

Location-based Services



Objectives

- Explain the location-based services
- Understand and explain key concepts of Google Play services location API

Location-based Services

- Mobile users take their devices with them everywhere
- Location awareness can be used to offer contextual experience
- Android uses Google Play services, Location APIs

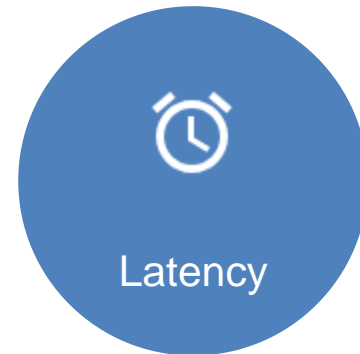
Location Strategies

- Two ways to obtain a user location: GPS and Wi-Fi or cell towers

Type	GPS	Wi-Fi or cell towers
Accuracy	High	Low
Speed	Slow	Fast
Power Consumption	High	Low
Environment	Outdoor	Indoor and outdoor

Battery Drain

- Location gathering and battery drain are affected by these aspects:



Accuracy

- The precision of the location data
- The higher the accuracy, the higher the battery drain

Priority	Constant	Description
High Accuracy	<u>PRIORITY_HIGH_ACCURACY</u>	Most accurate location data Enables GPS, Wi-Fi and cell
Balance Power	<u>PRIORITY_BALANCED_POWER_ACCURACY</u>	Provides accurate location data. May use GPS. Typically uses Wi-Fi and cell
Low Power	<u>PRIORITY_LOW_POWER</u>	Relies on cell towers mostly.
No Power	<u>PRIORITY_NO_POWER</u>	Receives location data from other apps

Types of priority

Type	Precision	Hardware Use	Power
Balanced Power Priority	City block (100 m)	Wi-Fi or cell tower	Less
High Accuracy	Most precise location possible	GPS	High
Low Power	City-level (10 km)	Wi-Fi or cell tower	Less
No power	Receives locations from other apps	None	Very minimum

Frequency

1. `setInterval()` : interval to compute location data
 - Use the largest possible value for background location
 - Use small value for foreground
2. `setFastestInterval()` : interval to compute location data for other app

Frequency

- Interval of update could be achieved using:

Method	Description
setInterval	<p>Sets the rate in milliseconds</p> <p>Speed of update may vary; faster, slower or no update at all</p>
setFastestInterval	<p>Sets the fastest rate or the upper limit in milliseconds</p> <p>Must be set to prevent UI flicker or data overflow</p>
setPriority	Sets the priority of the request

Example

Kotlin

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

Java

```
protected void createLocationRequest() {  
    LocationRequest locationRequest = LocationRequest.create();  
    locationRequest.setInterval(10000);  
    locationRequest.setFastestInterval(5000);  
    locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);  
}
```

Latency

- `setMaxWaitTime()` : delays location delivery
- set a value several times larger than the `setInterval()`

Question?

Identify a location model (accuracy, frequency, and latency) suitable for each of the use cases:

- a. A mapping app
- b. A weather app
- c. A retailer app that notify users who are within proximity to a retail store
- d. A fitness app that can track user activity

Location best practices

1. Remove location updates
2. Set timeouts
3. Batch requests
4. Passive location updates

Location best practices

1. Remove location updates

Life Cycle	Call
onStart() / onResume	Call requestLocationUpdates()
onPause() / onStop()	Call removeLocationUpdates()

Location best practices

2. Set timeouts

Method	Description
<u>setExpirationDuration()</u>	The time in milliseconds since the method was last called
<u>setExpirationTime()</u>	The expiration time in milliseconds since the system last boot

Location best practices

3. Batch requests

- Batch multiple requests together.
- Suitable for non-foreground use cases

E.g. Location is computed every ten minutes:

```
val request = LocationRequest()  
  
request.setInterval(10 * 60 * 1000)  
  
request.setMaxWaitTime(60 * 60 * 1000)
```


Location best practices

4. Passive location updates

- Obtain location data from another foreground app

E.g. Location is computed every 15 minutes:

```
val request = LocationRequest()

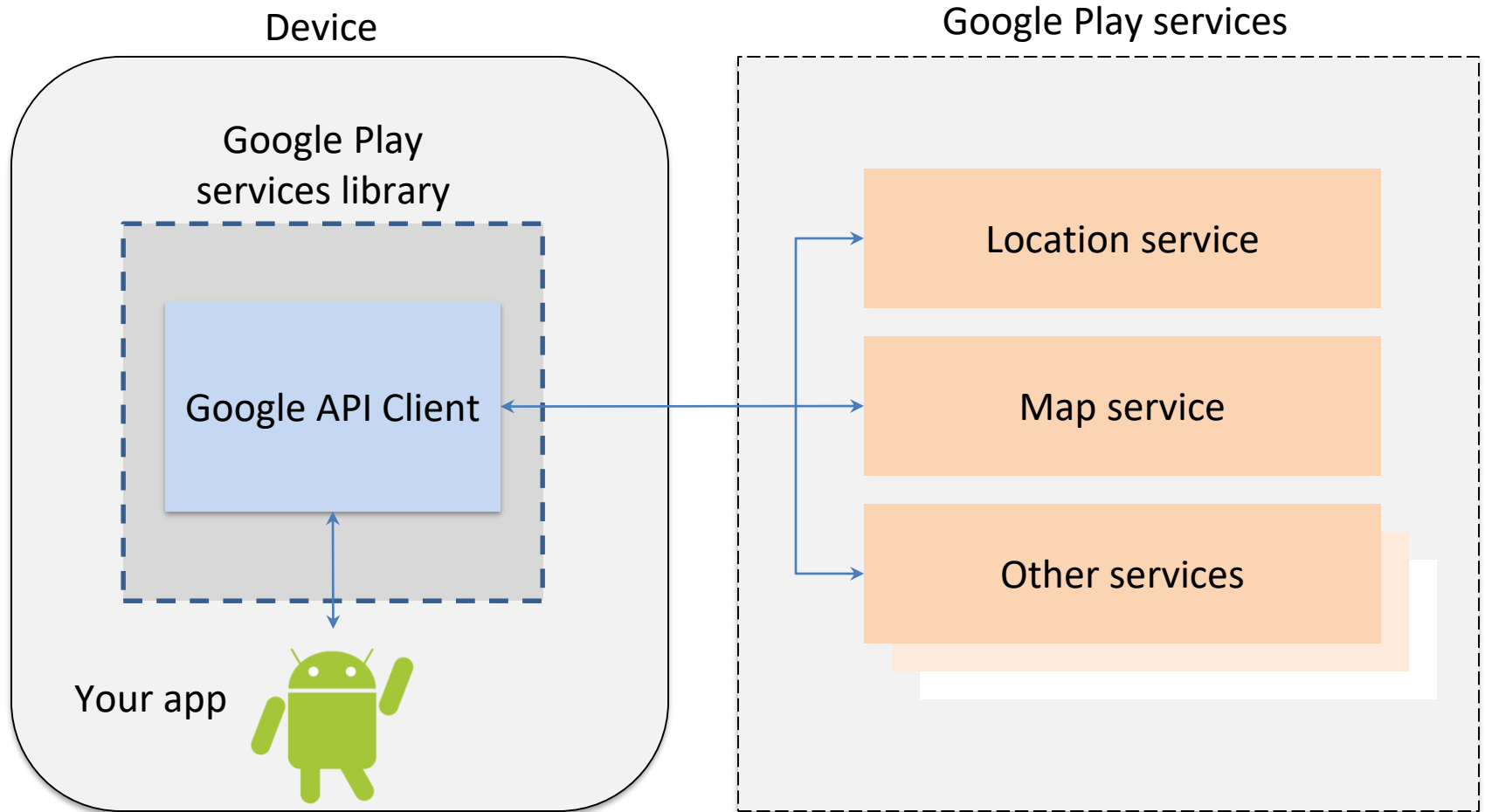
request.setInterval(15 * 60 * 1000)

request.setFastestInterval(2 * 60 * 1000)
```

Last Known Location

- Current location = last known location
- Basic setup:
 1. Google Play Service – SDK Manager
 2. Add library to your project

Google Play service



Google Play Service

- Insert permission to manifest file:

Permission	Description
Coarse Location	Allows an app to access approximate location. Returns a location with an accuracy approximately equivalent to a city block.
Fine Location	Allows an app to access precise location.
Background	For Android 10 (API 29) and above

Permission

- For Android 6 or higher

```
<manifest ...>  
  <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>  
</manifest>
```

- For Android 10 or higher

```
<manifest ... >  
  <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />  
  <uses-permission android:name="android.permission.ACCESS_BACKGROUND_LOCATION" />  
</manifest>
```

Google Play service – Your App

- onCreate() method creates an instance of Google API Client

```
Kotlin private lateinit var fusedLocationClient: FusedLocationProviderClient
...
override fun onCreate(savedInstanceState: Bundle?) {
    ...
    fusedLocationClient = LocationServices.getFusedLocationProviderClient(this)
}
```

```
Java private FusedLocationProviderClient fusedLocationClient;
...
@Override
protected void onCreate(Bundle savedInstanceState) {
    ...
    fusedLocationClient = LocationServices.getFusedLocationProviderClient(this);
}
```

Communicate with Google Services

- Connect to service in onStart() method.

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

```
Java fusedLocationClient.getLastLocation()
    .addOnSuccessListener(this, new OnSuccessListener<Location>() {
        @Override
        public void onSuccess(Location location) {
            // Got last known location.
            // In some rare situations this can be null.
            if (location != null) {
                // Logic to handle location object
            }
        }
    });
```

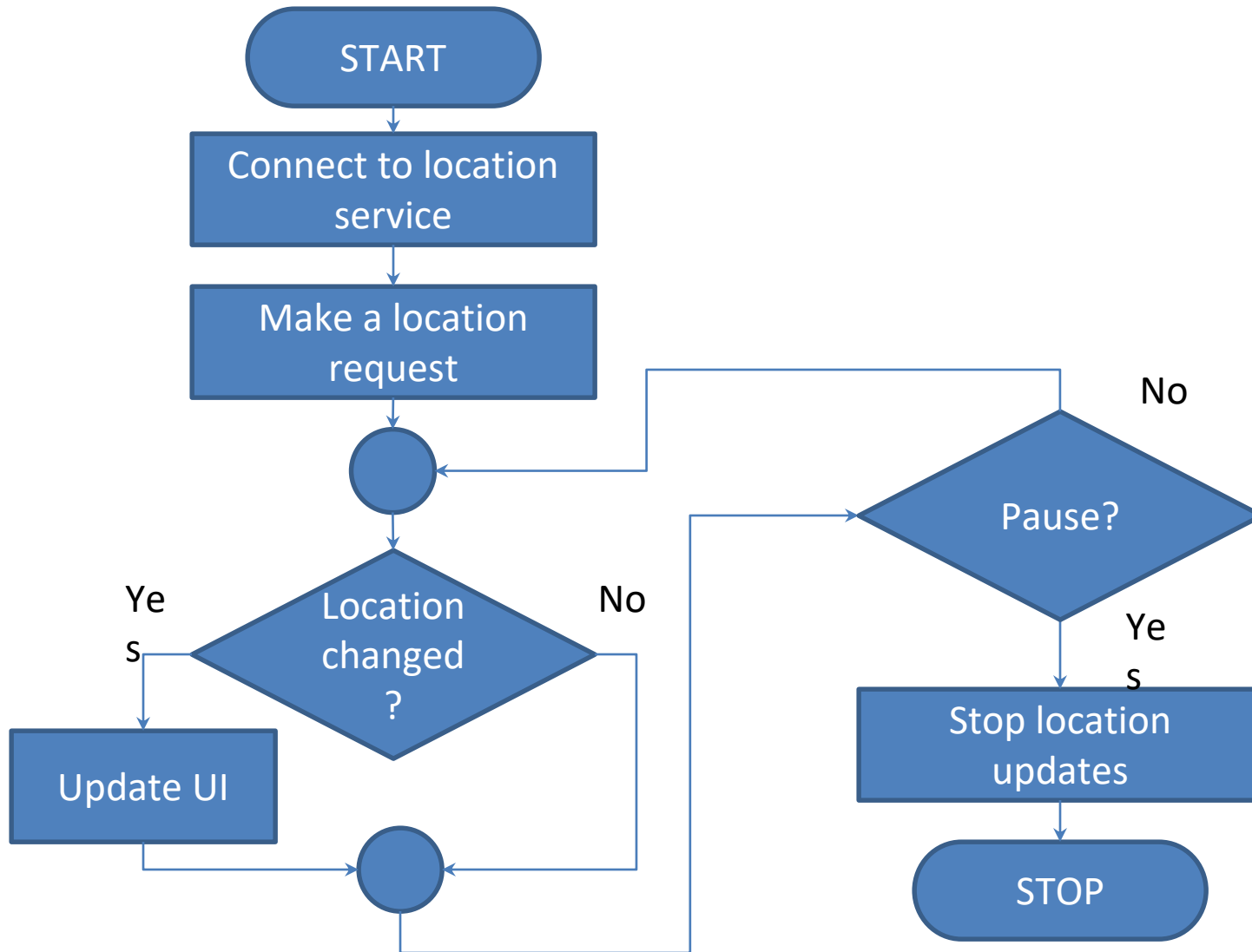
Question?

1. Among the three location permissions (Fine, Coarse, and Background), which one is suitable for
 - a. a mapping app?
 - b. a weather app?
2. Why it is necessary to start a location request in the `onStart()` method?
3. In which Activity's lifecycle method that you should turn off the Location service?

Location Strategies

- Challenges in determining user location:
 - Multitude of location sources
 - User movement
 - Varying accuracy

Location Updates



Permission

- The new Permissions model (Android 6.0 and above) changes the way that permissions are allocated to your app by the user.
- App must ask the user for individual permissions at runtime.

Mock Location Data

- Use mock location for testing purposes
- Providing mock location data by injecting GPS location data
- Using DDMS to set location to AVD

Geocoding

- Geocoding = converting a geographic location to an address
- Possible errors:
 - No location data provided
 - Invalid latitude or longitude used
 - No geocoder available
 - No address found

Review Questions

1. Compare the TWO methods to obtain a user location.
2. What are the factors that could affect accuracy of location-based services?
3. “It is important to create a model of best performance for a location-based app”
Comment on this statement.

Review Questions

4. “The best way to ensure accuracy of location-based services is to test your app using real device.” Comment on this statement.
5. Location awareness is one of the unique features of mobile devices. Discuss how this feature could be utilized to enhance user experience. Use a suitable example to explain your answer. (6 marks)

Review Questions

6. MyChild.com is a company that provides child tracking system using Global Positioning System (GPS). Due to some limitations of GPS, the company is looking for a mobile solution to better link its existing tracking technologies with mobile technologies.
- Describe THREE (3) limitation of GPS. (6 marks)
 - Suggest and describe a technique to overcome the limitations of GPS. (4 marks)