Introduction to Android Camera

CS 436 Software Development on Mobile

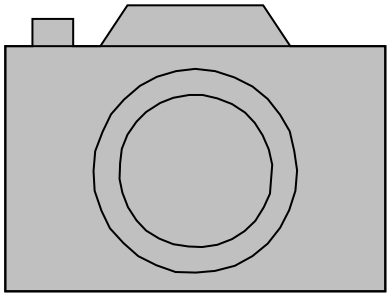
Dr.Paween Khoenkaw

Department of Computer Science
Maejo University

Camera

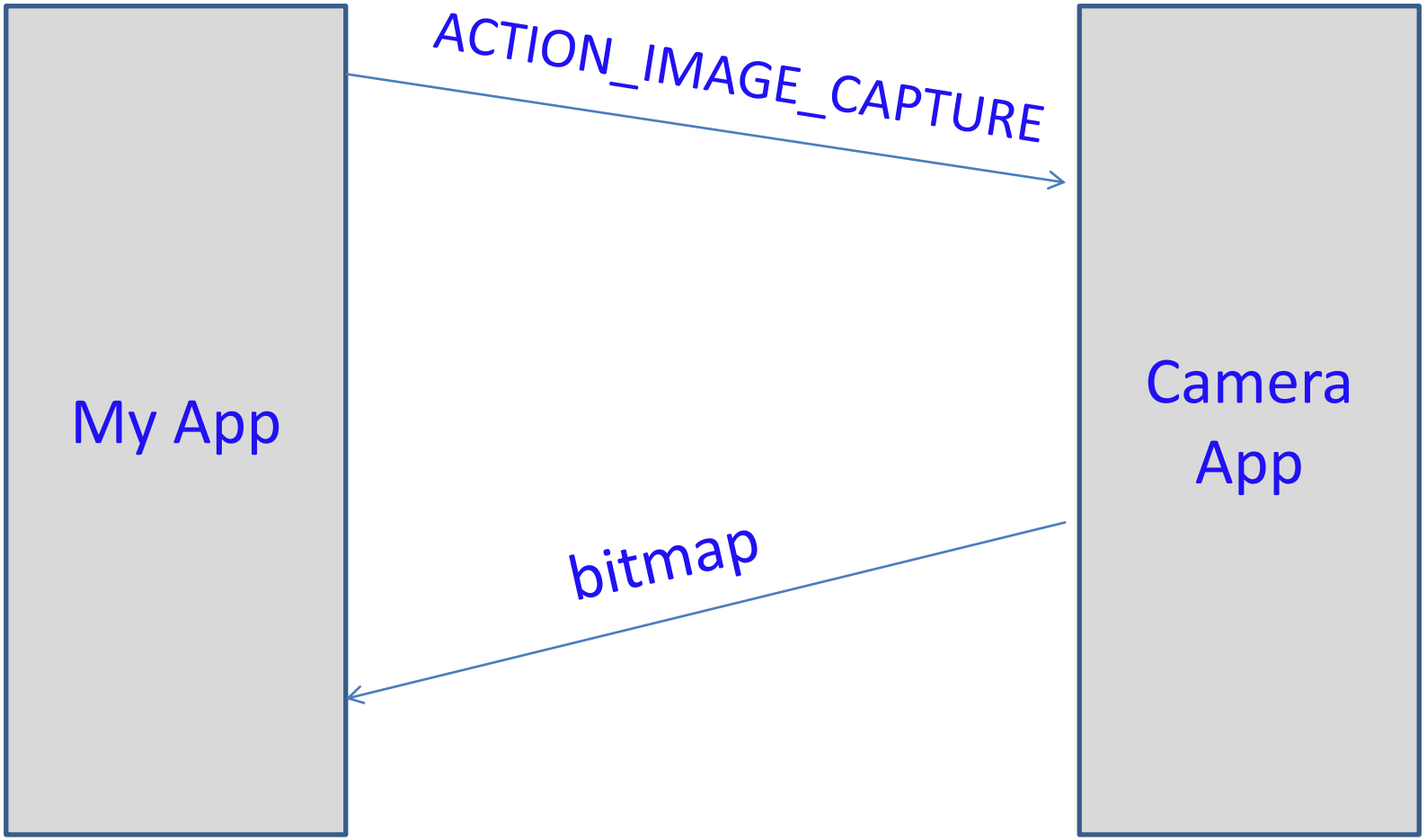
How to take a picture

1) Use camera intent
2) Use camera API



Camera

Camera Intent



Camera

Camera Intent

```
private static final int REQUEST_CODE = 1;
private Bitmap bitmap;

Intent intent = new
Intent(MediaStore.ACTION_IMAGE_CAPTURE);
 startActivityForResult(intent, REQUEST_CODE);
```

Camera

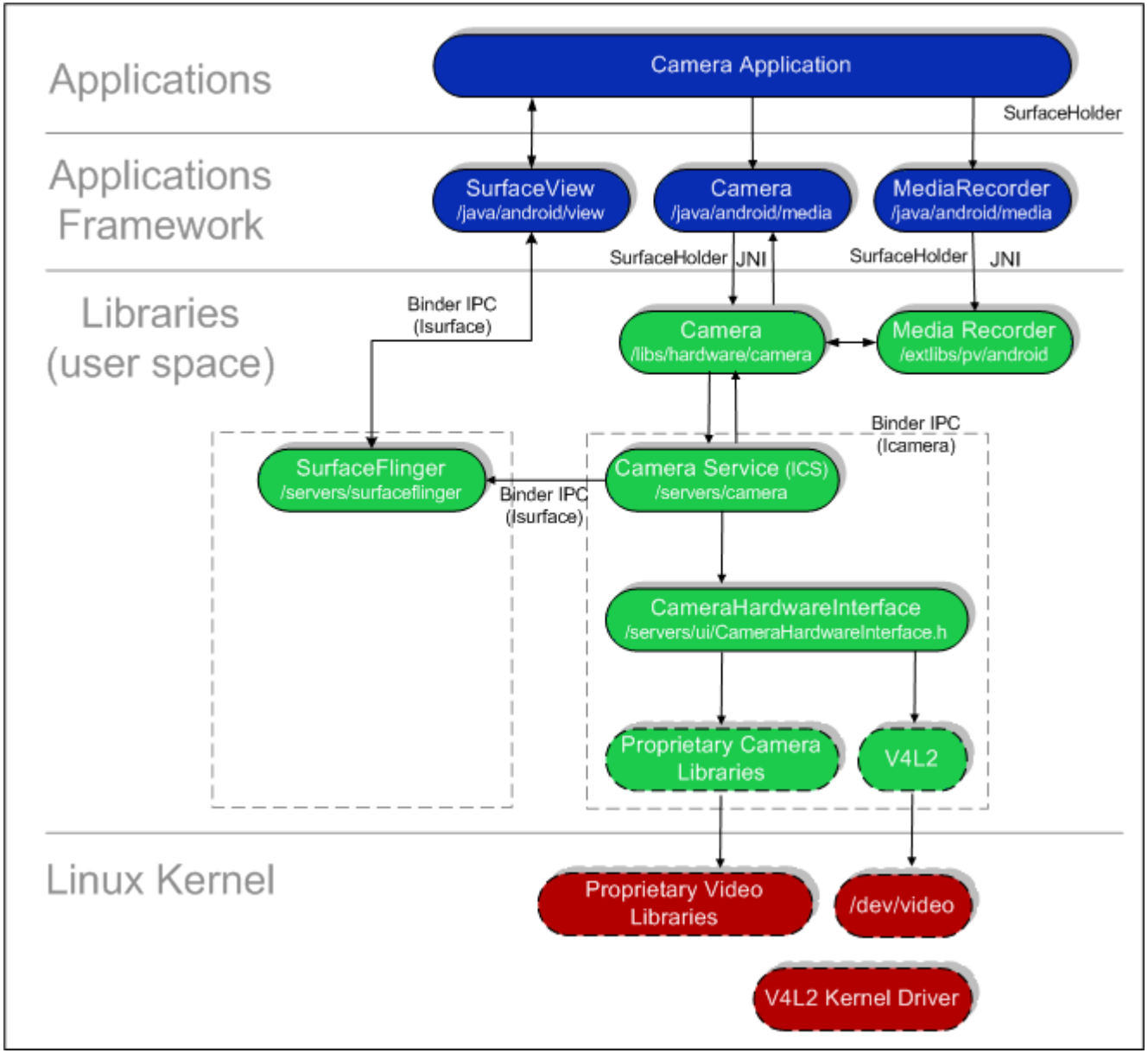
Camera Intent

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
                    if (requestCode == REQUEST_CODE && resultCode ==
Activity.RESULT_OK)
                        if (bitmap != null) {
                          bitmap.recycle();
                        }
                    Bundle extras = data.getExtras();
          bitmap = extras.getParcelable("data");
 imageView1.setImageBitmap(bitmap);
          super.onActivityResult(requestCode, resultCode, data);
          }
```

Camera

Take a picture using camera API

Camera API


Applications
Camera Application
SurfaceHolder
Applications Framework
SurfaceView
/java/android/view
Camera
/java/android/media
MediaRecorder
/java/android/media
SurfaceHolder
JNI
SurfaceHolder
JNI
Libraries (user space)
Binder IPC
(Isurface)
Camera
/libs/hardware/camera
Media Recorder
/extlibs/pv/android
Binder IPC
(Icamera)
SurfaceFlinger
/servers/surfaceflinger
Binder IPC
(Isurface)
Camera Service (ICS)
/servers/camera
CameraHardwareInterface
/servers/ui/CameraHardwareInterface.h
Proprietary Camera Libraries
V4L2
Linux Kernel
Proprietary Video Libraries
/dev/video
V4L2 Kernel Driver

Camera API

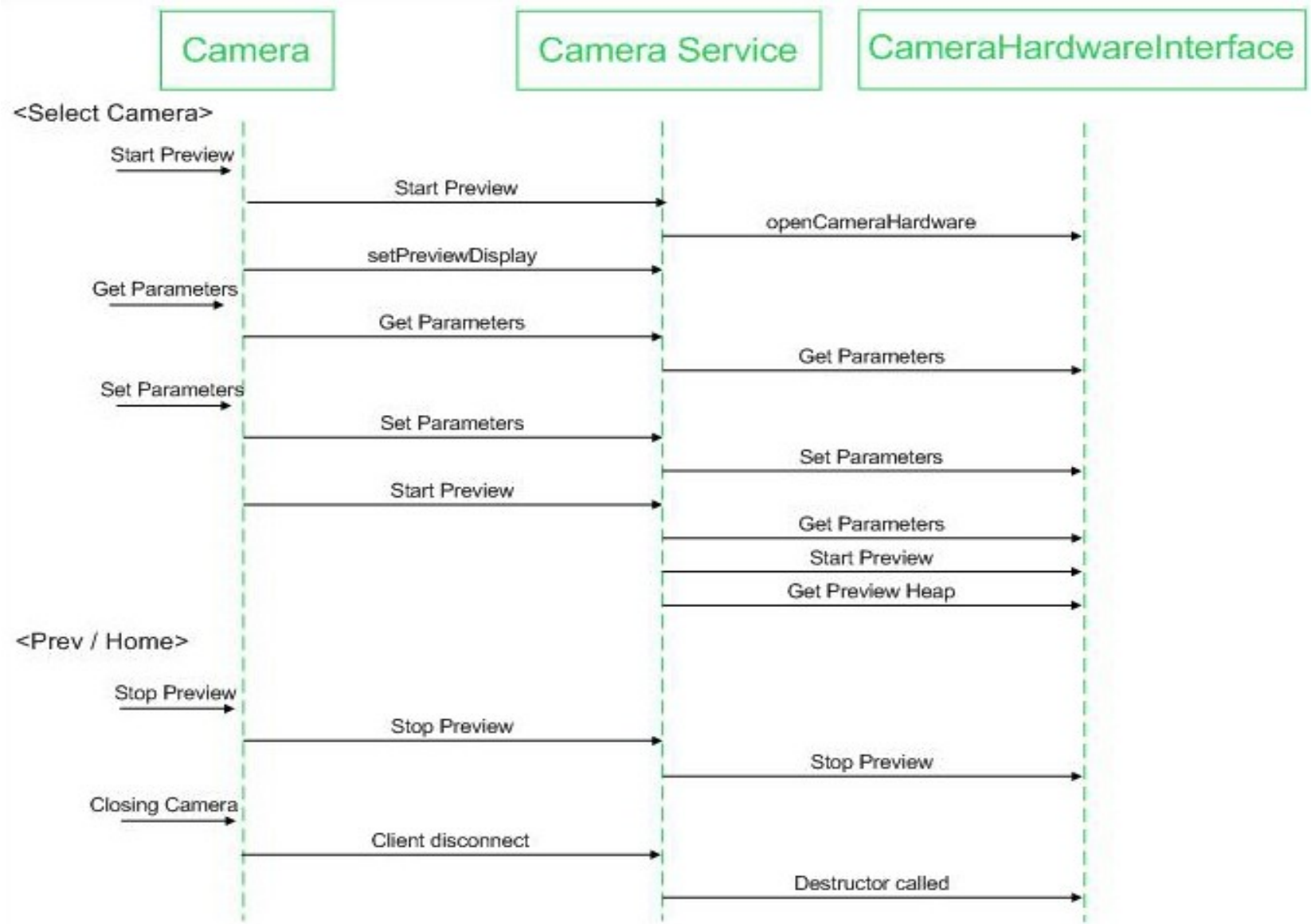
Take picture with camera API

1) Select camera
2) Create Preview
3) Start Preview
4) Take Picture
5) Stop preview
6) Release camera

!!Warning camera is unshareable resource!!

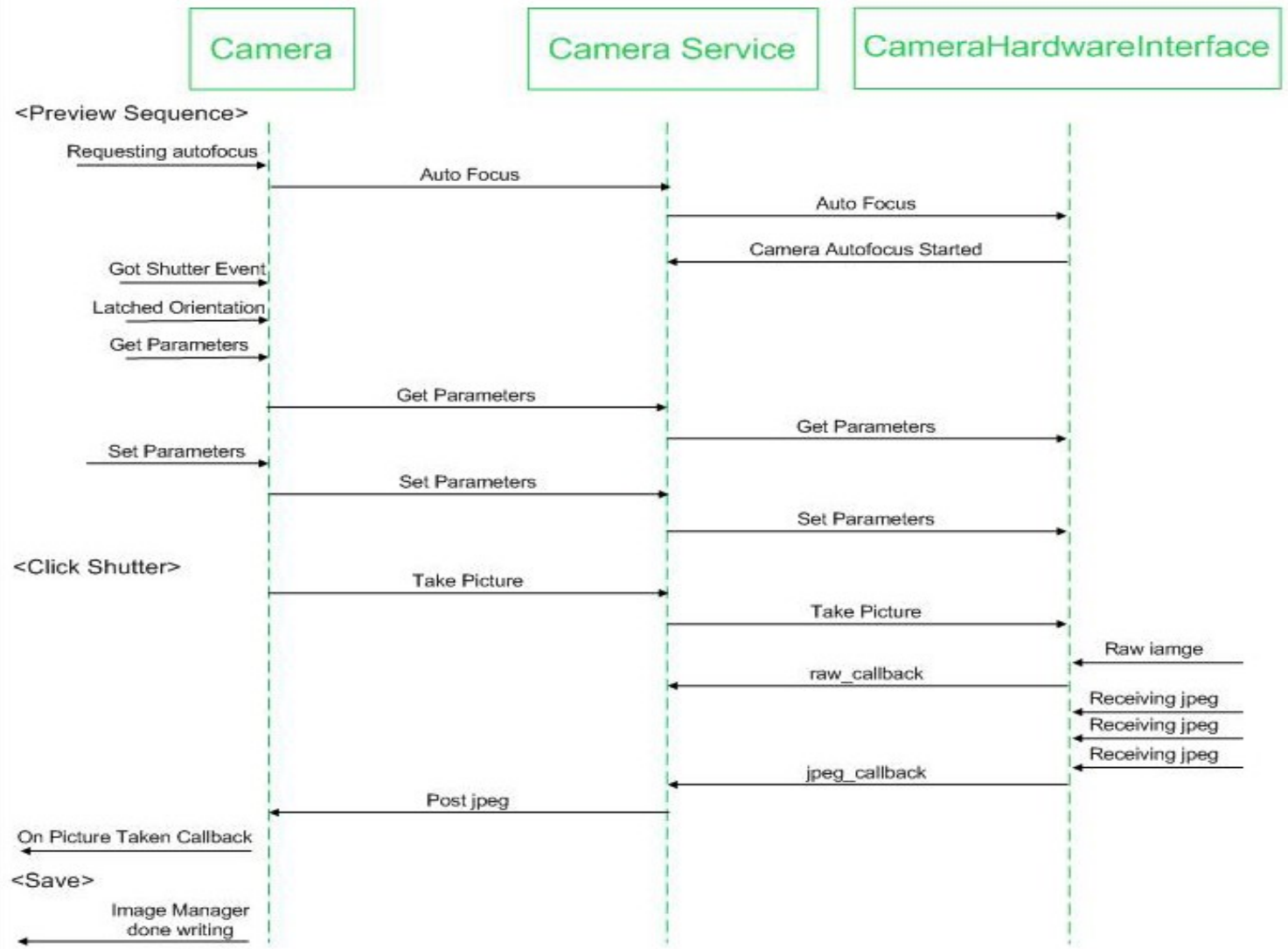
Camera API

Preview



Camera API

Take picture

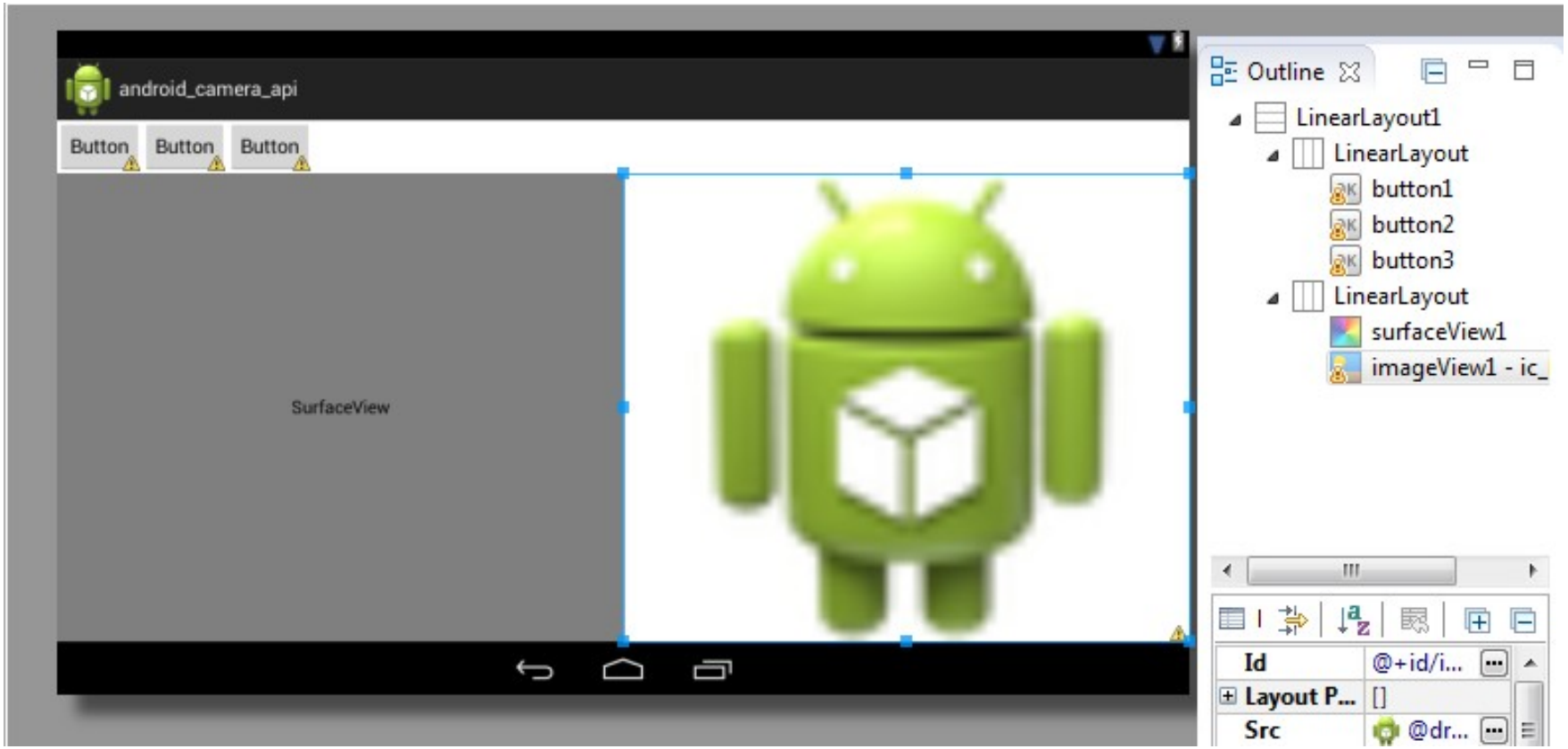


Camera API

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="com.example.android_camera_api"
   android:versionCode="1"
   android:versionName="1.0" >
   <uses-sdk
      android:minSdkVersion="9"
      android:targetSdkVersion="14" />
<uses-permission android:name="android.permission.CAMERA"/>
   <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
   <application
      android:allowBackup="true"
      android:icon="@drawable/ic_launcher"
      android:label="@string/app_name"
      android:theme="@style/AppTheme" >
      <activity
         android:name="com.example.android_camera_api.CameraAPI"
         android:label="@string/app_name" >
         <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
         </intent-filter>
      </activity>
   </application>
</manifest>
```

Camera API


android_camera_api
Button
Button
Button
SurfaceView
Outline
LinearLayout1
LinearLayout
button1
button2
button3
LinearLayout
surfaceView1
imageView1 - ic_
Id
@+id/i...
Layout P...
[]
Src
@dr...

Camera API

Global variable

```
private SurfaceView preview=null;
private SurfaceHolder previewHolder=null;
private ImageView imageview1;
private Bitmap bitmap;
private Camera camera=null;
private boolean inPreview=false;
private boolean cameraConfigured=false;
```

Camera API

Prepare preview

```
previewHolder=preview.getHolder();
    previewHolder.addCallback(surfaceCallback);

previewHolder.setType(SurfaceHolder.SURFACE_TYPE_
PUSH_BUFFERS);
```

```
 SurfaceHolder.Callback surfaceCallback=new
SurfaceHolder.Callback() {
    public void surfaceCreated(SurfaceHolder holder) {
    }
    public void surfaceChanged(SurfaceHolder holder, int format,
                        int width, int height) {
      initPreview(width, height);
    }
    public void surfaceDestroyed(SurfaceHolder holder) {
    }  };
```

Camera API

Select camera to open

```
Camera.CameraInfo info=new Camera.CameraInfo();

    for (int i=0; i < Camera.getNumberOfCameras(); i++) {
      Camera.getCameraInfo(i, info);

      if (info.facing ==
Camera.CameraInfo.CAMERA_FACING_FRONT) {
        camera=Camera.open(i);
      }
    }
```

Camera API

Get the nearest preview size to match input size

```
private Camera.Size getBestPreviewSize(int width, int height,
                Camera.Parameters parameters) {
Camera.Size result=null;
for (Camera.Size size : parameters.getSupportedPreviewSizes()) {
if (size.width <= width && size.height <= height) {
if (result == null) {
result=size;
}else {
int resultArea=result.width * result.height;
int newArea=size.width * size.height;
if (newArea > resultArea) {
result=size;
}}}}
return(result);
}
```

Camera API

Smallest picture size support by camera

```
private Camera.Size getSmallestPictureSize(Camera.Parameters
parameters) {
Camera.Size result=null;
for (Camera.Size size : parameters.getSupportedPictureSizes()) {
if (result == null) {
result=size;
}else {
int resultArea=result.width * result.height;
int newArea=size.width * size.height;
if (newArea < resultArea) {
result=size;
}}}
return(result);
}
```

Camera API

initPreview

```
private void initPreview(int width, int height) {
   if (camera != null && previewHolder.getSurface() != null) {
      try {
camera.setPreviewDisplay(previewHolder);
} catch (IOException e) {
e.printStackTrace();
}   if (!cameraConfigured) {
      Camera.Parameters parameters=camera.getParameters();
      Camera.Size size=getBestPreviewSize(width, height, parameters);
      Camera.Size pictureSize=getSmallestPictureSize(parameters);
      if (size != null && pictureSize != null) {
        parameters.setPreviewSize(size.width, size.height);
       parameters.setPreviewFrameRate(8);//8,10,15,20,24,25,30
        parameters.setPictureSize(pictureSize.width,
                          pictureSize.height);
        parameters.setPictureFormat(ImageFormat.JPEG);
        camera.setParameters(parameters);
        cameraConfigured=true;
      }    }   } }
```

Camera API

Start preview

```
private void startPreview() {
   if (cameraConfigured && camera != null) {
     camera.startPreview();
     inPreview=true;
   }
 }
```

```
@Override
protected void onPause() {
  if (inPreview) {
     camera.stopPreview();
    }
    camera.release();
    camera=null;
    inPreview=false;
super.onPause();
}
```

Camera API

Take picture

```
if (inPreview) {
      camera.takePicture(null, null, photoCallback);
      inPreview=false;
     }
```

```
Camera.PictureCallback photoCallback=new
Camera.PictureCallback() {
    public void onPictureTaken(byte[] data, Camera camera) {
    bitmap=BitmapFactory.decodeByteArray(data, 0,
data.length);
   imageview1.setImageBitmap(bitmap);
      camera.startPreview();
      inPreview=true;
    }
  };
```

Thank you ☺