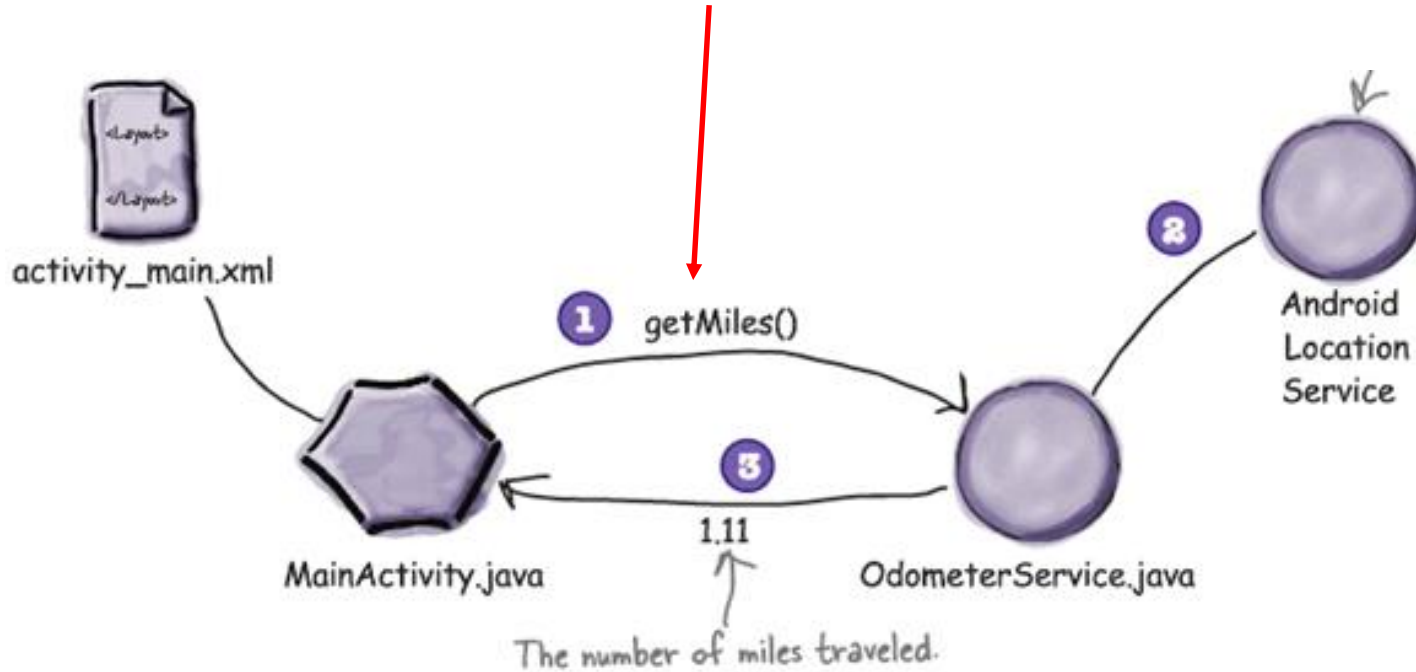


# Example: Odometer (Distance Travelled) updates as a Services

(Ref: Head First Android pg 541)



- Example odometer app that tracks distance travelled
- `getMiles()`, displays distance travelled every 10 seconds



**Study this example!!!**



# 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

Latitude: 37.422005 Longitude: -122.084095

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

- **Android Geocoding API:** access to **geocoding** and **reverse geocoding** services using HTTP requests



# Google Places API 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	<a href="#">Crumbs Bake Shop</a> ☆	<a href="http://www.crumbs.com">www.crumbs.com</a> - (212) 480-7500 - 52 reviews
B	<a href="#">Sugar Sweet Sunshine</a> ☆	<a href="http://www.sugarsweetsunshine.com">www.sugarsweetsunshine.com</a> - (212) 995-1960 - 255 reviews
C	<a href="#">Babycakes Nyc</a> ☆	<a href="http://www.babycakesnyc.com">www.babycakesnyc.com</a> - (212) 677-5047 - 172 reviews
D	<a href="#">Billy's Bakery</a> ☆	<a href="http://www.billysbakerynyc.com">www.billysbakerynyc.com</a> - (212) 647-9956 - 219 reviews
E	<a href="#">Magnolia</a> ☆	<a href="http://www.magnoliabakery.com">www.magnoliabakery.com</a> - (212) 462-2572 - 1055 reviews
F	<a href="#">Tribeca Treats</a> ☆	<a href="http://www.tribecatreats.com">www.tribecatreats.com</a> - (212) 571-0500 - 63 reviews
G	<a href="#">Butter Lane Cupcakes</a> ☆	<a href="http://www.butterlane.com">www.butterlane.com</a> - (212) 677-2880 - 78 reviews

Visit our website Sponsored

More results near **New York, NY** »

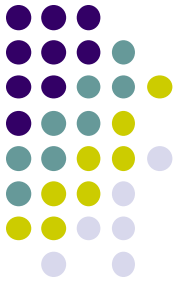
- **On-device caching:** Can cache places data locally on device to avoid roundtrip delays on future requests



# Google Places

- **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.
- **E.g:** name of place, address, geographical location, place ID, phone number, place type, website URL, etc.
- Compliments geographic-based services offered by Android location services

# Sample Place 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

hospital

insurance\_agency

jewelry\_store

laundry

lawyer

library

liquor\_store

local\_government\_office

locksmith

lodging

meal\_delivery

meal\_takeaway

mosque

movie\_rental

movie\_theater

moving\_company

museum

night\_club

painter

park

city\_hall

clothing\_store

convenience\_store

courthouse

dentist

department\_store

doctor

electrician

electronics\_store

embassy

establishment (deprecated)

finance (deprecated)

fire\_station

florist

food (deprecated)

funeral\_home

furniture\_store

gas\_station

general\_contractor (deprecated)

grocery\_or\_supermarket

gym

hair\_care

hardware\_store

health (deprecated)

hindu\_temple

home\_goods\_store

physiotherapist

place\_of\_worship (deprecated)

plumber

police

post\_office

real\_estate\_agency

restaurant

roofing\_contractor

rv\_park

school

shoe\_store

shopping\_mall

spa

stadium

storage

store

subway\_station

synagogue

taxi\_stand

train\_station

transit\_station

travel\_agency

university

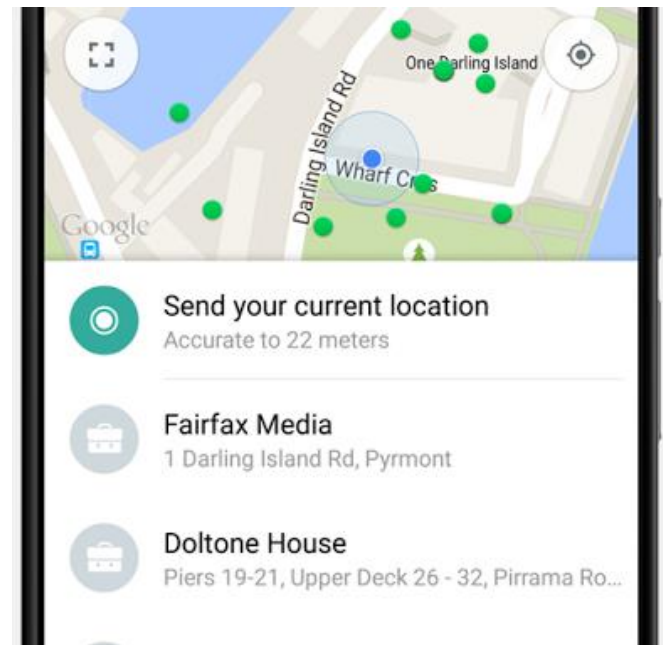
veterinary\_care

zoo



# Google Places API Overview

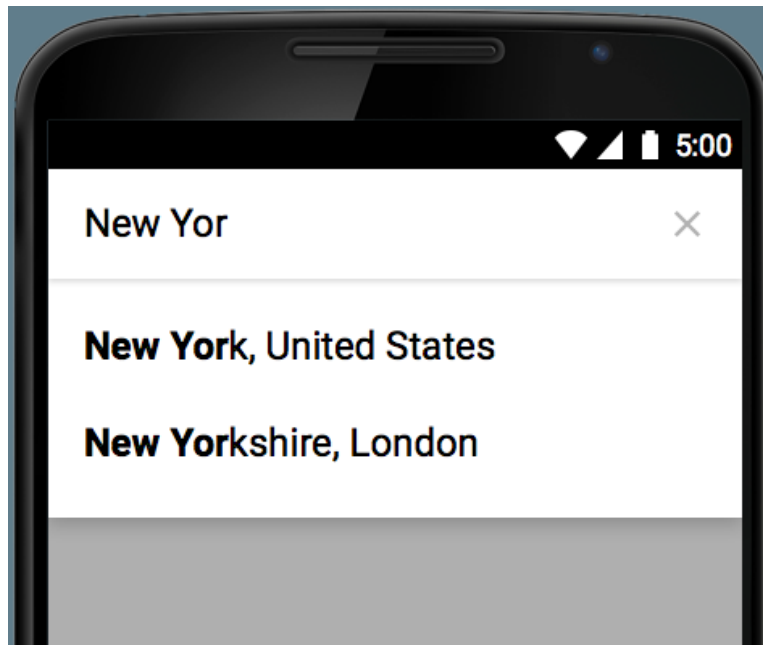
- **Use Place picker UI:** allows users select place from “possible place” on a map
- **Get current place:** place where device is last known to be located
  - Returns **list** of likely places + likelihood device is in that place





# Google Places API Overview

- **Autocomplete:** queries the location database as users type, suggests nearby places matching letters typed in





# Learning Google Places API

- Official Google Places website is “decent”, up to date:
  - <https://developers.google.com/places/>
- Two great references:
  - a) Getting started with Google Places API  
<https://developers.google.com/places/android-api/start>
  - b) Tutorial by Paul Trebilcox-Ruiz may be more readable:
    - <http://code.tutsplus.com/articles/google-play-services-using-the-places-api--cms-23715>



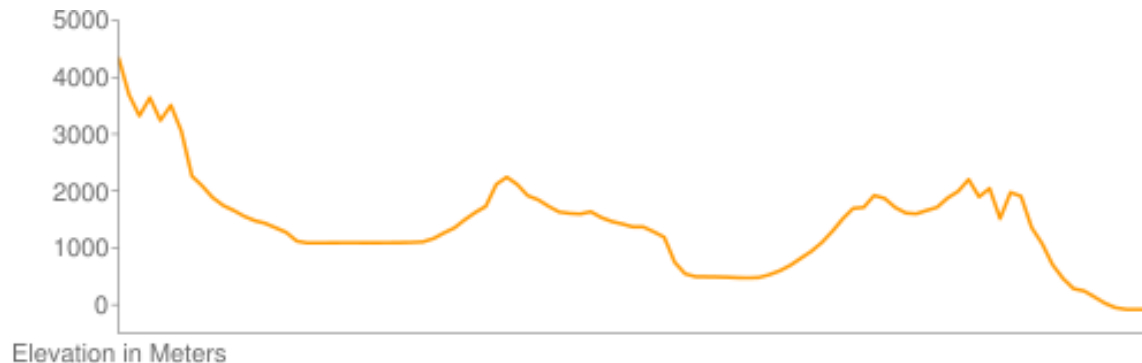
# Other Useful Google Maps/Location APIs





## Other Maps/Useful Location APIs

- **Maps Directions API:** calculates directions between locations (walking, driving) as well as public transport directions
- **Distance Matrix API:** Calculate travel time and distance for multiple destinations
- **Elevation API:** Query locations on earth for elevation information, calculate elevation changes along routes



# Other Useful Maps/Location APIs



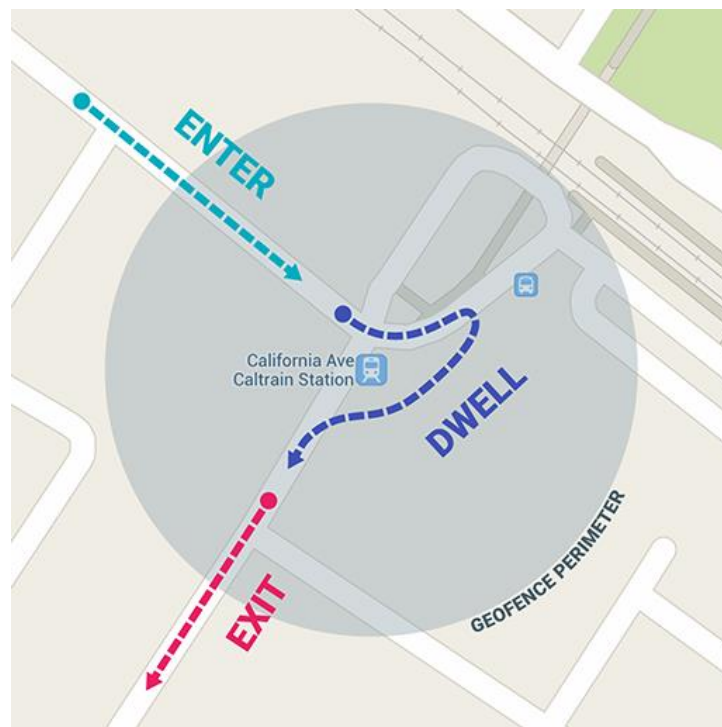
- **Roads API:**
  - snaps set of GPS coordinates to road user was likely travelling on (best fit)
  - Returns posted speed limits for any road segment (premium plan)
- **Time Zone API:** request time zone for location on earth

# GeoFencing

<https://developer.android.com/training/location/geofencing.html>



- **Geofence:** Sends alerts when user is within a certain radius to a location of interest
- Can be configured to send:
  - **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



# GeoFencing

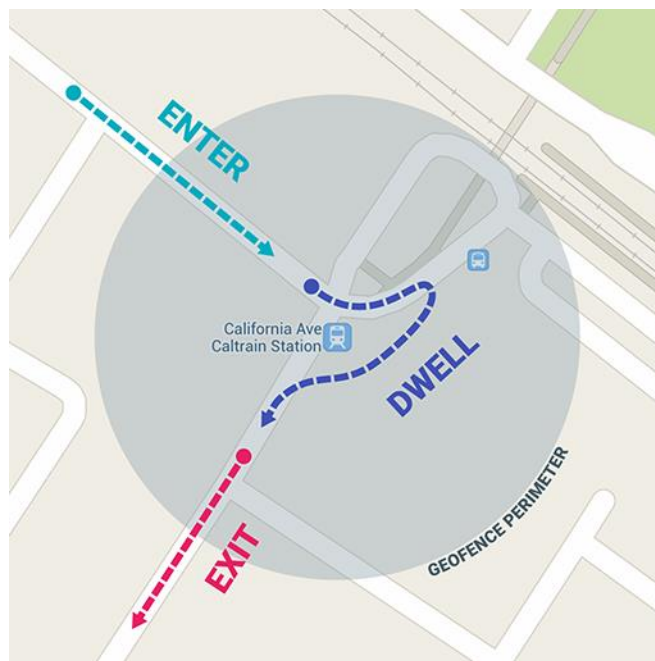
<https://developer.android.com/training/location/geofencing.html>



- **Great reference:**

- How to work with GeoFences on Android by Tin Megali

<https://code.tutsplus.com/tutorials/how-to-work-with-geofences-on-android--cms-26639>



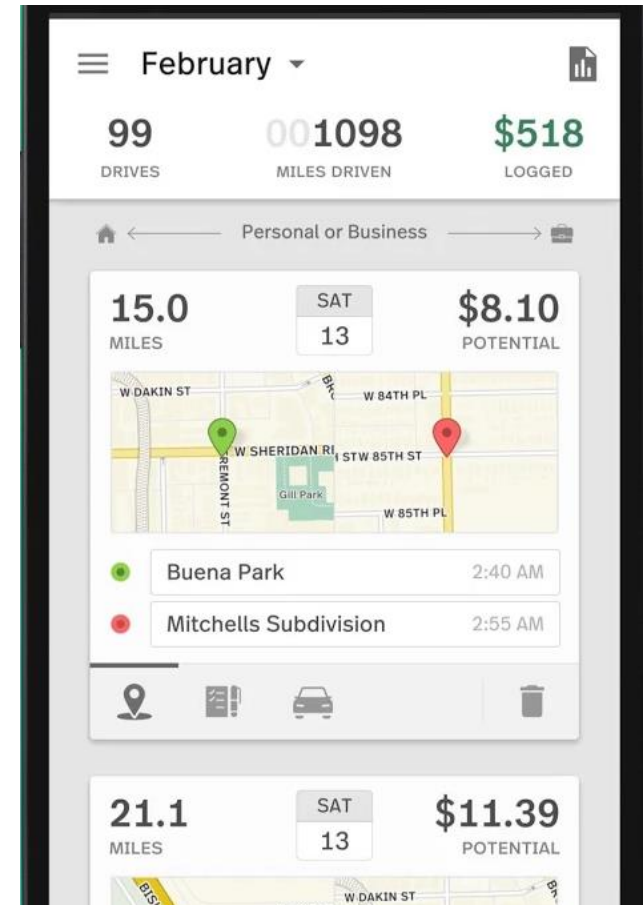


# Some Interesting Location-Aware Apps



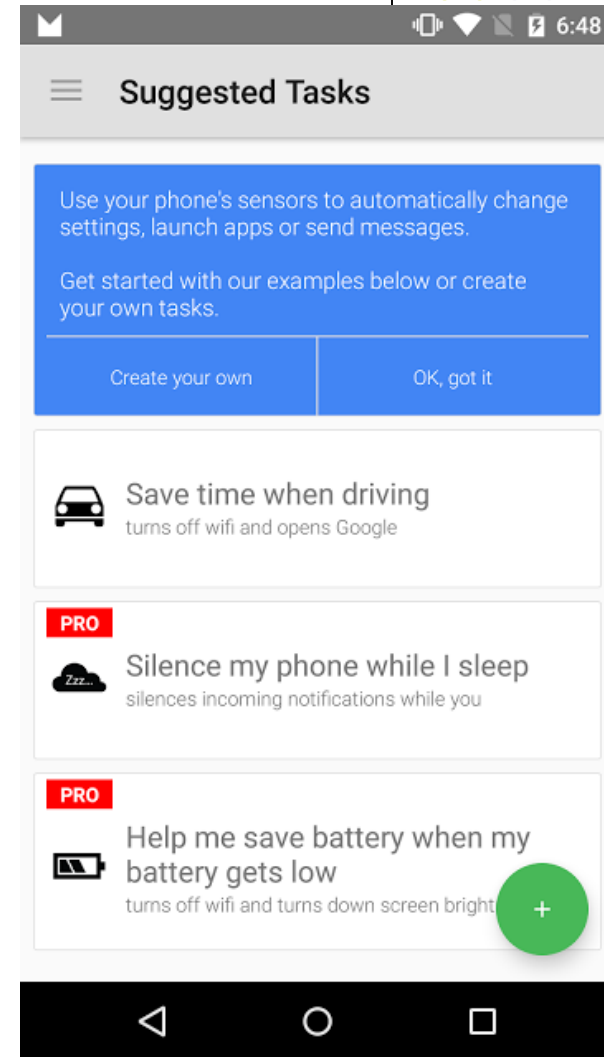
# MileIQ

- **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
- Project idea? Implement some of this functionality
- **How Android modules? For what?**
- **What stats to decide if this is tackling important problem?**



# Trigger

- Use geofences, NFC, bluetooth, WiFi connections, etc to set auto-behaviors
  - Battery low -> turn off bluetooth + auto sync
  - Silence phone every morning when you get to work
  - Turn off mobile data when you connect to your home WiFi
  - Silence phone and set alarm once I get into bed
  - Use geofence for automatic foursquare checkin
  - Launch maps when you connect to your car's bluetooth network
- Project idea? Implement subset of these features
- **What triggers would be useful for a WPI student?**





# References

- John Corpuz, 10 Best Location Aware Apps
- Liane Cassavoy, 21 Awesome GPS and Location-Aware Apps for Android,
- Head First Android
- Android Nerd Ranch, 2<sup>nd</sup> edition
- Busy Coder's guide to Android version 6.3
- CS 65/165 slides, Dartmouth College, Spring 2014
- CS 371M slides, U of Texas Austin, Spring 2014