

Introduction to Android Sensors

CS 436 Software Development on Mobile

Dr.Paween Khoenkaw

Department of Computer Science
Maejo University



What is sensor

- Hardware that feed in data from the physical world
- Read only
- Single or combined
- Physical or Virtual

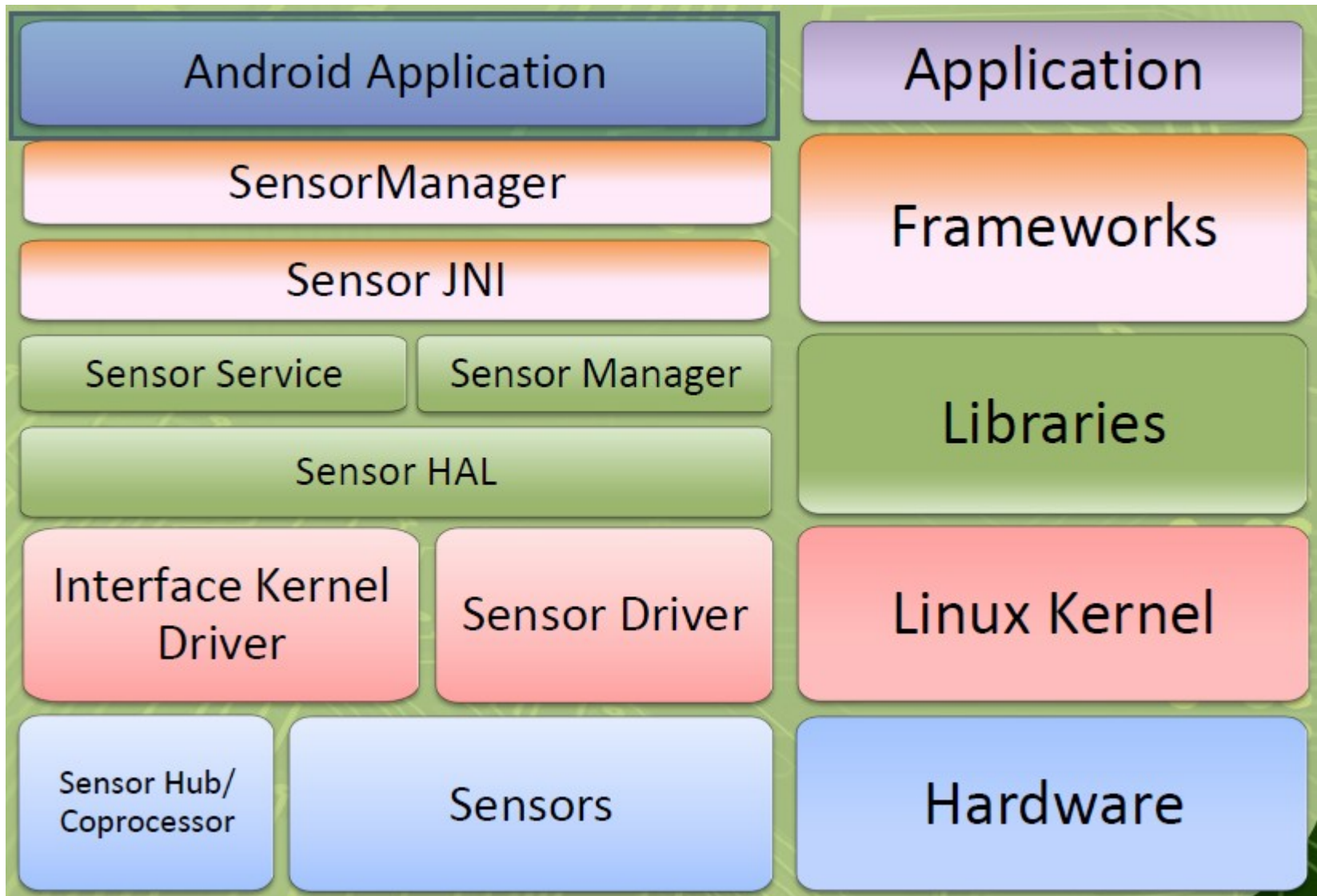
Types of sensors

- Motion sensors
- Environmental sensor
- Position

Types of sensors

Sensor	Android 4.0 (API Level 14)	Android 2.3 (API Level 9)	Android 2.2 (API Level 8)	Android 1.5 (API Level 3)
TYPE_ACCELEROMETER	Yes	Yes	Yes	Yes
TYPE_AMBIENT_TEMPERATURE	Yes	n/a	n/a	n/a
TYPE_GRAVITY	Yes	Yes	n/a	n/a
TYPE_GYROSCOPE	Yes	Yes	n/a ¹	n/a ¹
TYPE_LIGHT	Yes	Yes	Yes	Yes
TYPE_LINEAR_ACCELERATION	Yes	Yes	n/a	n/a
TYPE_MAGNETIC_FIELD	Yes	Yes	Yes	Yes
TYPE_ORIENTATION	Yes ²	Yes ²	Yes ²	Yes
TYPE_PRESSURE	Yes	Yes	n/a ¹	n/a ¹
TYPE_PROXIMITY	Yes	Yes	Yes	Yes
TYPE_RELATIVE_HUMIDITY	Yes	n/a	n/a	n/a
TYPE_ROTATION_VECTOR	Yes	Yes	n/a	n/a
TYPE_TEMPERATURE	Yes ²	Yes	Yes	Yes

How Android sensor works



How to use sensors

- 1) Check sensor available
 - Direct method
 - Indirect method
- 2) Create an instance of the sensor service (SensorManager class)
- 3) Create an instance of a specific sensor (Sensor class)
- 4) Create two callback methods that receive sensor notifications
(SensorEventListener interface)
- 5) Get sensor data from sensor event object (SensorEvent)
- 6) Start listening
- 7) Unregister when no longer need

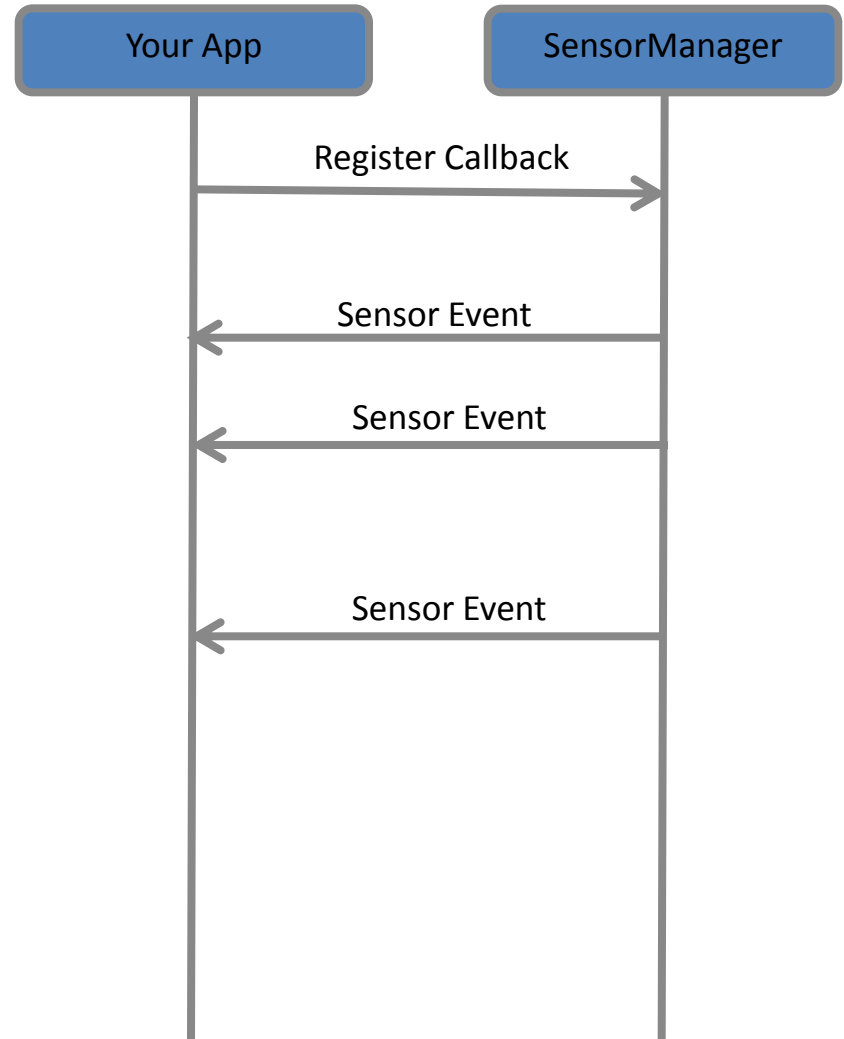
Async Callbacks

- Android's sensors are controlled by external services and only send events when they choose to

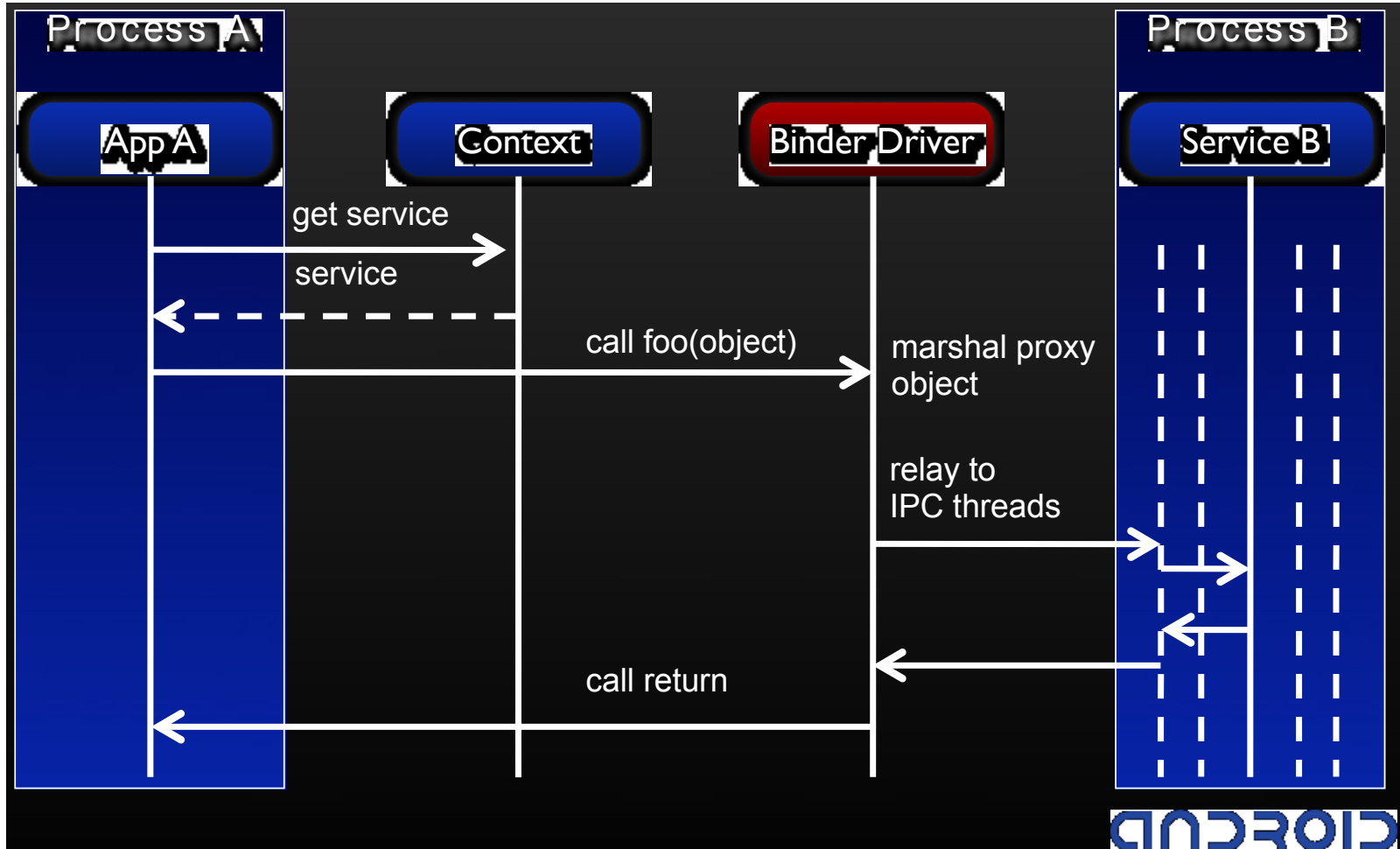
- An app must register a callback to be notified of a sensor event

- Each sensor has a related XXXListener interface that your callback must implement

 - E.g. `LocationListener`



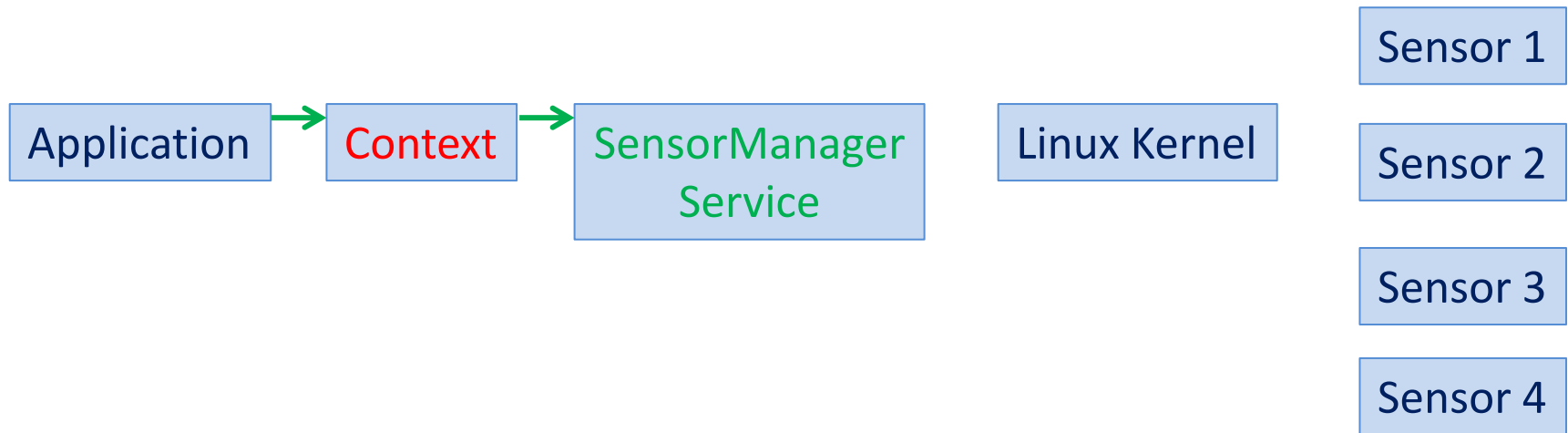
Async Callbacks



Start reading sensor

Step 1 Get the manager

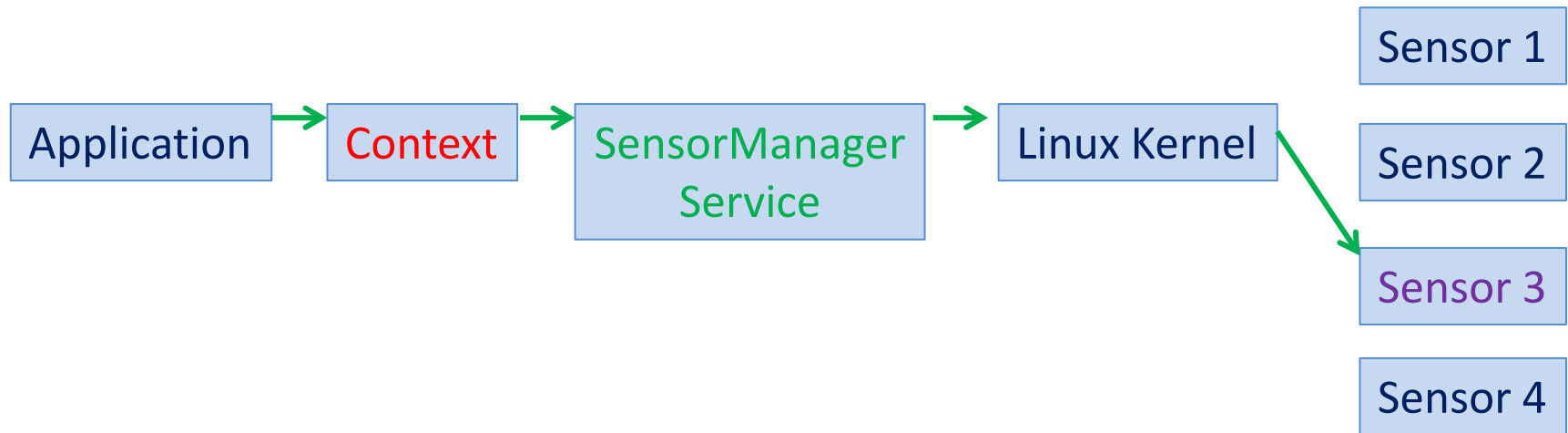
```
SensorManager manager=(SensorManager)  
    getSystemService(Context.SENSOR_SERVICE);
```



Start reading sensor

Step 2 Get the sensor

```
Sensor sensor = manager.getDefaultSensor(type);
```

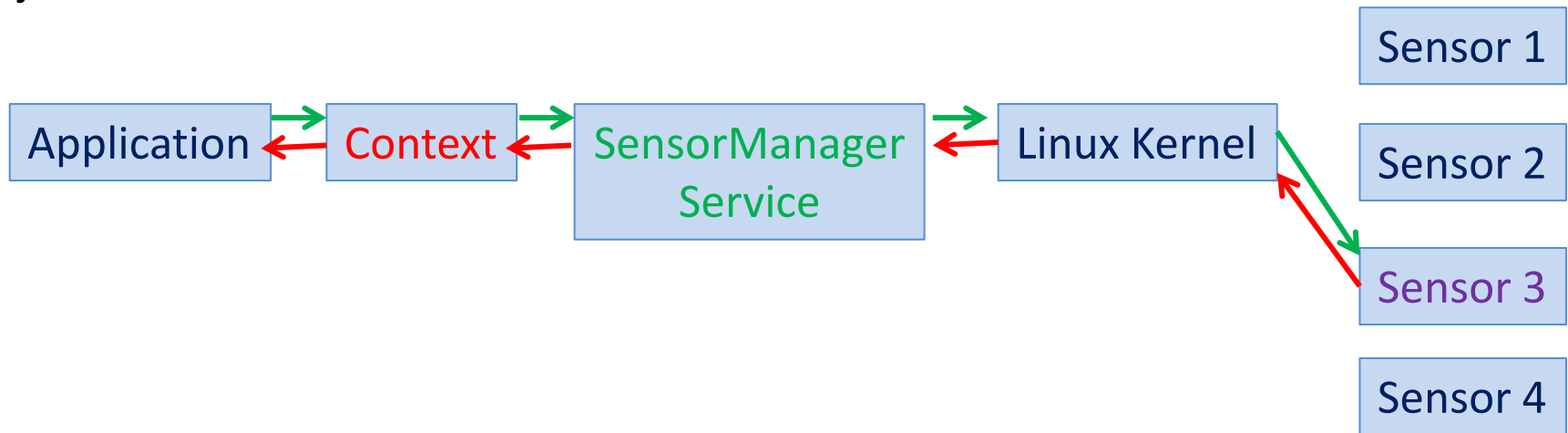


Start reading sensor

Step 3 Implement listener

```
public void onAccuracyChanged(Sensor sensor, int accuracy) {  
}
```

```
public void onSensorChanged(SensorEvent event) {  
  
}
```

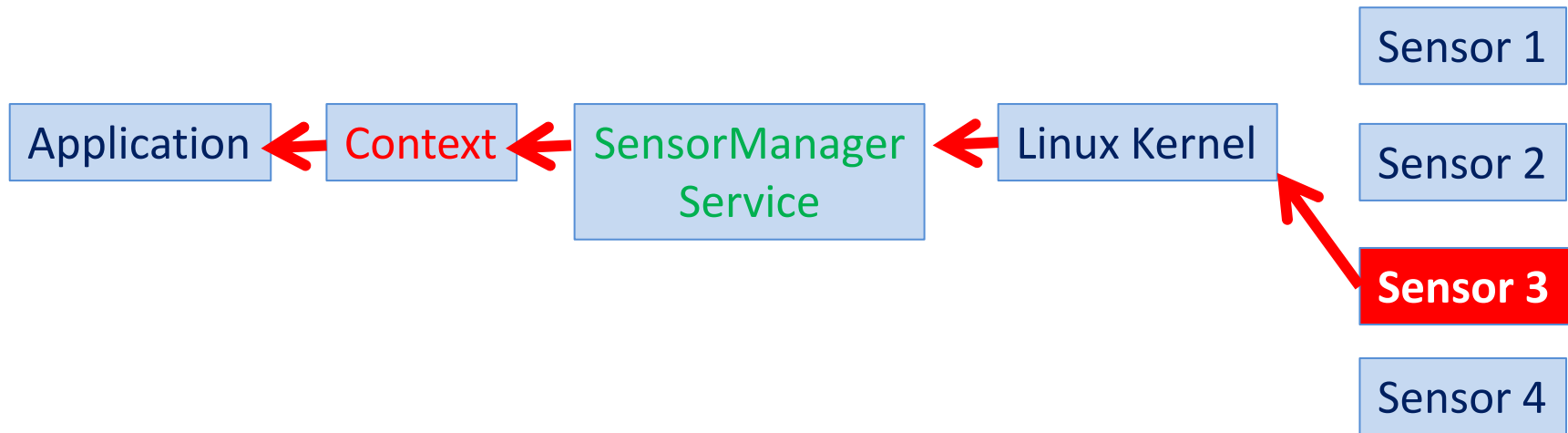


Start reading sensor

Step 4 Start sensor

```
manager.registerListener(context, sensor, delay mode);
```

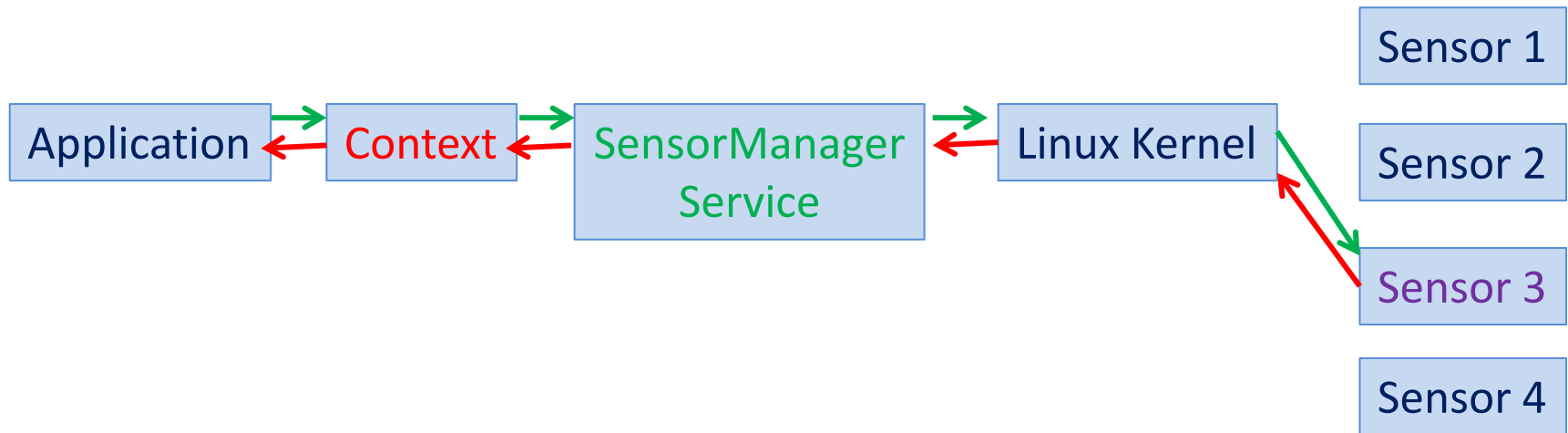
Sampling rate



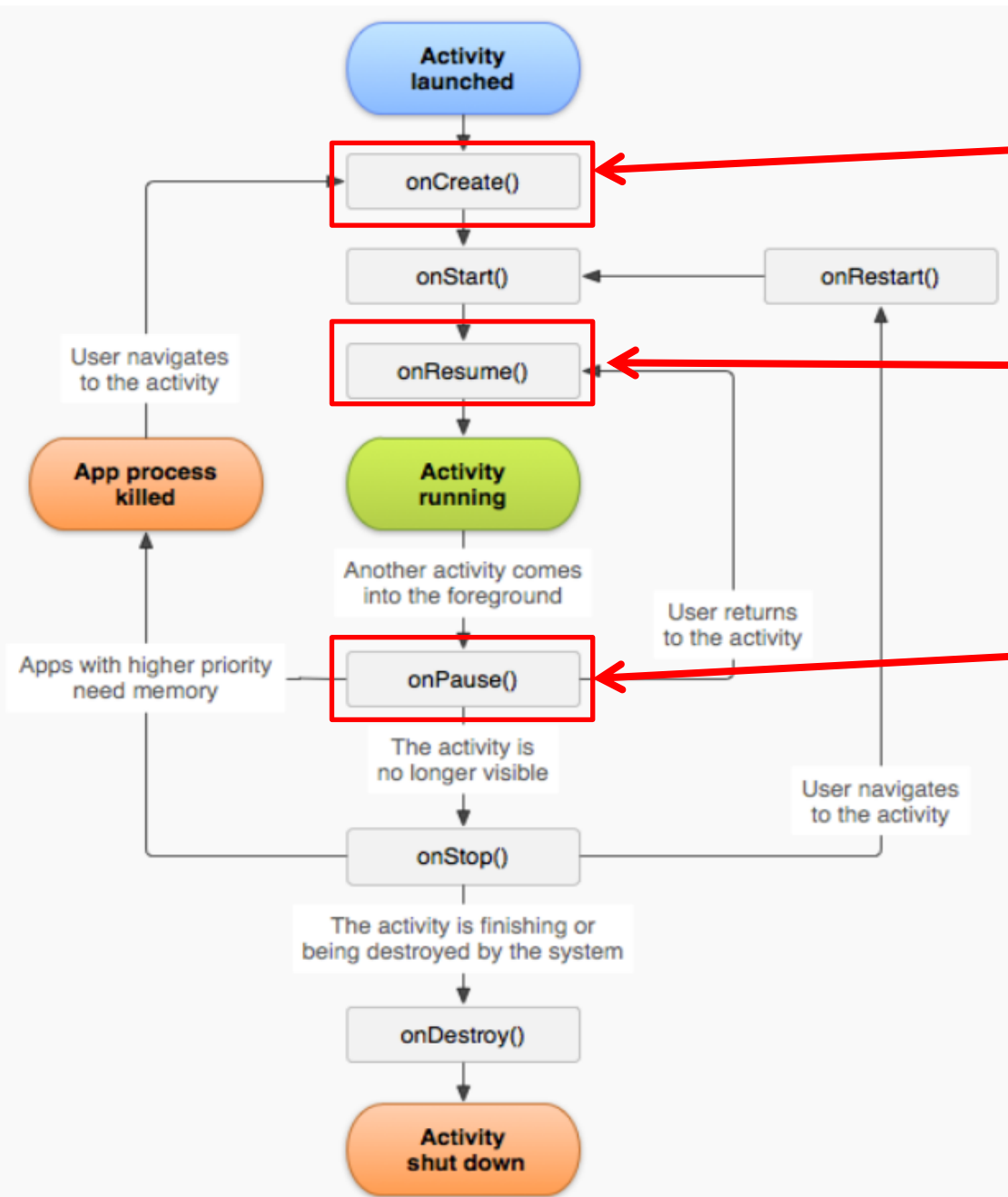
Start reading sensor

Step 5 Turn off sensor

```
manager.unregisterListener(context);
```



Put them all together



Step 1 Get the manager

Step 2 Get the sensor

Step 3 Implement listener

Step 4 Start sensor

Step 5 Turn off sensor

List all available sensors



List all available sensors

Step 1) Get the sensor manager

Step 2) Get the sensor list (return as List of Sensor)

Step 3) Get the sensor (for each)

List all available sensors

Step 1) Get the sensor manager

Step 2) Get the sensor list (return as List of Sensor)

Step 3) Get the sensor (for each)

List all available sensors

Step 1) Get the sensor manager

```
mSensorManager= (SensorManager) getSystemService(SENSOR_SERVICE);
```

Step 2) Get the sensor list (return as List of Sensor)

```
List<Sensor> msensorList = mSensorManager.getSensorList(Sensor.TYPE_ALL);
```

List all available sensors

Step 3) Get the sensor (for each)

```
int i;  
Sensor tmp;  
for (i=0;i<msensorList.size();i++){  
    tmp = msensorList.get(i);  
}
```

List all available sensors

Sensor class

Public Methods	
float	<u>getMaximumRange()</u>
int	getMinDelay()
String	getName()
float	getPower()
float	<u>getResolution()</u>
int	<u>getType()</u>
String	getVendor()
int	getVersion()
String	toString() Returns a string containing a concise, human-readable description of this object.

```

public class MainActivity extends Activity {
    SensorManager mSensorManager;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button button1=(Button)findViewById(R.id.button1);
        TextView editText1=(TextView)findViewById(R.id.textView1);
        mSensorManager= (SensorManager)
        getSystemService(SENSOR_SERVICE);
        List<Sensor> msensorList =
        mSensorManager.getSensorList(Sensor.TYPE_ALL);
        int i;
        Sensor tmp;
        for (i=0;i<msensorList.size();i++){
            tmp = msensorList.get(i);
            Log.d("slist", tmp.getName()); // Add the sensor name to
            the string of sensors available
            editText1.setText(editText1.getText().toString() +
            tmp.getName()+"\n"
            +tmp.getVendor()+"\n" + "-----\n");

        }

        button1.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                finish();
            }
        });
    }
}

```

```

<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity" >

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:text="Button" />

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/button1"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:text="Large Text"
        android:textAppearance="?android:attr/textAppearanceLarge"
    />

</RelativeLayout>

```

Thank you 😊