

# Мобильная платформа Android. Что может быть проще?



Цытович Павел Леонидович  
Южно-Уральский государственный  
университет  
E-mail: [pavel.tsytovich@item74.ru](mailto:pavel.tsytovich@item74.ru)



# Коротко о главном

- Средства создания интерфейса пользователя
- Намерения пользователя
- Сервисы
- Что для разработчика?

# Архитектура платформы

## Applications and Widgets

Home

Contacts

Browser

Widgets

Your App Here

## Application Framework

Activity Manager

Window Manager

Content Providers

View System

Notification Manager

Package Manager

Telephony Manager

Resource

Location Manager

Sensor Manager

## Libraries

Surface Manager

Media Framework

SQLite

OpenGL | ES

FreeType

WebKit

SGL

SSL

libc

## Android Runtime

Core Libraries

Dalvik Virtual Machine

## Linux Kernel

Display Driver

Bluetooth Driver

Camera Driver

Flash Memory Driver

Binder (IPC) Driver

Keypad Driver

USB Driver

WiFi Driver

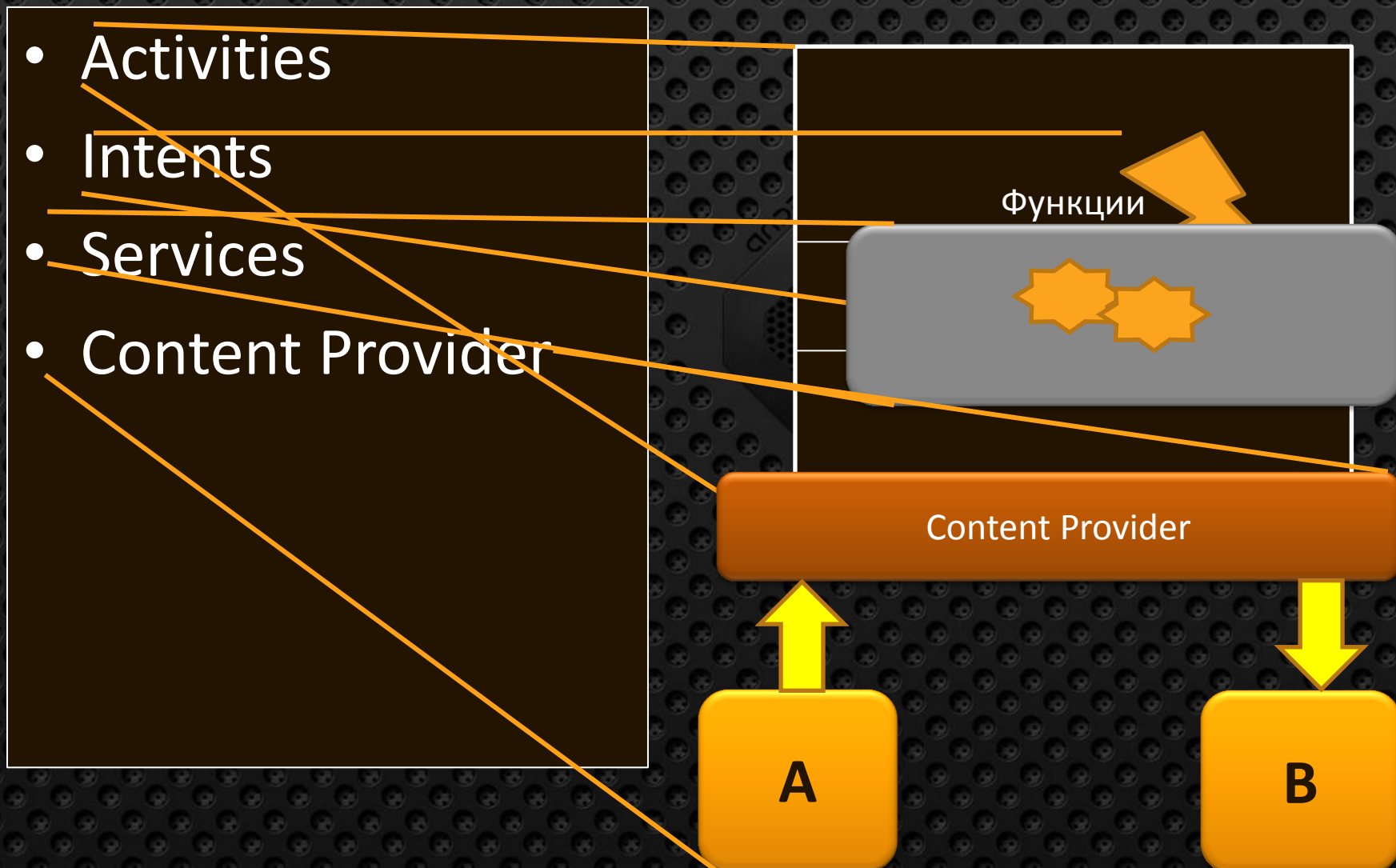
Audio Drivers

Power Management

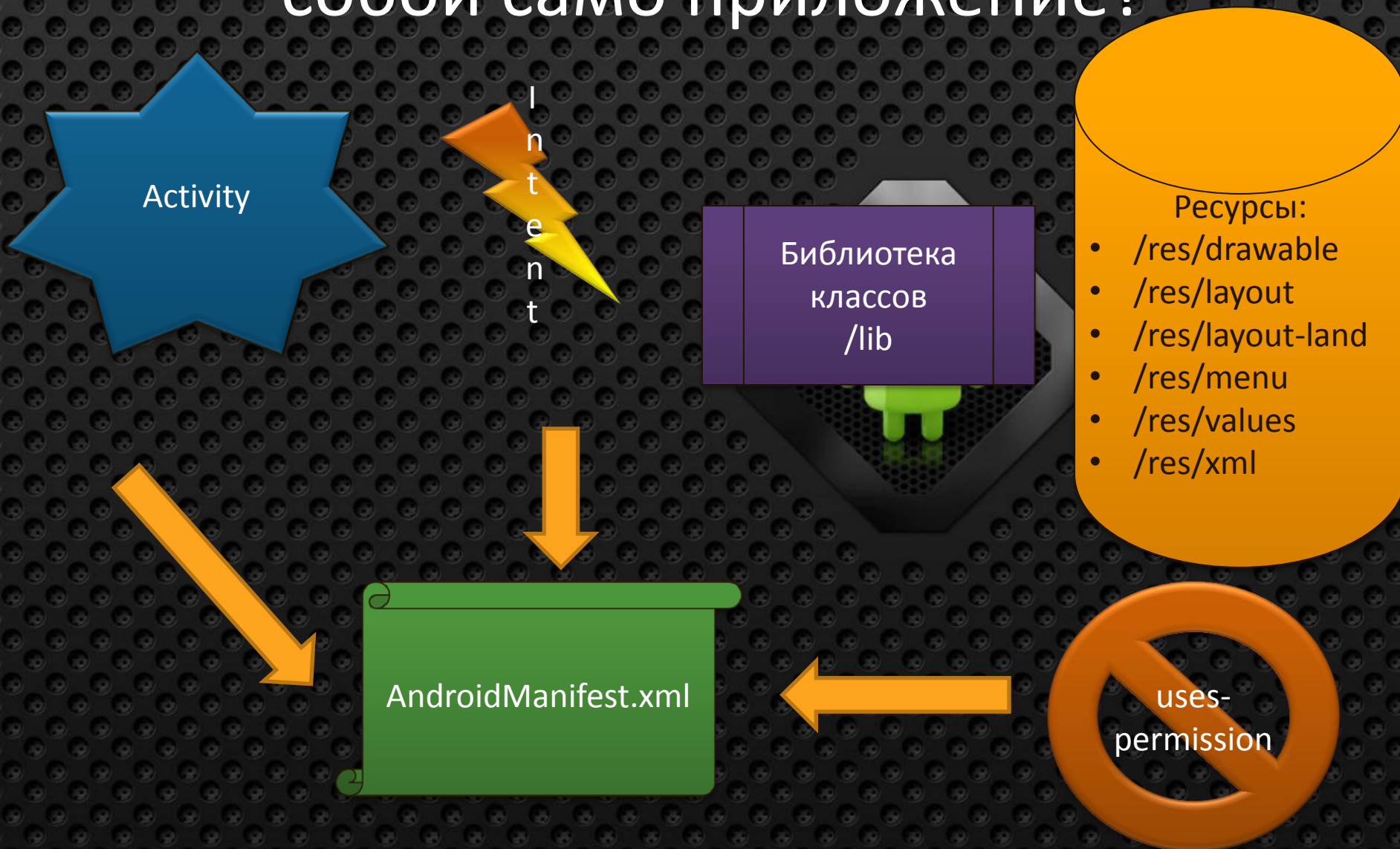


# Что нам стоит приложение построить?

- Activities
- Intents
- Services
- Content Provider



# А что же логически представляет собой само приложение?



# А физически?..

Application.dpk

META-INF

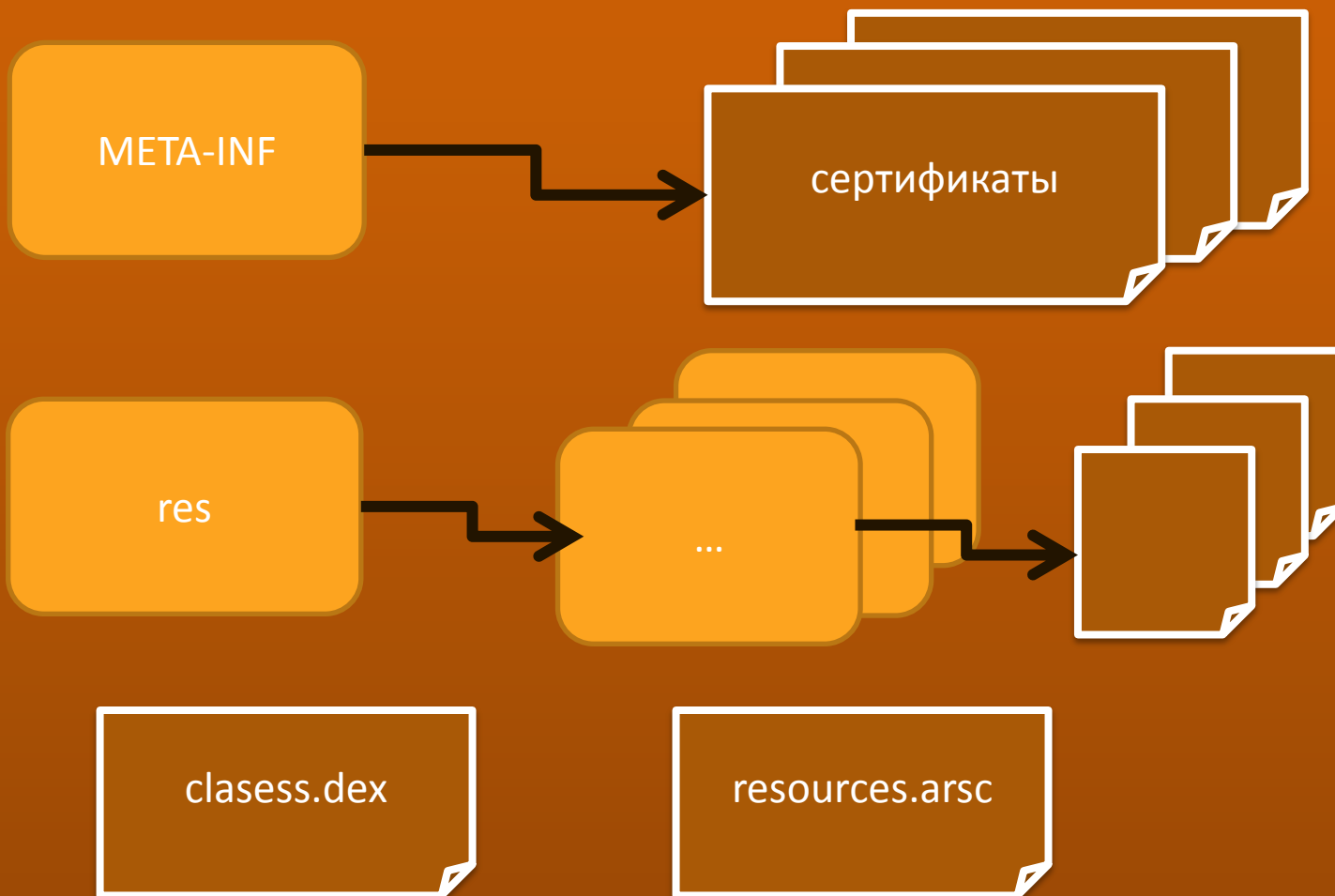
сертификаты

res

...

classes.dex

resources.arsc





# Face-to-Face

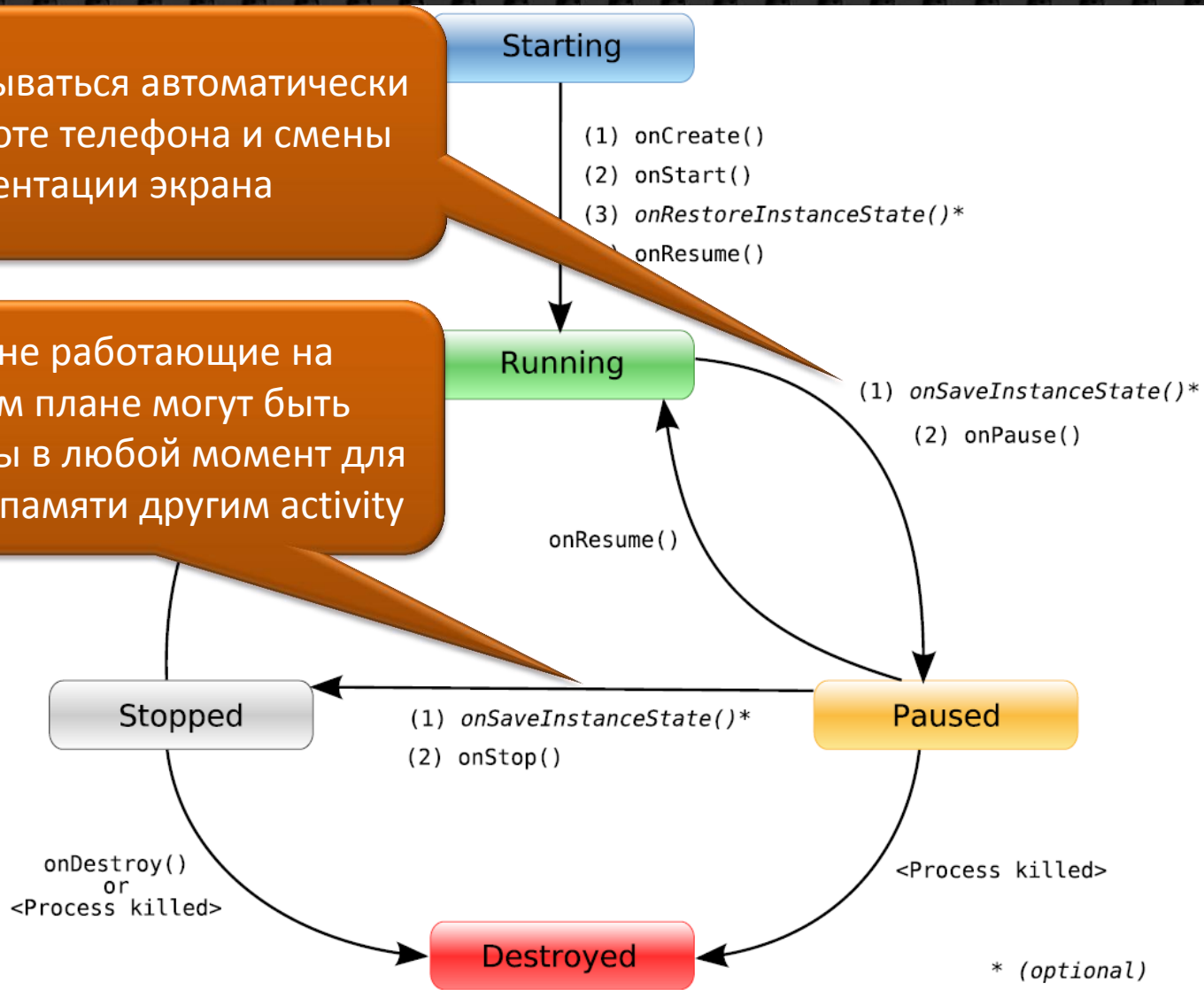
## Activity



# Activities – основа UI

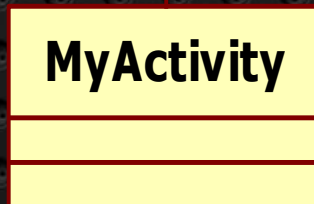
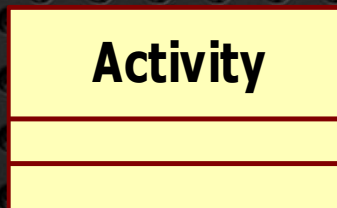
Может вызываться автоматически при повороте телефона и смены ориентации экрана

Activity не работающие на переднем плане могут быть остановлены в любой момент для выделения памяти другим activity





# Activities



```
package PavlinCo.Sample;
```

```
import android.app.Activity;
```

```
import android.os.Bundle;
```

```
import android.widget.Button;
```

```
public class MyActivity extends Activity {
```

```
    /** Called when the activity is first created. */
```

```
    @Override
```

```
    public void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        Button btn = new Button(this);
```

```
        setContentView(btn);    }
```

```
}
```

# Добавляем интерактивности в Activity

```
package PavlinCo.Sample;

import android.app.Activity;
import android.os.Bundle;
import android.widget.Button;
import android.view.View;
import java.util.Date;

public class Main extends Activity
implements View.OnClickListener {
    private Button btn;

    /** Called when the activity is first created.
    */
    @Override
    public void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
        btn = new Button(this);
```

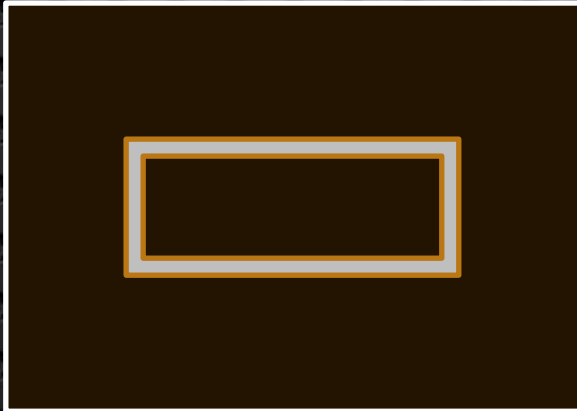
```
        btn.setOnClickListener(this);
        setContentView(btn);
        updateTime();
        //setContentView(R.layout.main);
    }

    private void updateTime() {
        btn.setText(new Date().toString());
    }

    @Override
    public void onClick(View v) {

        updateTime();
    }
}
```

# Декларативное описание интерфейса



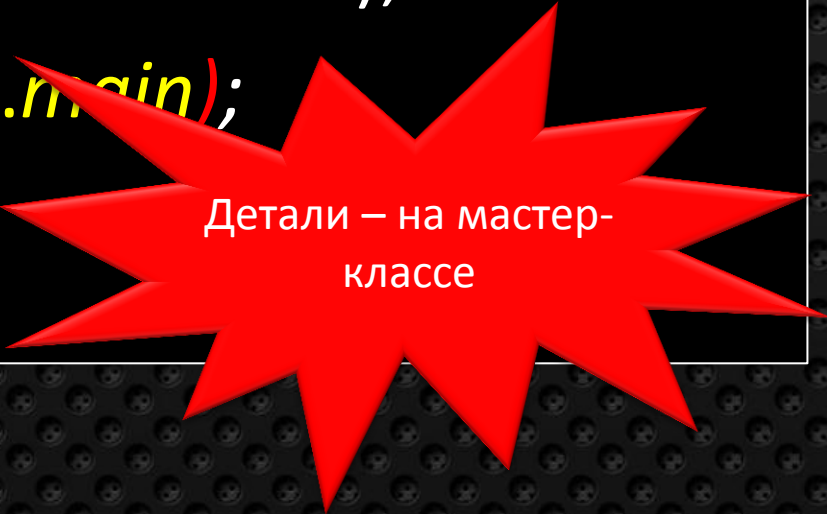
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res
    /android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
  >
  <TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello"
  />
</LinearLayout>
```

Загрузка из ресурсов



# Привязка декларативного описания в коде

```
public class Main extends Activity {  
    private static final int ID_ABOUT = 0xF;  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
    }  
}
```



Детали – на мастер-  
классе

# Базовые Activity

- Activity – простейшее activity
- ListActivity – Отображение списка элементов
- TabActivity – activity с закладками
- PreferenceActivity – activity для конфигурационных страниц приложений



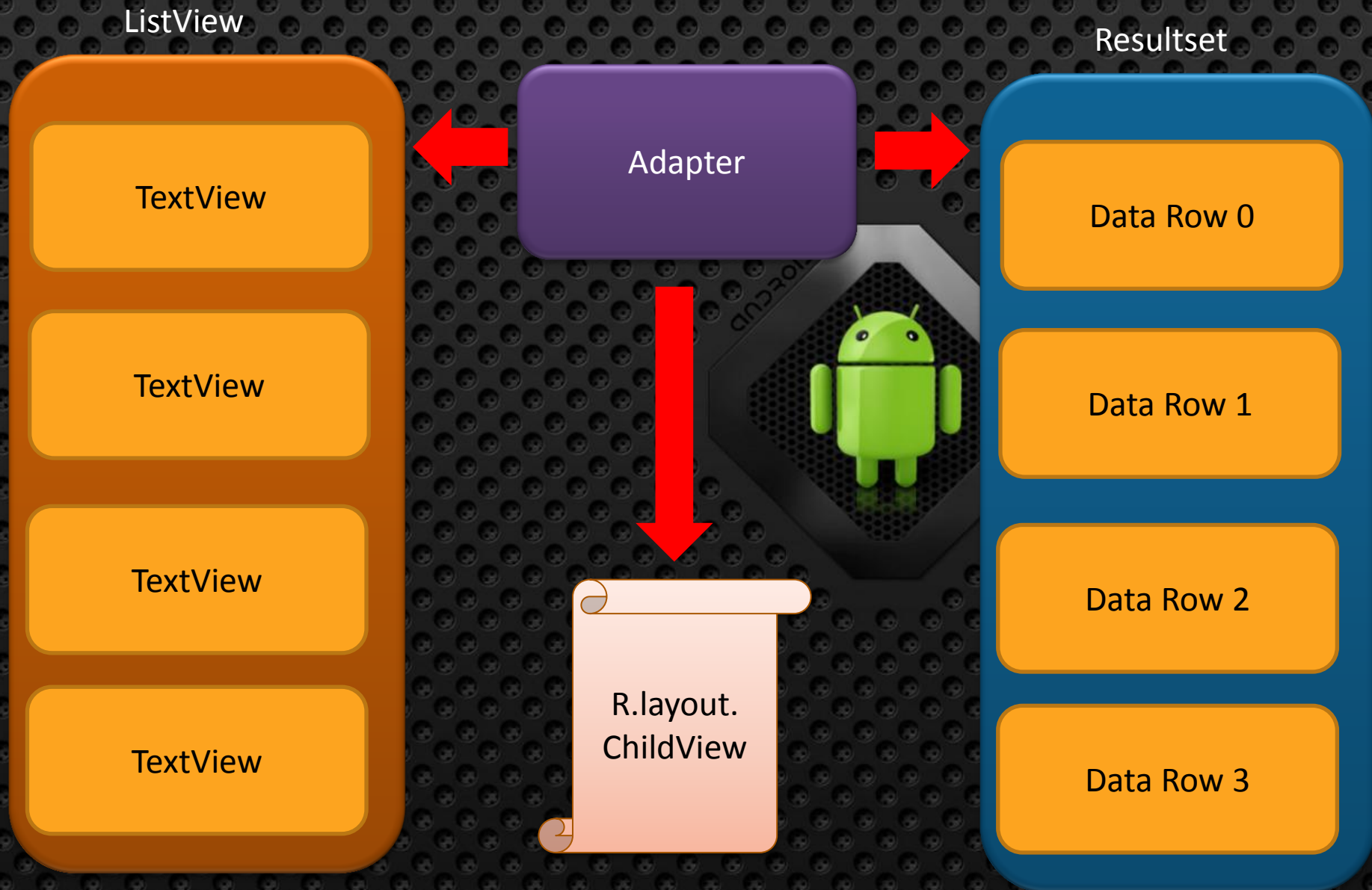
Детали – на мастер-классе

# Менеджеры компоновки

- `LinearLayout` – располагает элементы в одну строку/столбец в зависимости от значения свойства `orientation`
- `TableLayout` – располагает элементы в ячейки таблицы (Строки – `TableRow`)
- `AbsoluteLayout` – располагает элементы в заданных их свойствами `layout_x` и `layout_y` координатах
- `RelativeLayout` – располагает элементы в зависимости от других элементов (свойство `layout_below`)
- `FrameLayout` – отображает только один элемент. Если есть еще элементы, то они «находятся» за ним



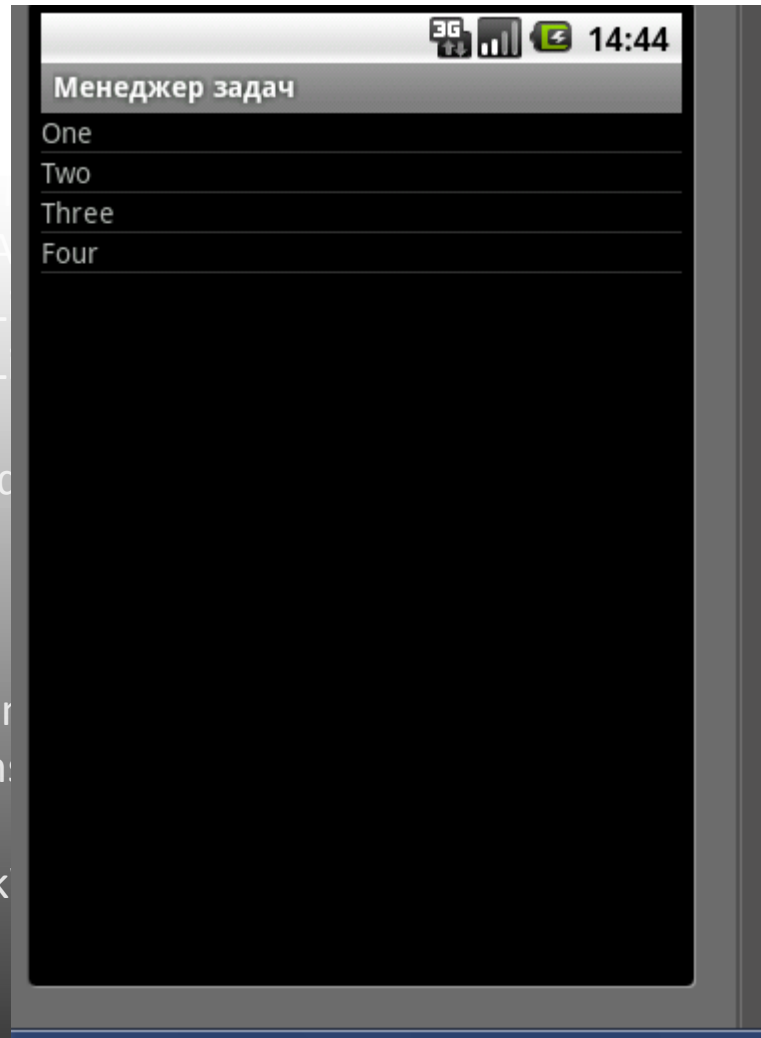
# Привязка данных



# Привязка данных

```
import android.os.Bundle;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ListView;

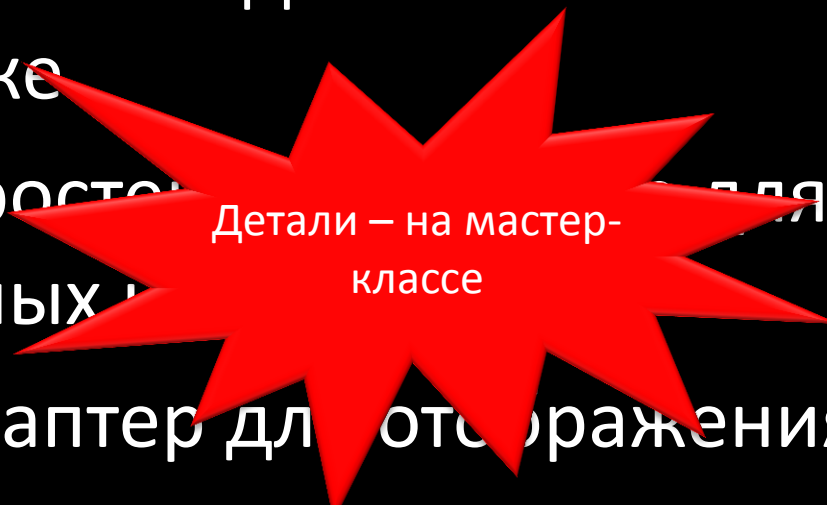
public class WorkWithAdapter extends
    AppCompatActivity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ListView lv = new
            ListView(this);
```



```
    String[] items = {"One", "Two", "Three", "Four"};
    ArrayAdapter<String> adapter = new
        ArrayAdapter<String>(this,
            R.layout.list_item, items);
    lv.setAdapter(adapter);
```

# Адаптеры

- ArrayAdapter<T> – список элементов произвольного типа. Каждый элемент приводится к строке
- SimpleAdapter – простейший адаптер для отображения данных
- CursorAdapter – адаптер для отображения данных, полученных как результат запроса к БД или другому источнику данных



Детали – на мастер-классе

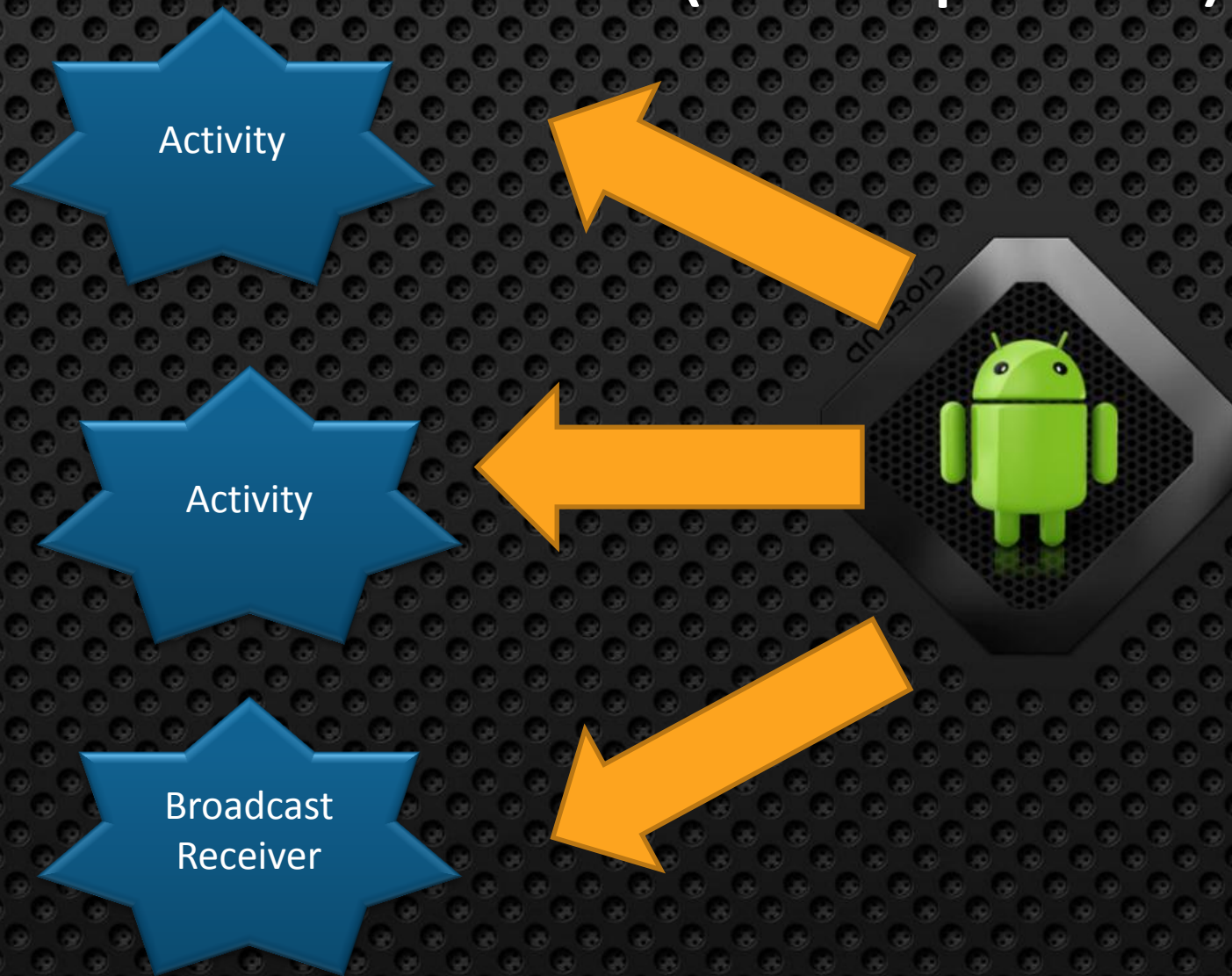


От пользователя  
можно много чего  
ожидать...

Intent



# Intents (намерение)



# Состав Intent

- Category – категория Intent (LAUNCHER, ALTERNATIVE, DEFAULT, ...)
- type – указывает MIME-тип данных, передаваемых вместе с Intent
- extras – данные Intent
- Data – Uri, по которому извлекаются данные



# Открытие Activity

```
public static invokeMyApplication  
    (Activity parentActivity)  
{  
    String actionName= "  
com.androidbook.intent.action.ShowBasicView ";  
    Intent intent = new Intent(actionName);  
    parentActivity.startActivity(intent);  
}
```

Имя класса Activity

# Открытие диалога вызова абонента

```
public static void dial(Activity activity)
{
    Intent intent = new Intent(Intent.ACTION_DIAL);
    activity.startActivity(intent);
}
```

# Набор номера

```
public static void call(Activity activity)
{
    Intent intent = new Intent(Intent.ACTION_CALL);
    intent.setData(Uri.parse("tel:555-555-5555"));
    activity.startActivity(intent);
}
```



# Открытие обозревателя

```
public static void invokeWebBrowser  
(Activity activity)  
{  
    Intent intent = new Intent(Intent.ACTION_VIEW);  
    intent.setData(Uri.parse("http://www.google.com"))  
};  
activity.startActivity(intent);  
}
```

# Получение SMS

```
public class MySMSMonitor extends BroadcastReceiver
{
    private static final String ACTION = "android.provider.Telephony.SMS_RECEIVED";
    @Override
    public void onReceive(Context context, Intent intent)
    {
        if(intent!=null && intent.getAction()!=null &&
        ACTION.compareToIgnoreCase(intent.getAction())==0)
        {
            Object[] pduArray= (Ob
            SmsMessage[] messag
            for (int i = 0; i<pduArra
            messages[i] = SmsMes
            }
            Log.d("MySMSMonitor
            }
        }
    }
}
```

**Androidmanifest.xml**

```
<receiver android:name="MySMSMonitor">
<intent-filter>
<action
android:name="android.provider.Telephony.SMS_RECEIVED
"/>
</intent-filter>
</receiver>
```

Служу андроиду...

Services





# Services

- Сервис – приложение, работающее в фоновом режиме без участия пользователя
- Сервис позволяет в фоновом режиме выполнять определенные задачи (local service)
- Сервис позволяет обеспечить межпроцессное взаимодействие (remote service)

# Service

```
import android.app.Service;
public class TestService1 extends Service
{
    private static final String TAG = "TestService1";
    @Override
    public void onCreate() {
        Log.d(TAG, "onCreate");
        super.onCreate();
    }
    @Override
    public IBinder onBind(Intent intent) {
        Log.d(TAG, "onBind");
        return null;
    }
}
```

Androidmanifest.xml

```
<service android:name="TestService1"></service>
```

# Local Service

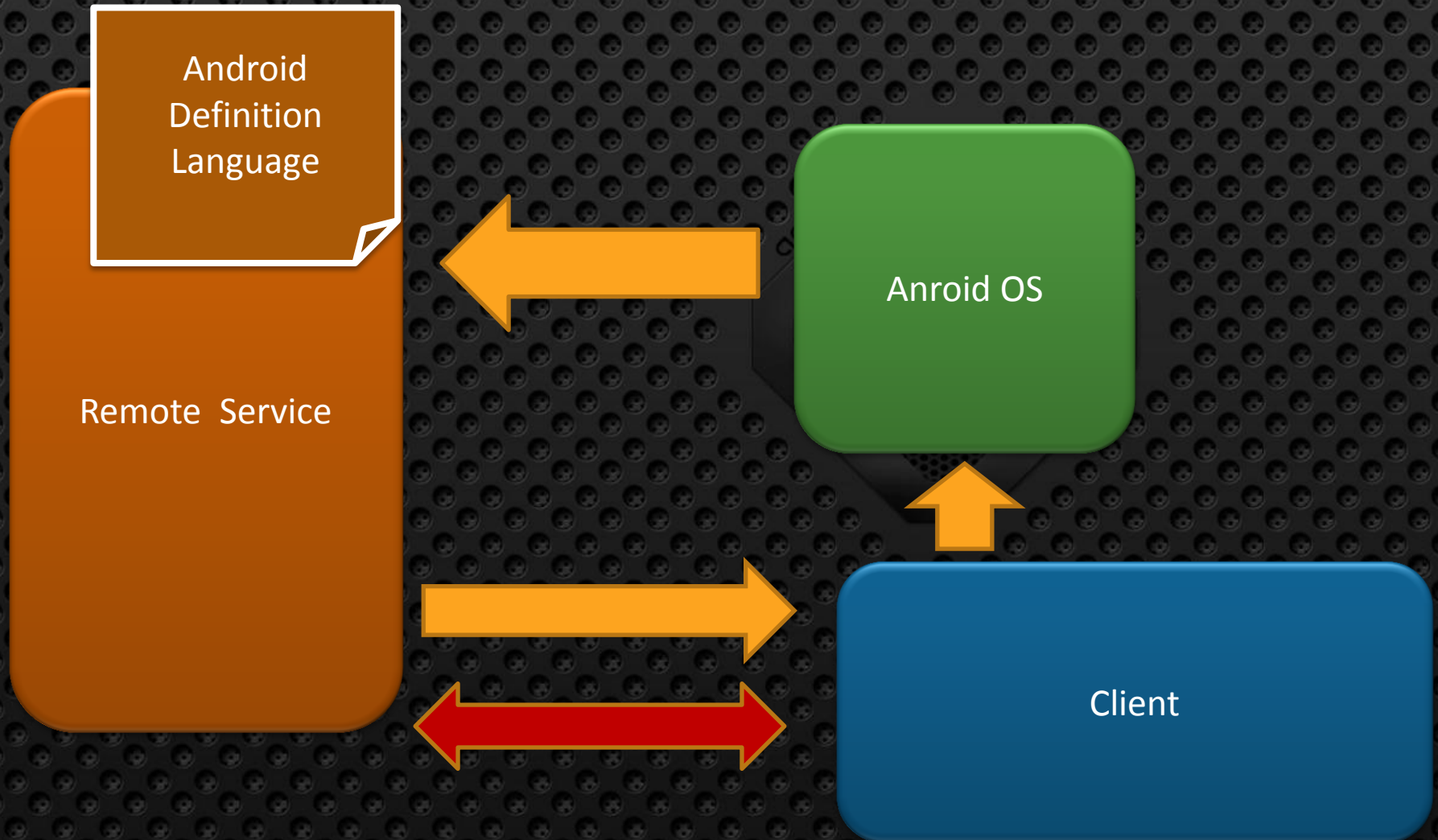
```
class MyService extends Service  
{  
  ...  
}
```



```
class ServiceWorker implements  
Runnable  
{  
  ...  
}
```



# Remote Service



# О бедном разработчике замолвите слово

SDK, Eclipse и все, все, все...



# С чего начать

- Установите Anroid SDK - <http://bit.ly/h7wUZf>
- Установите плагин к Eclipse - <http://bit.ly/ghgPt4>
- Настройте путь к Android SDK в Eclipse
- Создайте эмулятор телефона
- Создайте свой первый проект



Детали – на мастер-классе



# Пути распространения приложения

- Экспорт в Android-приложение
- Android Market
- Обычная установка

А что дальше делать?



# Рекомендации от автора

- Пути программиста неисповедимы
- Брать примеры и изучать
- Сделать свое первое приложение и загрузить его в маркет (сам бы это сделал!)
- Создать комьюнити и проводить регулярные встречи (Гугл нам поможет?)



# Полезные ссылки



<http://developer.android.com/> -  
Android SDK, примеры



<http://ibm.co/eK5ebG> -  
IBM DeveloperWorks (подборка  
статей по разработке под Android)



<http://bit.ly/hFdWnS> –  
Набор бесплатных иконок для  
Android

# Полезные ссылки



Блог Дарьи Ряжских -  
<http://bit.ly/dJIFof>



Сайт по программированию для  
мобильных устройств –  
<http://bit.ly/fMTI6f>



Android Development Tutorial  
Gingerbread –  
<http://bit.ly/fYIq0Y>

# КНИГИ

- Mark L. Murphy. Beginning Android
- Sayed Y. Hashimi and Satya Komatineni. Pro Android
- Ed Burnette. Hello Android



Спасибо за внимание!



Ваши вопросы?

# Встречаемся в...



[http://twitter.com/pavlin\\_co](http://twitter.com/pavlin_co)  
<http://pavlin-co.livejournal.com>  
<http://www.item74.ru>