

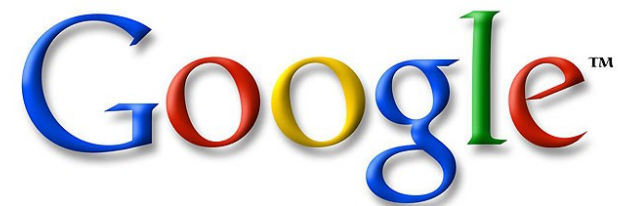
Introduction to Android SMS & Broadcast receiver

CS 436 Software Development on Mobile

By Dr.Paween Khoenkaw

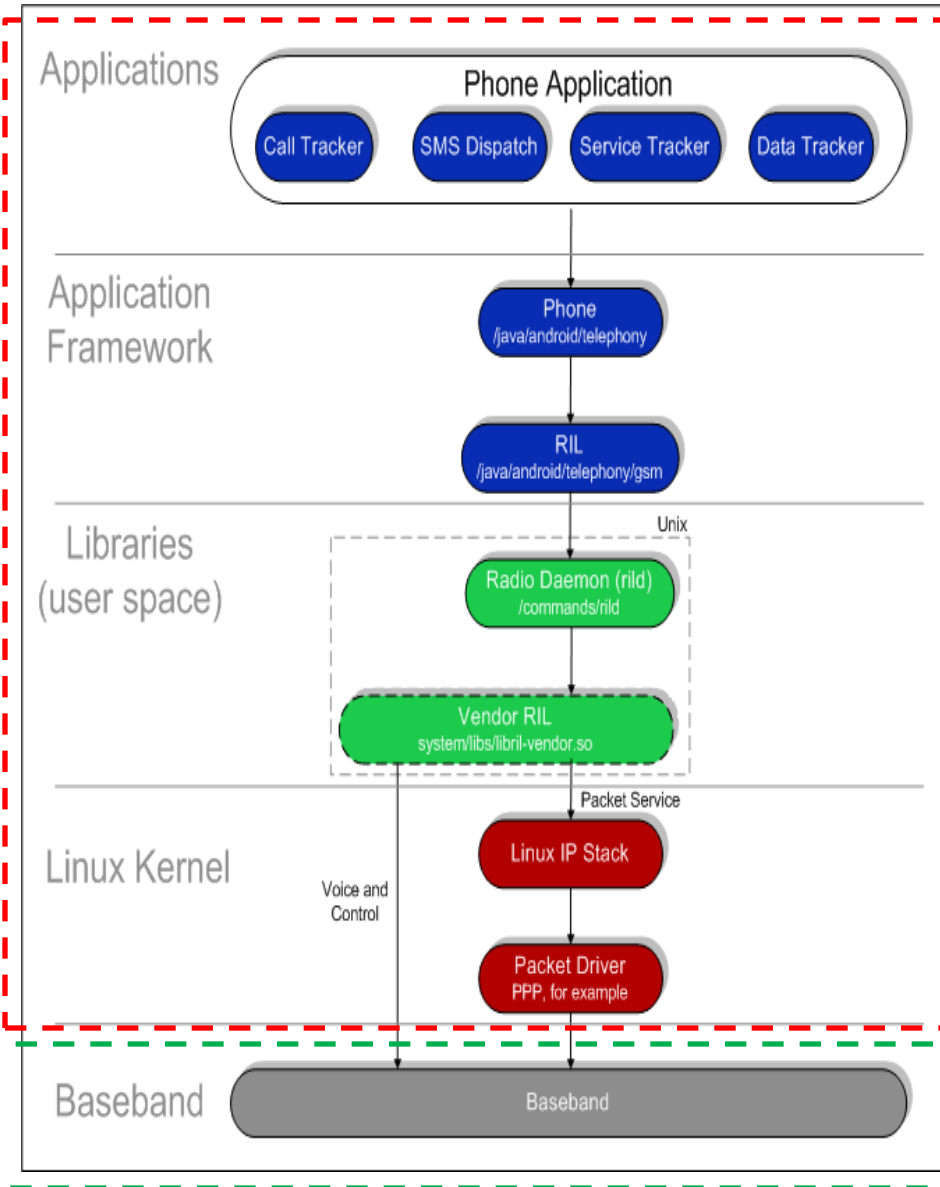


ANDROID

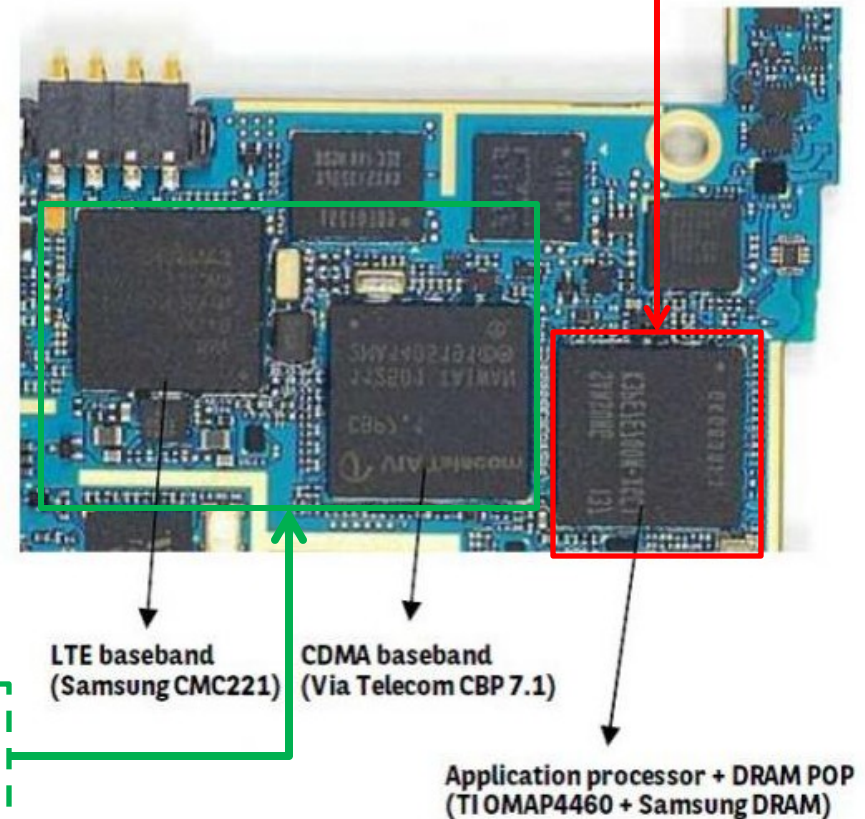
The image shows the Google logo, featuring the word "Google" in its characteristic multi-colored font (blue, red, yellow, blue, green, red) with a trademark symbol.

SMS Manager

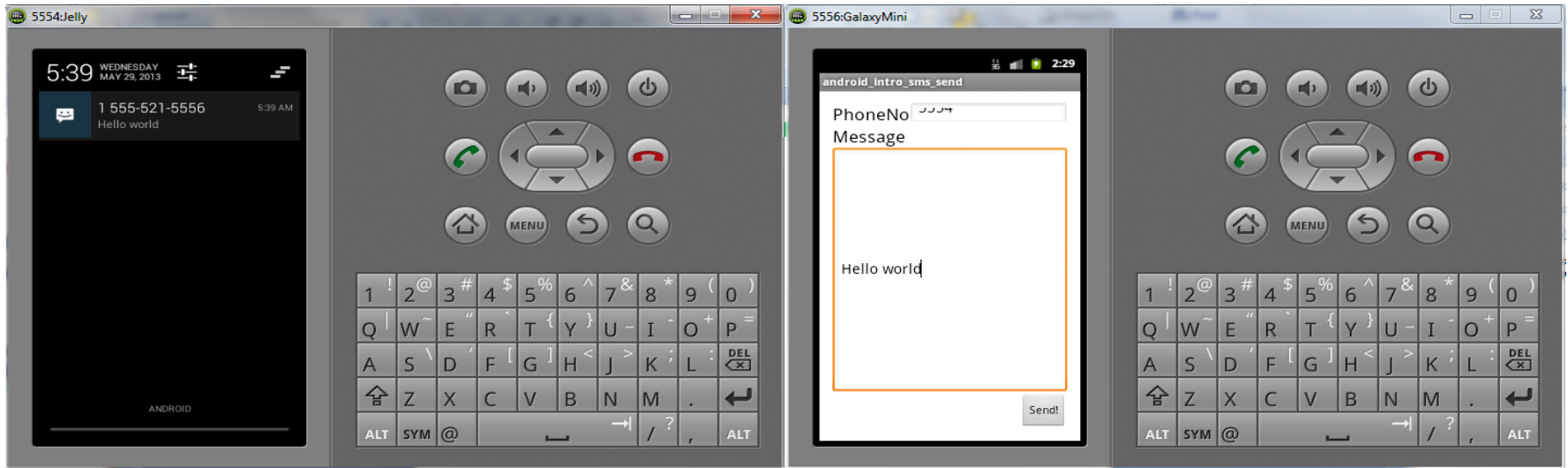
Radio Layer Interface



Galaxy Nexus



SMS Manager



Project: Android_intro_sms_send

SMS Manager

Send SMS in Android is simple!

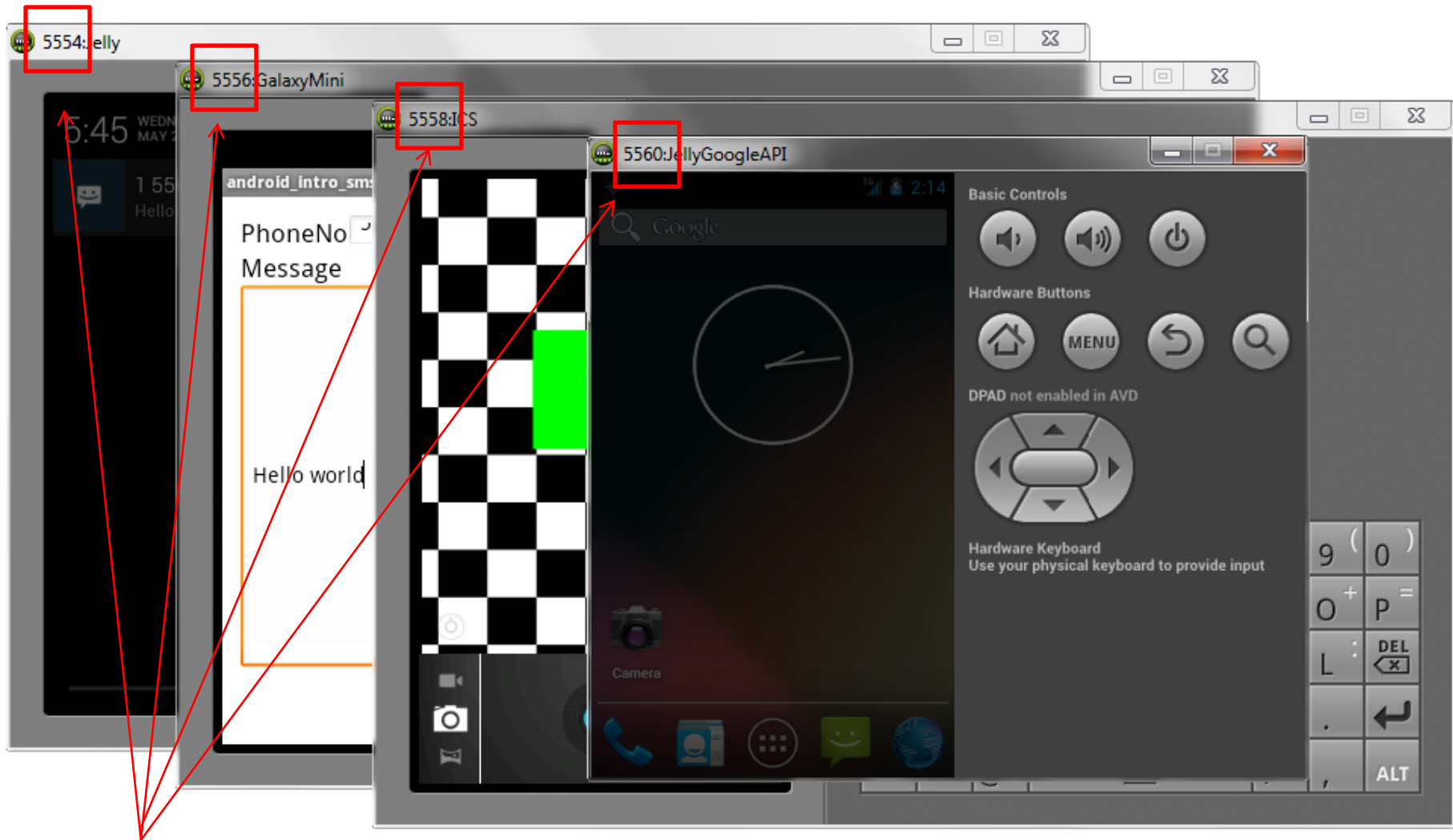
```
SmsManager.sendTextMessage (String destinationAddress, String scAddress,  
String text, PendingIntent sentIntent, PendingIntent deliveryIntent)
```

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

Example code:

```
SmsManager sms = SmsManager.getDefault();  
sms.sendTextMessage("089123456", null, "Hello world", null, null);
```

SMS Manager

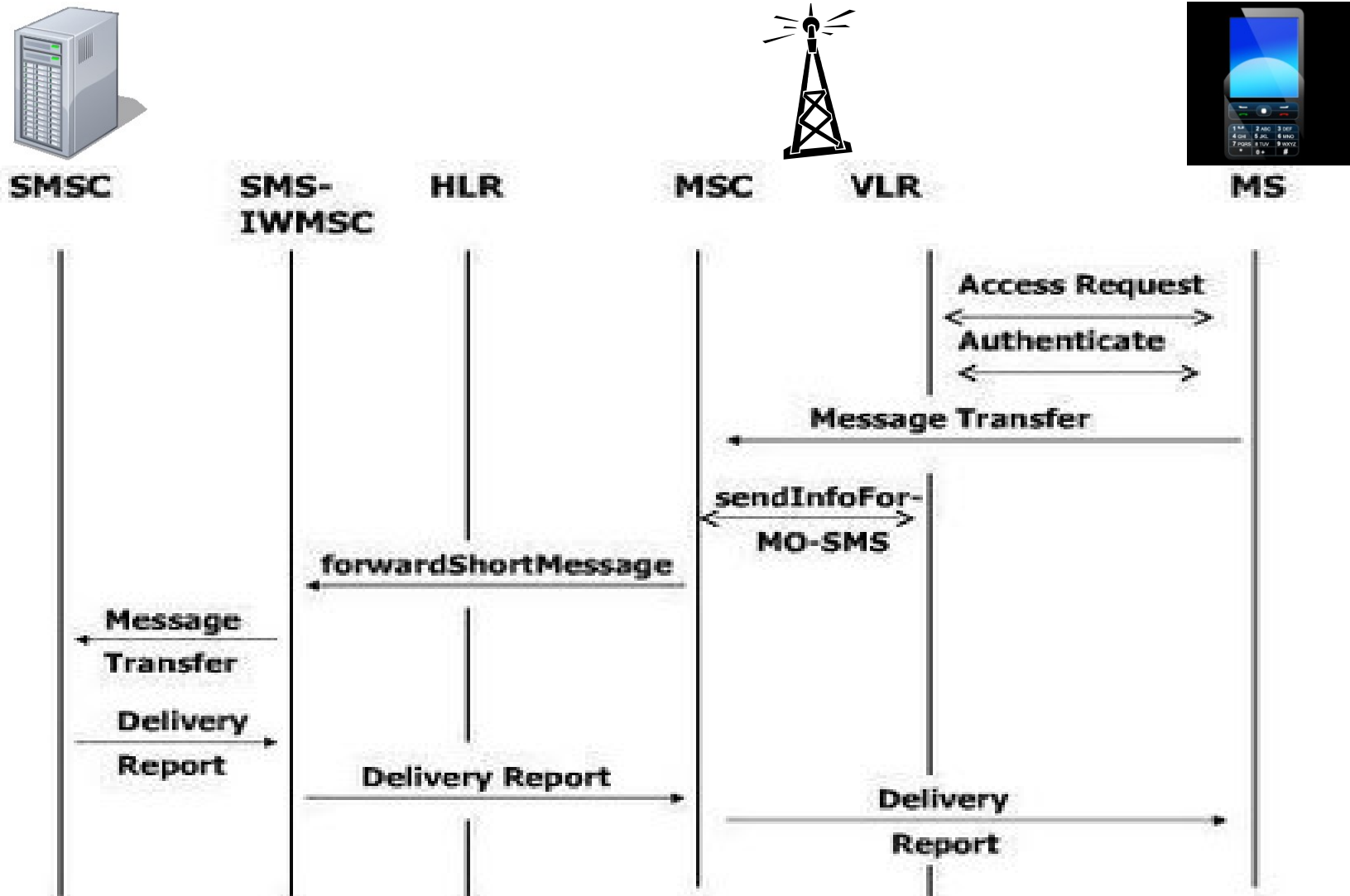


Emulator 's phone number

Can we track SMS sent result ?

How to track SMS sent result

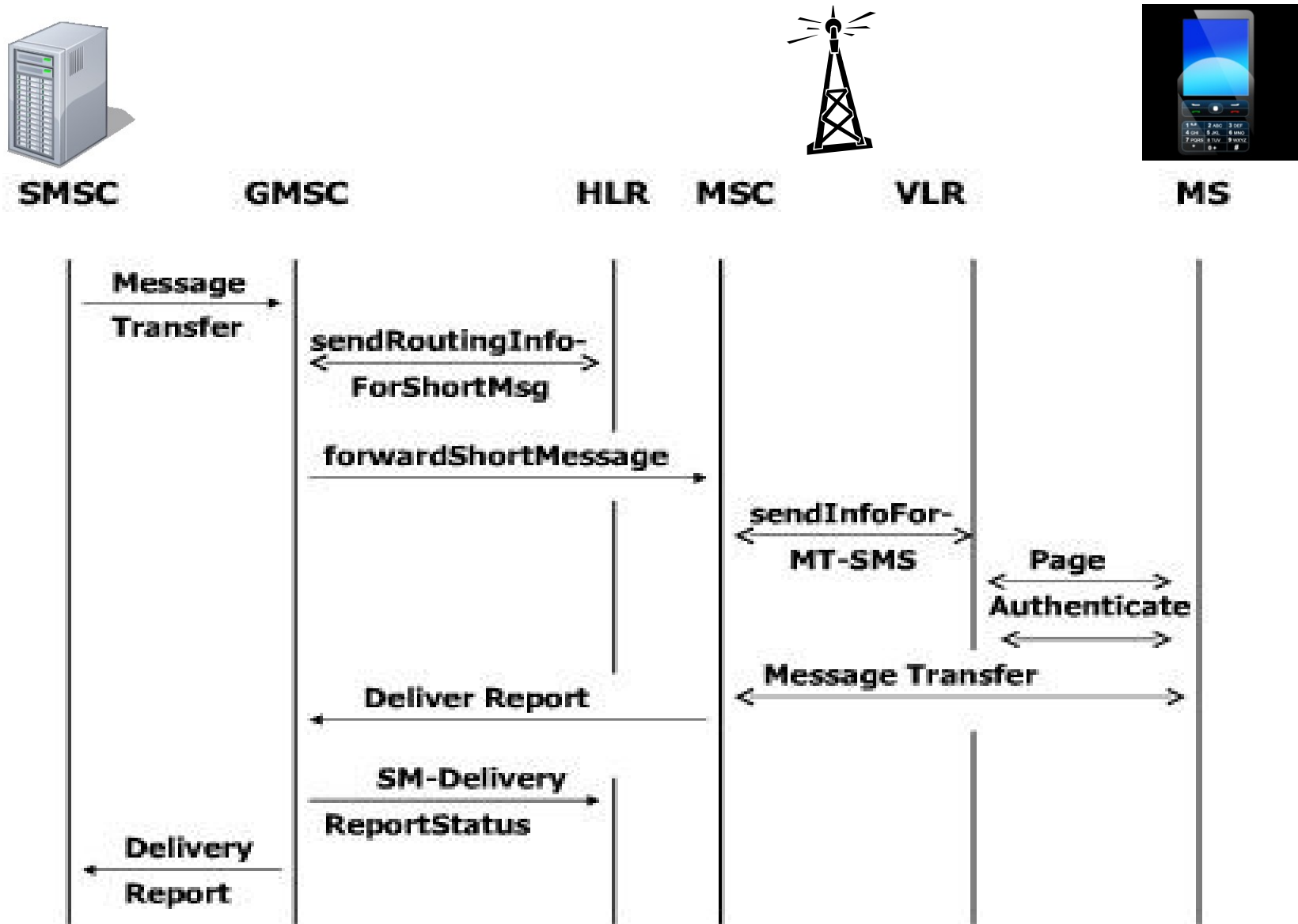
Mobile Originated (MO) SMS Flow



Mobile Origination (MO) SMS Flow

How to track SMS sent result

Mobile Terminated (MT) SMS Flow



How to track SMS sent result

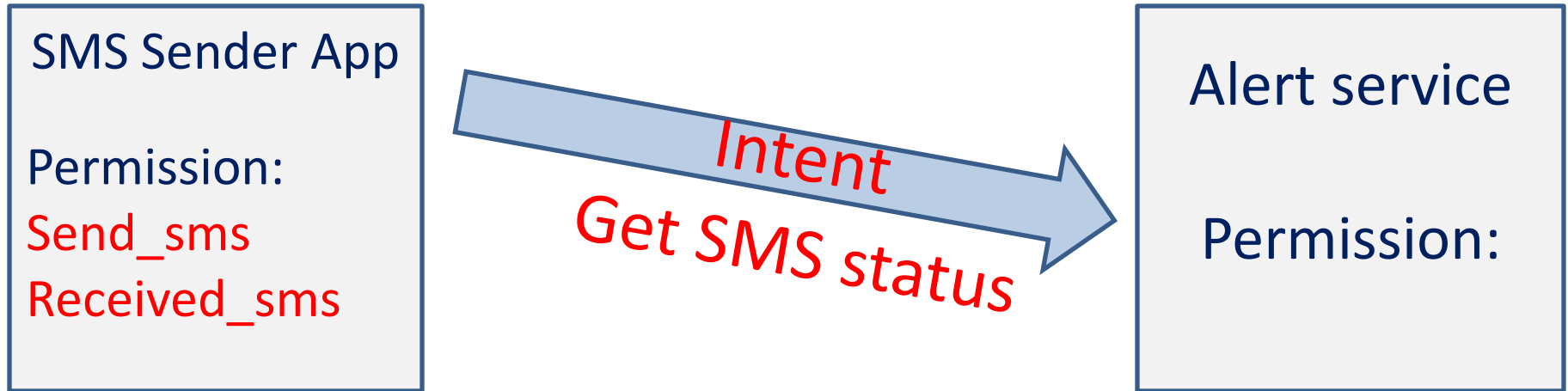
We request SMS delivery status using
Pending Intent & Broadcast receiver

```
SmsManager.sendTextMessage (String destinationAddress, String scAddress,  
String text, PendingIntent sentIntent, PendingIntent deliveryIntent)
```

```
<uses-permission android:name="android.permission.SEND_SMS"/>  
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
```

Pending intent

By giving a PendingIntent to another application, you are granting it the right to perform the operation you have specified as if the other application was yourself



Error: Permission denied!

Pending intent

By giving a PendingIntent to another application, you are granting it the right to perform the operation you have specified as if the other application was yourself



SMS delivered!

Pending intent

```
public static PendingIntent getBroadcast (Context context, int  
requestCode, Intent intent, int flags)
```

Example code:

```
PendingIntent sentPI = PendingIntent.getBroadcast(  
this, 0, new Intent("sms sent"), 0);
```

This pending intent sentPI was ready to broadcast !!

Broadcast receiver

Broadcast receiver

A **broadcast receiver** is an Android component which allows you to register for system or application events. All registered receivers for an event will be notified by the Android runtime once this event happens.

- Received Intents from Application and OS
- Activate task / application
- Auto start application after boot
- Static implement (*AndroidManifest*)
- Dynamic implement (*Context.registerReceiver()*)

Broadcast receiver

Receiver class

```
private class myReceiver extends BroadcastReceiver
{
    @Override
    public void onReceive(Context context, Intent intent) {
        Do your task here !!!
    }
}
```

Register the Receiver class

```
registerReceiver(new myReceiver(), new IntentFilter("sms sent"));
```


How to track SMS result

How to track SMS result

- 1) Create Pending intent for SMS sent status and SMS delivered status
- 2) Create broadcast receiver for SMS send intent and delivered intent
- 3) Register both of two broadcast receivers
- 4) Send SMS by pass two pending intents

Notice!

- Emulator do not fire SMS delivered intent
- Emulator version grater than 4.0 fail in airplane mode
- Recommend using Emulator android version 2.3.3

How to track SMS result

Create PendingIntent

```
String SENT = "SMS_SENT";  
String DELIVERED = "SMS_DELIVERED";
```

```
PendingIntent sentPI = PendingIntent.getBroadcast(  
    MainActivity.this, 0, new Intent(SENT), 0);
```

```
PendingIntent deliveredPI = PendingIntent.getBroadcast(  
    MainActivity.this, 0, new Intent(DELIVERED), 0);
```

How to track SMS result

Create SMS sent Broadcast receiver

```
private class SMSsent extends BroadcastReceiver
{
    @Override
    public void onReceive(Context context, Intent intent) {
        switch (getResultCode()) {
            case Activity.RESULT_OK:
                Toast.makeText(getBaseContext(), "OK SMS sent", Toast.LENGTH_SHORT).show(); break;
            case SmsManager.RESULT_ERROR_GENERIC_FAILURE:
                Toast.makeText(getBaseContext(), "ERROR Generic failure", Toast.LENGTH_SHORT).show();
                break;
            case SmsManager.RESULT_ERROR_NO_SERVICE:
                Toast.makeText(getBaseContext(), "ERROR No service", Toast.LENGTH_SHORT).show(); break;
            case SmsManager.RESULT_ERROR_NULL_PDU:
                Toast.makeText(getBaseContext(), "ERROR Null PDU", Toast.LENGTH_SHORT).show(); break;
            case SmsManager.RESULT_ERROR_RADIO_OFF:
                Toast.makeText(getBaseContext(), "ERROR Radio off", Toast.LENGTH_SHORT).show(); break;
        }
    }
}
```

How to track SMS result

Create SMS delivered Broadcast receiver

```
private class SMSdelivered extends BroadcastReceiver
{
    @Override
    public void onReceive(Context context, Intent intent) {
        switch (getResultCode()) {
            case Activity.RESULT_OK:
                Toast.makeText(getBaseContext(), "SMS delivered",
                    Toast.LENGTH_SHORT).show();
                break;
            case Activity.RESULT_CANCELED:
                Toast.makeText(getBaseContext(),
                    "SMS not delivered", Toast.LENGTH_SHORT)
                    .show();
                break;
        }
    }
}
```

How to track SMS result

Register the receivers

```
registerReceiver(new SMSdelivered(), new IntentFilter(DELIVERED));  
registerReceiver(new SMSsent(), new IntentFilter(SENT));
```

Send SMS

```
SmsManager sms = SmsManager.getDefault();  
sms.sendTextMessage("089123456", null, "Hello world",  
sentPI, deliveredPI);
```

How to receive SMS ?

Receive SMS using broadcast receiver

Create Broadcast receiver

```
private class SMSreceived extends BroadcastReceiver
{
    @Override
    public void onReceive(Context context, Intent intent) {
        Log.v("SMS received!", intent.toString());
    }
}
```

Register Broadcast receiver

```
registerReceiver(new SMSreceived(), new
IntentFilter("android.provider.Telephony.SMS_RECEIVED"));
```


Receive SMS using broadcast receiver

How to decode PDUs

Raw PDU data

```
0:20:b:ffffff81:51:55:25:51:55:ffffff6:0:0:31:60:10:51:24:7
4:ffffff82:b:ffffffc8:32:ffffff9b:fffffffd:6:ffffffdd:ffffffdf:72:36
:19:
```

String message

```
From:15555215556
Message:Hello world
```

Receive SMS using broadcast receiver

```
private String decodePDU(Bundle bundle)
{
    SmsMessage[] msgs = null; String str = "";
    if (bundle != null) {
        Object[] pdus = (Object[]) bundle.get("pdus");
        msgs = new SmsMessage[pdus.length];
        for (int i = 0; i < msgs.length; i++) {
            msgs[i] = SmsMessage.createFromPdu((byte[]) pdus[i]);
            str += "SMS from " + msgs[i].getOriginatingAddress();
            str += " :";
            str += msgs[i].getMessageBody().toString();
            str += "\n";
        }
        return str;
    }
}
```

Why do we need array for PDU ?

Receive SMS using broadcast receiver

```
private class SMSreceived extends BroadcastReceiver
{
    @Override
    public void onReceive(Context context, Intent intent) {
        Log.v("SMS received!", decodePDU( intent.getExtras() ));
    }
}
```

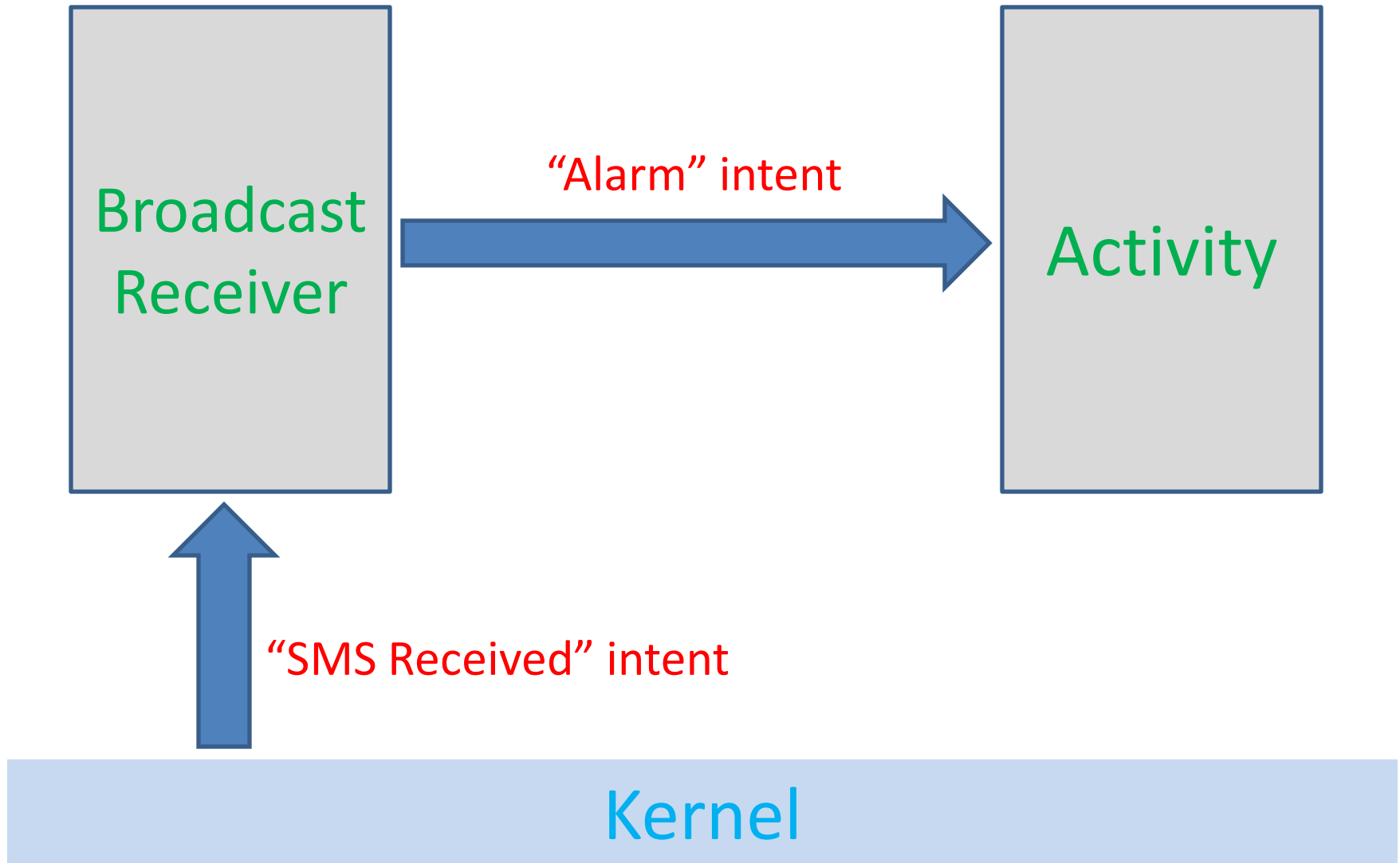
How to control a phone using SMS ?

- Application activate automatically
- Using same permission as SMS sender program
- Everything was done in background

How to control a phone using SMS

- 1) Create broadcast receiver class
- 2) Create Activity class
- 3) Register intent filter in AndroidManifest
- 4) Listen to `android.provider.Telephony.SMS_RECEIVED`
- 5) Fire Activity to run task if message or keyword is received

How to control a phone using SMS



How to control a phone using SMS

Activity class : *just play sound on new intent*

```
public class MainActivity extends Activity {  
    MediaPlayer mPlayer;  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        mPlayer = MediaPlayer.create(this, R.raw.alarm);  
    }  
    @Override  
    protected void onNewIntent(Intent intent) {  
        super.onNewIntent(intent);  
        if (intent.getBooleanExtra("alarm", false) == true) {  
            mPlayer.start();  
        }  
    }  
}
```

How to control a phone using SMS

Broadcast receiver

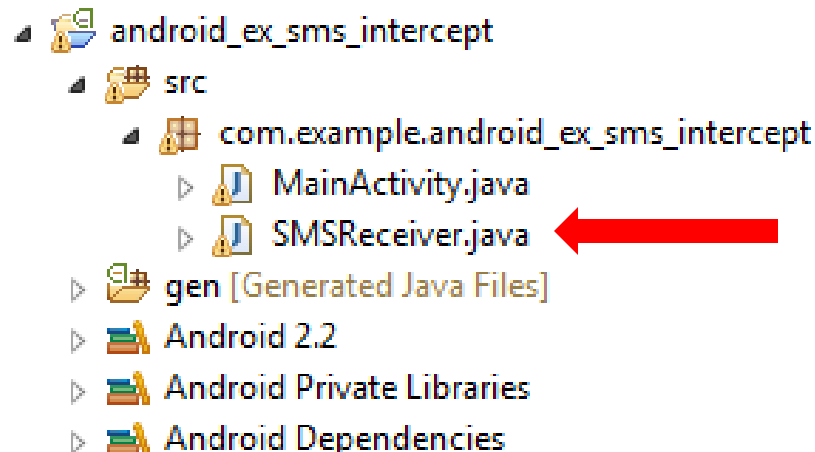
```
public class SMSReceiver extends BroadcastReceiver {  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        Bundle bundle = intent.getExtras();  
        Intent intent2 = new Intent(context, MainActivity.class);  
        intent2.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);  
        intent2.addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP);  
        SmsMessage[] msgs = null;String str = "";  
        if (bundle != null) {  
            Object[] pdus = (Object[]) bundle.get("pdus");  
            msgs = new SmsMessage[pdus.length];  
            for (int i = 0; i < msgs.length; i++) {  
                msgs[i] = SmsMessage.createFromPdu((byte[]) pdus[i]);  
                if (msgs[i].getMessageBody().equals("alarm")) {  
                    intent2.putExtra("alarm", true);  
                    context.startActivity(intent2);  
                    intent2.putExtra("alarm", true);  
                    context.startActivity(intent2);  
                }  
            }  
        }  
    }  
}
```


How to control a phone using SMS

AndroidManifest

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

```
<receiver  
    android:name=".SMSReceiver"  
    android:enabled="true"  
    android:exported="true" >  
    <intent-filter>  
        <action android:name="android.provider.Telephony.SMS_RECEIVED" />  
    </intent-filter>  
</receiver>
```



Thank you 😊