

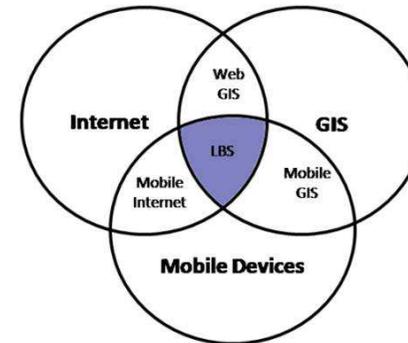
# ANDROID APPS DEVELOPMENT FOR MOBILE AND TABLET DEVICE (LEVEL II)

## Lecture 3: Location Based Services

Peter Lo

## What is LBS?

- **Location Based Service (LBS)** is an information system driven by the ability of the central system to detect the geographical position of the mobile devices.



4T025-2-A @ Peter Lo 2014

2

## Introduction to LBS

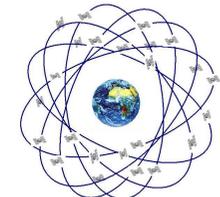
- Location Based Services are used in a variety of situations, such as commercial, entertainment, emergency, health, work, personal life, etc.
- Examples:
  - Locate the nearest bank, restaurant, gas station, hotel, golf course, hospital, police station, etc.
  - Provide transportation information on how to go from 'here' to 'there'.
  - Social networking is used to locate and reach events, friends and family members.

4T025-2-A @ Peter Lo 2014

3

## What is GPS?

- Created by DOD-USA under the name NAVSTAR (Navigation System for Timing and Ranging) but it is commonly known as **Global Positioning System (GPS)**.
- The system's backbone consists of 27 Earth-orbiting satellites (24 in operation and 3 in stand-by mode)
- Each satellite circles the globe at about 12,000 miles, making two complete rotations every day.
- The disposition of orbiting satellites is set so that at any time there are at least four of them in range to any point on earth.



4T025-2-A @ Peter Lo 2014

4

## How the GPS Works?

- The three circles intersect on the point over Central America.
- The actual location is: San Jose, Costa Rica.

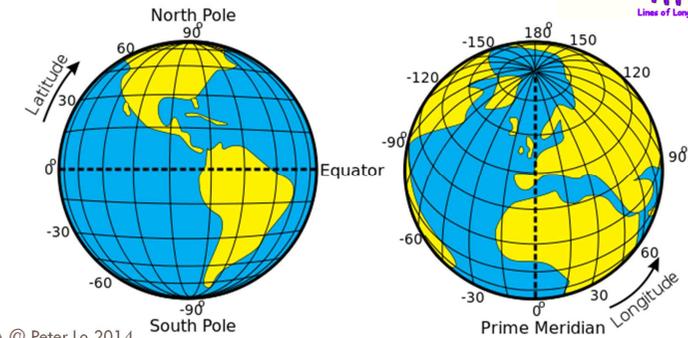
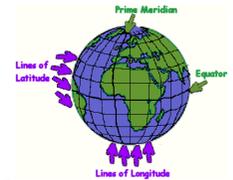


4T025-2-A @ Peter Lo 2014

5

## Latitude and Longitude

- Latitude in GPS-Decimal notation:
  - ▣ +90.00000 (North) to -90.00000 (South)
- Longitude GPS-Decimal notation:
  - ▣ +180.00000 (East) to -180.00000 (West)



4T025-2-A @ Peter Lo 2014

6

## Main Component of LBS

- The Location-Based API includes two packages
  - ▣ Google Map API ([com.google.android.maps](http://com.google.android.maps))
  - ▣ Location API ([android.location](http://android.location))
- They provide an initial look at the support in the Android platform for building location-based services.
- These API work over the internet to invoke services from Google servers

Location-Based Service

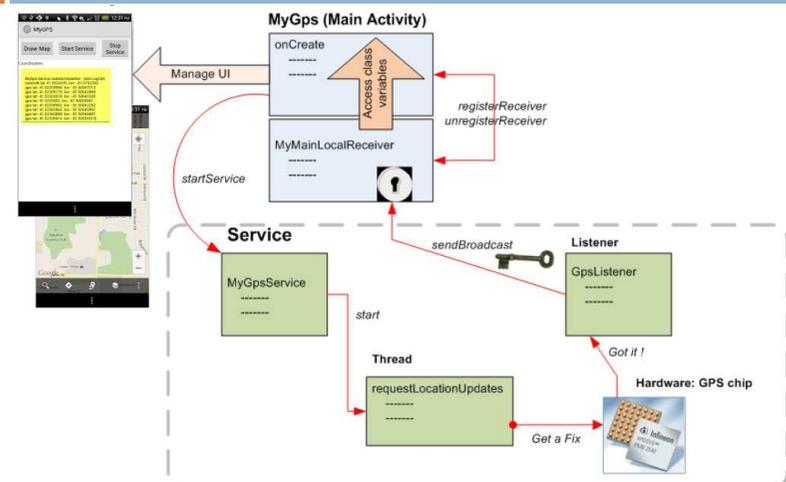
Location API  
android.location

Google Map API  
com.google.android.map

4T025-2-A @ Peter Lo 2014

7

## Android GPS Algorithm



4T025-2-A @ Peter Lo 2014

8

## Location Manager

- This class provides access to the system location services.
- These services allow applications
  - ▣ To obtain periodic updates of the device's geographical location,
  - ▣ Fire an application-specified Intent when the device enters the proximity of a given geographical location.
- You do not instantiate this class directly; instead, retrieve it through
  - ▣ `Context.getSystemService(Context.LOCATION_SERVICE)`

## Location Listener

- This class used for receiving notifications from the **LocationManager** when the location has changed.
- These methods are called if the **LocationListener** has been registered with the location manager service using the method:
  - ▣ `requestLocationUpdates(Provider, minTime, minDistance, LocationListener)`

## Define Location Listener and Request Location Updates

The diagram illustrates the process of defining a location listener and requesting updates. It consists of a code block with several callouts:

- Define the criteria how to select the location provider:** Points to the `Criteria` object creation and the `getBestProvider` call.
- Acquire a reference to the system Location Manager:** Points to the `getSystemService` call.
- Define a listener that responds to location updates:** Points to the `LocationListener` implementation.
- Called when a new location is found by the location provider:** Points to the `onLocationChanged` method.
- Register the listener with the Location Manager to receive location updates:** Points to the `requestLocationUpdates` call.

```

locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
criteria = new Criteria();
provider = locationManager.getBestProvider(criteria, false);

LocationListener locationListener = new LocationListener() {
    public void onLocationChanged(Location location) {
        // Make use of new location
    }

    public void onStatusChanged(String provider, int status, Bundle extras) {}

    public void onProviderEnabled(String provider) {}

    public void onProviderDisabled(String provider) {}
};

locationManager.requestLocationUpdates(provider, 0, 0, locationListener);
    
```

## Requesting User Permissions

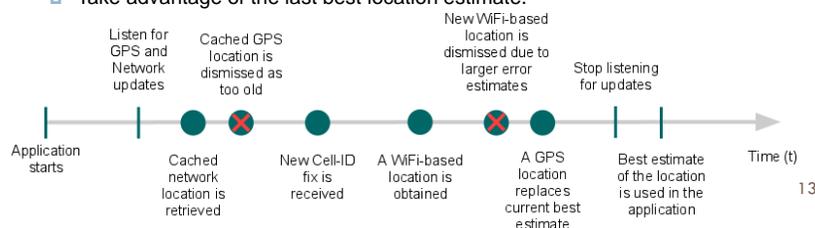
- In order to receive location updates from network provider or GPS provider, you must request user permission by declaring corresponding permission in your Android manifest file.

Permission	Provider
ACCESS_FINE_LOCATION	Allows the API to use the GPS to determine the device's location to within a very small area: <ul style="list-style-type: none"> <li>• NETWORK_PROVIDER</li> <li>• GPS_PROVIDER</li> </ul>
ACCESS_COARSE_LOCATION	Allows the API to use WiFi or mobile cell data (or both) to determine the device's location: <ul style="list-style-type: none"> <li>• NETWORK_PROVIDER</li> </ul>

Without these permissions, your application will fail at runtime when requesting location updates.

## Obtaining User Location with Best Performance

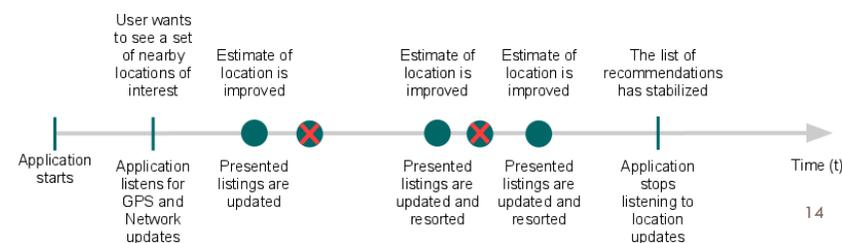
- To overcome the obstacles of obtaining a good user location while preserving battery power, you must define a consistent model that specifies how your application obtains the user location.
- The typical flow of procedures for obtaining the user location:
  - Start application.
  - Start listening for updates from desired location providers.
  - Maintain a "current best estimate" of location by filtering out new, but less accurate fixes.
  - Stop listening for location updates.
  - Take advantage of the last best location estimate.



13

## Helping the User decide on Where to Go

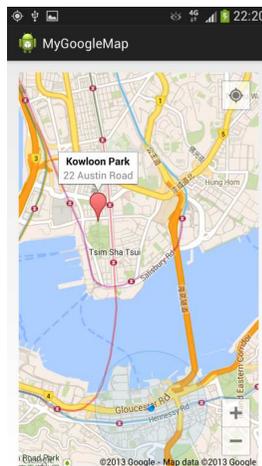
- You might be creating an application that attempts to provide users with a set of options about where to go.
- To accommodate such a flow, you might choose to:
  - Rearrange recommendations when a new best estimate is obtained
  - Stop listening for updates if the order of recommendations has stabilized



14

## Google Maps

- The Google Maps API for Android provides developers with the means to create apps with localization functionality.
- Google Maps API V2 was released at the end of 2012 and it introduced a range of new features including 3D, improved caching, and vector tiles.



15

## Google Maps Android API Getting Started

- Creating a new Android application that uses the Google Maps Android API v2 requires several steps.
- The overall process of adding a map to an Android application is as follows:
  - Download and configure the Google Play services SDK, which includes the Google Maps Android API.
  - Obtain an API key. You will need to register a project in the Google APIs Console, and get a signing certificate for your app.
  - Add the required settings in your application's manifest.
  - Add a map to your application.
  - Publish your application.

If you use the Google Maps Mobile SDK for Business you must download and configure the Google Maps Mobile SDK for Business static library.

16

## Step 1: Configure Google Play Services SDK

- Google Maps Android API v2 is distributed as part of the Google Play services SDK. You can download it via the Android SDK Manager.
- Summary of the steps you will need to take:
  - Install the Google Play services SDK.
  - Add Google Play services as an Android library project.
  - Reference the Google Play services in your app's project.
  - Edit your application's **AndroidManifest.xml** file, and add the following declaration within the `<application>` element.

```
<meta-data
    android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version" />
```

## Step 2: Get an Android Certificate and the Google Maps API Key

- Maps API keys are linked to specific certificate / package pairs, rather than to users or applications.
- Only need one key for each certificate, no matter how many users for an application.
- Apps that use the same certificate can use the same API key.
- Obtaining a key requires several steps:
  - Retrieve information about your application's certificate.
  - Register a project in the Google APIs Console and add the Maps API as a service for the project.
  - Request API keys.
  - Add your key to your application and begin development.

However, the recommended practice is to sign each of your applications with a different certificate and get a different key for each one

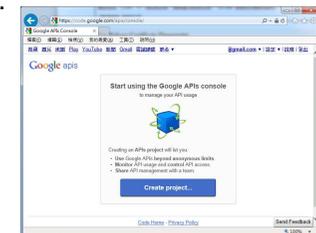
## Step 2A: Obtain your App's Certificate Information

- The Maps API key is based on a short form of your application's digital certificate, known as its **SHA-1 Fingerprint**.
- The fingerprint is itself unique, Google Maps uses it as a way to identify your application.
- To display the SHA-1 fingerprint for your certificate, first ensure that you are using the right certificate. You may have two certificates:
  - **Debug Certificate**
    - The Android SDK tools generate this certificate automatically when you do a "Debug" build from the command line, or when you build and run a project from Eclipse without exporting it as a released application.
    - Only use this certificate with apps that you're testing; do not attempt to publish an app that's signed with a debug certificate.
  - **Release Certificate**
    - The Android SDK tools generate this certificate when you do a "Release" build with either ant program or Eclipse.

```
keytool -list -v -keystore your_keystore_name -alias your_alias_name
```

## Step 2B: Create an API Project in Google APIs Console

- To create or modify a project for your application in the Google APIs Console and register for the Maps API.
  - Navigate to the Google APIs Console (<https://code.google.com/apis/console>).
  - In the list of services displayed in the center of the page, scroll down until you see Google Maps Android API v2.
  - This displays the Google Maps Android API Terms of Service. If you agree to the terms of service.



## Step 2C: Obtain a Google Maps API key

- If your application is registered with the Google Maps Android API service, then you can request an API key.
  - Navigate to your project in the Google APIs Console.
  - In the left navigation bar, click API Access.
  - In the resulting page, click Create New Android Key
  - In the resulting dialog, enter the SHA-1 fingerprint, then a semicolon, then your application's package name.
- The Google APIs Console responds by displaying Key for Android apps followed by a forty-character API key:

```
BB:0D:AC:74:D3:21:E1:43:67:71:9B:62:91:AF:A1:66:6E:44:5D:75;com.example.android.mapexample
```

```
AIzaSyBdVl-cTICSwYKrZ95SuvNw7dbMuDt1KGO
```

It's possible to register more than one key per project

## Step 2D: Add the API Key to Application

- The Maps API reads the key value and passes it to the Google Maps server, which then confirms that you have access to Google Maps data.
  - In AndroidManifest, add the following element as a child of the `<application>` element, by inserting it just before the closing tag `</application>`, then save and re-build your application.

```
<meta-data
  android:name="com.google.android.maps.v2.API_KEY"
  android:value="API_KEY"/>
```

Substitute your API key for API\_KEY. This element sets the key `com.google.android.maps.v2.API_KEY` to the value `API_KEY` and makes the API key visible to any `MapFragment` in your application.

## Step 3: Specify App Settings in Manifest

- An Android application that uses the Google Maps Android API should specify the following settings in its manifest file:
  - A reference to the Google Play services version.
  - The Maps API key for the application.
  - Permissions that give the application access to Android system features and to the Google Maps servers.
- Notification that the application requires OpenGL ES version 2.

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"/>
<!-- The following two permissions are not required to use
Google Maps Android API v2, but are recommended. -->
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

```
<uses-feature
  android:glEsVersion="0x00020000"
  android:required="true"/>
```

## Step 4: Add a Map Fragment

- Add the map fragment in the app's layout XML file.

```
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
  android:id="@+id/map"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:name="com.google.android.gms.maps.MapFragment"/>
```

This code is only workable in an application targeting Android API 12 or later.

## Step 5: Creating the Map

