

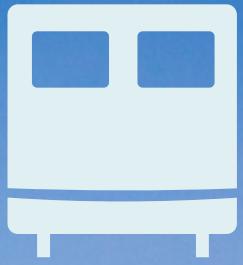
Getting around with

GOOGLE MAPS ANDROID API V2

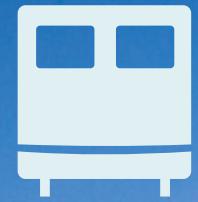
@ CYRILMOTTIER



capitaine train



capitaine train



Book and buy train tickets

<https://www.capitainetrain.com/search/>

capitaine train

Search Cart Tickets Help Cyril Mottier

Where can we take you?

From Paris

To Antwerpen-Centraal (Anvers)

Depart Saturday 19 October at 10 AM

Return

Passengers Cyril **GV Plus** **Jeune** Annick No card Antoine No card Cécile **Jeune** **Voyageur** Marie-Astrid **Jeune** **Voyageur** Mylène **Jeune** **Voyageur** Ophélie **GV**

Paris	
Paris-Gare-de-Lyon	Lyon → Paris
Paris-Gare-Montparnasse	Paris → Lyon
Aéroport-Paris-Roissy-Charles-de-Gaulle CDG	Brest → Marseille
Paris-Gare-du-Nord	Lyon → Angers St-Laud
Paris-Gare-de-l'Est	Rennes → Lyon
	Paris → Flers

I once made a

LOVELY & GOOD-LOOKING

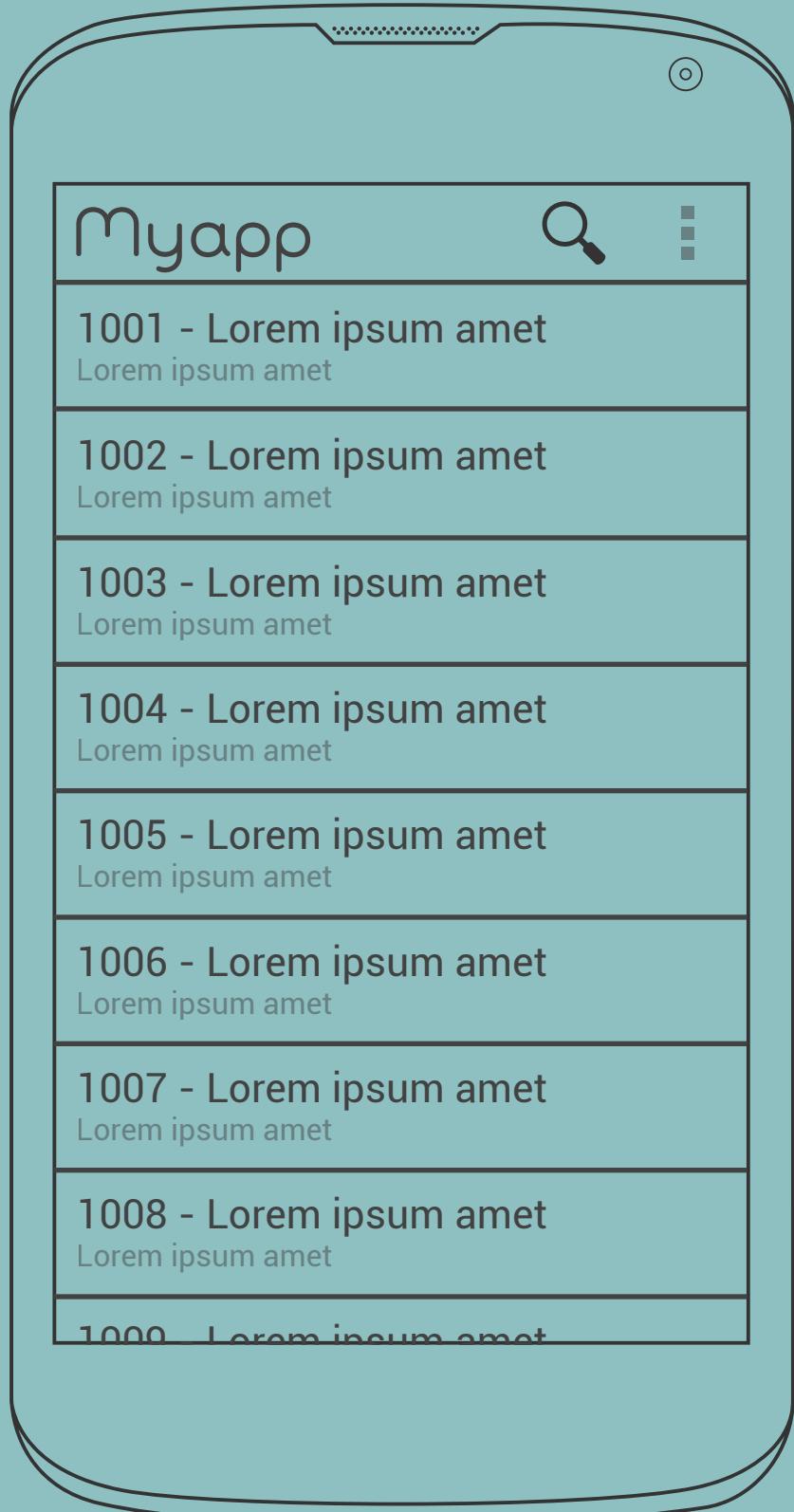
mobile app

I once made a

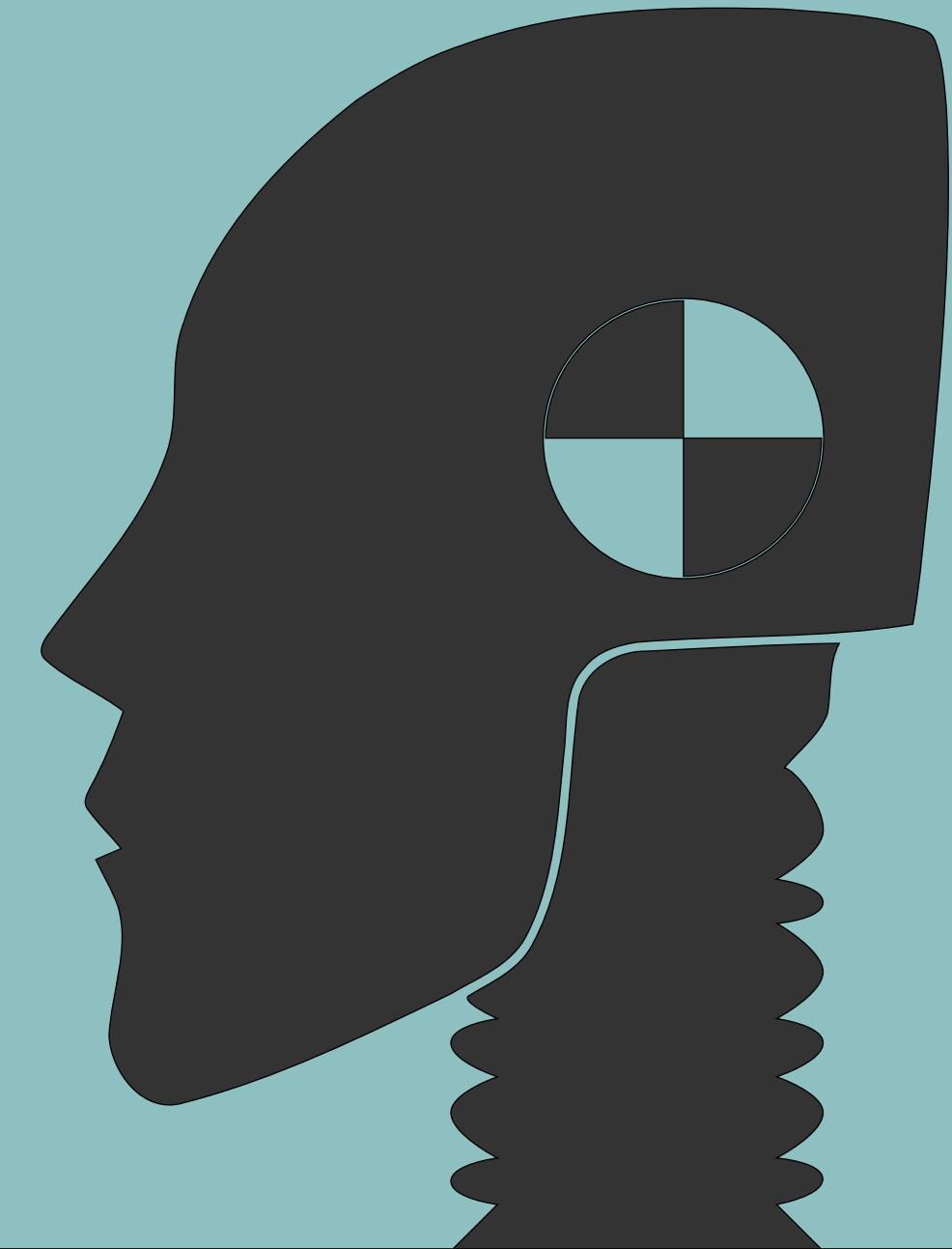
LOVELY & GOOD-LOOKING

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

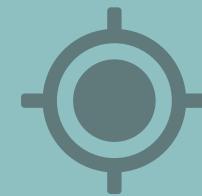


USER LOCATION

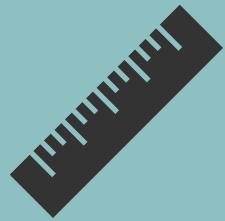
Maps give the user some clear context on



POI LOCATION



USER LOCATION



DISTANCES

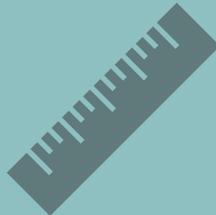
Maps give the user some clear context on



POI LOCATION



USER LOCATION



DISTANCE

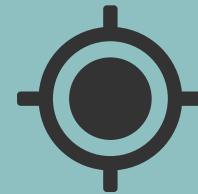


DIRECTIONS

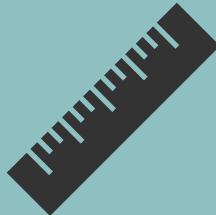
Maps give the user some clear context on



POI LOCATION



USER LOCATION



DISTANCES



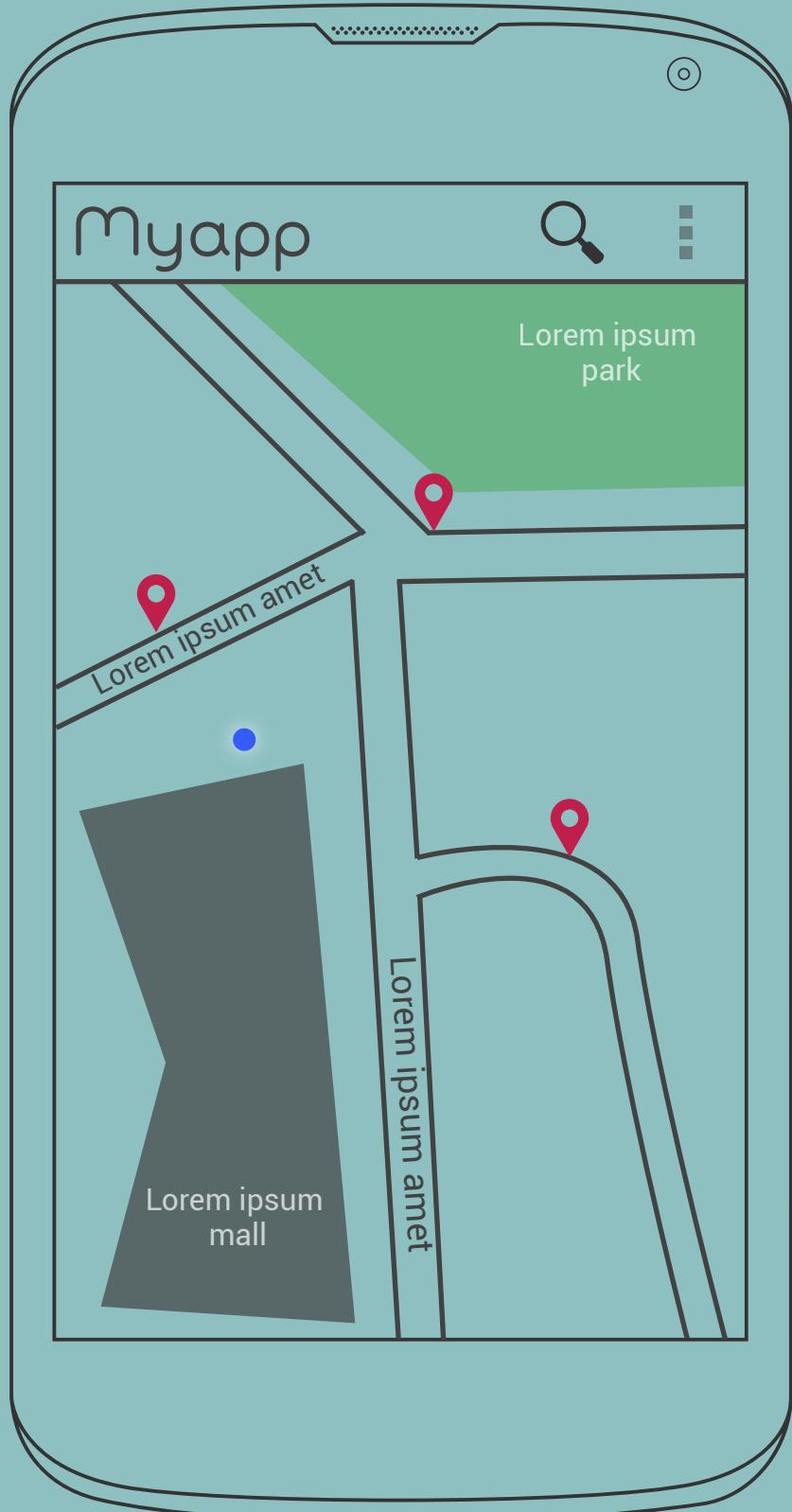
DIRECTIONS



GOOGLE MAPS ANDROID API V2 TO THE RESCUE

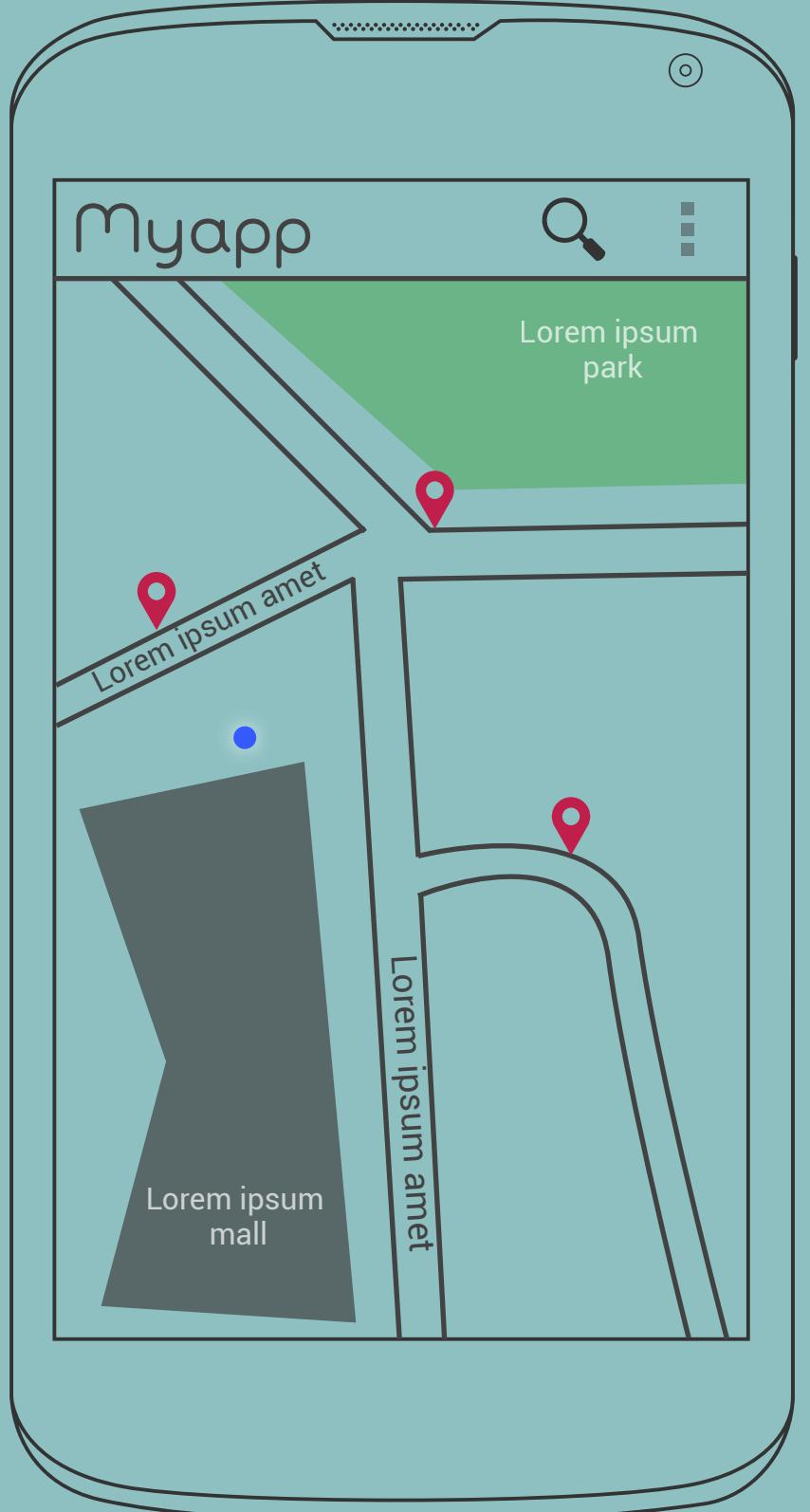
are immersive

Map-based apps



are immersive
and that's awesome :)

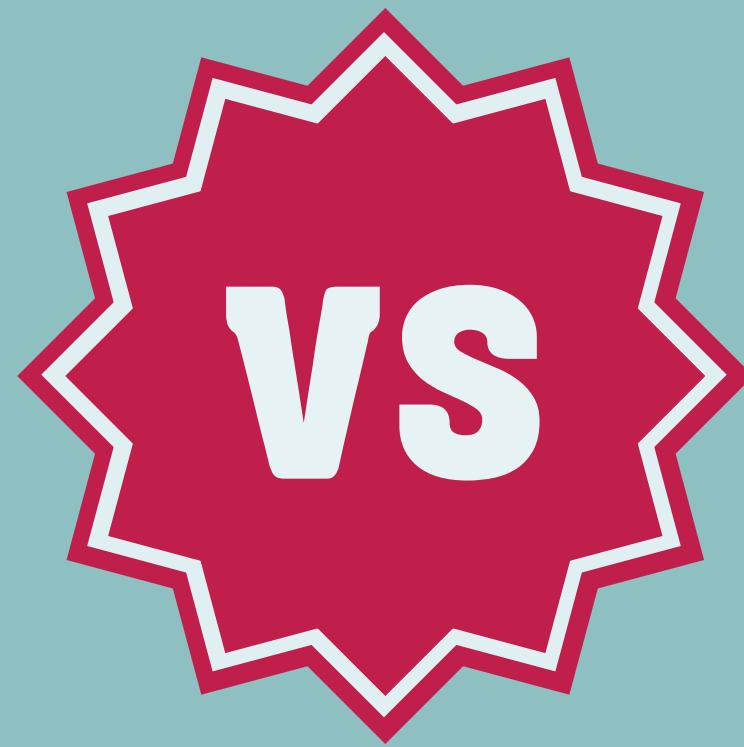
Map-based apps





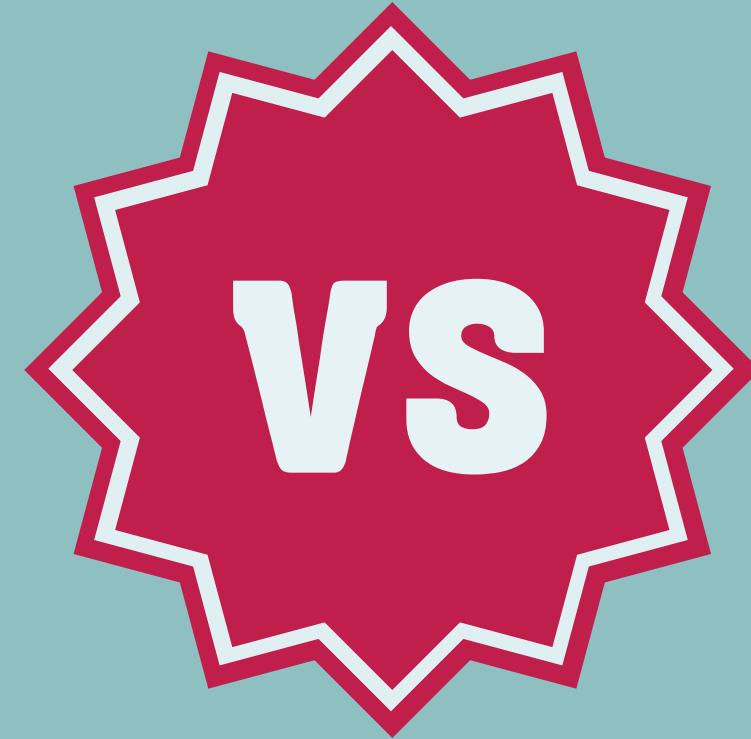
Introducing
MAPS API V2

v1



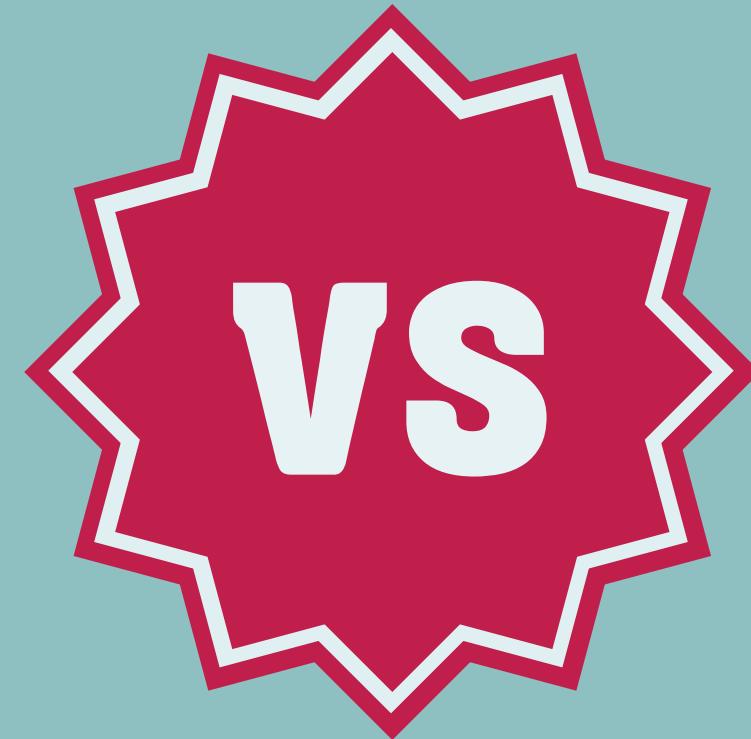
v2

v1



v2



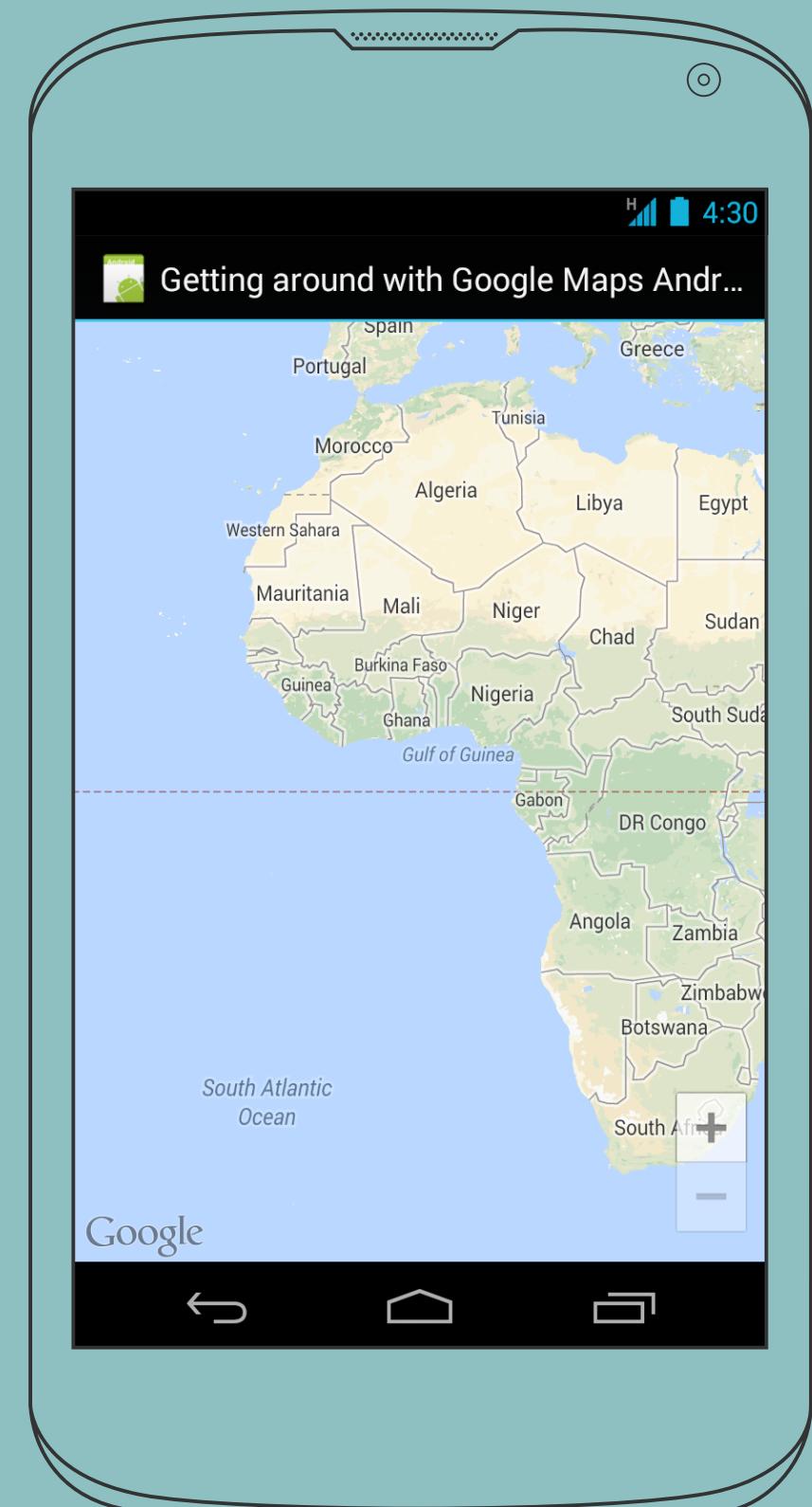


**EXTREMELY
STRAIGHTFORWARD**

A P I

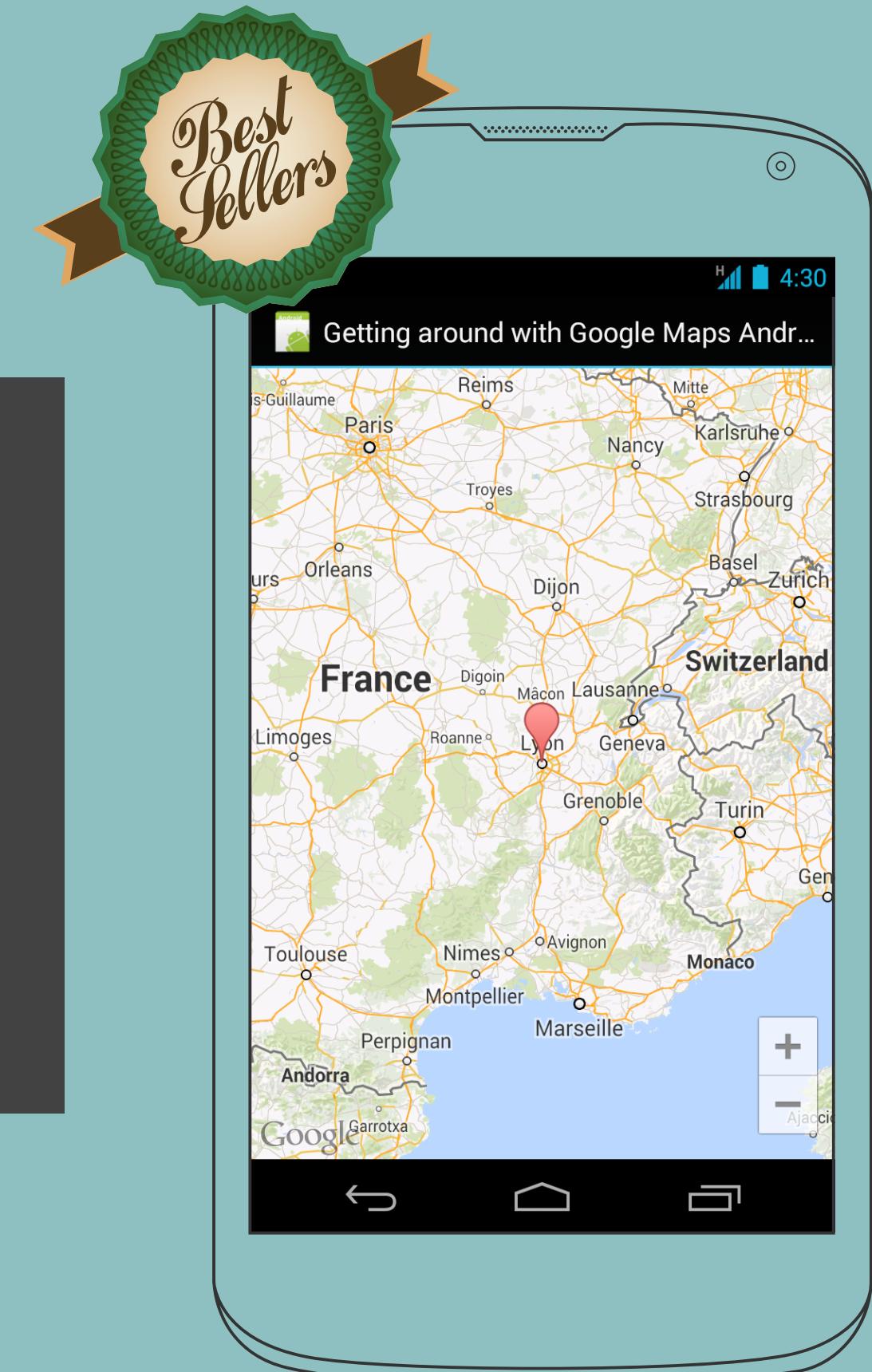
Initial [Support]MapFragment setup

```
1 public class TestActivity extends FragmentActivity {  
2     private GoogleMap mMap;  
3  
4     @Override  
5     protected void onCreate(Bundle savedInstanceState) {  
6         super.onCreate(savedInstanceState);  
7         setContentView(R.layout.activity_marker);  
8         setUpMapIfNeeded();  
9     }  
10  
11    @Override  
12    protected void onResume() {  
13        super.onResume();  
14        setUpMapIfNeeded();  
15    }  
16  
17    private void setUpMapIfNeeded() {  
18        if (mMap == null) {  
19            mMap = ((SupportMapFragment) getSupportFragmentManager().  
20                  findFragmentById(R.id.map)).getMap();  
21        if (mMap != null) {  
22            setUpMap();  
23        }  
24    }  
25  
26    private void setUpMap() { /* TODO */ }  
27  
28 }
```



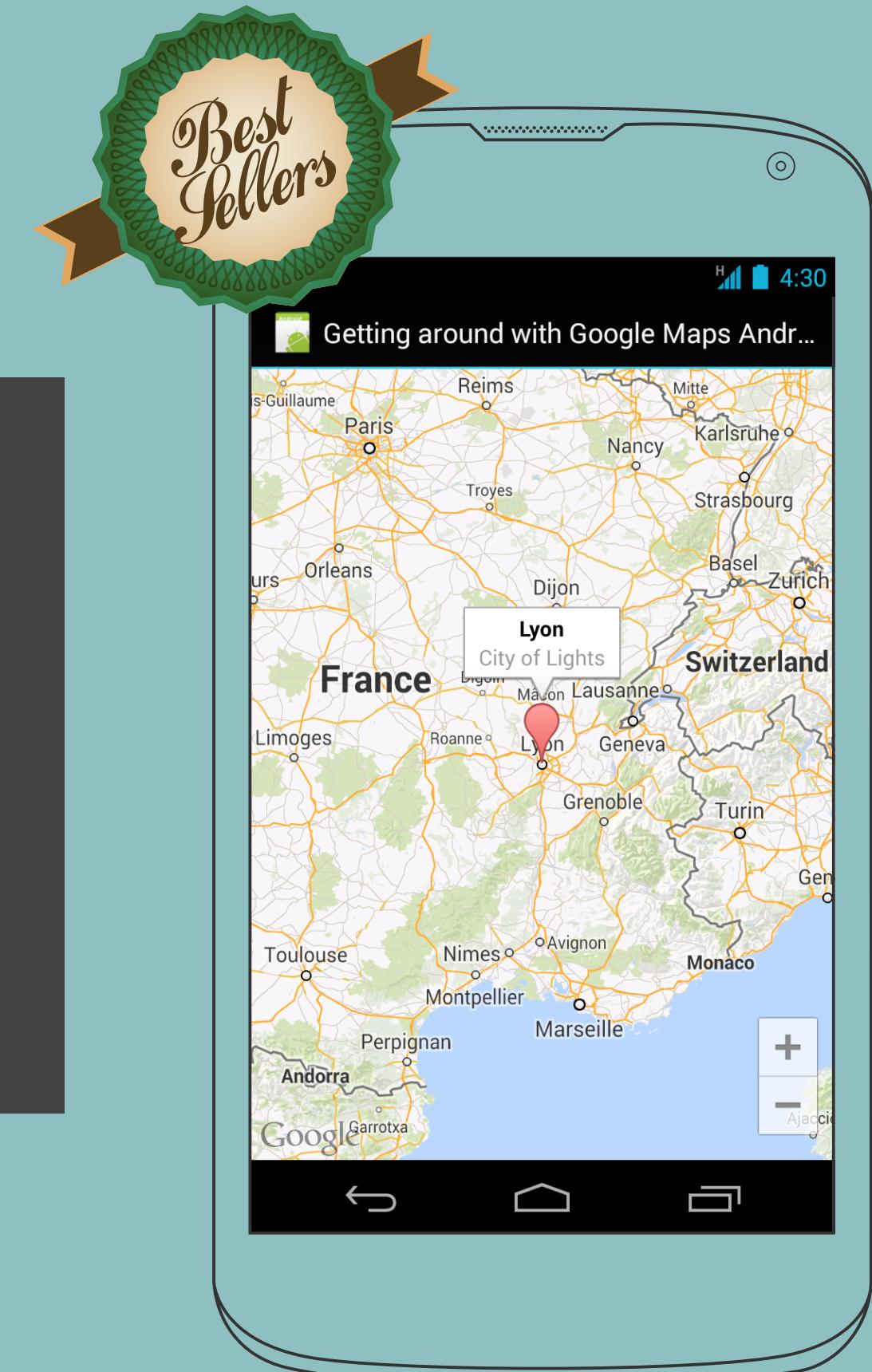
Annotating with Markers

```
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

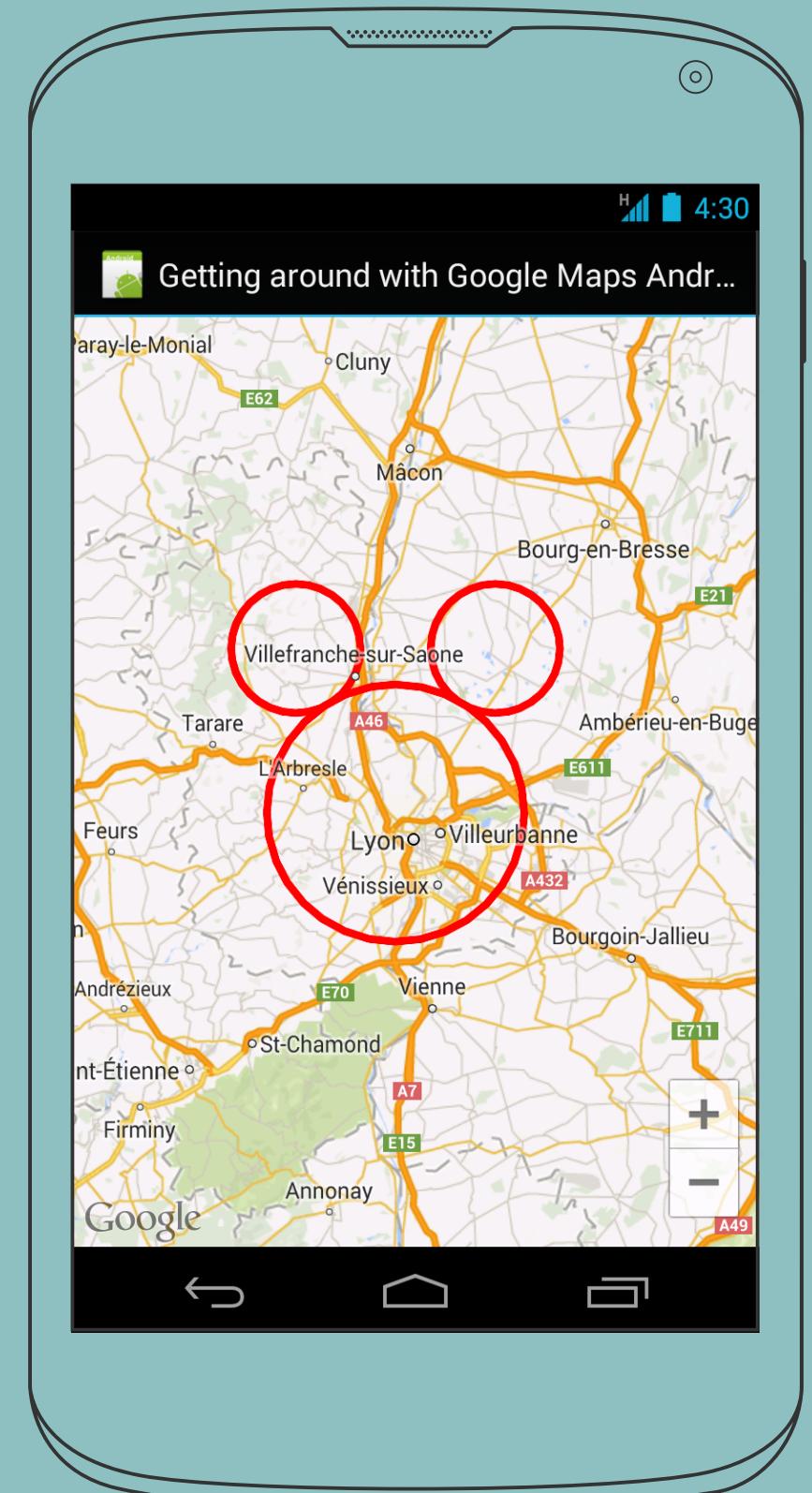
Drawing shapes on **MapView**

3 POSSIBLE SHAPES

Polyline | Polygon | Circle

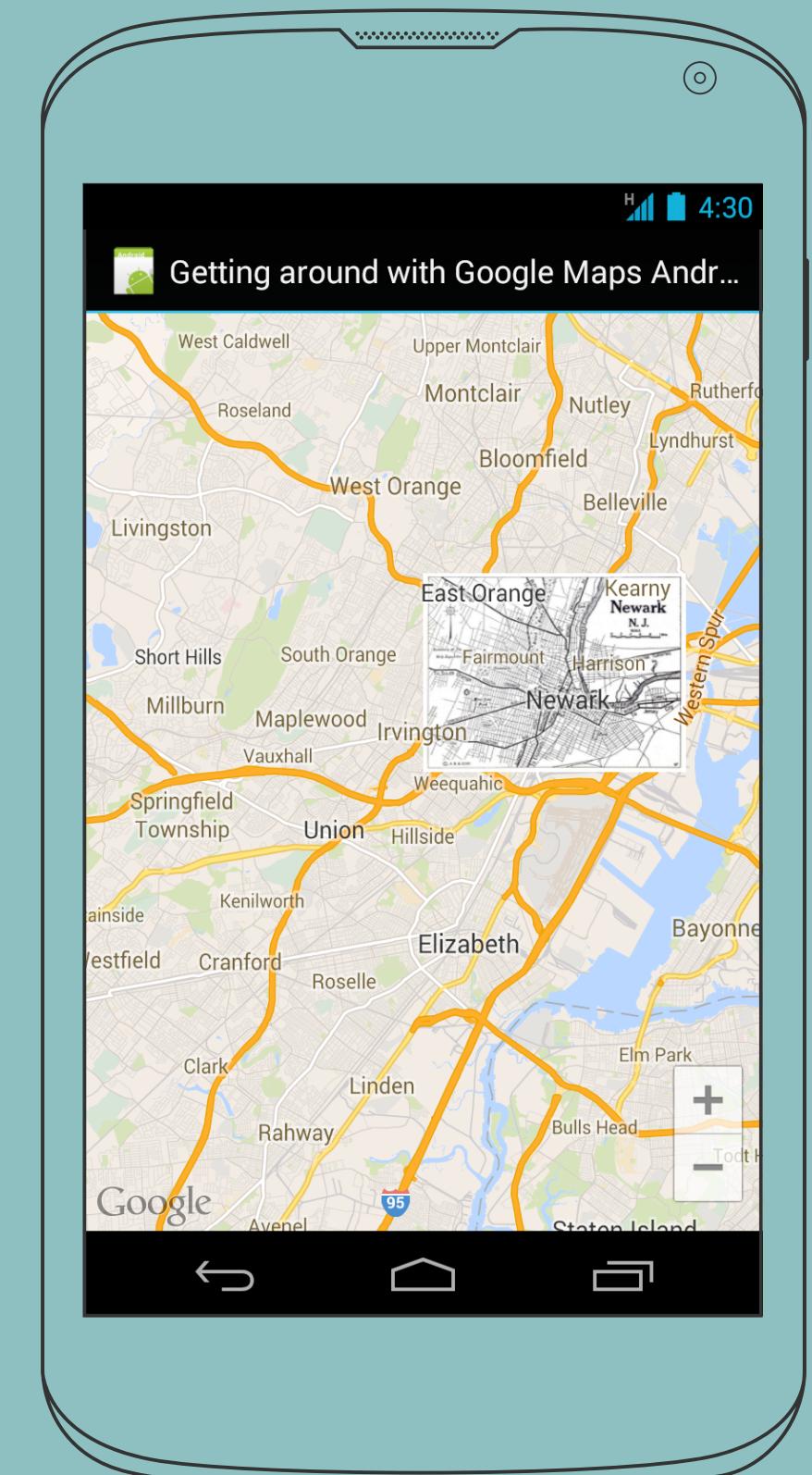
Drawing shapes on MapView

```
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 }
```



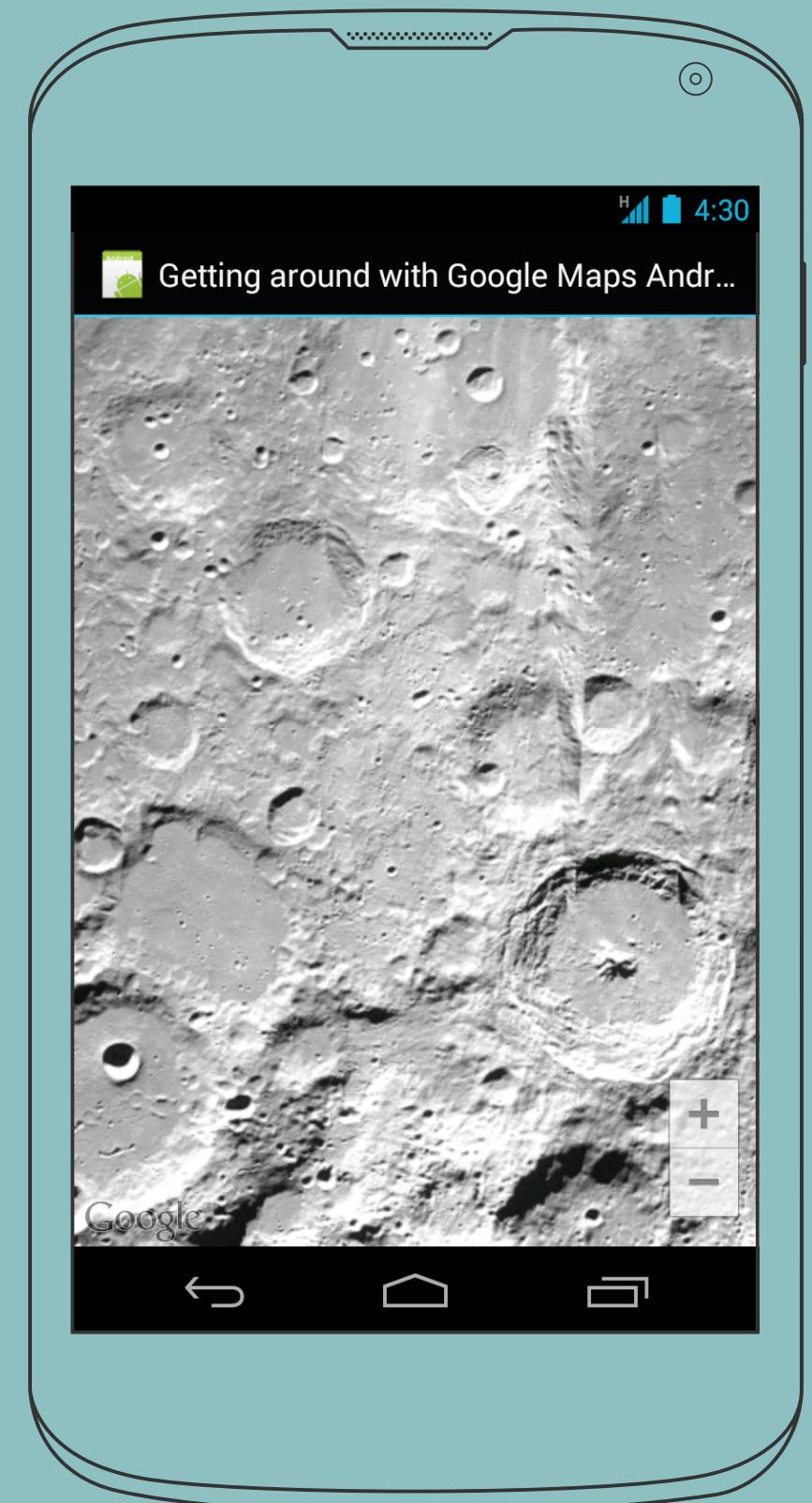
Overlaying map with GroundOverlay

```
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 }
```



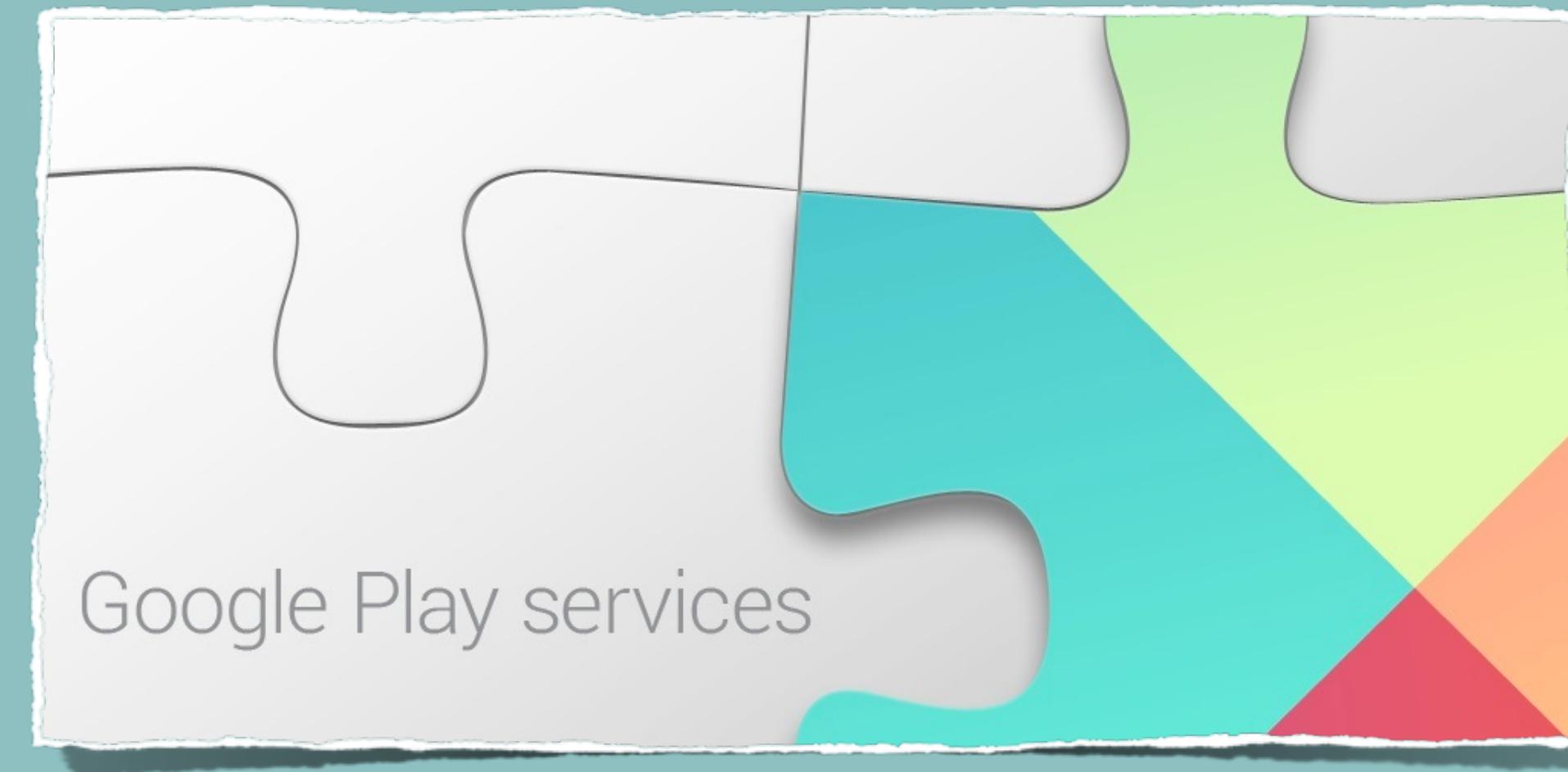
Rendering custom tiles with TileOverlay

```
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 }
```

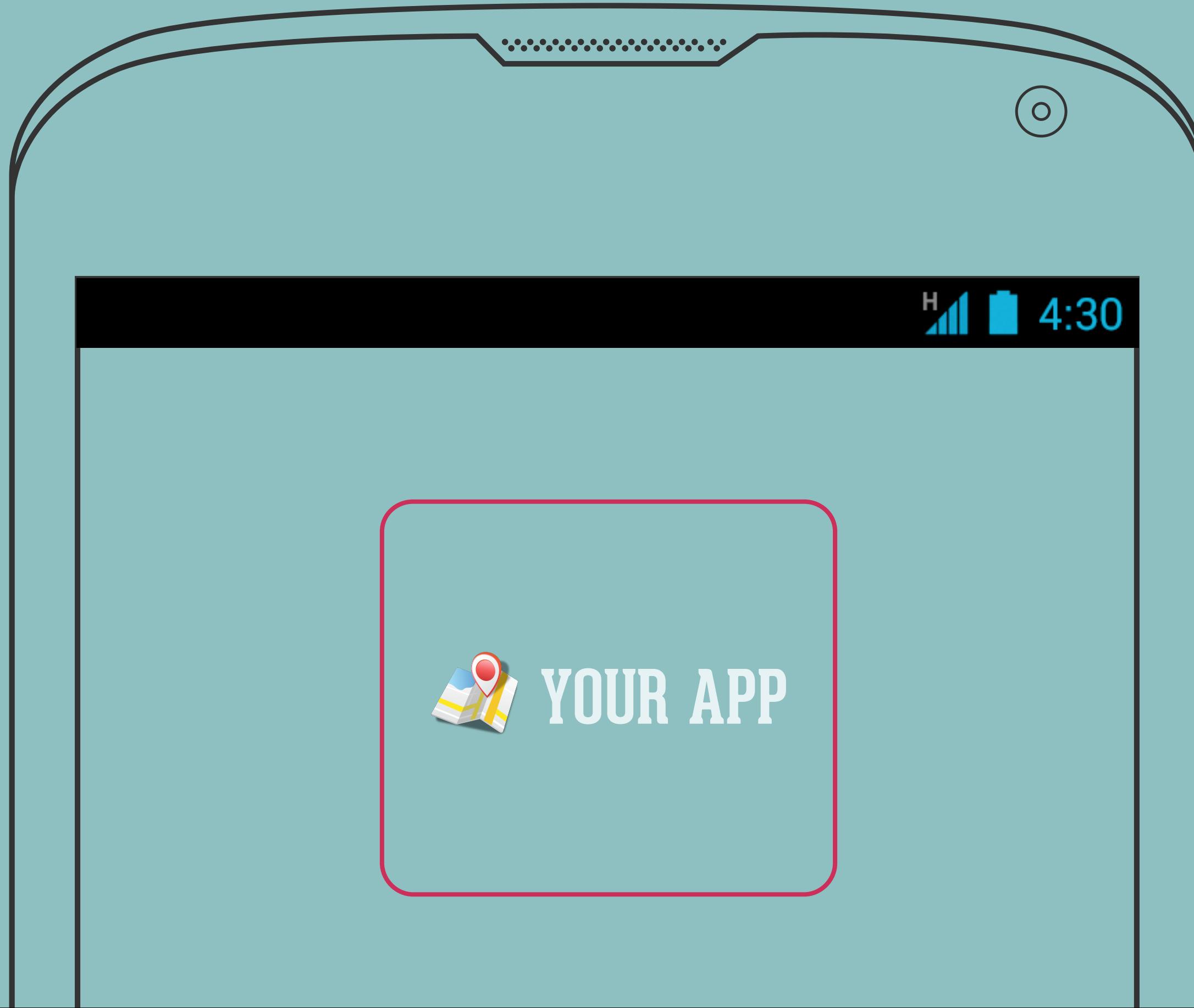


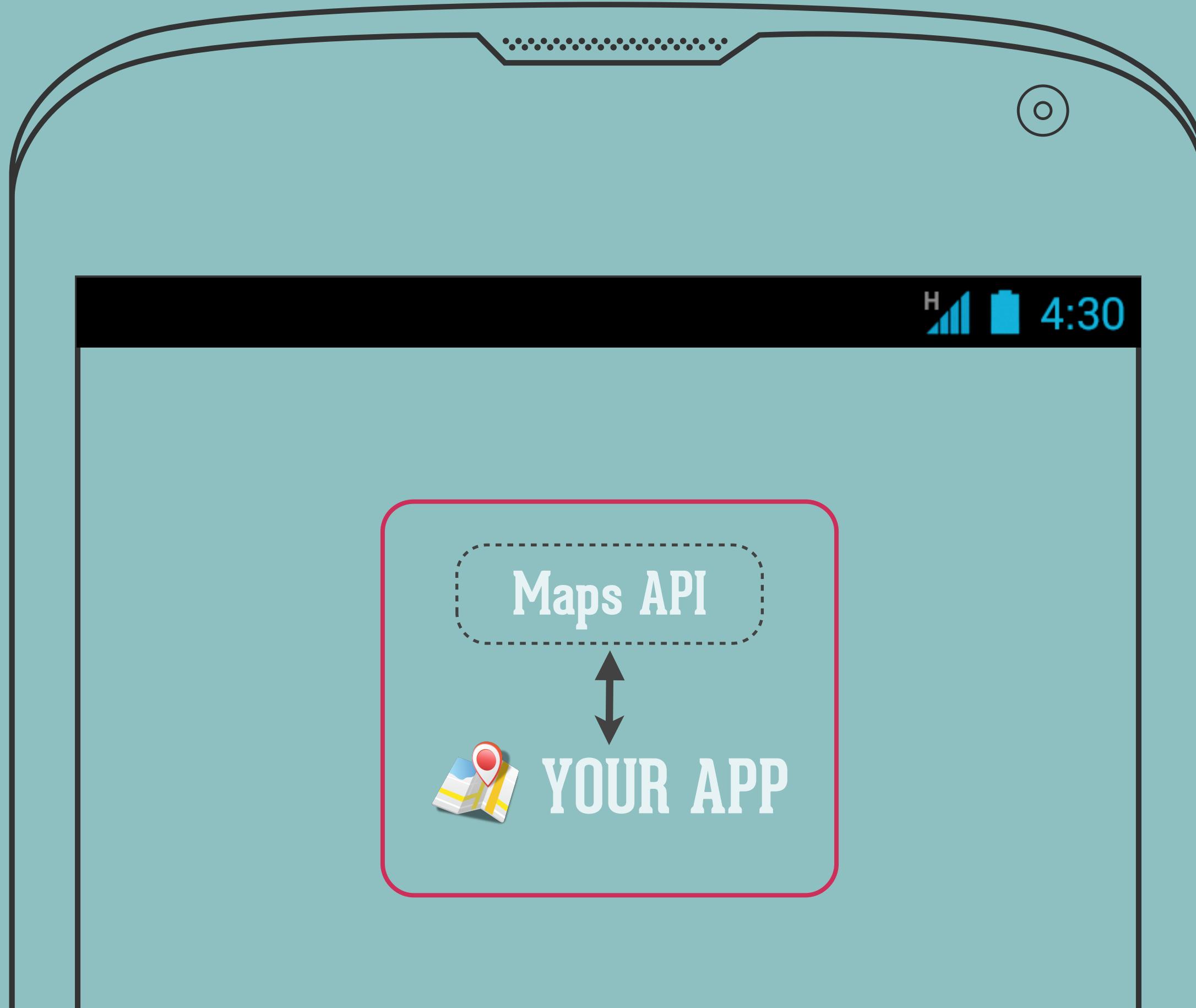


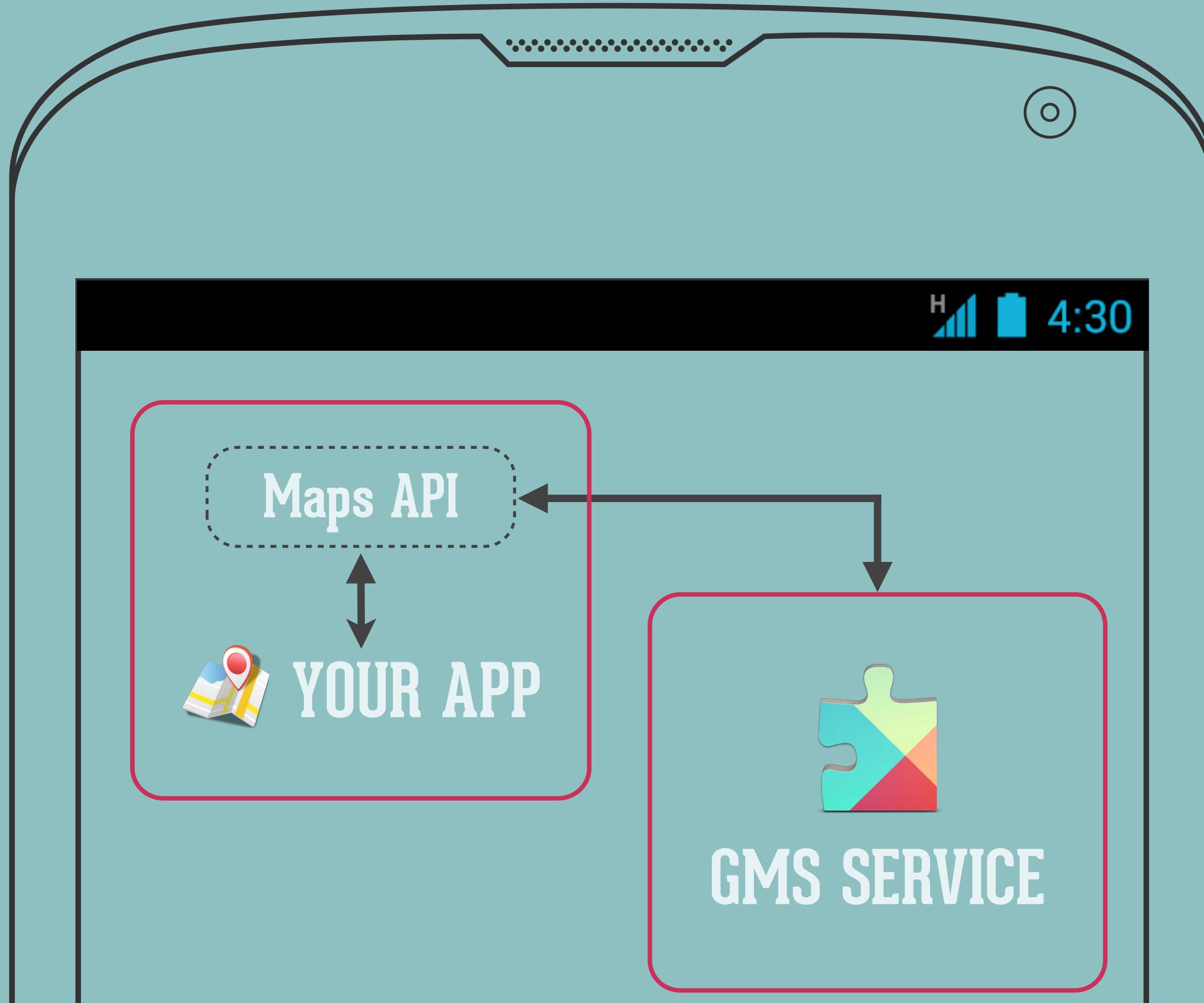
Deep dive into
THE INTERNALS

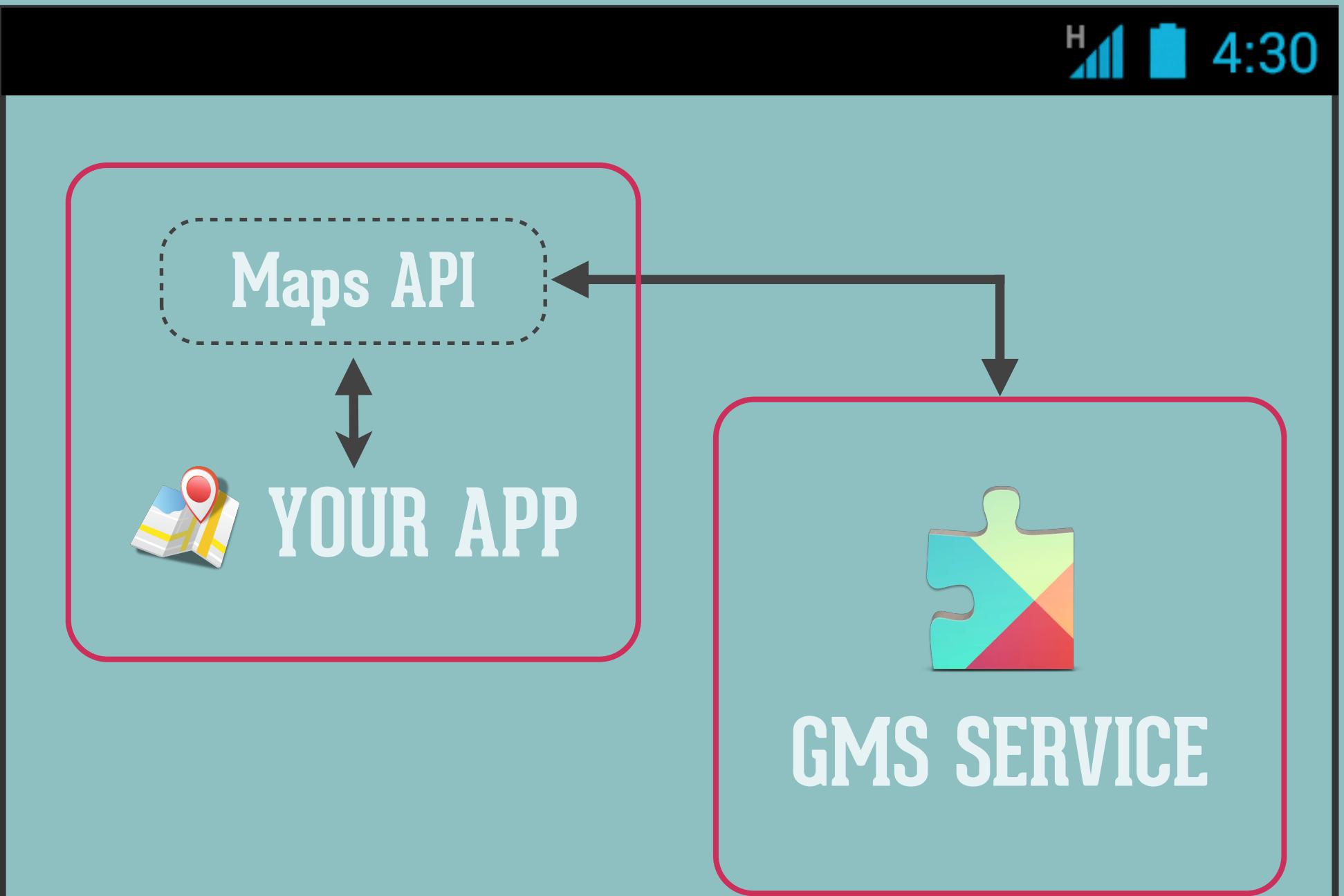


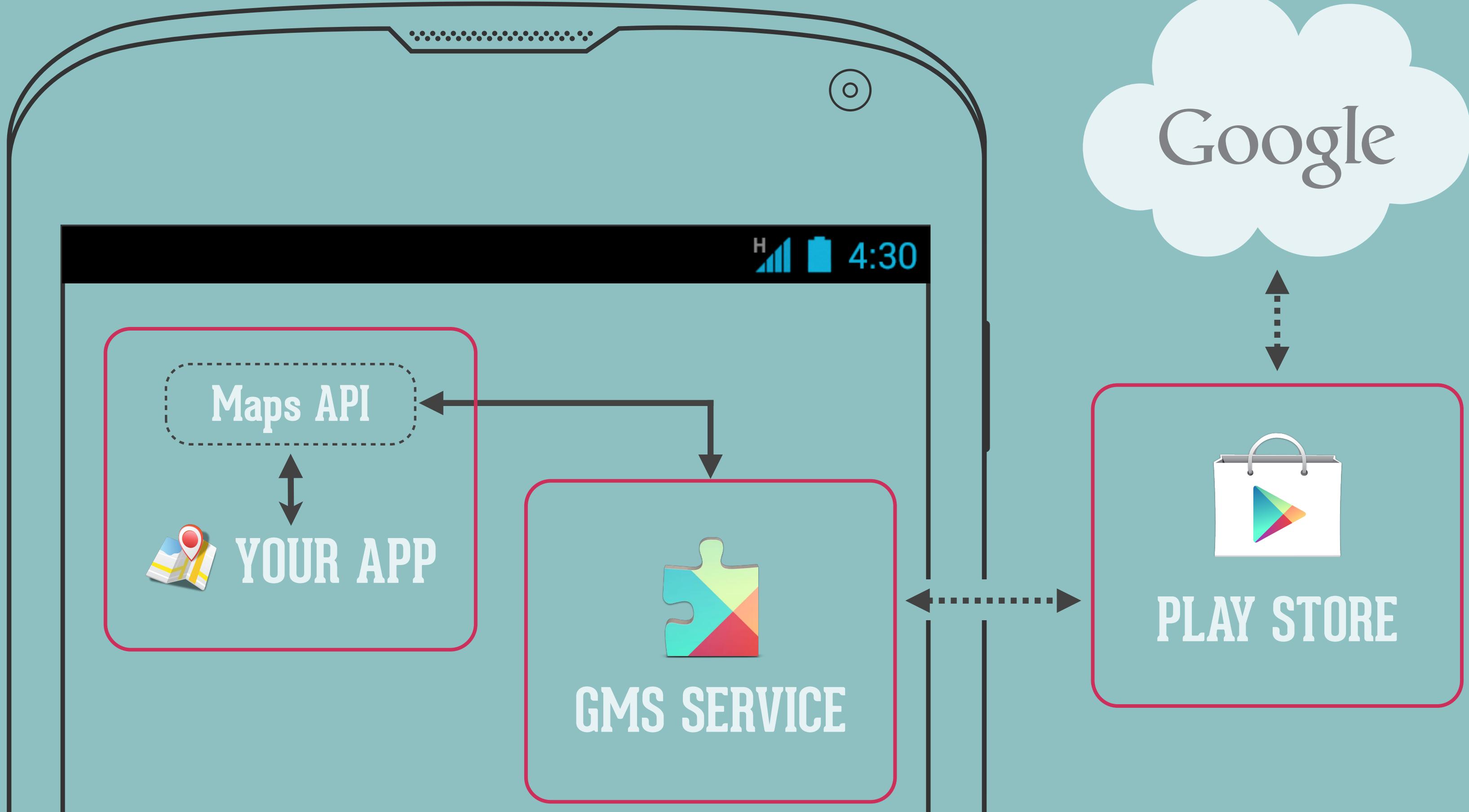
Google Maps Android API v2
is part of
Google Play Services











**EVERYTHING
EVERYTHING
EVERYTHING
EVERYTHING IS PARCELABLE**

MarkerOptions | Android X

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**

Summary: Inherited Constants | Fields | Ctors | Methods | Inherited Methods | [Expand All]

extends Object

implements Parcelable

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

MarkerOptions

extends Object

implements Parcelable

java.lang.Object

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

Summary: Inherited Constants

Class Overview

public final class

MarkerOptions

extends Object

implements **Parcelable**

java.lang.Object

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

Summary: Inherited Constants

Class Overview

*Each call to the API
is a done through a Binder ...*

... and it may be





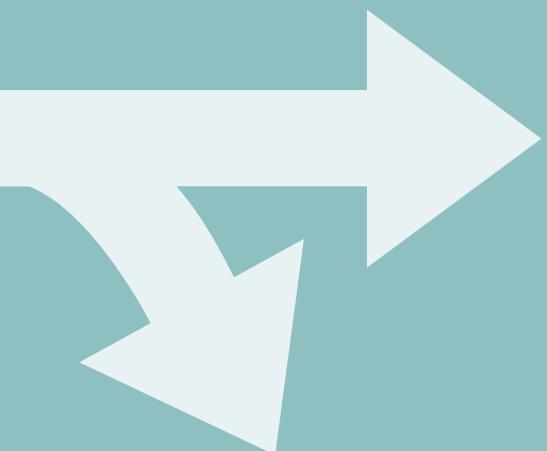
A VIEW
MAY HIDE
ANOTHER



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

The screenshot shows a web browser window displaying the Android Developers website at developer.android.com/reference/com/google/android/gms/maps/model/Marker.html. The page is titled "Marker" and is part of the "Google Services" section. The left sidebar lists various classes and interfaces, including `BitmapDescriptorFactory`, `CameraPosition`, `CameraPosition.Builder`, `Circle`, `CircleOptions`, `GroundOverlay`, `GroundOverlayOptions`, `LatLng`, `LatLngBounds`, `LatLngBounds.Builder`, `Marker` (which is highlighted in blue), `MarkerOptions`, `Polygon`, `PolygonOptions`, `Polyline`, `PolylineOptions`, `Tile`, `TileOverlay`, and `TileOptions`. The main content area provides a detailed overview of the `Marker` class, stating it is a public final class that extends `Object` and implements `java.lang.Object`. It also includes a link to the `Marker` interface. The "Class Overview" section describes a marker as an icon placed on a map's surface, oriented against the device's screen. It then lists properties: `Anchor` (the point on the image placed at the `LatLang` position), `Position` (the `LatLang` value for the marker's position), and `Title`.

Marker | Android Developers

developer.android.com/reference/com/google/android/gms/maps/model/Marker.html

Developers | Design | Develop | Distribute

Training API Guides Reference Tools Google Services

Summary: Methods | Inherited Methods | [Expand All]

Marker

extends Object

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

Class Overview

An icon placed at a particular point on the map's surface. A marker icon is drawn oriented against the device's screen rather than the map's surface; i.e., it will not necessarily change orientation due to map rotations, tilting, or zooming.

A marker has the following properties:

Anchor

The point on the image that will be placed at the `LatLang` position of the marker. This defaults to 50% from the left of the image and at the bottom of the image.

Position

The `LatLang` value for the marker's position on the map. You can change this value at any time if you want to move the marker.

Title

public final class

Marker

extends Object

java.lang.Object

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

scriptorFactory
position
position.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 }
```

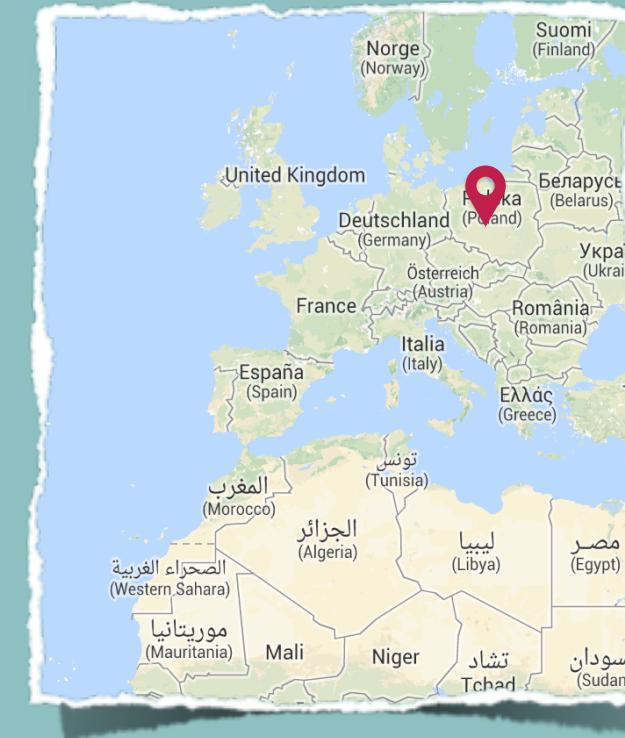
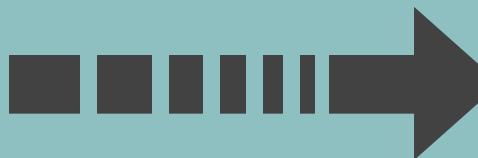
Attaching information to Markers

```
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(getApplicationContext() /* ... */);
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

b lazy when possible
mindful on memory
nice with users

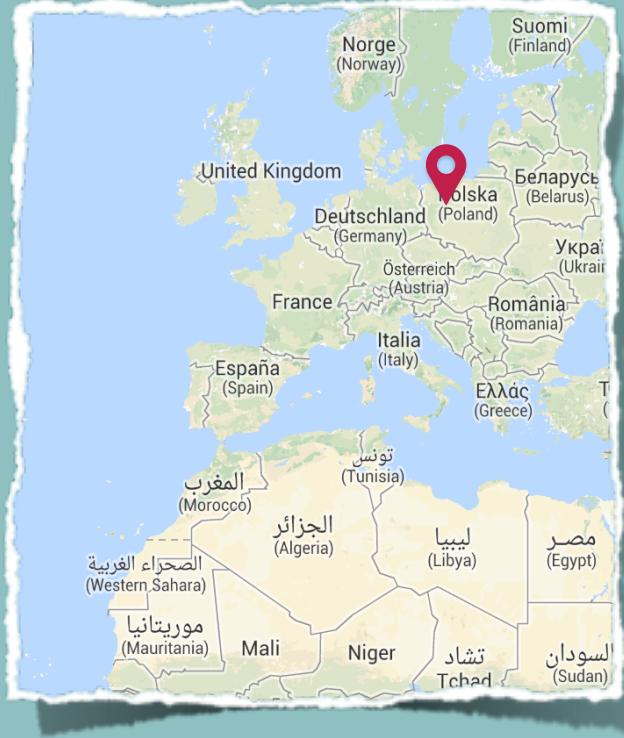
```
1 private void setUpMap() {
2     mMap.setInfoWindowAdapter(new GoogleMap.InfoWindowAdapter() {
3         @Override
4         public View getInfoWindow(Marker marker) {
5             return null;
6         }
7
8         @Override
9         public View getInfoContents(Marker marker) {
10            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!

```
1 public class LatLngEvaluator implements TypeEvaluator<LatLng> {  
2  
3     @Override  
4     public LatLng evaluate(float fraction, LatLng startValue, LatLng endValue) {  
5         double endValueLng = endValue.longitude;  
6  
7         // Take the shortest path across the 180th meridian.  
8         double lngDelta = startValue.longitude - endValue.longitude;  
9         if (Math.abs(lngDelta) >= 180) {  
10             endValueLng = endValue.longitude + Math.signum(lngDelta) * 360;  
11         }  
12  
13         double newLat = eval(fraction, startValue.latitude, endValue.latitude);  
14         double newLng = eval(fraction, startValue.longitude, endValueLng);  
15         return new LatLng(newLat, newLng);  
16     }  
17  
18     private static double eval(float fraction, double startValue, double endValue) {  
19         return startValue + fraction * (endValue - startValue);  
20     }  
21 }
```

```
1 private static final LatLngEvaluator LAT_LNG_EVALUATOR = new LatLngEvaluator();
2 private static final Property<Marker, LatLng> MARKER_POSITION_PROPERTY =
3     Property.of(Marker.class, LatLng.class, "position");
4
5 private static final LatLng LYON = new LatLng(45.764043, 4.835659);
6 private static final LatLng PARIS = new LatLng(48.856614, 2.3522219);
7
8 private LatLng mCurrentDest = LYON;
9 private Marker mMarker;
10 private ObjectAnimator mMarkerAnimator;
11
12 private void setUpMap() {
13     mMap.setOnMarkerClickListener(mOnMarkerClickListener);
14     mMarker = mMap.addMarker(new MarkerOptions().position(mCurrentDest));
15 }
16
17
18
19
20
21
22
23
```

```
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 Poppletton

Badge by Edward Boatman

Snail by aLf

Fonts

Geared Slab

Mission Script