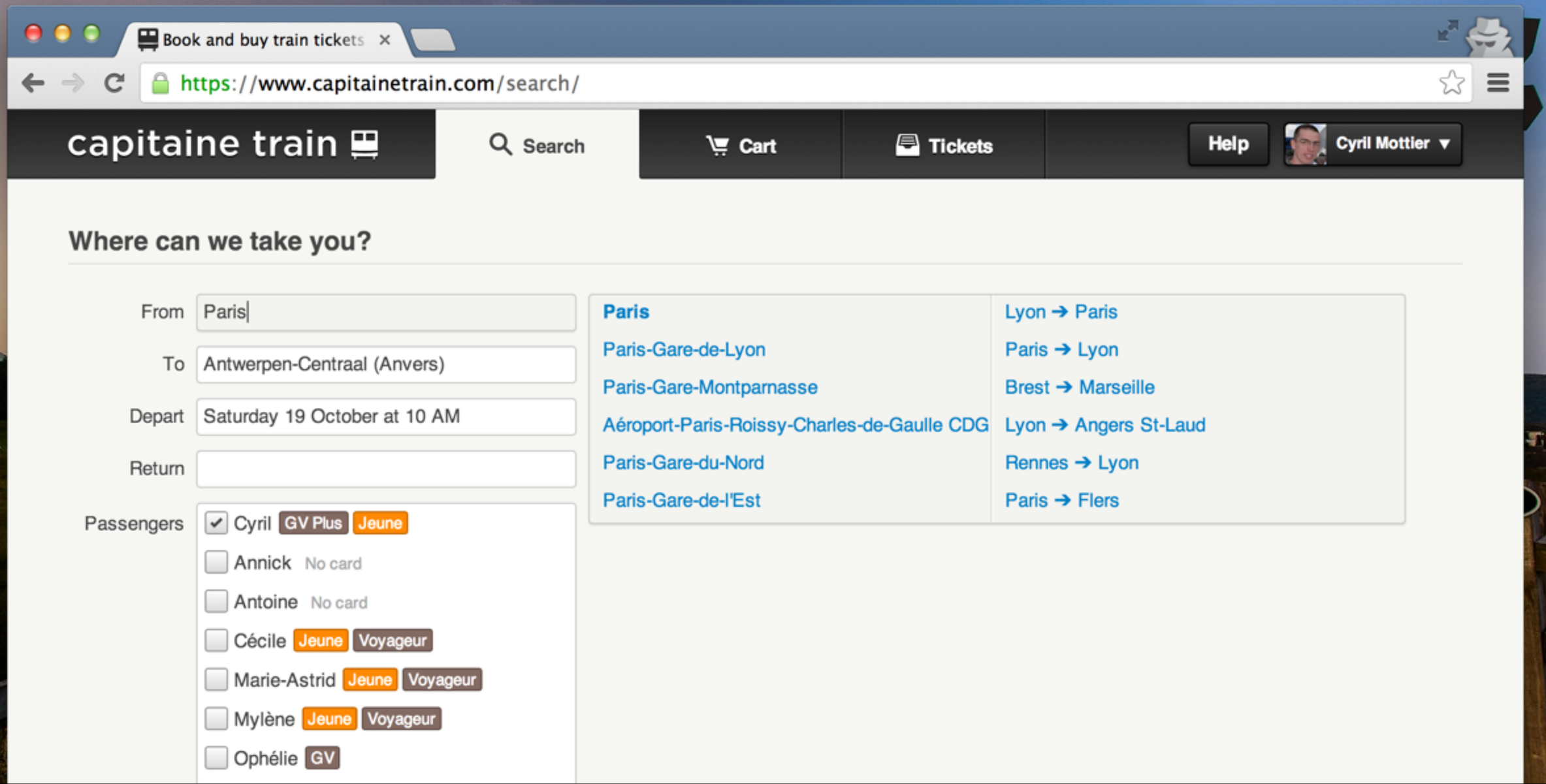
Getting around with

GOOGLE MAPS ANDROID API V2


@ CYRILMOTTIER


capitaine train
LC
54,67
WHR

capitaine train



I once made a

LOVELY & GOOD-LOOKING

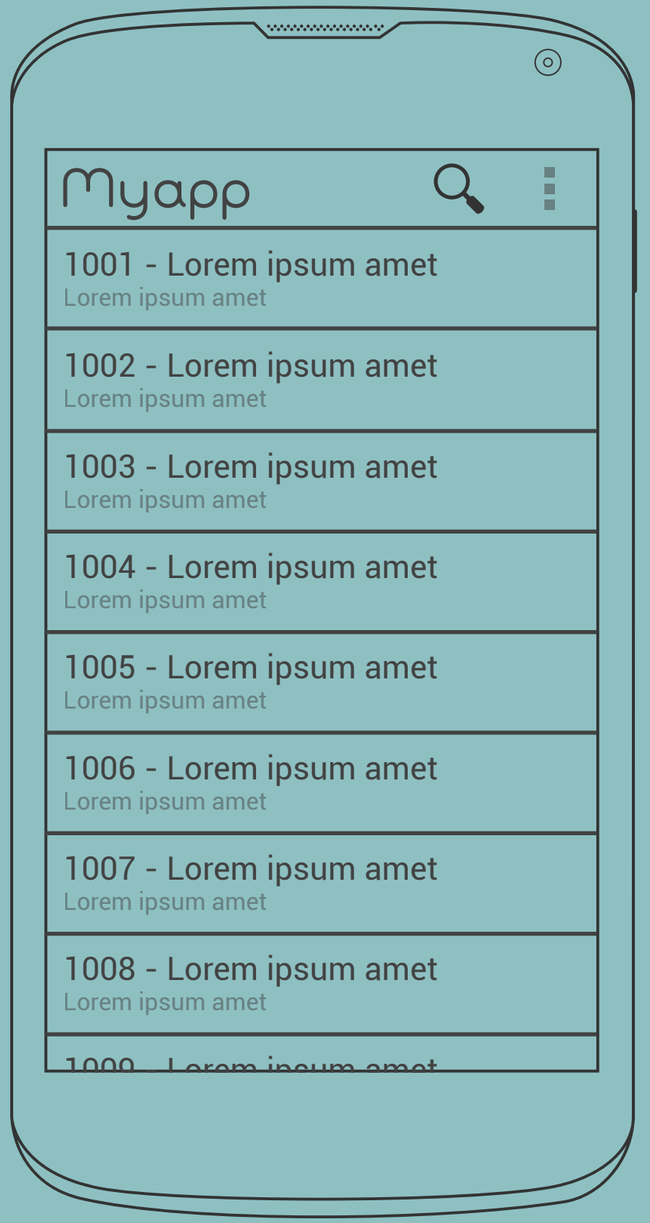
mobile app

I once made a

LOVELY & GOOD-LOOKING

mobile app

IT LOOKED LIKE THIS



I started to do some quick

USER TESTING

“That’s a pretty nice list of thingy things”

– John Doe


“It sucks”
- John Doe’s translator

Lists-based screens

are boring and

give no geographic context

Maps give the user some clear context on

POI LOCATION

Maps give the user some clear context on

POI LOCATION

USER LOCATION

Maps give the user some **clear context on**

POI LOCATION

USER LOCATION

DISTANCES

Maps give the user some clear context on

POI LOCATION

USER LOCATION

DISTANCE

DIRECTIONS

Maps give the user some **clear context on**

POI LOCATION

USER LOCATION

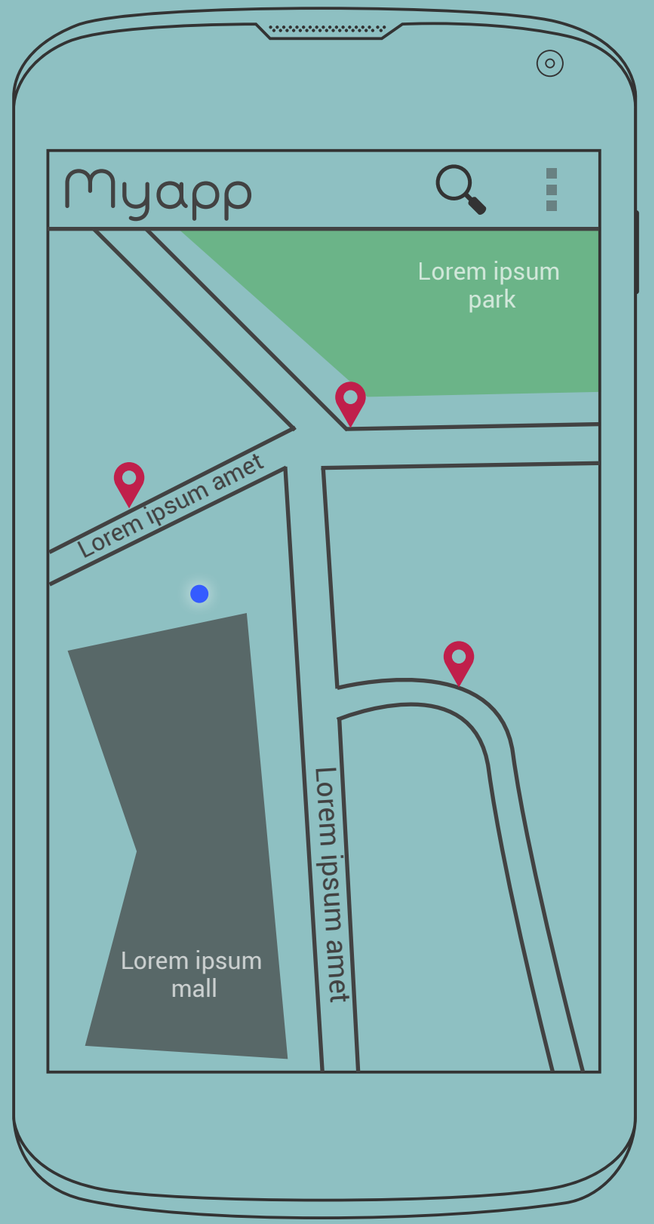
DISTANCES

DIRECTIONS

GOOGLE MAPS ANDROID API V2 TO THE RESCUE

Map-based apps

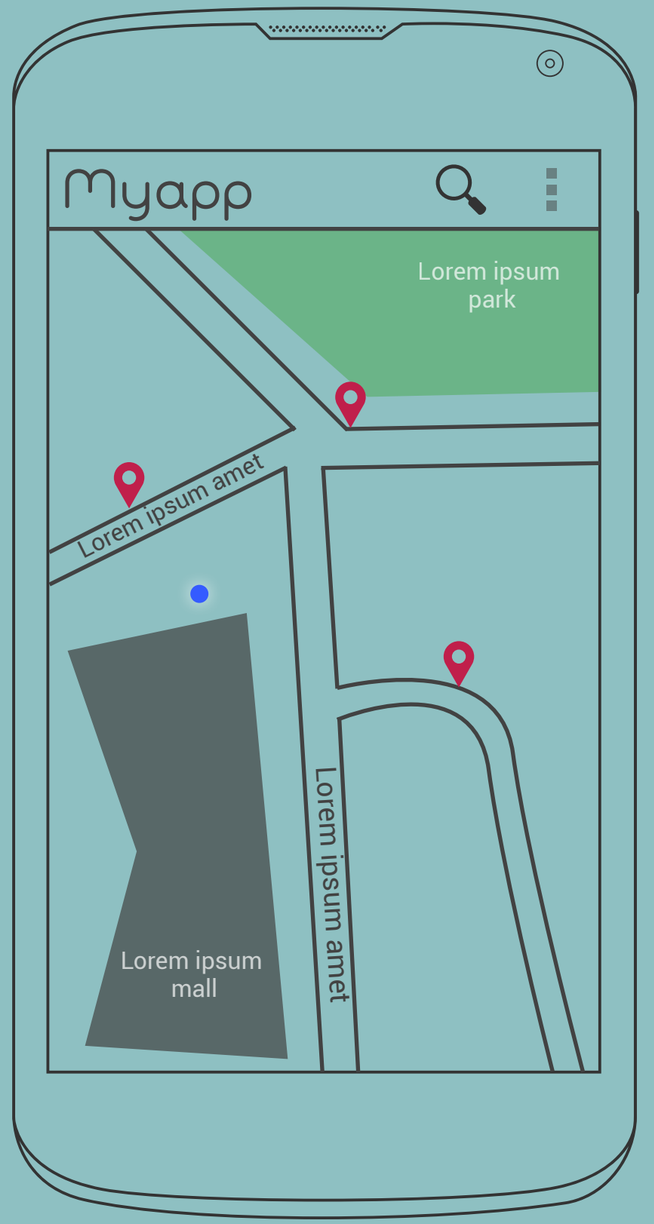
are immersive



Map-based apps

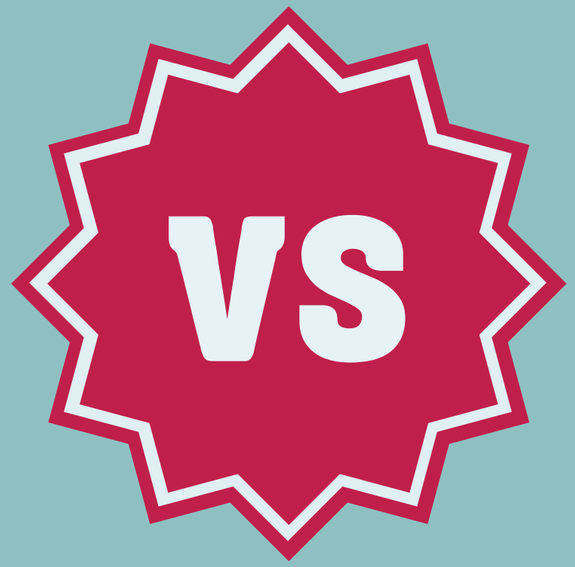
are immersive

and that's awesome :)

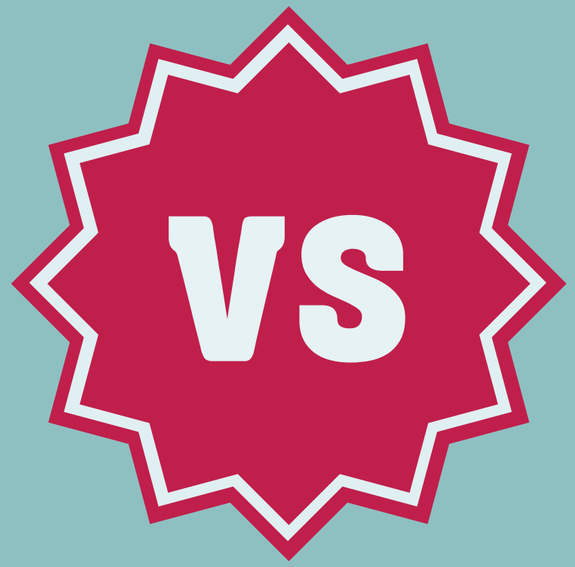


Introducing

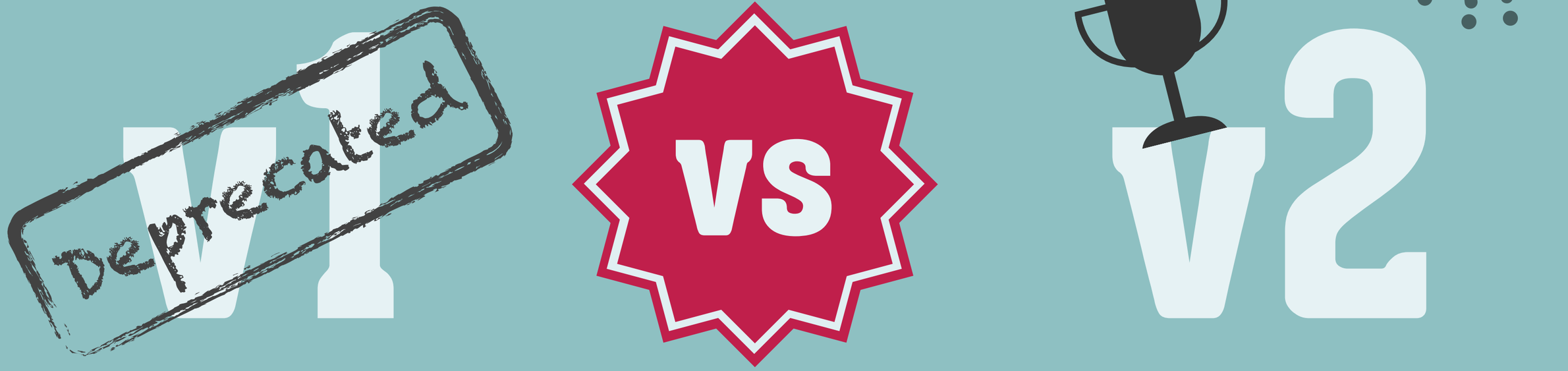
MAPS API V2

v1  v2

v1



v2


v1
Deprecated
VS
v2

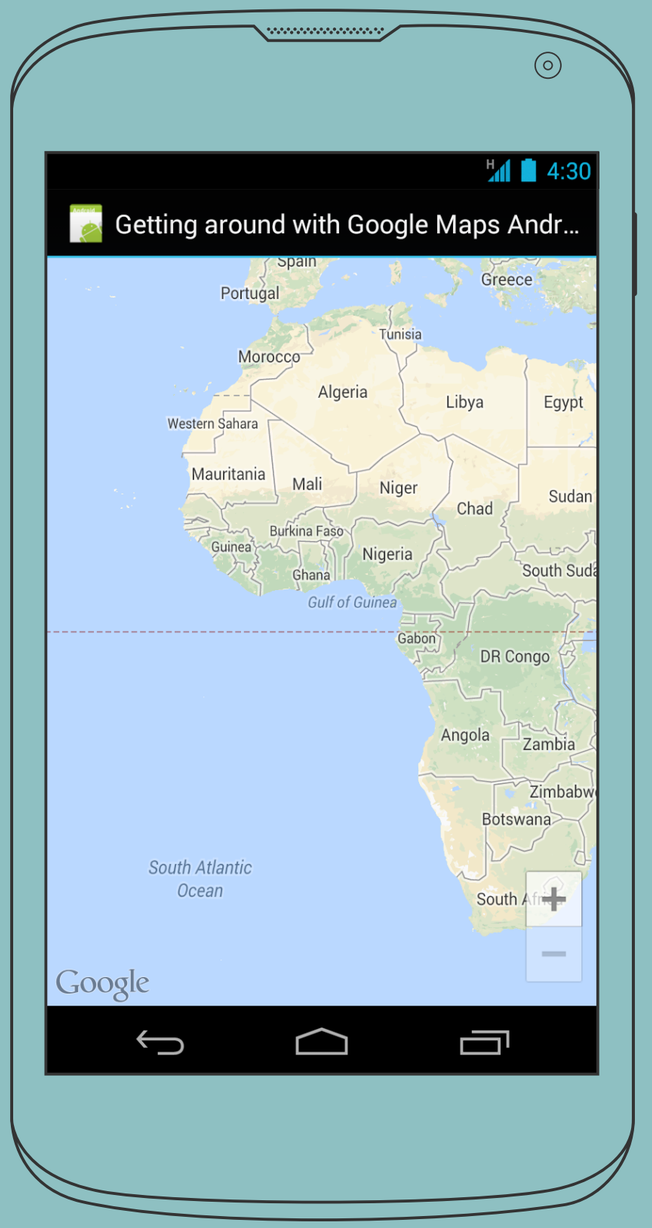
EXTREMELY

STRAIGHTFORWARD

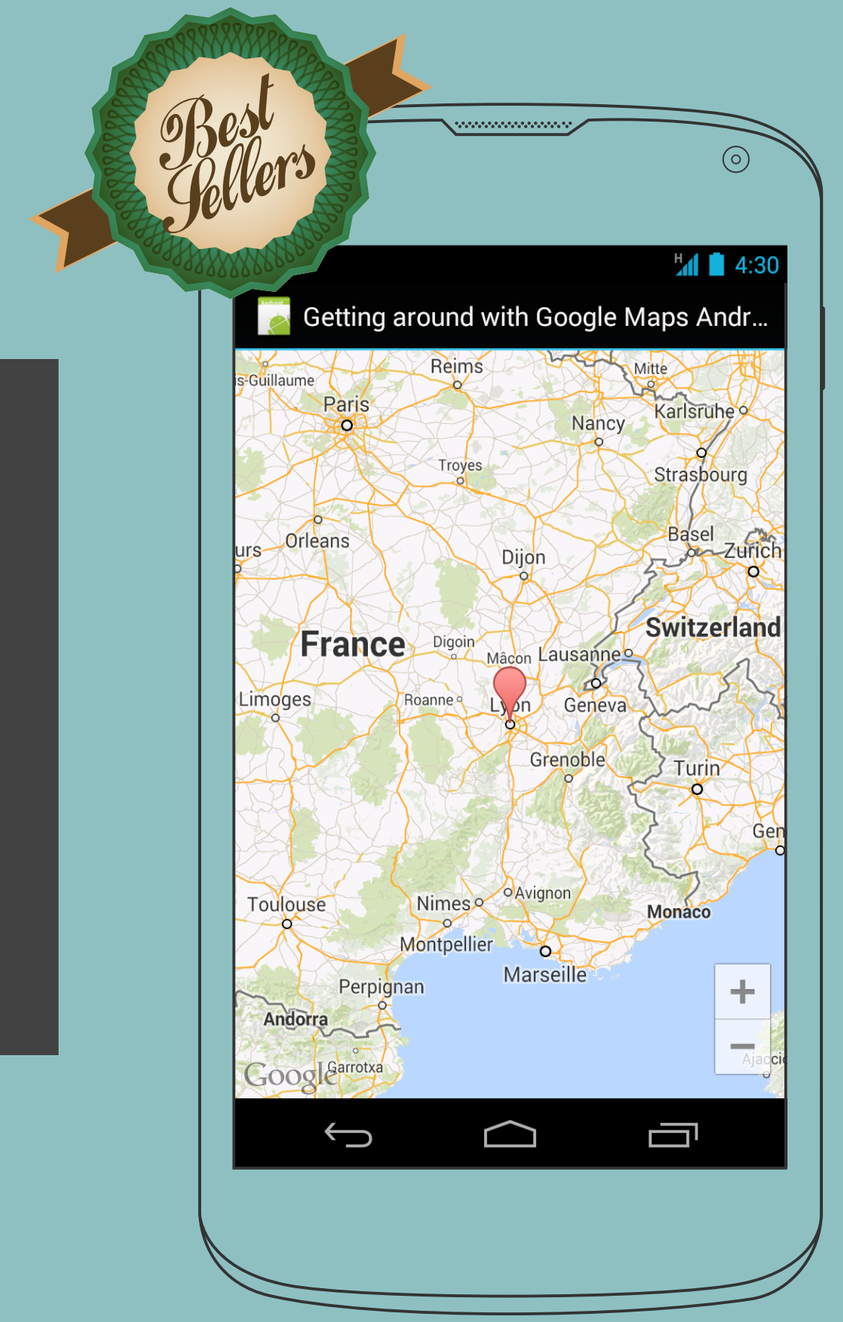
API

Initial [Support]MapFragment setup

```java
public class TestActivity extends FragmentActivity {
    private GoogleMap mMap;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_marker);
        setUpMapIfNeeded();
    }

    @Override
    protected void onResume() {
        super.onResume();
        setUpMapIfNeeded();
    }

    private void setUpMapIfNeeded() {
        if (mMap == null) {
            mMap = ((SupportMapFragment) getSupportFragmentManager().
                    findFragmentById(R.id.map)).getMap();
            if (mMap != null) {
                setUpMap();
            }
        }
    }

    private void setUpMap() { /* TODO */ }
}
```



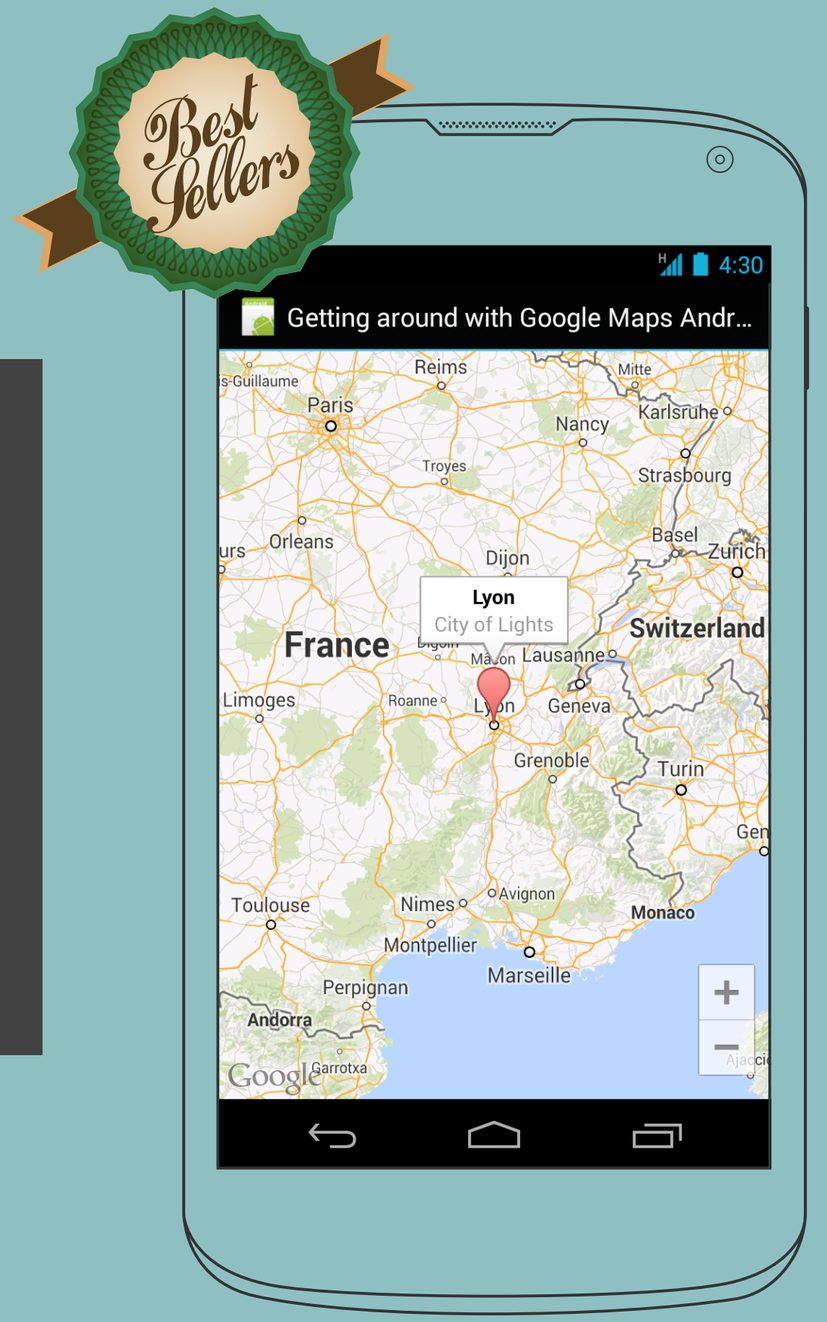
Annotating with Markers




```
1  private static final LatLng LYON = new LatLng(45.764043, 4.835659);
2
3  private Marker mMarker;
4
5  private void setUpMap() {
6      mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(LYON, 6));
7
8      mMarker = mMap.addMarker(new MarkerOptions().
9              position(LYON).
10             title("Lyon").
11             snippet("City of Lights").
12             icon(BitmapDescriptorFactory.
13                     defaultMarker(BitmapDescriptorFactory.HUE_RED)));
14 }
```

Annotating with Markers

```
1  private static final LatLng LYON = new LatLng(45.764043, 4.835659);
2
3  private Marker mMarker;
4
5  private void setUpMap() {
6      mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(LYON, 6));
7
8      mMarker = mMap.addMarker(new MarkerOptions().
9              position(LYON).
10             title("Lyon").
11             snippet("City of Lights").
12             icon(BitmapDescriptorFactory.
13                     defaultMarker(BitmapDescriptorFactory.HUE_RED)));
14 }
```



Drawing shapes on MapView

3 POSSIBLE SHAPES

Drawing shapes on MapView

3 POSSIBLE SHAPES

Polyline

Drawing shapes on MapView

3 POSSIBLE SHAPES

Polyline | Polygon

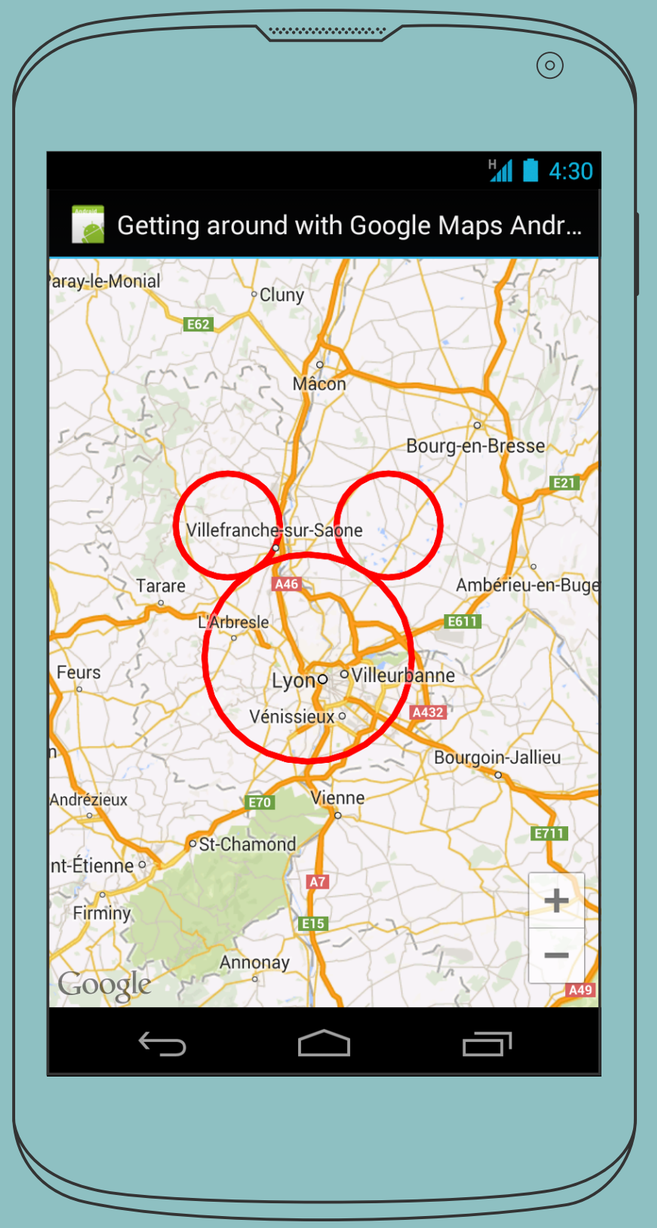
Drawing shapes on MapView

3 POSSIBLE SHAPES

Polyline | Polygon | Circle

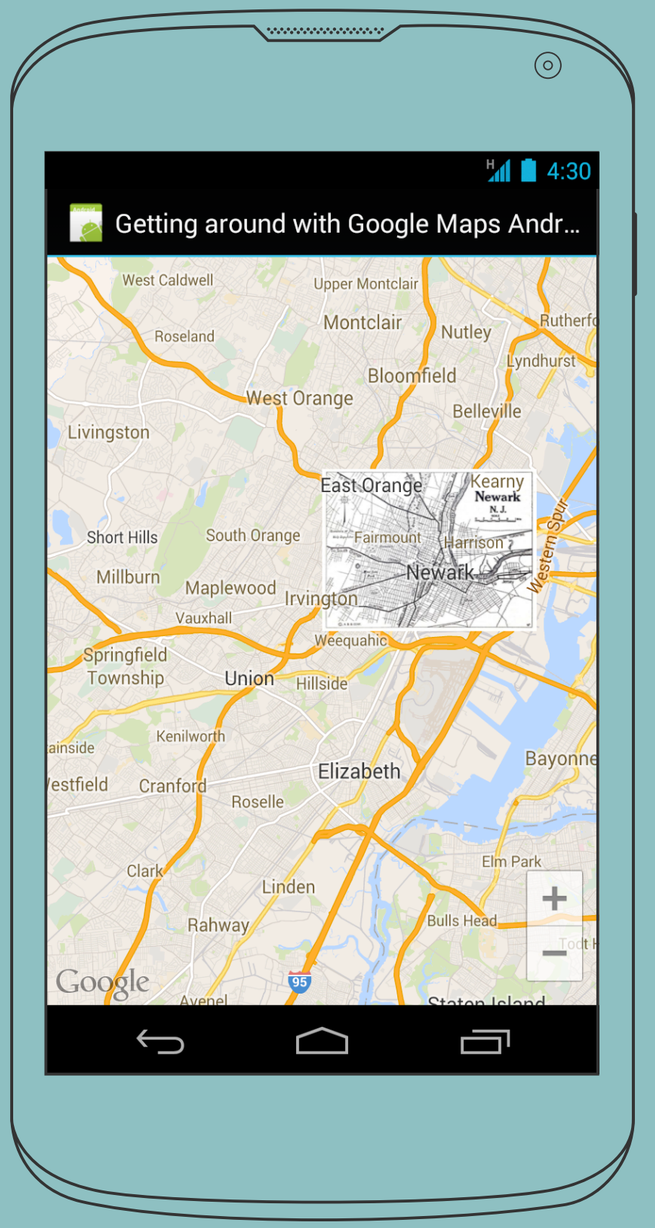
Drawing shapes on MapView

```
1  private void setUpMap() {
2      mMap.addCircle(getCircleOptions(new LatLng(45.8, 4.8), 20000));
3      mMap.addCircle(getCircleOptions(new LatLng(46.03, 4.6), 10000));
4      mMap.addCircle(getCircleOptions(new LatLng(46.03, 5), 10000));
5  }
6
7  private CircleOptions mCircleOptions = new CircleOptions();
8
9  private CircleOptions getCircleOptions(LatLng center,
10                                         double radius) {
11     final float w = 4 * getResources().getDisplayMetrics().density;
12     return mCircleOptions.
13             center(center).
14             radius(radius).
15             strokeWidth(w).
16             strokeColor(Color.RED);
17 }
```



Overlaying map with GroundOverlay

```
1  private static final LatLng NEWARK = new LatLng(40.71408, -74.22869);
2
3  private void setUpMap() {
4      mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(NEWARK, 11));
5
6      mMap.addGroundOverlay(new GroundOverlayOptions().
7              image(BitmapDescriptorFactory.
8                      fromResource(R.drawable.newark_1922)).
9              anchor(0, 1).
10             position(NEWARK, 8600f, 6500f));
11 }
```

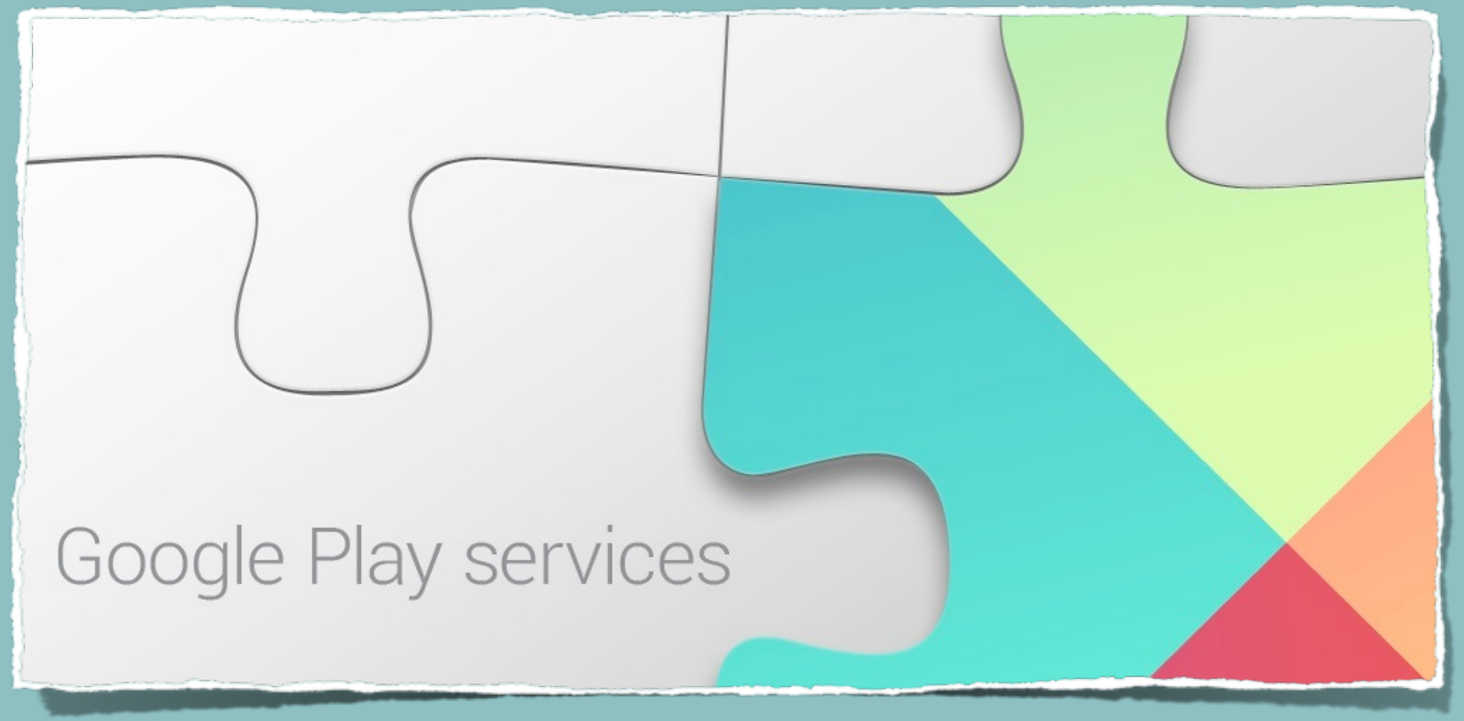


Rendering custom tiles with TileOverlay

```
1  private static final String MOON_MAP_URL_FORMAT =
2          "http://www.google.moon/bw/%d/%d/%d.jpg";
3
4  private void setUpMap() {
5      mMap.setMapType(GoogleMap.MAP_TYPE_NONE);
6
7      TileProvider tileProvider = new UrlTileProvider(256, 256) {
8          @Override
9          public synchronized URL getTileUrl(int x, int y, int zoom) {
10             String s = String.format(Locale.US,
11                     MOON_MAP_URL_FORMAT, x, y, zoom);
12             try {
13                 return new URL(s);
14             } catch (MalformedURLException e) {
15                 throw new AssertionError(e);
16             }
17         }
18     };
19
20     mMap.addTileOverlay(new TileOverlayOptions().
21             tileProvider(tileProvider));
22 }
```

Deep dive into

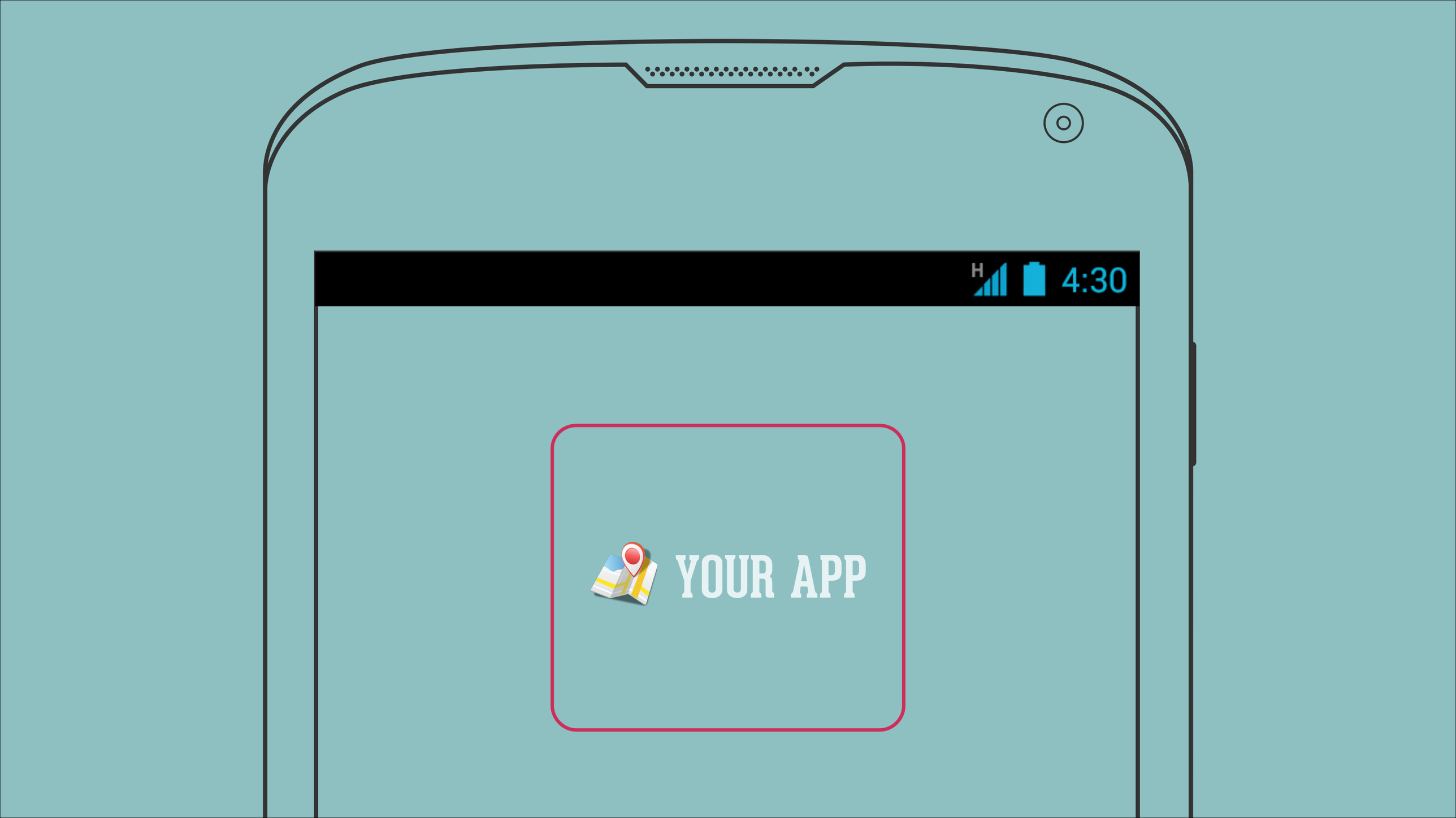
THE INTERNALS

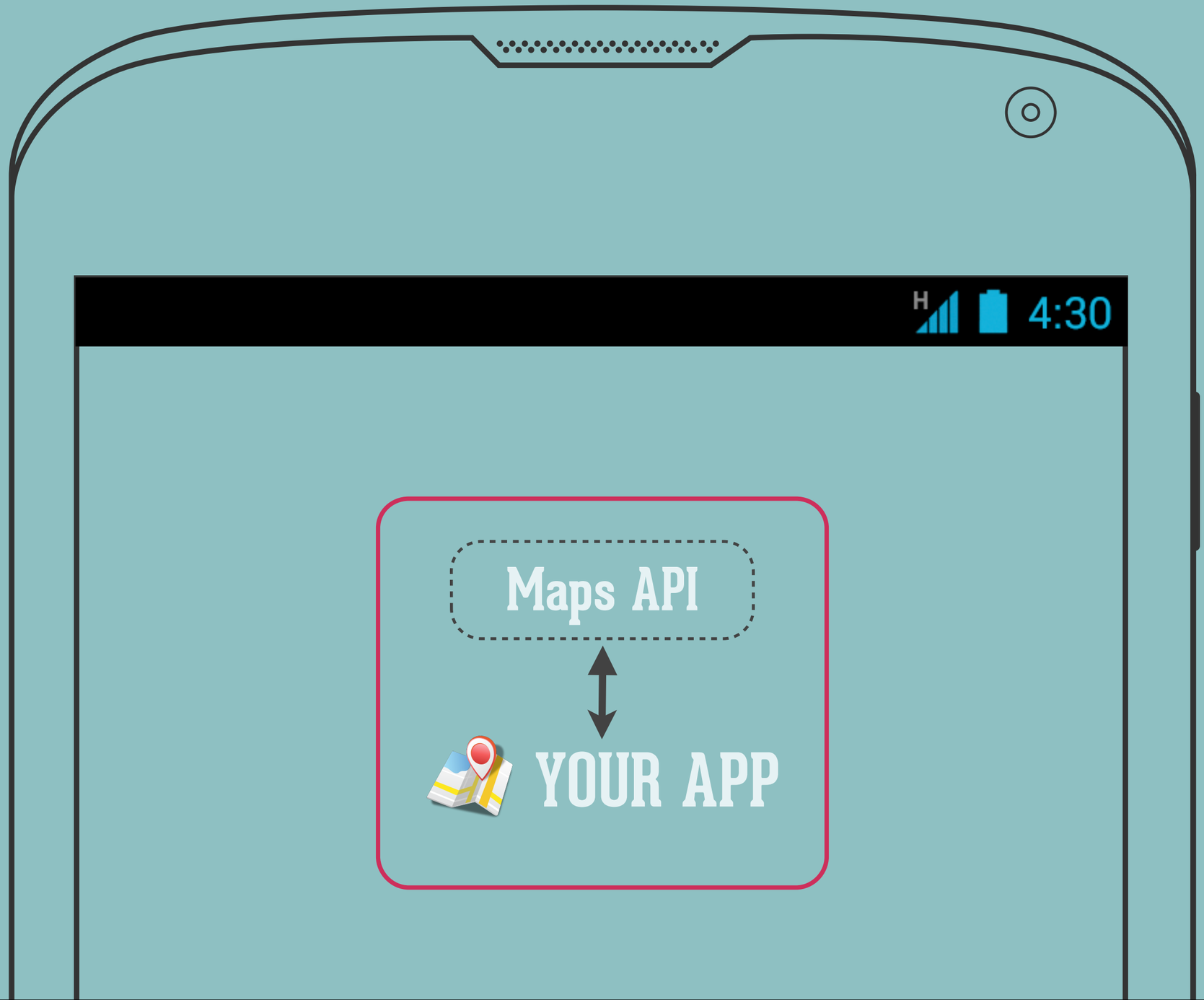


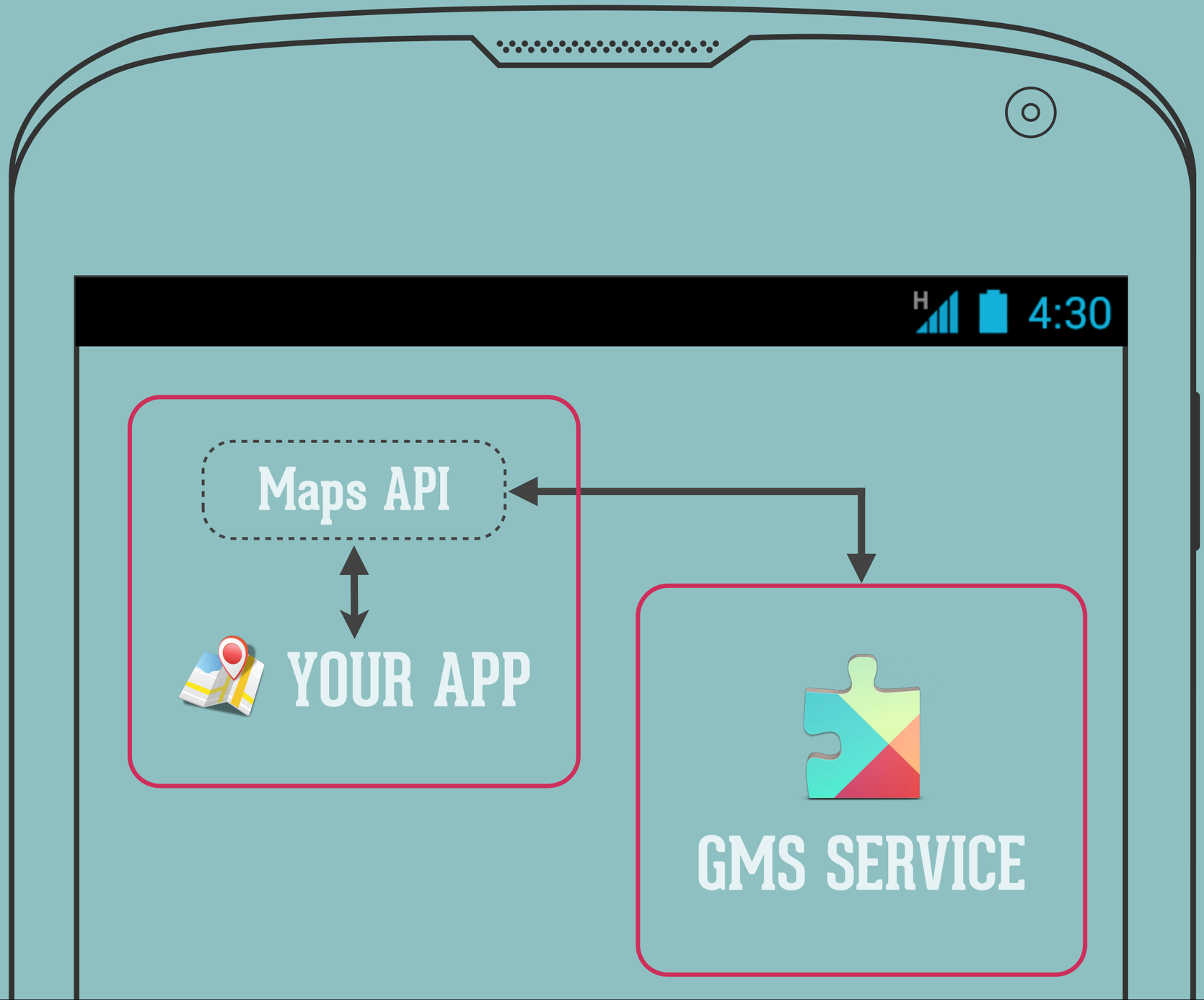
Google Maps Android API v2

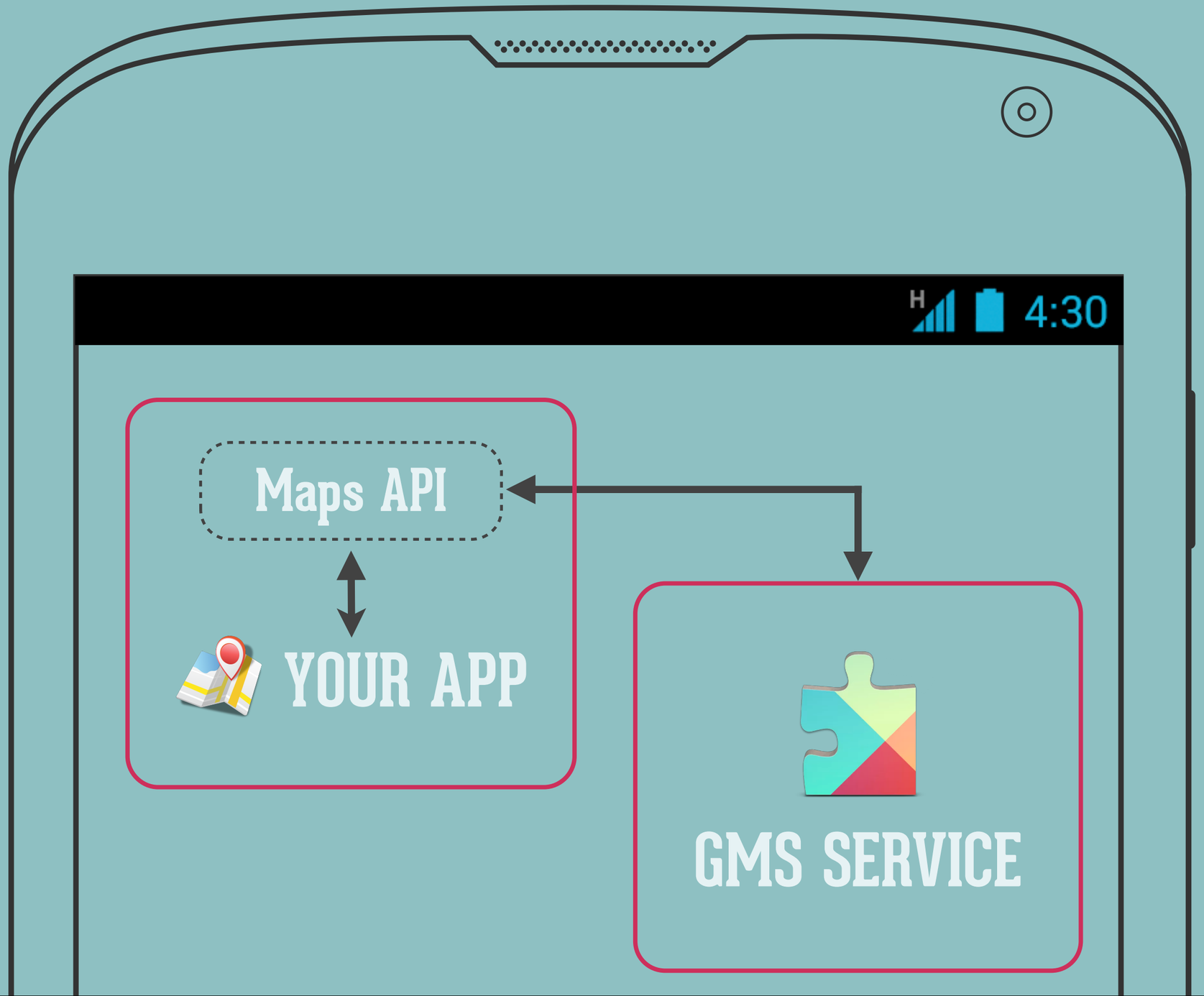
is part of

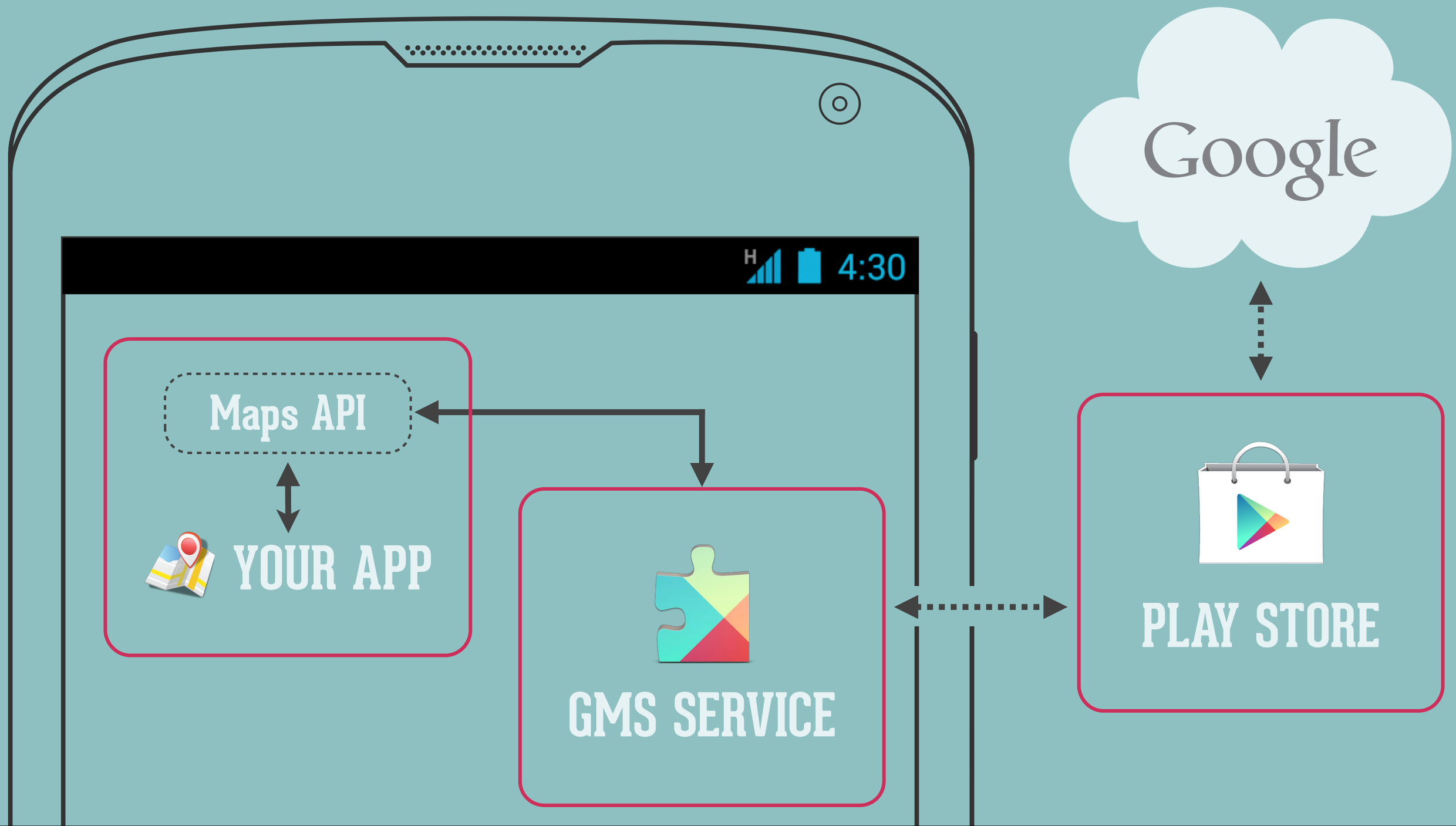
Google Play Services


H
4:30
YOUR APP

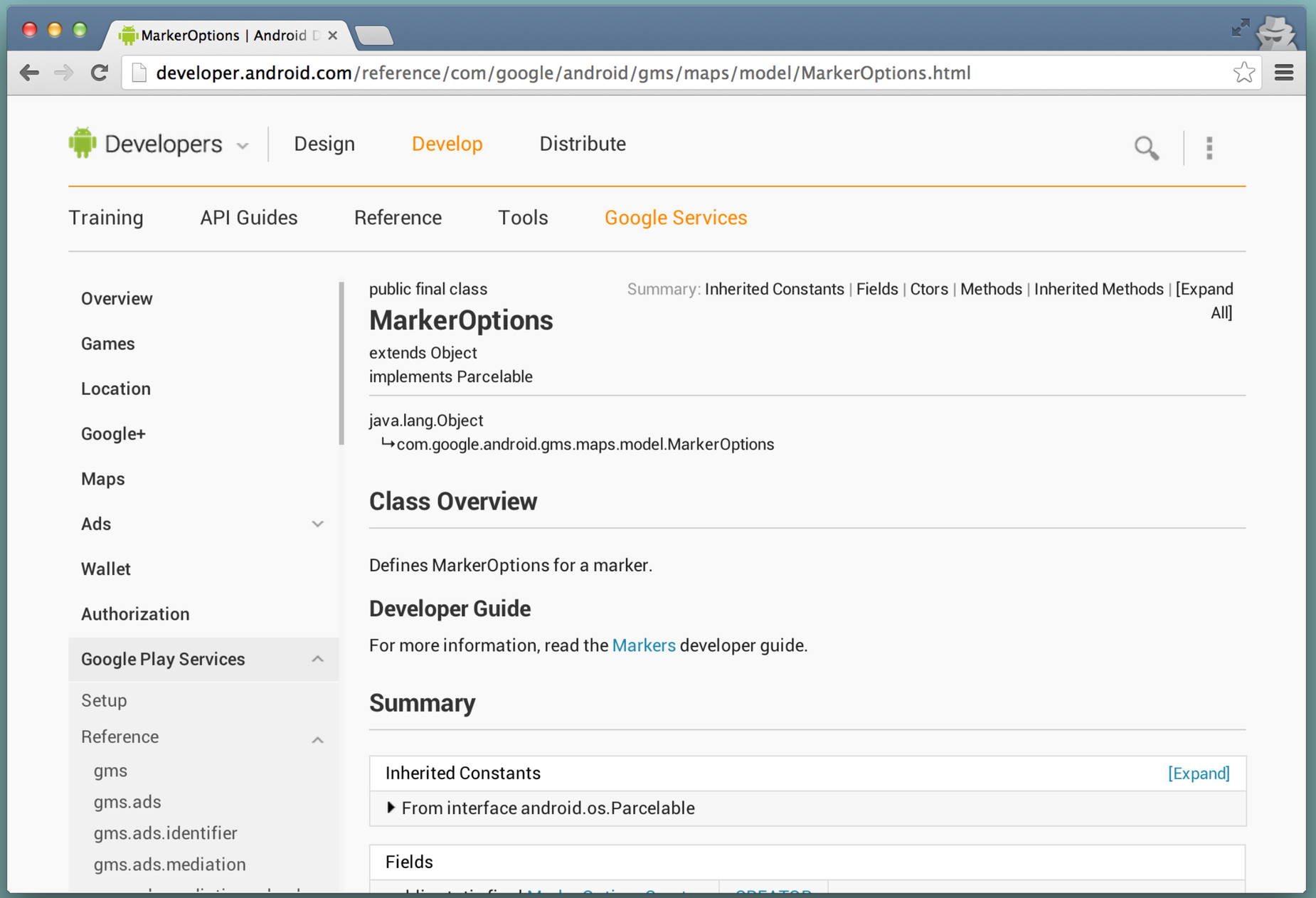

H
4:30
Maps API
YOUR APP


H
4:30
Maps API
YOUR APP
GMS SERVICE


H
4:30
Maps API
YOUR APP
GMS SERVICE


Google
4:30
Maps API
YOUR APP
GMS SERVICE
PLAY STORE

EVERYTHING
EVERYTHING
EVERYTHING
EVERYTHING IS PARCELABLE
EVERYTHING
EVERYTHING
EVERYTHING
EVERYTHING


MarkerOptions | Android
developer.android.com/reference/com/google/android/gms/maps/model/MarkerOptions.html
Developers
Design
Develop
Distribute
Training
API Guides
Reference
Tools
Google Services
Overview
Games
Location
Google+
Maps
Ads
Wallet
Authorization
Google Play Services
Setup
Reference
gms
gms.ads
gms.ads.identifier
gms.ads.mediation
public final class
MarkerOptions
extends Object
implements Parcelable
Summary: Inherited Constants | Fields | Ctors | Methods | Inherited Methods | [Expand All]
java.lang.Object
↳com.google.android.gms.maps.model.MarkerOptions
Class Overview
Defines MarkerOptions for a marker.
Developer Guide
For more information, read the Markers developer guide.
Summary
Inherited Constants
[Expand]
From interface android.os.Parcelable
Fields

public final class

Summary: Inherited Constants

MarkerOptions

extends Object

implements Parcelable

java.lang.Object

↳com.google.android.gms.maps.model.MarkerOptions

Class Overview

public final class

Summary: Inherited Constants

MarkerOptions

extends Object

implements Parcelable

java.lang.Object

↳com.google.android.gms.maps.model.MarkerOptions

Class Overview

Each call to the API

is a done through a Binder ...

... and it may be

A VIEW MAY HIDE ANOTHER

A VIEW MAY HIDE ANOTHER

I.E. THIS IS NOT A [REGULAR] VIEW

TextureView

API 16+

SurfaceView

API 15-

1 No direct Canvas-based rendering

2 Only Bitmaps can be rendered

3 No Drawable/View automatic refresh

You must be on the **UI THREAD**

WHEN CALLING ANY METHOD ON

any **View**

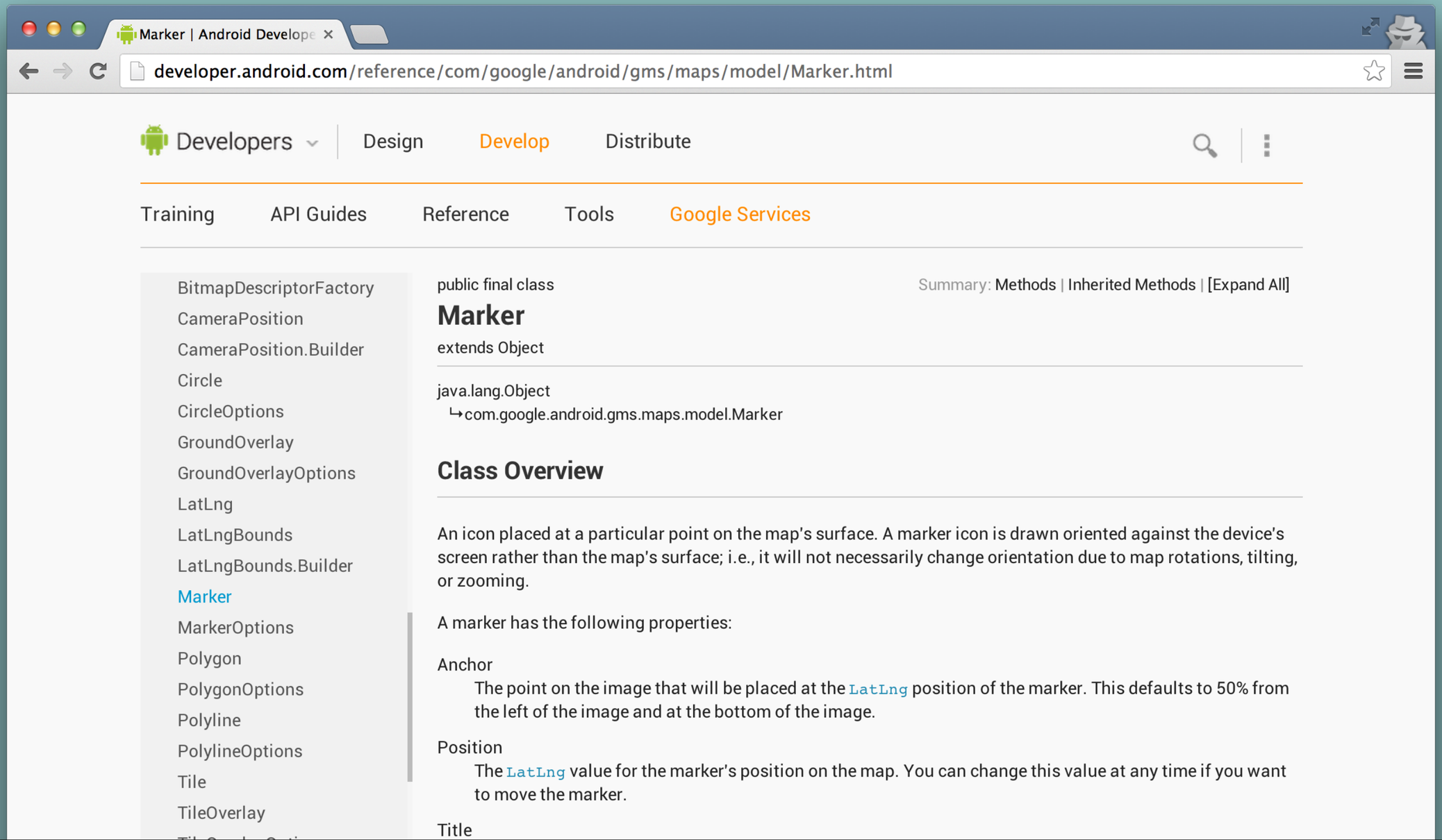
You must be on the **UI THREAD**

WHEN CALLING ANY METHOD ON

any **GoogleMap**

Google Maps Android API v2
TIPS AND TRICKS

Attaching information to Markers



scriptorFactory
sition
sition.Builder
ons
erlay

public final class

Marker

extends Object

java.lang.Object
↳com.google.android.gms.maps.model.M

scriptorFactory
sition
sition.Builder
ons
erlay

public final class

Marker

extends Object

java.lang.Object
↳com.google.android.gms.maps.model.M

Use a MAPPING COLLECTION

Map<K, V>

HashMap | LinkedHashMap | ArrayMap

Use a MAPPING COLLECTION

Map<K, V>

HashMap | LinkedHashMap | ArrayMap

NEW

Attaching information to Markers

```
1  private Map<Marker, Integer> mMarkerPositions = new HashMap<Marker, Integer>();
2  private Cursor mCursor;
3
4  private void setUpMap() {
5      updatePois();
6  }
7
8  private void updatePois() {
9      mMarkerPositions.clear();
10     if (mMap != null) {
11         mMap.clear();
12         if (mCursor != null) {
13             final MarkerOptions markerOptions = new MarkerOptions();
14             while (mCursor.moveToNext()) {
15                 final Marker marker = mMap.addMarker(markerOptions.
16                         position(new LatLng(
17                                 mCursor.getDouble(PoisQuery.LAT),
18                                 mCursor.getDouble(PoisQuery.LNG))));
19                 mMarkerPositions.put(marker, mCursor.getPosition());
20             }
21         }
22     }
23 }
```

Attaching information to Markers

```
21          }
22      }
23  }
24
25  private final LoaderManager.LoaderCallbacks<Cursor> mLoaderCallbacks =
26          new LoaderManager.LoaderCallbacks<Cursor>() {
27      @Override
28      public Loader<Cursor> onCreateLoader(int id, Bundle args) {
29          return new CursorLoader(getBaseContext() /* ... */);
30      }
31
32      @Override
33      public void onLoadFinished(Loader<Cursor> loader, Cursor cursor) {
34          mCursor = cursor;
35          updatePois();
36      }
37
38      @Override
39      public void onLoaderReset(Loader<Cursor> loader) {
40          mCursor = null;
41          updatePois();
42      }
43  };
```

The "be" attitude

BE

lazy when possible
mindful on memory
nice with users

```
1  private void setUpMap() {
2      mMap.setInfoWindowAdapter(new GoogleMap.InfoWindowAdapter() {
3          @Override
4          public View getInfoWindow(Marker marker) {
5              return null;
6          }
7
8          @Override
9          public View getInfoContents(Marker marker) {
10             Integer position = mMarkerPositions.get(marker);
11             if (position == null || mCursor == null) {
12                 return null;
13             }
14             mCursor.moveToPosition(position);
15             marker.setTitle(mCursor.getString(PoisQuery.TITLE));
16             marker.setSnippet(mCursor.getString(PoisQuery.SUBTITLE));
17             return null;
18         }
19     });
20     //...
21 }
```

Animate state transitions

Animate state transitions

Animate state transitions

No actual animation support in

GOOGLE MAPS ANDROID API V2

SOLUTION

Do it

ON YOUR OWN!

```java
public class LatLngEvaluator implements TypeEvaluator<LatLng> {

    @Override
    public LatLng evaluate(float fraction, LatLng startValue, LatLng endValue) {
        double endValueLng = endValue.longitude;

        // Take the shortest path across the 180th meridian.
        double lngDelta = startValue.longitude - endValue.longitude;
        if (Math.abs(lngDelta) >= 180) {
            endValueLng = endValue.longitude + Math.signum(lngDelta) * 360;
        }

        double newLat = eval(fraction, startValue.latitude, endValue.latitude);
        double newLng = eval(fraction, startValue.longitude, endValueLng);
        return new LatLng(newLat, newLng);
    }

    private static double eval(float fraction, double startValue, double endValue) {
        return startValue + fraction * (endValue - startValue);
    }
}
```

```java
private static final LatLngEvaluator LAT_LNG_EVALUATOR = new LatLngEvaluator();
private static final Property<Marker, LatLng> MARKER_POSITION_PROPERTY =
        Property.of(Marker.class, LatLng.class, "position");

private static final LatLng LYON = new LatLng(45.764043, 4.835659);
private static final LatLng PARIS = new LatLng(48.856614, 2.3522219);

private LatLng mCurrentDest = LYON;
private Marker mMarker;
private ObjectAnimator mMarkerAnimator;

private void setUpMap() {
    mMap.setOnMarkerClickListener(mOnMarkerClickListener);
    mMarker = mMap.addMarker(new MarkerOptions().position(mCurrentDest));
}
```

```
19
20
21
22
23
24
25 private final GoogleMap.OnMarkerClickListener mOnMarkerClickListener =
26         new GoogleMap.OnMarkerClickListener() {
27     @Override
28     public boolean onMarkerClick(Marker marker) {
29         mCurrentDest = mCurrentDest.equals(LYON) ? PARIS : LYON;
30         if (mMarkerAnimator == null) {
31             mMarkerAnimator = ObjectAnimator.ofObject(mMarker,
32                     MARKER_POSITION_PROPERTY,
33                     LAT_LNG_EVALUATOR,
34                     mCurrentDest);
35         } else {
36             mMarkerAnimator.setObjectValues(mCurrentDest);
37         }
38         mMarkerAnimator.start();
39         return true;
40     }
41 };
```

config.proguard

```
-keepclassmembers public class
    com.google.android.gms.maps.model.Marker {
   void setPosition(***);
   *** getPosition();
}
```

Need more?

googlemaps.github.io/android-maps-utils/

Thank you!

@cyrilmottier ★ cyrilmottier.com

Resources

Map Marker by Andrew Onorato
Dressed for Iceland by Cécile Bernard
Moelwynion, Eryri, Cymru by Marc Popletton
Badge by Edward Boatman
Snail by aLf

Fonts

Geared Slab
Mission Script