JAOO conference
Oct. 3 - 8 in Aarhus, DK 2010

Android Geek Night

Application framework

software pilots
TRIFORK.

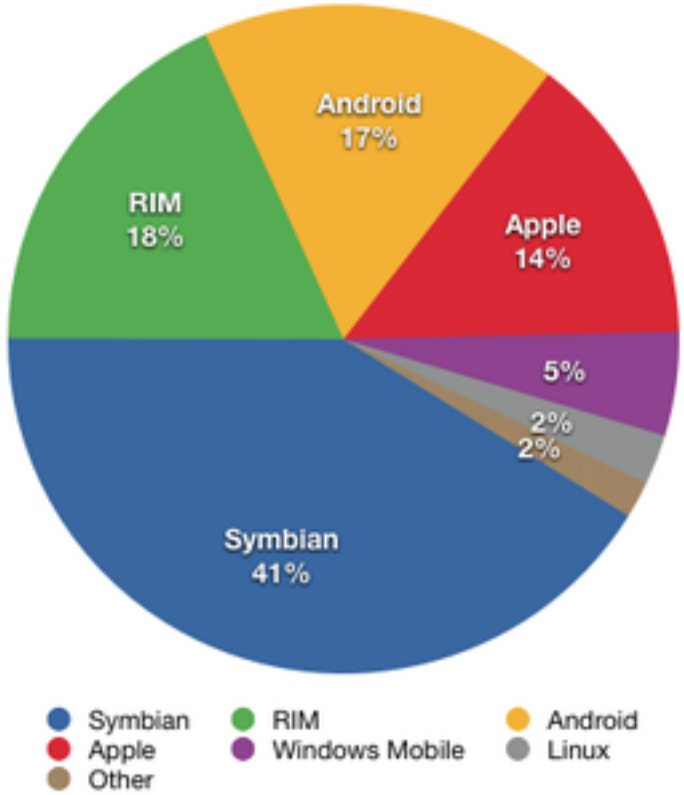
Agenda

JAOO conference
Oct. 3 - 8 in Aarhus, DK 2010

1. Presentation
    1. Trifork
    2. JAOO 2010
2. Google Android headlines
3. Introduction to an Android application
4. New project using ADT
5. Main building blocks of an application
    - Activities, Services, Broadcast, Intent, Content Providers
6. Tools
7. Interfacing with sensors, GPS and SD card
8. QA
9. Recordbeater by Ole
10. Sandwich

Google Android

- Operating system targeting mobile devices
- Linux based - with additions
- Open source under the Apache License
- Allows development in Java
  - Or Scala, JRuby, Groovy ..

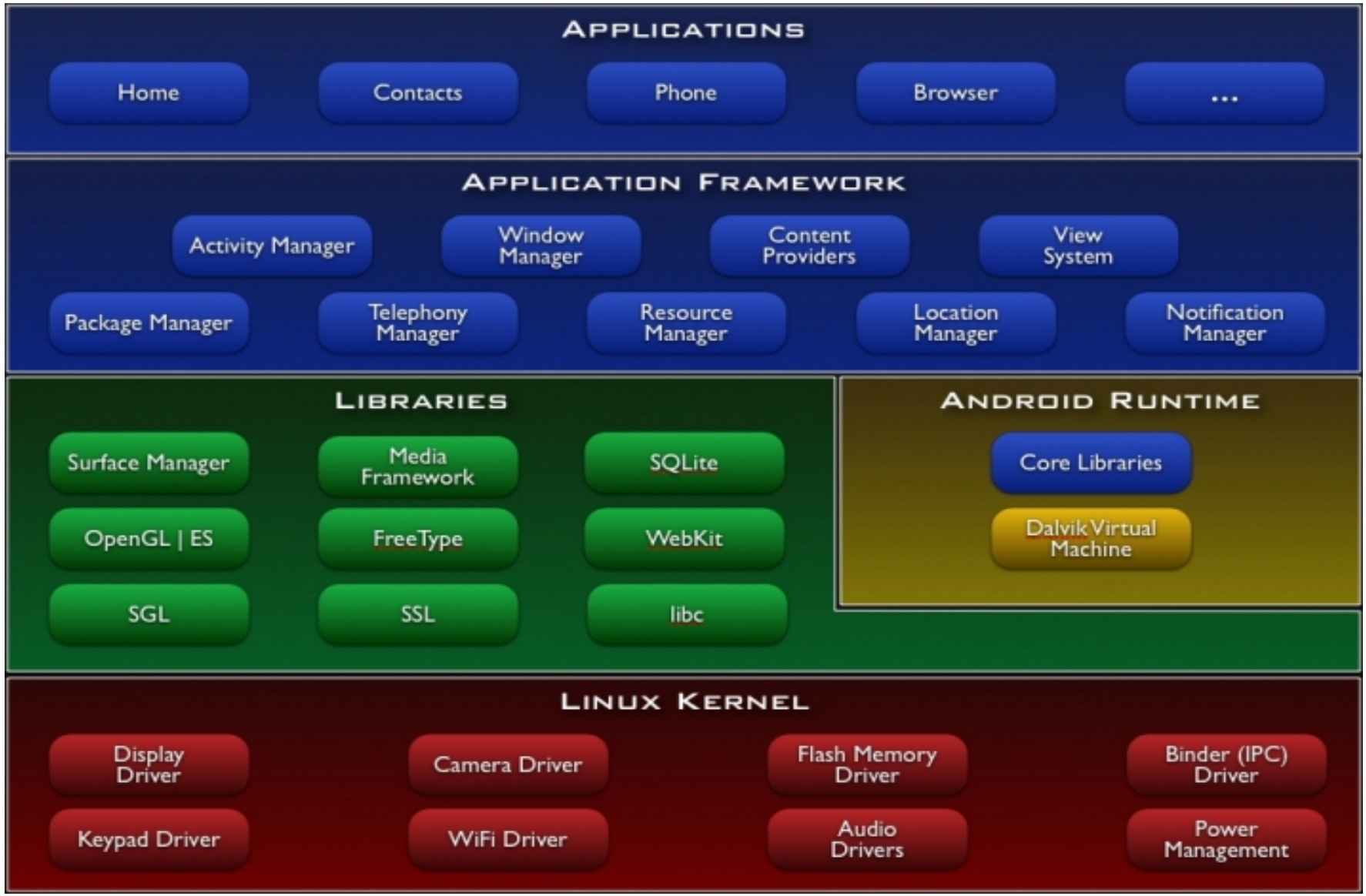


Share of 2010 Q2 smartphone sales to end users by operating system, according to Gartner.

Dalvik Virtual Machine

- Virtual machine developed by Google for mobile devices
- Uses the Dalvik Executable (.dex) format
- Designed for limited processing power and memory
- Register-based architecture
  - as opposed to stack machine Java VMs
- JIT from 2.2
- Class library based on Apache Harmony
  - No AWT, Swing
  - No Java MEDisplay data, handle user input, launch other activities

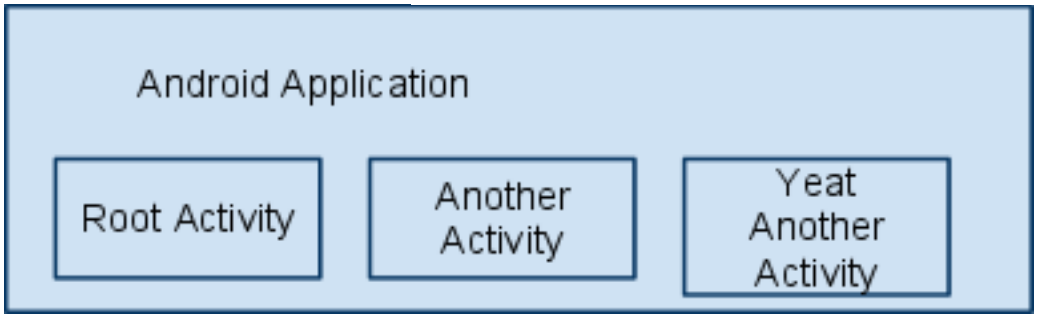
Architecture



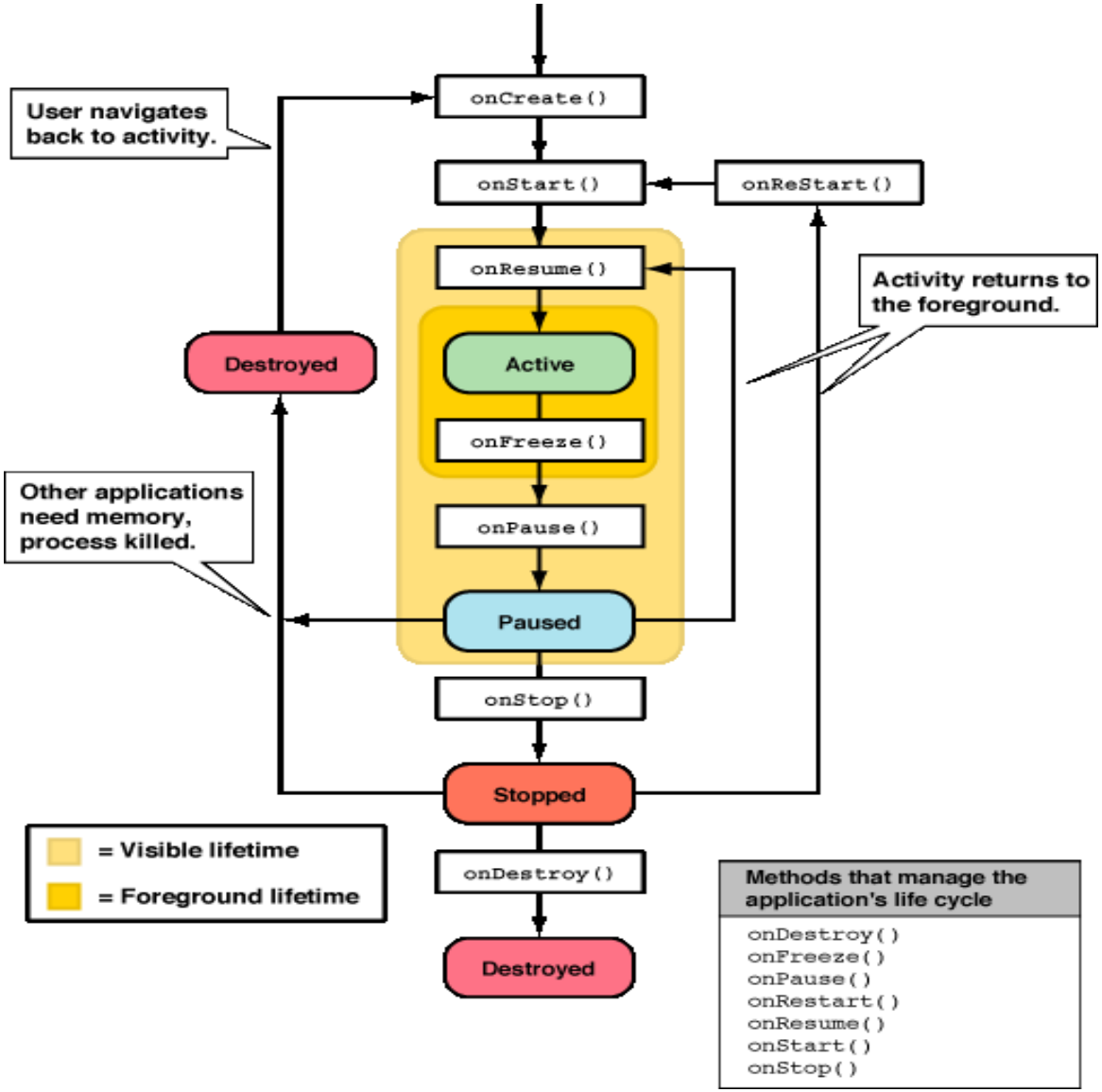
Main Building Blocks

Activities

- Display data, handle user input, launch other activities
- An Activity is for an Android Application like a webpage for a webapp.
- Correspond (you might say) to a UI screen.

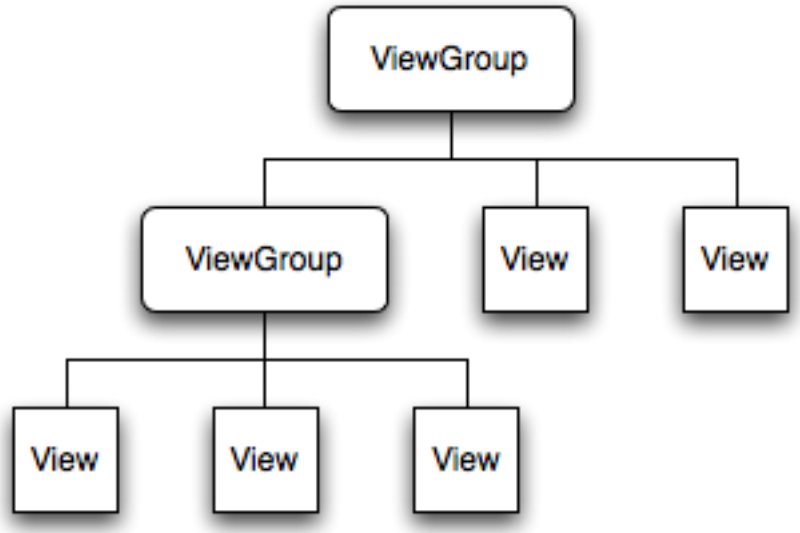


Activity Lifecycle



Views

- Activity UI is build using hierarchy of Views
- View is (mostly) defined in XML layout files
  - Can be defined in code as well..
- Components Layouts and UI Components
- Example HorizontalLayout, Button



Intent

- Intent objects are *asynchronous* messages used to activate Activities, Services and BroadcastReceivers
- Intent can be explicit (Give the class that you want started, Android OS will start)

```
startActivity(new Intent(getApplicationContext(),MyActivity.class));
```

Intent 2

- Intent can be implicit
  - The Android OS resolved the action using IntentFilter (more on this later)
  - Intent consist of a Action and Data
- (Implicit) Intent therefore says:
  - "Do this" Intent.Action on "That" Intent.Data
- This is nice .. then we can do something like!

```
Uri number = Uri.parse("tel:24234232");
Intent dial = new Intent(Intent.ACTION_DIAL,
number);
startActivity(dial);
```

Intent Filter

- Intent is matched against IntentFilters.
- Specified in the AndroidManifest.xml (*)
- Can contain the following elements.

```
<action android:name="....." />
<data android:mimeType="....." />
<category android:name="....." />
```

* The one exception would be filters for broadcast receivers that are registered dynamically by calling Context.registerReceiver(); they are directly created as IntentFilter objects.)

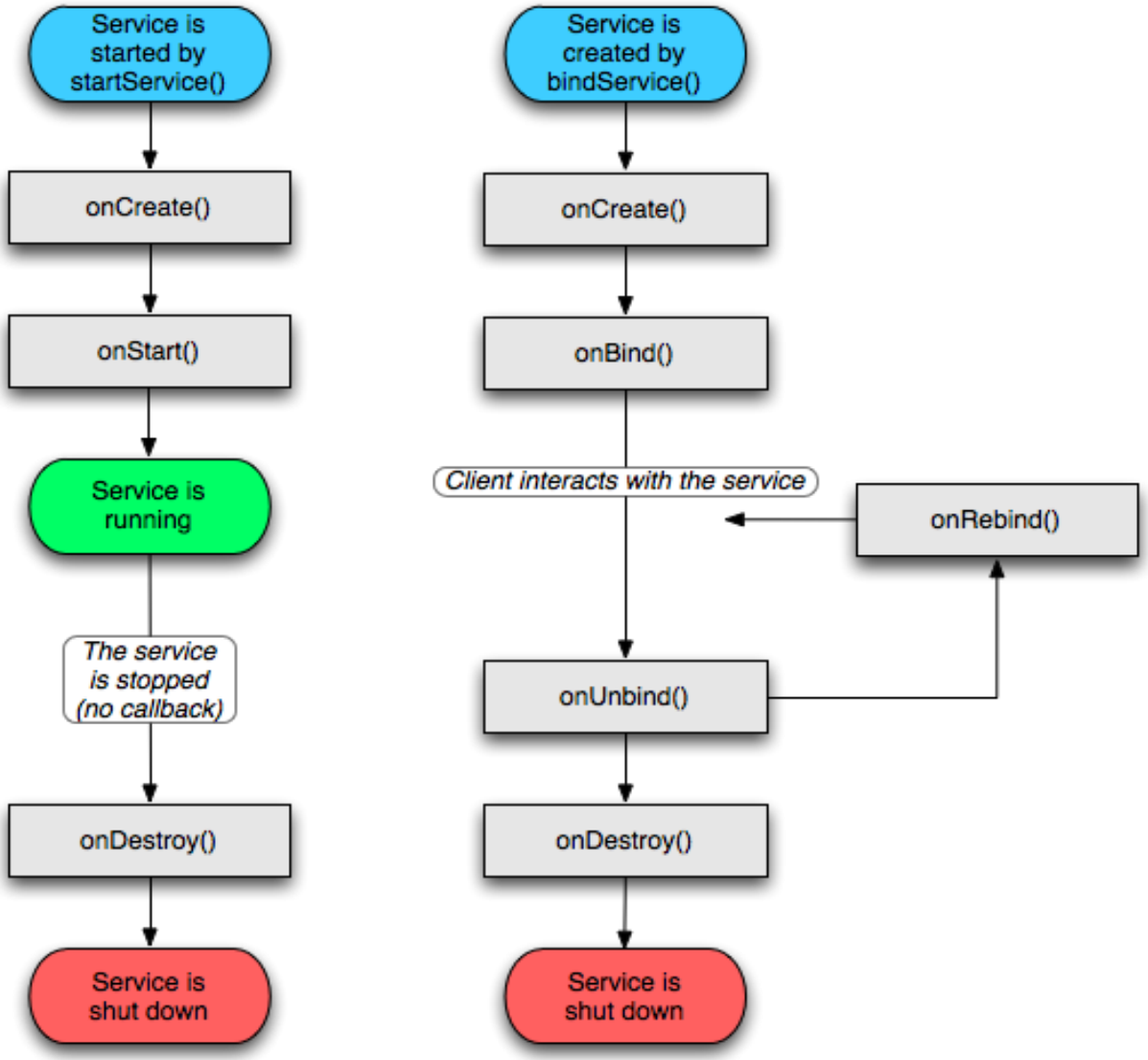
Intent Resolution

- Done by the Android OS
- Find the 'Best' component to activated from an Intent
- Native Android application components are part of the Intent resolution process in exactly the same way as third-party applications.
- My Little sample app.
  - Country List
  - My Jaoo conference (More than one app can respond to intent)

Service

- An activity without a UI
- Should be defined in the manifest
- Can run as a background process
- Start by either
  - startService
  - bindService
- Remember it runs in the main thread...

Service Lifecycle

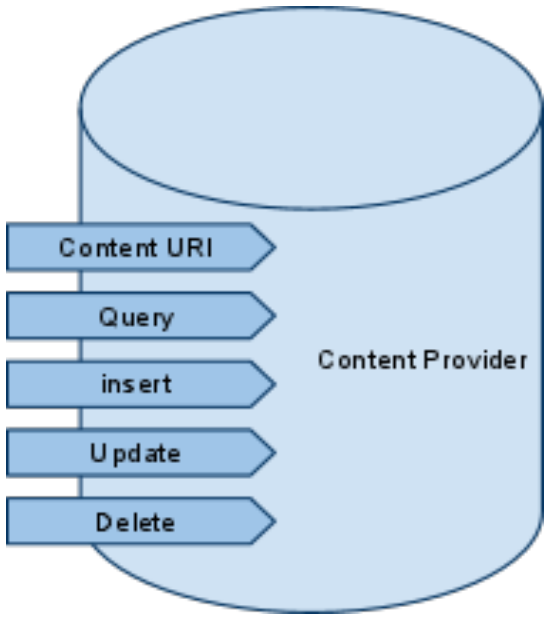


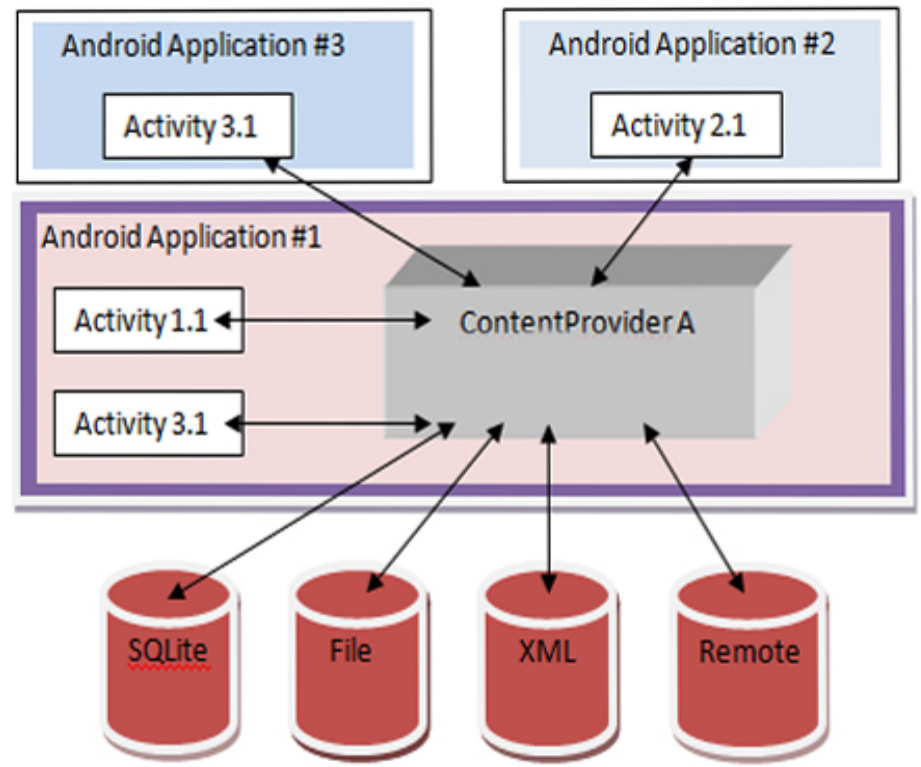
Broadcast Receivers

- Receive system wide event broadcast
- Defined in the manifest.xml (as well)
- No user interface
  - But may start an activity, play sound, show notification, start a service.
- Examples receives and reacts to broadcast announcements
  - SMS received
  - Timezone changed
  - Battery low
  - User setting changes

Content Providers

- Access through ContentResolver
- Content is record based
- Access data accross process boundaries
- Should be implemented thread save
- Example of builtin . Contact, Mediastore





Security and Permissions

- Applications
  - ... must be signed
  - ... run in their own sandbox
- No application has (default) permission to perform changes that will impact other applications.
- Declared statically and known at install time.
  - Therefore you can not change after install
- Example use permissions:

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

- You can declare you own permissions.

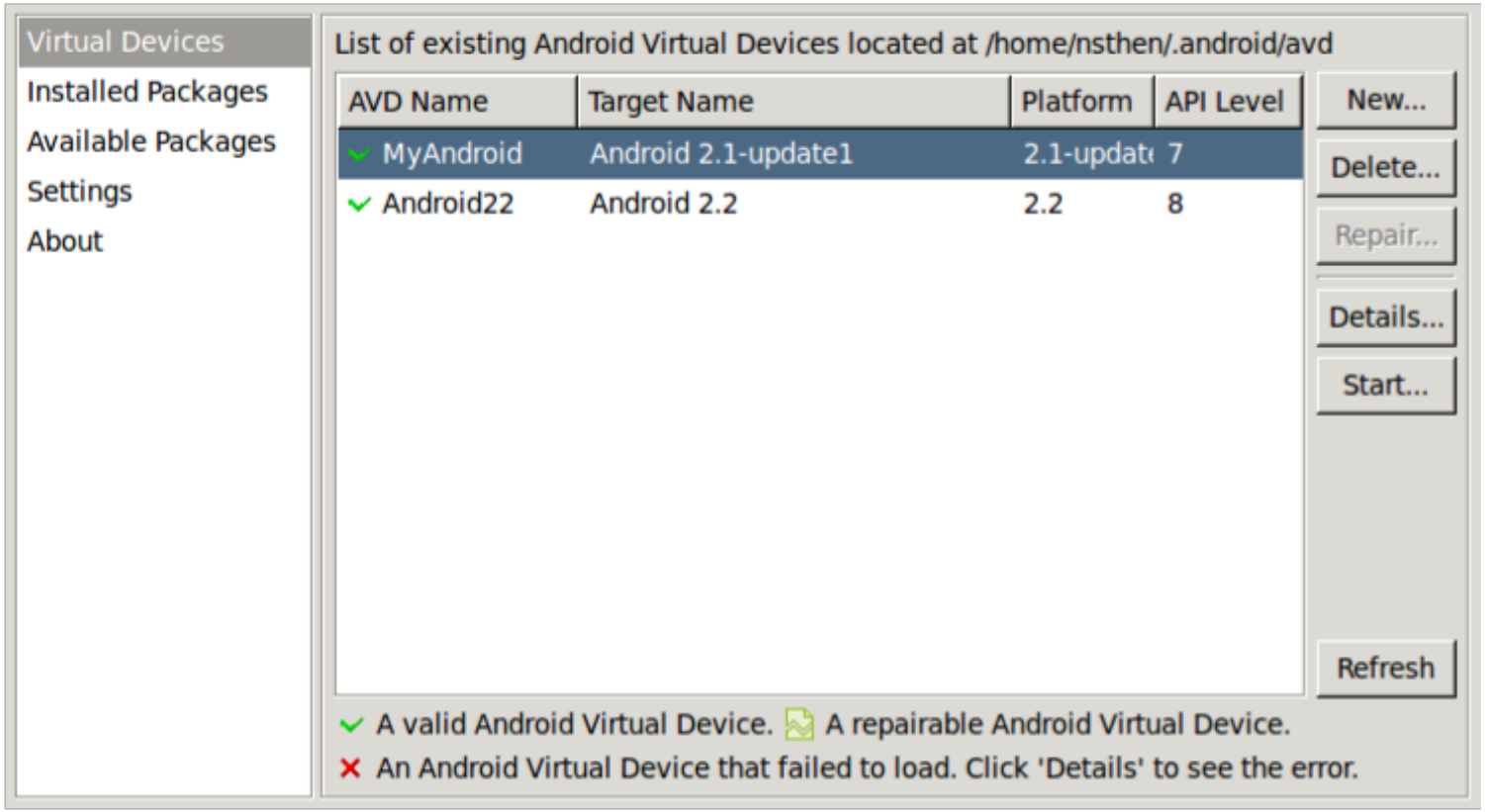
```
<permission android:name="com.me.myapp.permission.DEADLY_ACTIVITY"
    android:label="@string/permlab_deadlyActivity"
    android:description="@string/permdesc_deadlyActivity"
    android:permissionGroup="android.permission-group.COST_MONEY"
    android:protectionLevel="dangerous" />
```

Android Tools

- Virtual Devices
- Debugging & profiling
- GUI
- Build & packaging
- ... a lot of this is available from ADT

Android SDK and AVD Manager

$ android



Build & Packaging

- dx
  - .class bytecode -> Android bytecode
- apkbuilder
  - package Android application
- zipalign
  - align data in apk for more efficient mem mapping

GUI

- hierarchyviewer
  - Inspect View hierarchy, magnifier (pixel level)
- draw9patch
  - NinePatch images (e.g. for background imgs)
- layoutopt
  - detect inefficiencies and other problems in layout

Android Debug Bridge

- logcat
- Install, pull, push
- jdwp
- port forwarding
- shell

Debugging Cont'd

- ddms - Dalvik Debug Monitor Server
  - threads
  - logcat
  - fileviewer
- traceview
  - trace log viewer

sqlite3

- Simple but powerful, one-file-based relational database
- Can be accessed with sqlite3 tool from shell

Android with ant

- Generated ant build file
- Uses SDK tools to compile, debug and run

```
$ android create project –target ...
```

Android & Maven

- No official repository
  - Deploy in your own repo or install locally
- Uses android.jar and command line tools
- Modules & libraries works as in any other maven project
- maven-android-plugin

Android Hardware

How to interface to special hardware present on most Android devices

- Various sensors
- GPS
- SD card

and considerations regarding one piece of hardware present on all mobile devices ...

Android Hardware

Android is build on top of the Linux kernel which is responsible for traditional OS stuff:

- Hardware drivers
  - WIFI, sensors, telephony, display, ...
- Process management
- Memory management

Different hardware vendors will provide drivers for their specific hardware

Android Hardware

Writing software to interface with the hardware of an Android device a funny and challenging task.

Many applications need only to worry about one single piece of hardware...

Android Hardware

JAOO conference
Oct. 3 - 8 in Aarhus, DK 2010

THE BATTERY

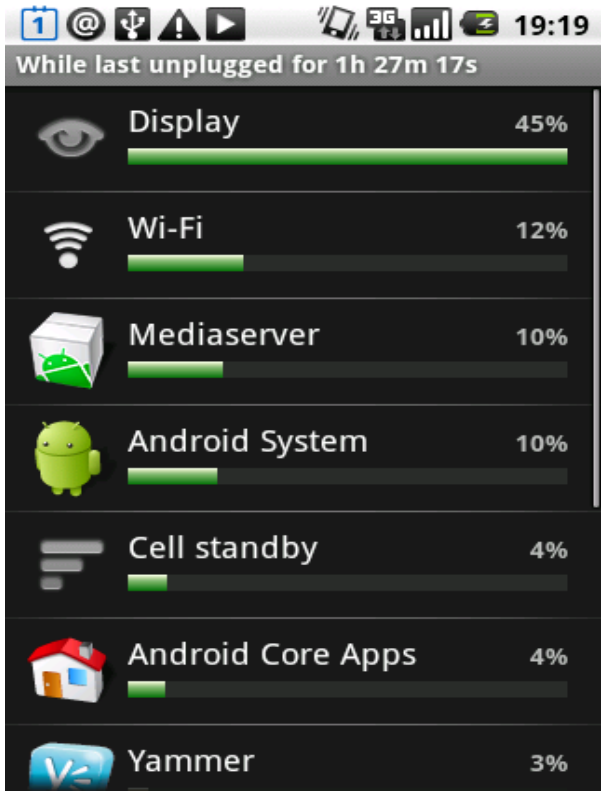
YOU SHOULD WORRY TOO !

Android Hardware

Read the guidelines on developer.android.com

http://developer.android.com/guide/practices/design/performance.html

Or your application might get un-installed quickly!



android.hardware

"Provides support for hardware devices that may not be present on every Android device."

Use

```
SensorManager.getDefaultSensor(int type)
```

or

```
SensorManager.getSensorList(int type)
```

to look for sensors of a specific type.

Sensors

The following sensors are defined in `Sensor`:

- **`TYPE_ACCELEROMETER`**
- `TYPE_GYROSCOPE`
- **`TYPE_LIGHT`**
- **`TYPE_MAGNETIC_FIELD`**
- **`TYPE_ORIENTATION`**
- `TYPE_PRESSURE`
- **`TYPE_PROXIMITY`**
- `TYPE_TEMPERATURE`

The highlighted ones are present on my Nexus One

Sensors

All sensors are monitored using the same API.

The example application shows the sensors found, and let the user select the one to watch.

<Example app: Senfo>

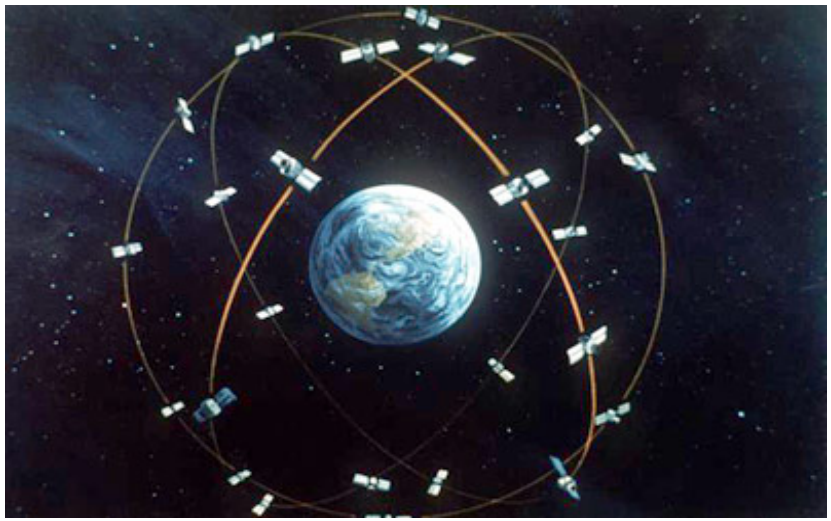
Location Providers

Access to location providers goes through the

```
android.location.LocationManager
```

Choose between the following types:

- `GPS_PROVIDER`
- `NETWORK_PROVIDER`
- `PASSIVE_PROVIDER`



Location Providers

<Example app: GetMeThere>

android.os.Environment


```
import android.os.Environment;
File root = getExternalStorageDirectory ();
```

Using external storage, you may need to listen for ACTION_MEDIA_ broadcast intents:

- ACTION_MEDIA_UNMOUNTED
- ACTION_MEDIA_MOUNTED
- ACTION_MEDIA_*

and check the state with getExternalStorageState()

JAOO conference
Oct. 3 - 8 in Aarhus, DK 2010

Q & A