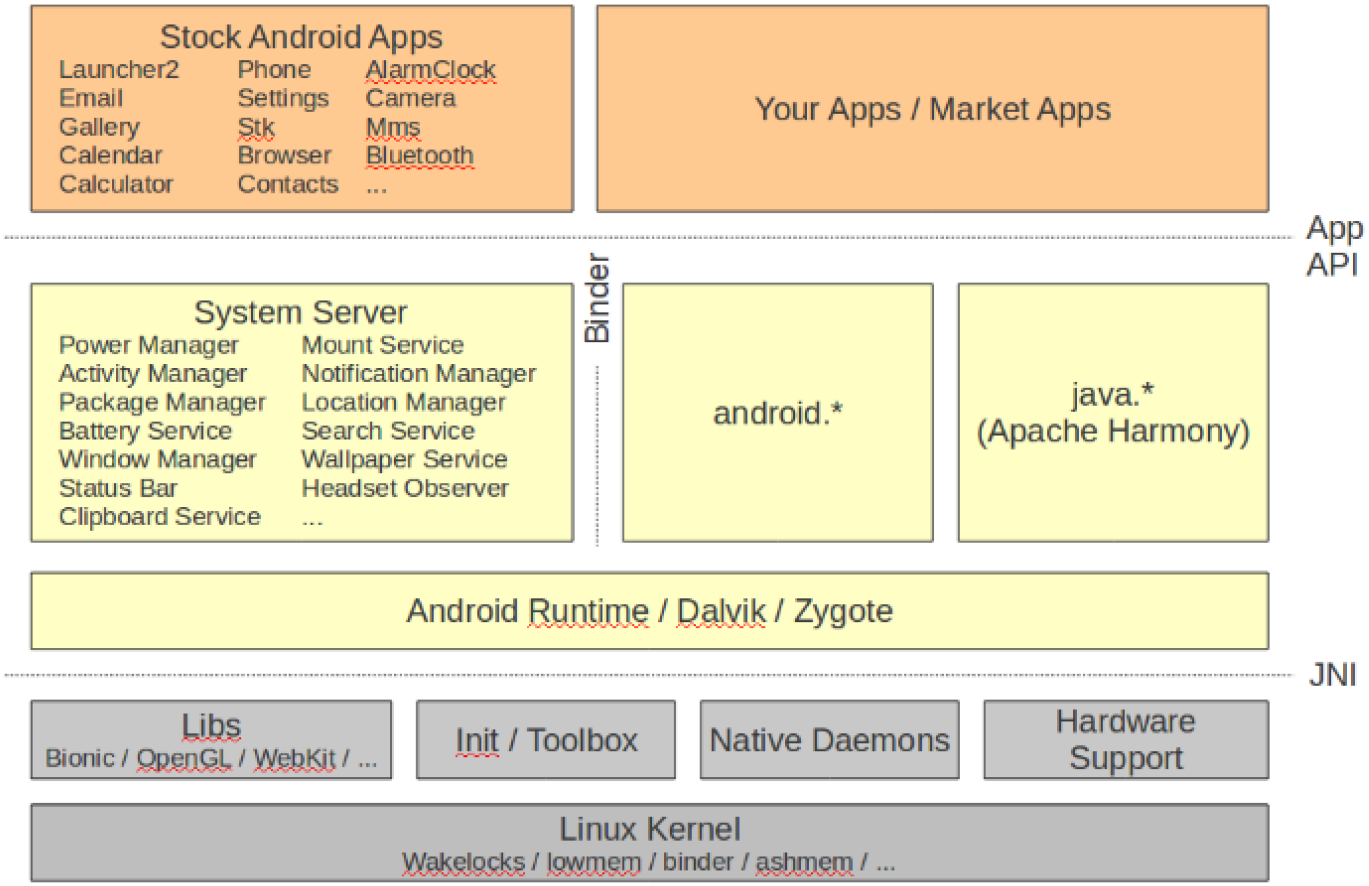
G54MDP
Mobile Device Programming

Lecture 4 – Android Application Components

Android Apps

- Applications are written in Java
  - But run on Google's own VM — Dalvik
  - Uses its own bytecode (DEX) format
- Code compiled using standard Java tools then convert to DEX format
  - Multiple class files in a single .dex file
- Code, data and resource files packed into a .apk file
  - Classes
  - Configuration
  - Resources



SDK, Eclipse, .apk

Manifest:
Perms / SDK ver.

.dex, ddms

NDK, rootfs, initrc, adb

GNU toolchain

(fastboot)

Android Programming Model

- Traditional OS applications
  - A single entry point
    - Main
  - OS loads the program into a process and executes it
- Java applications
  - A Java VM is instantiated
    - Loads all classes used by the application
    - Executes main

Android Programming Model

- Component based model
  - Multiple application entry points
    - The point through which the system can “enter” the application
  - Not all are entry points for the user
  - Each exists as a logical independent unique entity

Android Components

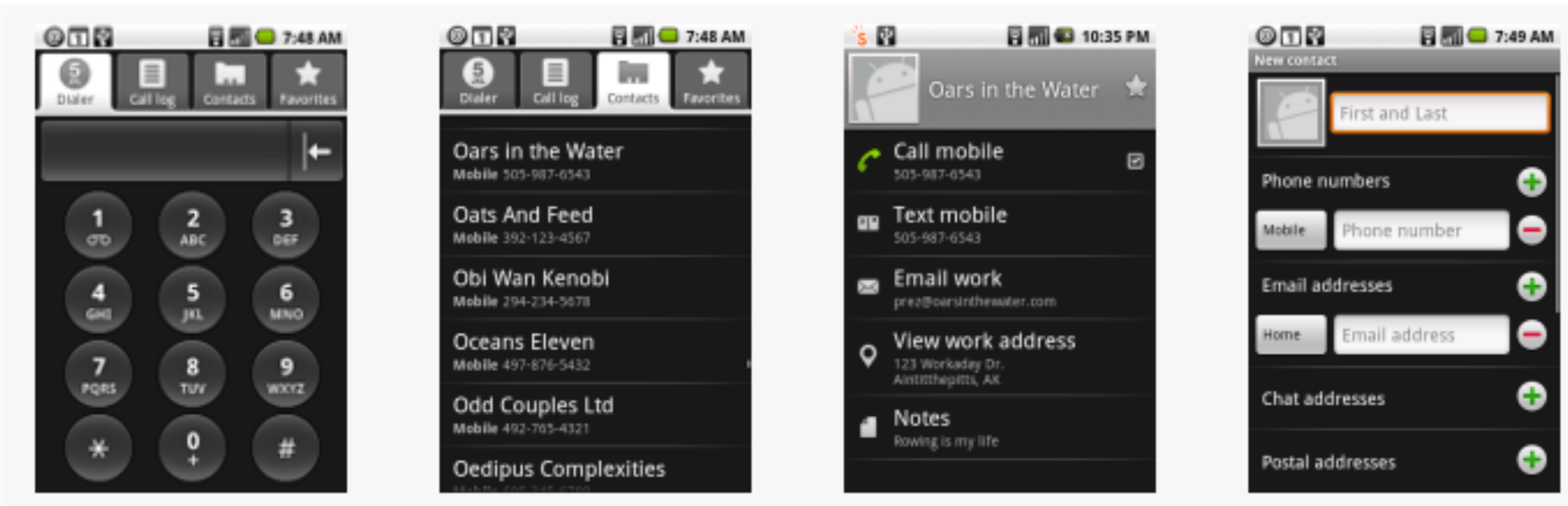
- Activities
  - UI components
- Services
  - Mechanism for doing something long-running in the background
- Broadcast Receivers
  - Respond to broadcast messages from the OS / other apps
- Content Providers
  - Make data available to / make use of data from other apps
  - No access to the file system
    - SD Card

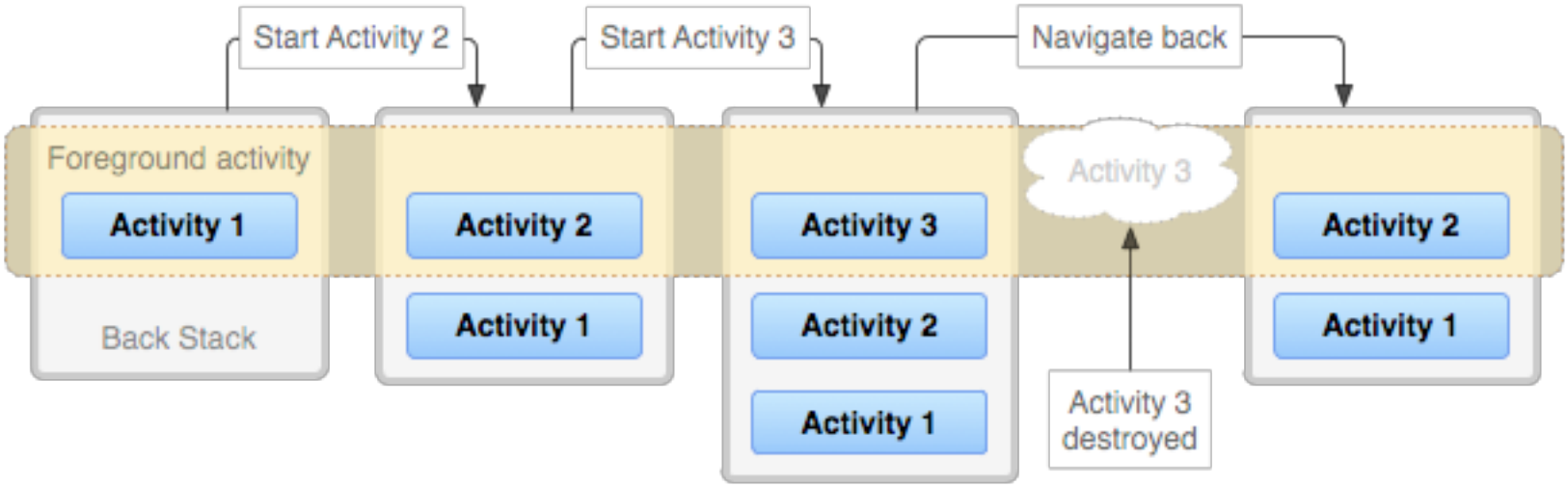
Activities

- Sub-classes of **android.app.Activity**
- Presents a visual UI
- Each Activity has its own “window”
  - Only one “window” on screen at once
- UI layout – a “View”
  - Specified in a separate XML file
  - Constructed programmatically
  - Call setContentView() to display it
- Apps usually have several Activities
  - **Context**
    - An abstract class representing the current application environment

Activities

- Activities can start other activities
- Forms a stack of Activities — current activity is on the top
- An activity should be an **atomic** part of a particular task
- Multiple activities form a Task
  - Like…
  - A Task may span multiple applications
- Activities in a task move as a unit from foreground to background and vice versa


7:48 AM
Dialer
Call log
Contacts
Favorites
1
2
ABC
3
DEF
4
GHI
5
JKL
6
MNO
7
PQRS
8
TUV
9
WXYZ
*
0
+
#
7:48 AM
Dialer
Call log
Contacts
Favorites
Oars in the Water
Mobile 505-987-6543
Oats And Feed
Mobile 392-123-4567
Obi Wan Kenobi
Mobile 294-234-5678
Oceans Eleven
Mobile 497-876-5432
Odd Couples Ltd
Mobile 492-765-4321
Oedipus Complexities
10:35 PM
Oars in the Water
Call mobile
505-987-6543
Text mobile
505-987-6543
Email work
View work address
Notes
Rowing is my life
7:49 AM
New contact
First and Last
Phone numbers
Mobile
Phone number
Email addresses
Home
Email address
Chat addresses
Postal addresses


Start Activity 2
Start Activity 3
Navigate back
Foreground activity
Activity 1
Back Stack
Activity 2
Activity 1
Activity 3
Activity 2
Activity 1
Activity 3
Activity 3 destroyed
Activity 2
Activity 1

Intent

- Activities are started by sending an **Intent**
- Represented by an **Intent** object
  - Contains the name of the action requested
  - And the URI of the data to act on

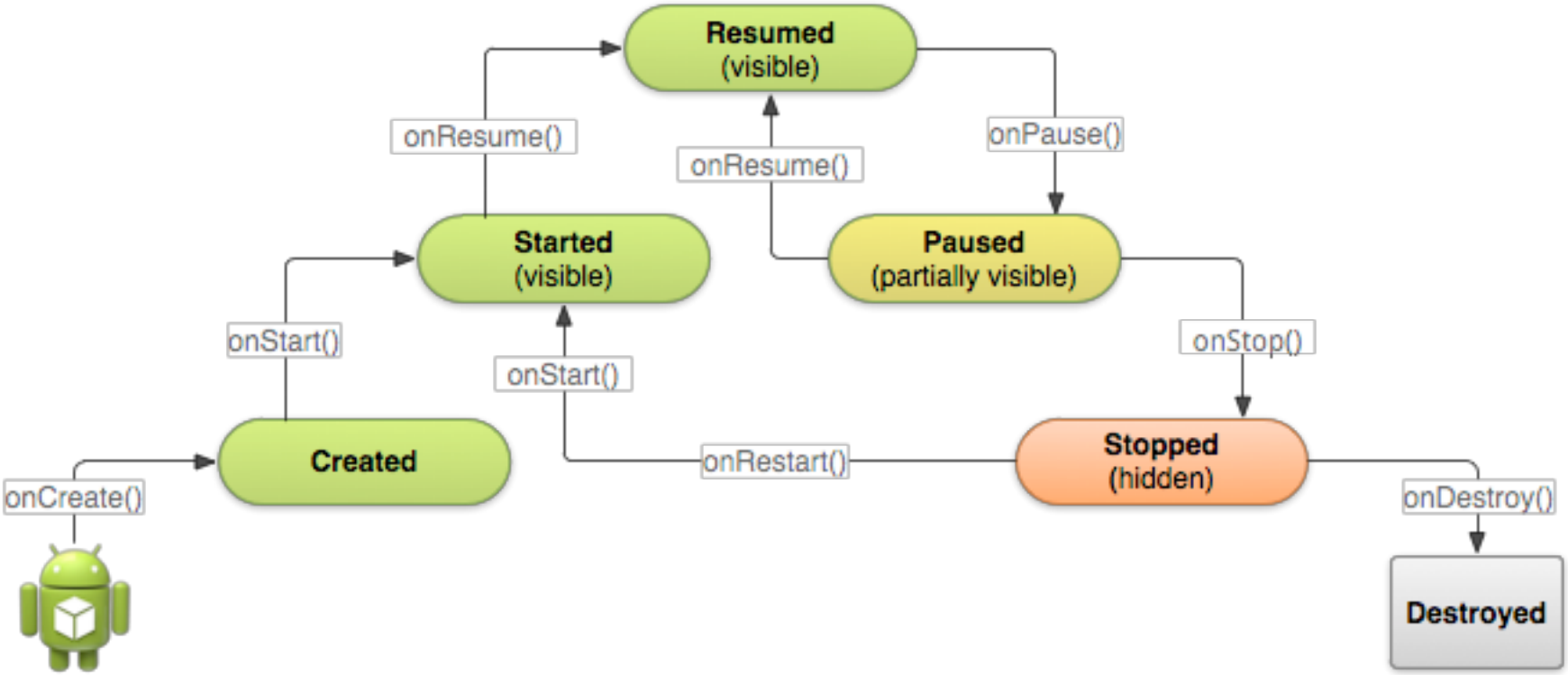
```
Uri webpage = Uri.parse("http://www.cs.nott.ac.uk");
Intent webIntent = new Intent(Intent.ACTION_VIEW, webpage);

Uri number = Uri.parse("tel:01151234567");
Intent callIntent = new Intent(Intent.ACTION_DIAL, number);

Intent myIntent = new Intent(this, otherActivity.class);
startActivity(myIntent);
```

Activity Lifecycle

- Essentially three states
  - Active
    - in the foreground
  - Paused
    - still visible, but not top
  - Stopped
    - obscured by another activity
- If paused or stopped, the system can drop the Activity from memory
  - Stopped activities are suspended in memory
    - Consume no processing resources
  - Inactive activities are destroyed if memory is required
    - Oldest first


Resumed
(visible)
onResume()
onResume()
onPause()
Started
(visible)
Paused
(partially visible)
onStart()
onStart()
onStop()
Created
onRestart()
Stopped
(hidden)
onCreate()
onDestroy()
Destroyed

Activity Lifecycle

- Memory is limited on mobile devices
- OS needs to manage its memory differently to a computer
- Java Object representing an Activity can be destroyed, while the app is notionally running
- Need to support this in our program

Services

- Subclass of android.app.Service
- No UI
- Run in background for an indefinite period of time
  - Persistently
  - While any activities are still using it
- Still runs on the main thread of the app
  - So might want to start a separate thread to avoid slowing UI

BroadcastReceiver

- Responds to broadcast announcements
- Either from System or other apps
- No UI, but can start a new Activity
- Or alert the user using a Notification
- i.e
  - Phone calls, SMS, social media

ContentProviders

- Subclasses android.content.ContentProvider
- Makes part of the application's data available to other apps / activities
- Data can be stored in the FS, or in a SQLite database etc.
- Not accessed directly, apps use a ContentResolver object

The Manifest

- Android needs to know about the contents of each application
  - What components does it contain?
  - How are these components started?
  - What does it do?
- .apk contains a manifest file (AndroidManifest.xml) that defines them
- XML syntax

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.helloworld"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.example.helloworld.MainActivity"
            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>
```

Let’s have a look…

References

- http://developer.android.com/sdk/index.html
- http://developer.android.com/guide/components/fundamentals.html
- http://developer.android.com/guide/components/activities.html