An Introduction to Android Application Development

Serdar Akın, Haluk Tüfekçi

ARDIC ARGE – http://www.ardictech.com

April 2011

Development Environment

Programming Languages

- **Java** (Officially supported)
- **C** (Android NDK Needed)
- **C++** (Android NDK Needed)

Android Software Development Kit (SDK)

- Dalvik Cross Assembler
- Android Debug Bridge (adb)
- Dalvik Debug Monitor Service (ddms)
- Android Emulator
- Native Development Kit (NDK)
- Class Library
- Documentation
- Sample Codes
- System Images
- USB Driver (OEM USB drivers may be installed)

Integrated Development Environment

- eclipse Galileo
- Java Development Tools
- Java Development Kit 5 or 6
- Android Development Tools Plugin

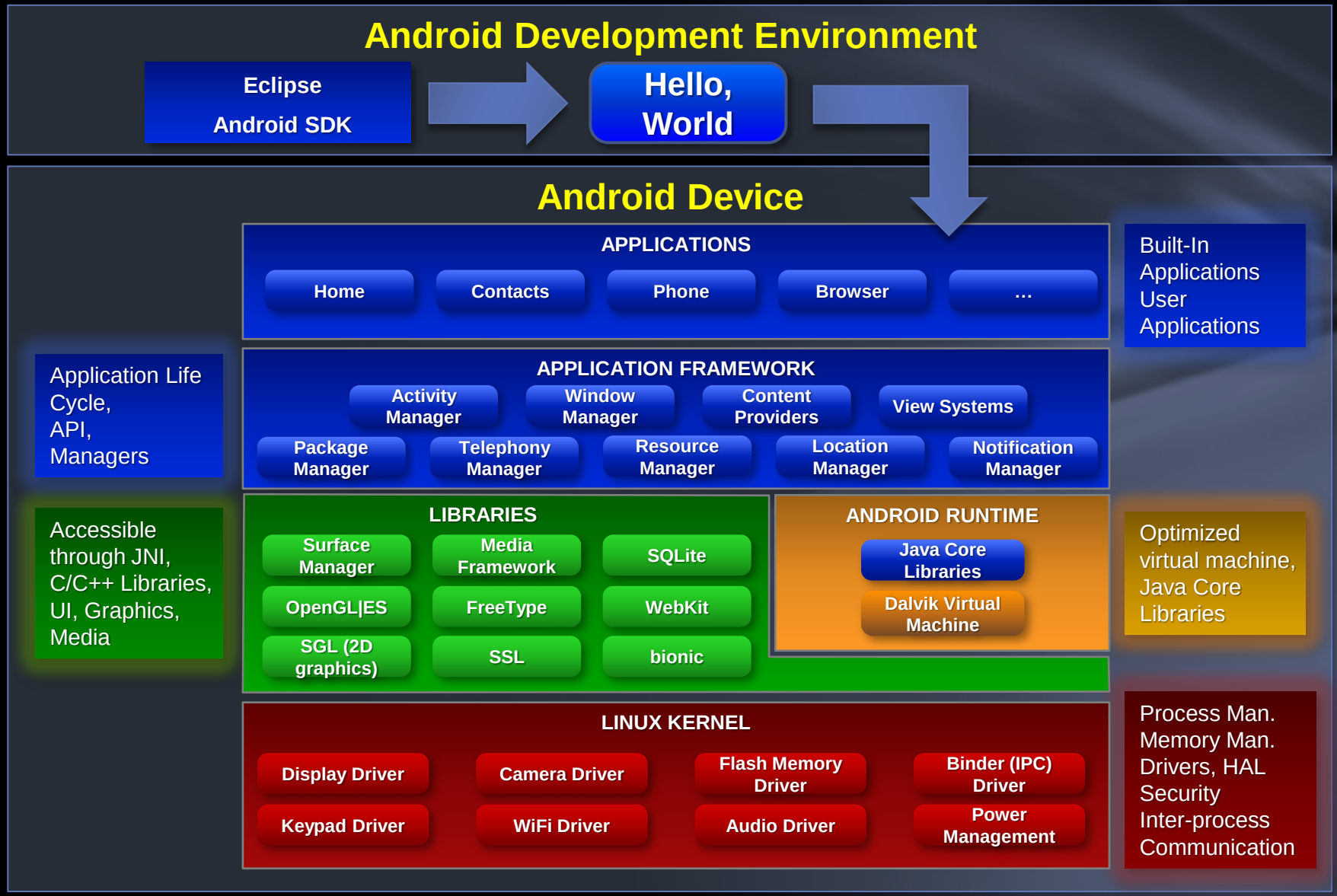
Operating System

- Microsoft Windows
- Mac OS X
- Linux
- Ubuntu

Computing System

- intel

Environments



Application Framework

Activity Manager	Application lifecycle Navigation
View Manager	Programmable controls List, Grid, Textbox, Button, Embedded web browser
Resource Manager	Localized strings Graphics Layout files
Content Providers	Access data Share data
Notification Manager	Custom alerts for end user
Window Manager	Screen organization Surface allocation for applications
Package Manager	Application Installation

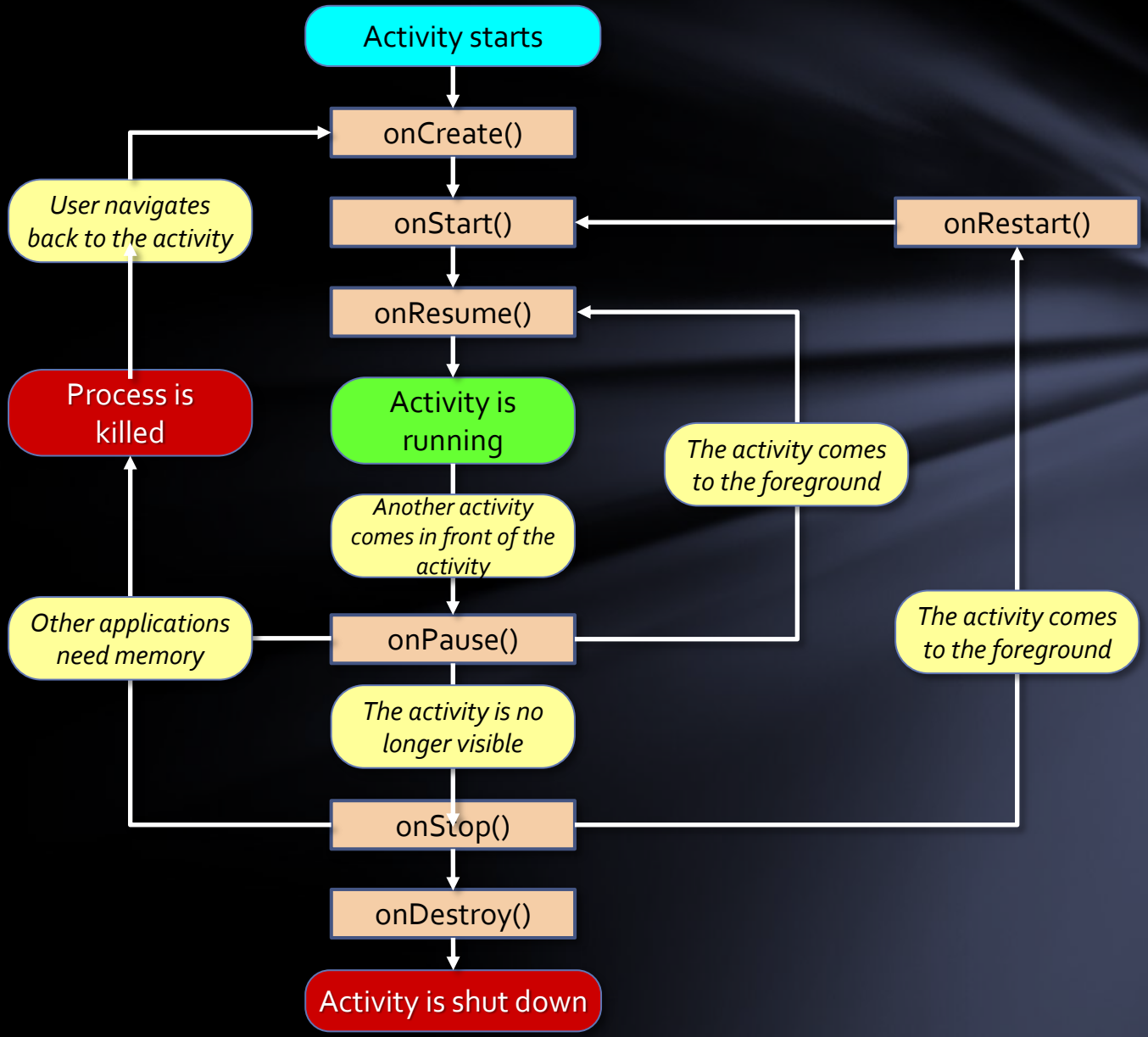
Hello, World


```
package com.google.android.HelloWorldActivity;

import android.app.Activity;
import android.os.Bundle;

public class HelloWorldActivity extends Activity {
    public HelloWorldActivity() {
    }
@Override
    public void onCreate(Bundle icicle) {
        super.onCreate(icicle);
        setContentView(R.layout.helloworld_activity);
    }
}
```

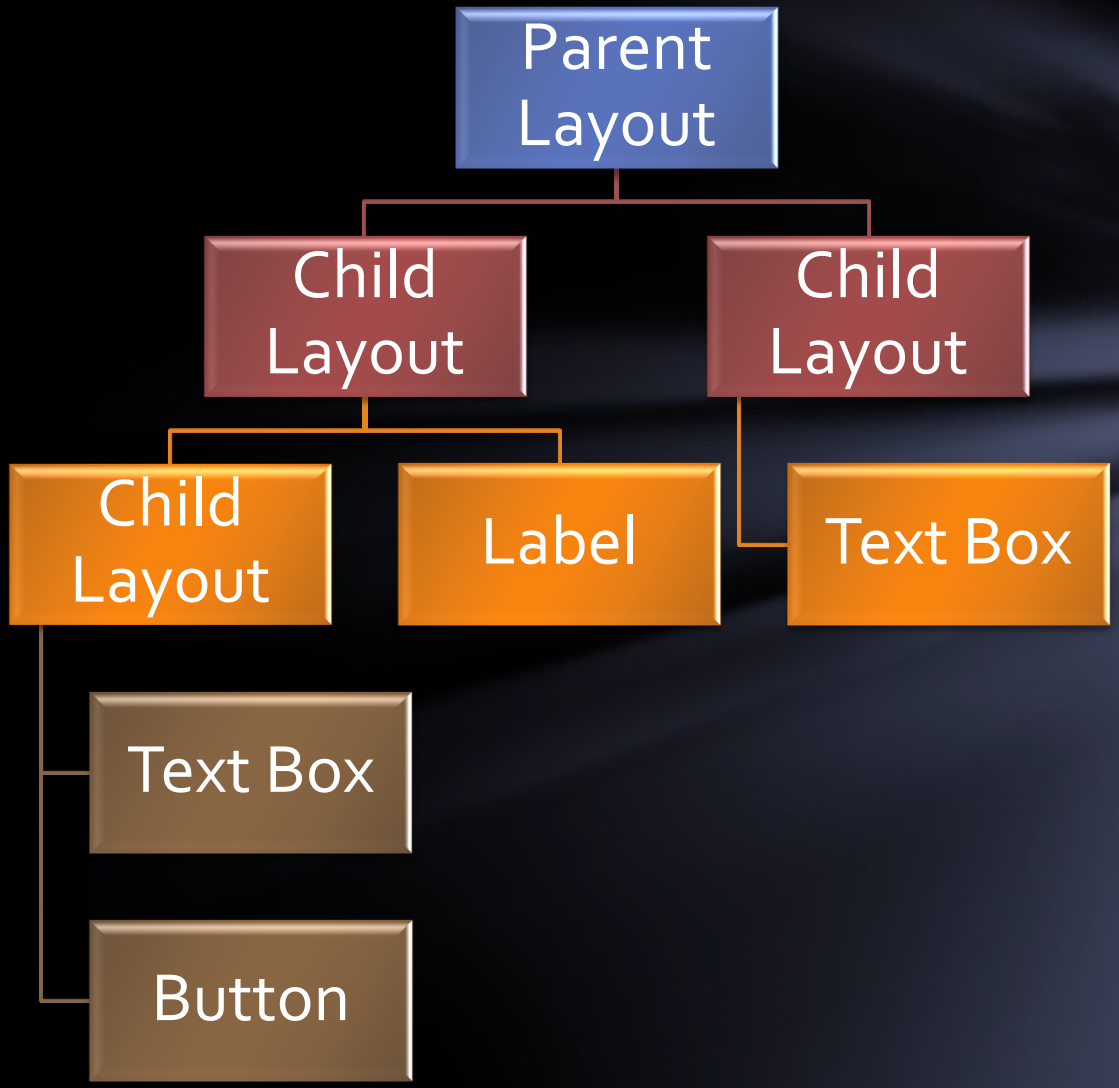
Application Structures

Activity	• A single, focused screen that the user can do
Intent	• Intents are used for inter-communication among activities or services • An intent is a data structure that stores a message
Broadcast Receiver	• A broadcast receiver triggers an intent to start an application
Content Provider	• A content provider is an interface to store and retrieve data and make it accessible to all applications
Service	• Services run in the background for an indefinite period of time • They have no user interface interaction
Resource	• Externalization of strings and graphics

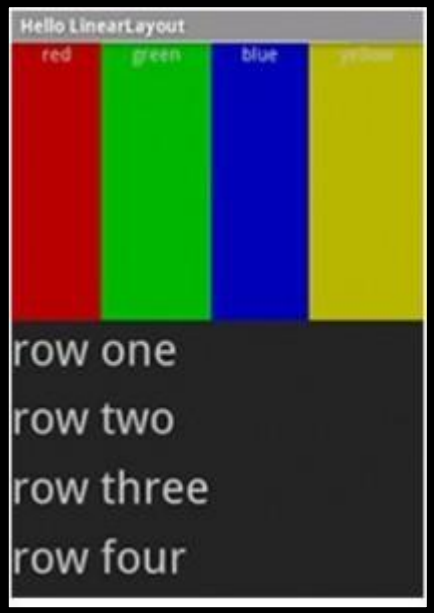
Application Lifecycle

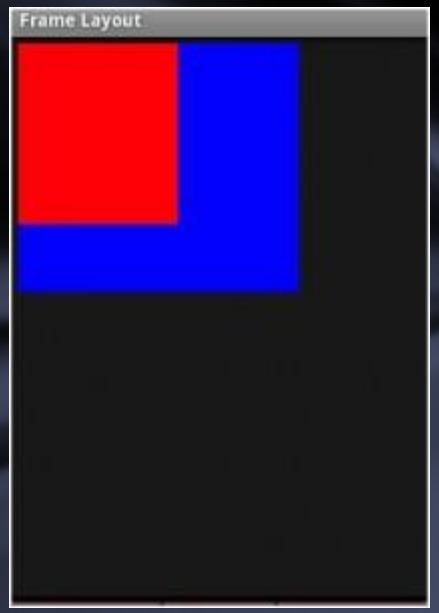


Views and Layouts



Layouts





Linear Layout

- Vertical
- Horizontal

Relative Layout

Table Layout

Frame Layout

* Absolute Layout is deprecated

View Examples

GridView

TabView

MapView

WebView

Key points In Android Application Development

- Performance and battery life is important in mobile devices
  - Recycle java objects
  - Avoid floating point arithmetic
  - Use efficient data format and parser
  - Reduce transferred data size, gzip text data
  - Keep memory small in order not to be killed
- Services
  - Do not use services as daemons
  - Start them with AlarmManager only when device is awake
  - Use receivers to awaken services
  - Check battery life and do not run heavy jobs if battery is critical

References

- http://developer.android.com/guide/index.html
- http://androidappdocs.appspot.com/index.html