G53MDP
Mobile Device Programming

Android Application Components

Activities

Microsoft gives up on Windows 10 Mobile

By Leo Kelion
Technology desk editor

9 October 2017 | Technology | 479 | Share

MICROSOFT

Microsoft said creating Windows 10 Mobile phones was not a "focus" for the company

Microsoft appears to have abandoned its smartphone operating system ambitions.

Top

Harve accus
US ma rape a which

47 m

Catal hold

16 m

React 'fudg

6 ho

Fea

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
- 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

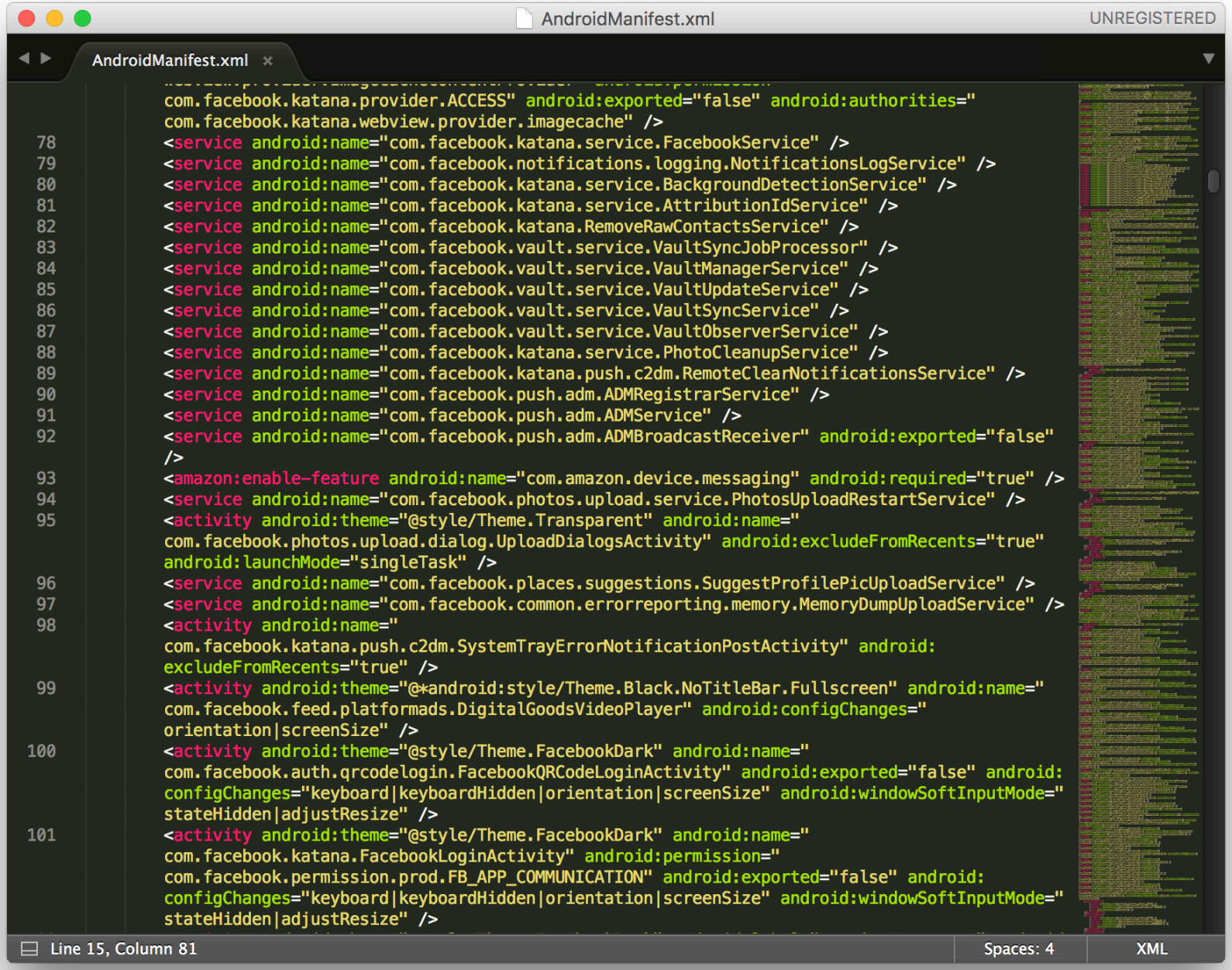
Android Components

- 4 main components
  - Communicating via specific interfaces
    - Inter- and intra- process
  - Bound at runtime
    - Each with a specific *lifecycle*
- **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
- Why?

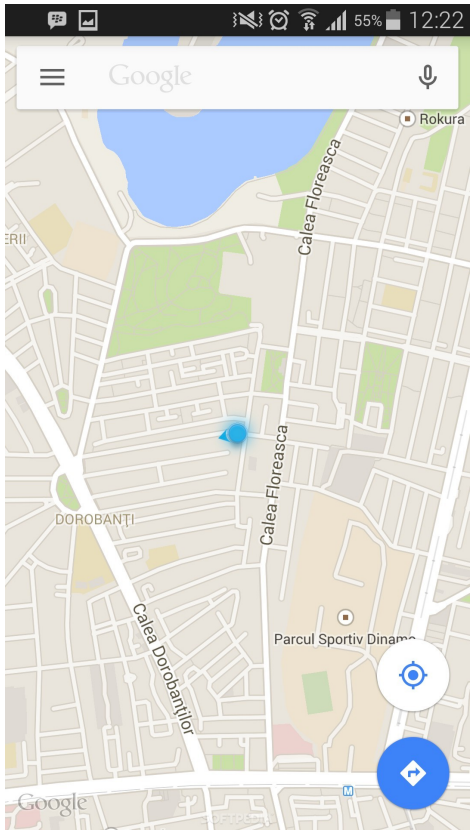
The Manifest

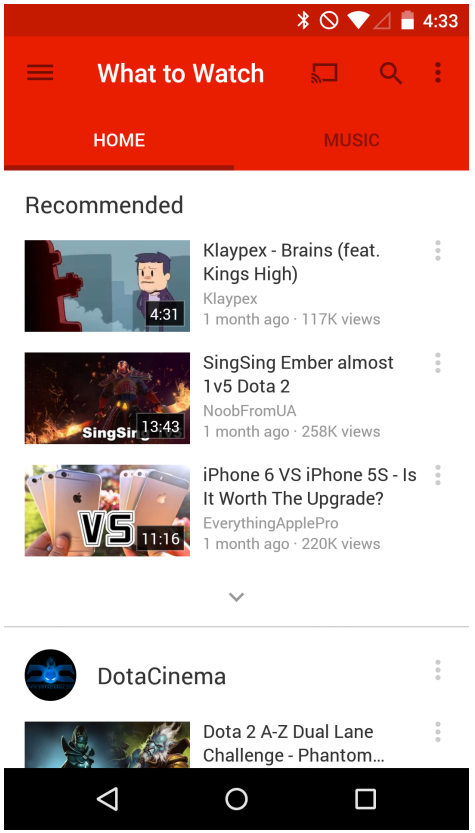
- A “list” of the application’s components
  - Classes that are not in the list are not seen as components
    - Cannot be started as such
      - Even those that look like components
    - Other Java classes
      - Not everything needs to be a component
- Used to specify entry points
  - How are they started
  - What can access it (inside and outside the application)
    - Permissions
  - What components does it contain?
    - All others
- Information about the application
  - What is it *allowed* to do
  - What are its *requirements*
    - Minimum, maximum SDK versions
- .apk contains the manifest file (AndroidManifest.xml)
  - XML syntax

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="HelloWorld"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```


AndroidManifest.xml
UNREGISTERED
AndroidManifest.xml
com.facebook.katana.provider.ACCESS" android:exported="false" android:authorities="
com.facebook.katana.webview.provider.imagecache" />
78 <service android:name="com.facebook.katana.service.FacebookService" />
79 <service android:name="com.facebook.notifications.logging.NotificationsLogService" />
80 <service android:name="com.facebook.katana.service.BackgroundDetectionService" />
81 <service android:name="com.facebook.katana.service.AttributionIdService" />
82 <service android:name="com.facebook.katana.RemoveRawContactsService" />
83 <service android:name="com.facebook.vault.service.VaultSyncJobProcessor" />
84 <service android:name="com.facebook.vault.service.VaultManagerService" />
85 <service android:name="com.facebook.vault.service.VaultUpdateService" />
86 <service android:name="com.facebook.vault.service.VaultSyncService" />
87 <service android:name="com.facebook.vault.service.VaultObserverService" />
88 <service android:name="com.facebook.katana.service.PhotoCleanupService" />
89 <service android:name="com.facebook.katana.push.c2dm.RemoteClearNotificationsService" />
90 <service android:name="com.facebook.push.adm.ADMRegistrarService" />
91 <service android:name="com.facebook.push.adm.ADMService" />
92 <service android:name="com.facebook.push.adm.ADMBroadcastReceiver" android:exported="false"
/>
93 <amazon:enable-feature android:name="com.amazon.device.messaging" android:required="true" />
94 <service android:name="com.facebook.photos.upload.service.PhotosUploadRestartService" />
95 <activity android:theme="@style/Theme.Transparent" android:name="
com.facebook.photos.upload.dialog.UploadDialogsActivity" android:excludeFromRecents="true"
android:launchMode="singleTask" />
96 <service android:name="com.facebook.places.suggestions.SuggestProfilePicUploadService" />
97 <service android:name="com.facebook.common.errorreporting.memory.MemoryDumpUploadService" />
98 <activity android:name="
com.facebook.katana.push.c2dm.SystemTrayErrorNotificationPostActivity" android:
excludeFromRecents="true" />
99 <activity android:theme="@*android:style/Theme.Black.NoTitleBar.Fullscreen" android:name="
com.facebook.feed.platformads.DigitalGoodsVideoPlayer" android:configChanges="
orientation|screenSize" />
100 <activity android:theme="@style/Theme.FacebookDark" android:name="
com.facebook.auth.qrcodelogin.FacebookQRCodeLoginActivity" android:exported="false" android:
configChanges="keyboard|keyboardHidden|orientation|screenSize" android:windowSoftInputMode="
stateHidden|adjustResize" />
101 <activity android:theme="@style/Theme.FacebookDark" android:name="
com.facebook.katana.FacebookLoginActivity" android:permission="
com.facebook.permission.prod.FB_APP_COMMUNICATION" android:exported="false" android:
configChanges="keyboard|keyboardHidden|orientation|screenSize" android:windowSoftInputMode="
stateHidden|adjustResize" />
Line 15, Column 81
Spaces: 4
XML


06:00
Squirtle
CP 268


55% 12:22
Google
Rokura
Calea Floreasca
DOROBANȚI
Calea Floreasca
Calea Dorobanților
Parcul Sportiv Dinamo
Google


4:33
What to Watch
HOME
MUSIC
Recommended
Klaypex - Brains (feat. Kings High)
Klaypex
1 month ago · 117K views
4:31
SingSing Ember almost 1v5 Dota 2
NoobFromUA
1 month ago · 258K views
13:43
iPhone 6 VS iPhone 5S - Is It Worth The Upgrade?
EverythingApplePro
1 month ago · 220K views
11:16
DotaCinema
Dota 2 A-Z Dual Lane Challenge - Phantom...

Activities

- Sub-classes of **android.app.Activity**
- Presents a visual UI
- Each Activity has its own “window”
  - Only one “window” on screen at once
  - **Granular management** of the resources of an application
- UI layout – a “View”
  - Specified in a separate XML file
  - Constructed programmatically
  - Call setContentView() to display it
- Apps usually have several Activities
  - **Context**
    - A class representing the current application runtime environment
    - context.startSomething(some other component)
  - *(Application* can be used as a singleton but generally bad practice)

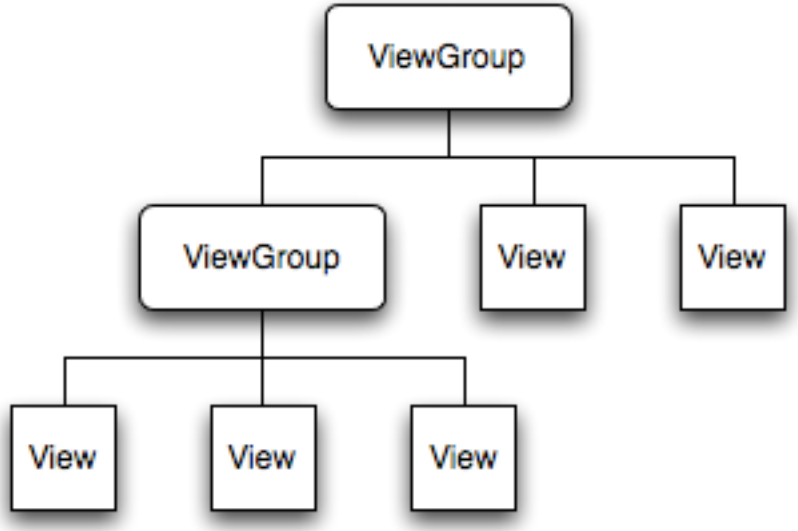
Android UI

- An Activity has a window associated with it
- Usually full screen
  - Can hover over another activity
  - Can be transparent
  - (Multi-window / picture-in-picture)
- Within the window there is a hierarchy of View objects
  - Set with setContentView()
  - Usually specified via an XML resource
    - /res/layout/mylayout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schema
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="example.com.helloworld.MainActivity">

    <LinearLayout
        android:layout_width="368dp"
        android:layout_height="495dp"
        android:layout_weight="1"
        android:orientation="vertical"
        tools:layout_editor_absoluteX="8dp"
        tools:layout_editor_absoluteY="8dp">

        <EditText
            android:id="@+id/editText1"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:ems="10"
            android:inputType="textPersonName"
            android:text="Name"
            tools:layout_editor_absoluteX="0dp"
            tools:layout_editor_absoluteY="0dp" />

        <EditText
            android:id="@+id/editText3"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
```

View Hierarchy

- Types of View subclasses
  - Those that display something
  - Those that do something (Widgets)
  - And ViewGroups which layout a collection of subviews
- Various layout types available — can specify in the XML resource



ViewGroups - Layouts

- FrameLayout
  - Simplest, contains a single object
- LinearLayout
  - Aligns all children in a single direction, based on the orientation attribute
- TableLayout
  - Positions children into rows and columns
- RelativeLayout
  - Lets the child views specify their position relative to the parent view or to each other
- ScrollView
  - A vertically scrolling view, like FrameLayout only contains a single element (e.g. a LinearLayout)
- (AbsoluteLayout)

Views - Widgets

- A child View that the user can (optionally) interact with
  - Button (a button)
  - EditText (text entry)
  - CalendarViewer (a calendar widget)
  - ImageView (displays an image)
  - …
- Handle appropriate UI events
  - In code, register setOnClickListener()
  - In XML layout, set android:onClick parameter
- Properties / parameters
  - Set via XML at build-time
  - Equivalent set / get methods for modifying at run-time

    android:text="@string/hello_world" />

    .getText(...), .setText(…)

Views - Parameters

- Parameters specify the details of particular Views
  - Width, height
    - Generally **not** in terms of absolute pixels
    - Relative to parent, percentages, offsets, margins
      - android:layout_width="match_parent”
      - android:layout_height="wrap_content"
  - ID
    - Used to generate a Java member variable we can refer to programmatically
      - android:id="@+id/my_button"
  - Methods
    - Used to automatically bind UI events to code
      - android:onClick=“myMethod”

Manipulating Views

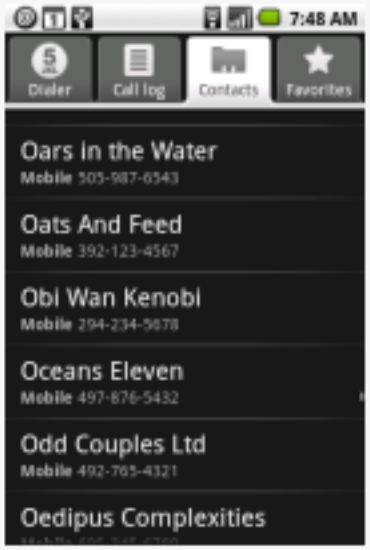
- We can alter the view hierarchy programmatically as the application runs
- ViewGroup provides methods
  - To add addView()
    - Need to either keep a reference to it or call setId() on the view so we can find it later
    - Or use references generated from layout XML for existing views
  - Or to remove removeView() children
    - Can add new buttons, layouts as required

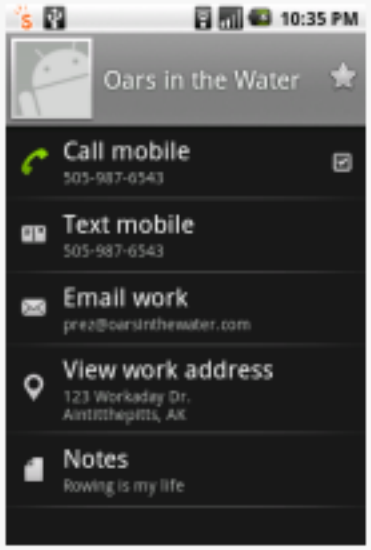
Let’s have a look…

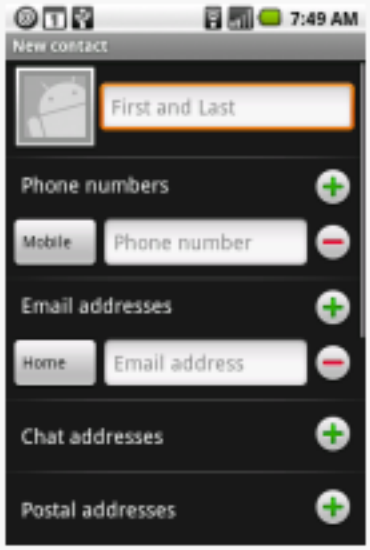
Tasks vs Activities

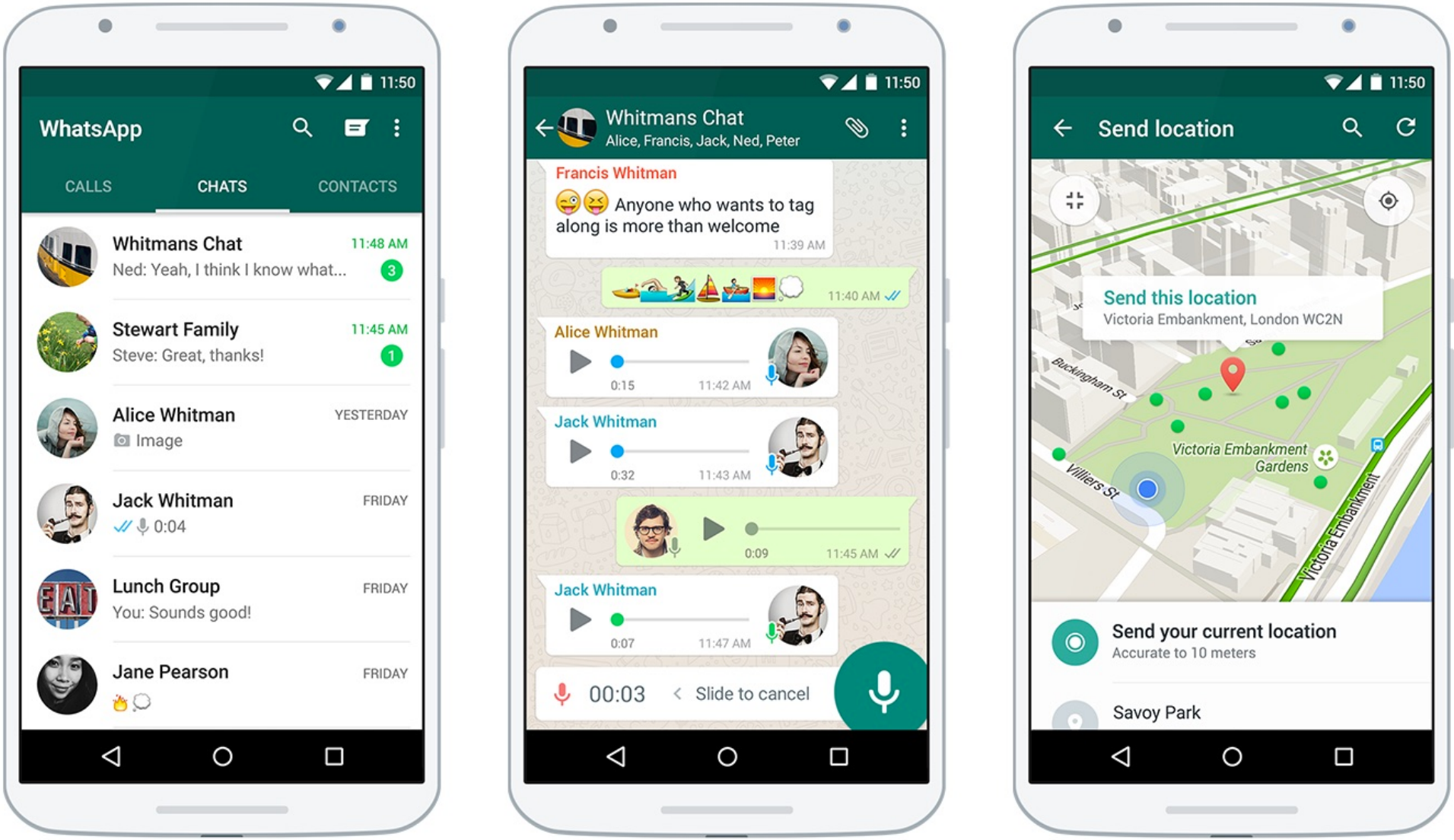
- Activities can start other activities
  - Forms a stack of Activities — current activity is on the top
    - The ”Back Stack”
- Multiple activities form a Task
  - Such as…?
  - A Task may span multiple applications
    - Make use of *standard* activities
    - Make use of *3rd party* activities
      - User choice
- An activity should be an **atomic** part of a particular task
  - Supports reuse, integration into multiple tasks
- Activities in a task move as a unit from foreground to background and vice versa
  - Switching task = switching stack

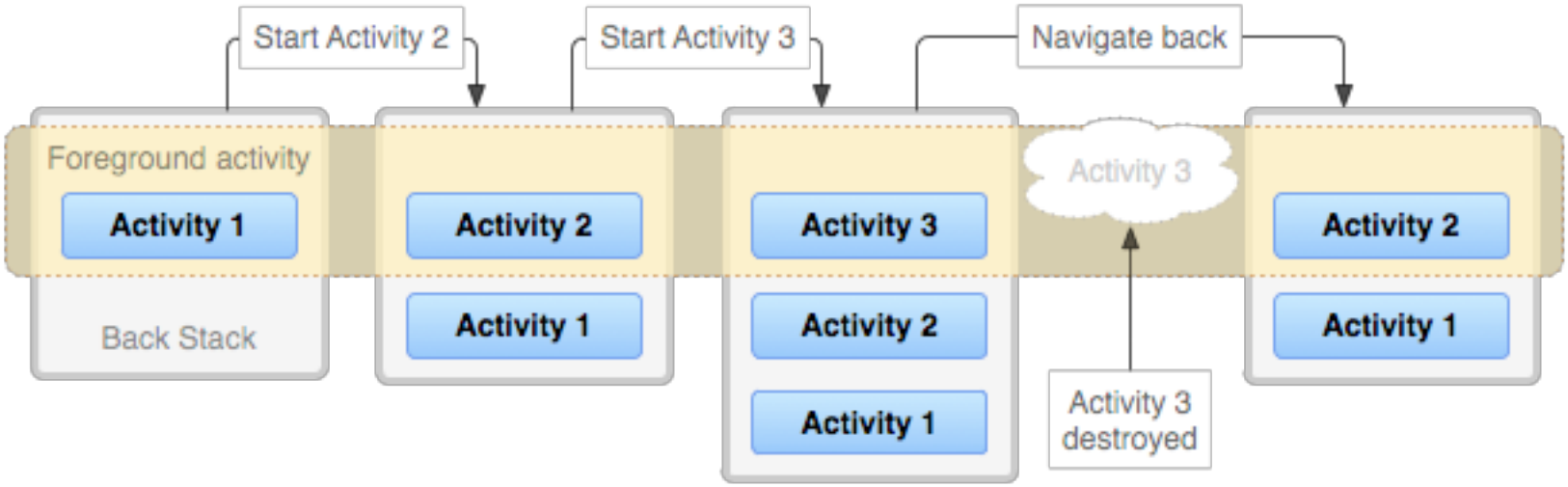

7:48 AM
Dialer
Call log
Contacts
Favorites
1
2 ABC
3 DEF
4 GHI
5 JKL
6 MNO
7 PQRS
8 TUV
9 WXYZ
*
0 +
#


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
Email work
prez@oarsinthewater.com
View work address
123 Workaday Dr.
Ainttheplits, AK
Notes
Rowing is my life

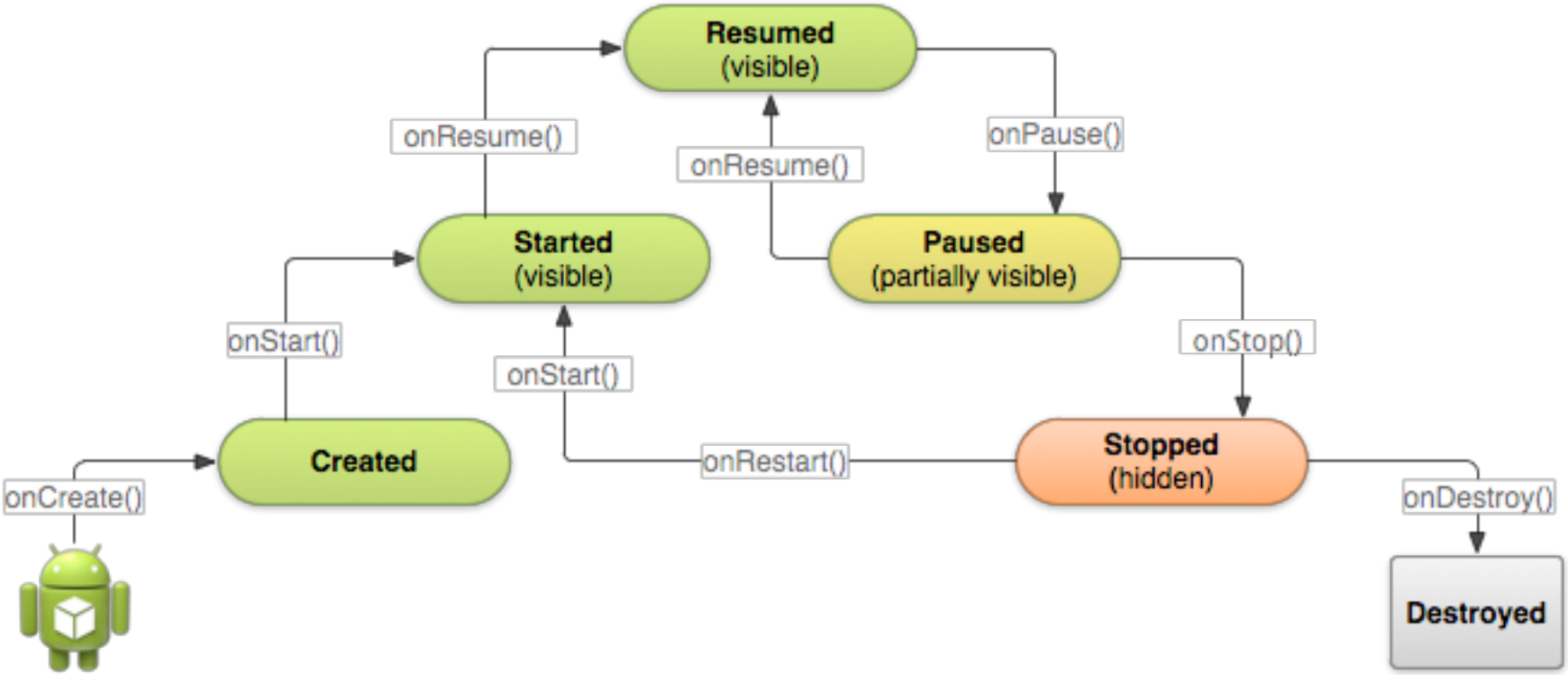

7:49 AM
New contact
First and Last
Phone numbers
Mobile
Phone number
Email addresses
Home
Email address
Chat addresses
Postal addresses


11:50
WhatsApp
CALLS
CHATS
CONTACTS
Whitmans Chat
11:48 AM
Ned: Yeah, I think I know what...
3
Stewart Family
11:45 AM
Steve: Great, thanks!
1
Alice Whitman
YESTERDAY
Image
Jack Whitman
FRIDAY
0:04
Lunch Group
FRIDAY
You: Sounds good!
Jane Pearson
FRIDAY
11:50
Whitmans Chat
Alice, Francis, Jack, Ned, Peter
Francis Whitman
Anyone who wants to tag along is more than welcome
11:39 AM
11:40 AM
Alice Whitman
0:15
11:42 AM
Jack Whitman
0:32
11:43 AM
0:09
11:45 AM
Jack Whitman
0:07
11:47 AM
00:03
Slide to cancel
11:50
Send location
Send this location
Victoria Embankment, London WC2N
Buckingham St
Victoria Embankment Gardens
Villiers St
Victoria Embankment
Send your current location
Accurate to 10 meters
Savoy Park


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

Activity Lifecycle

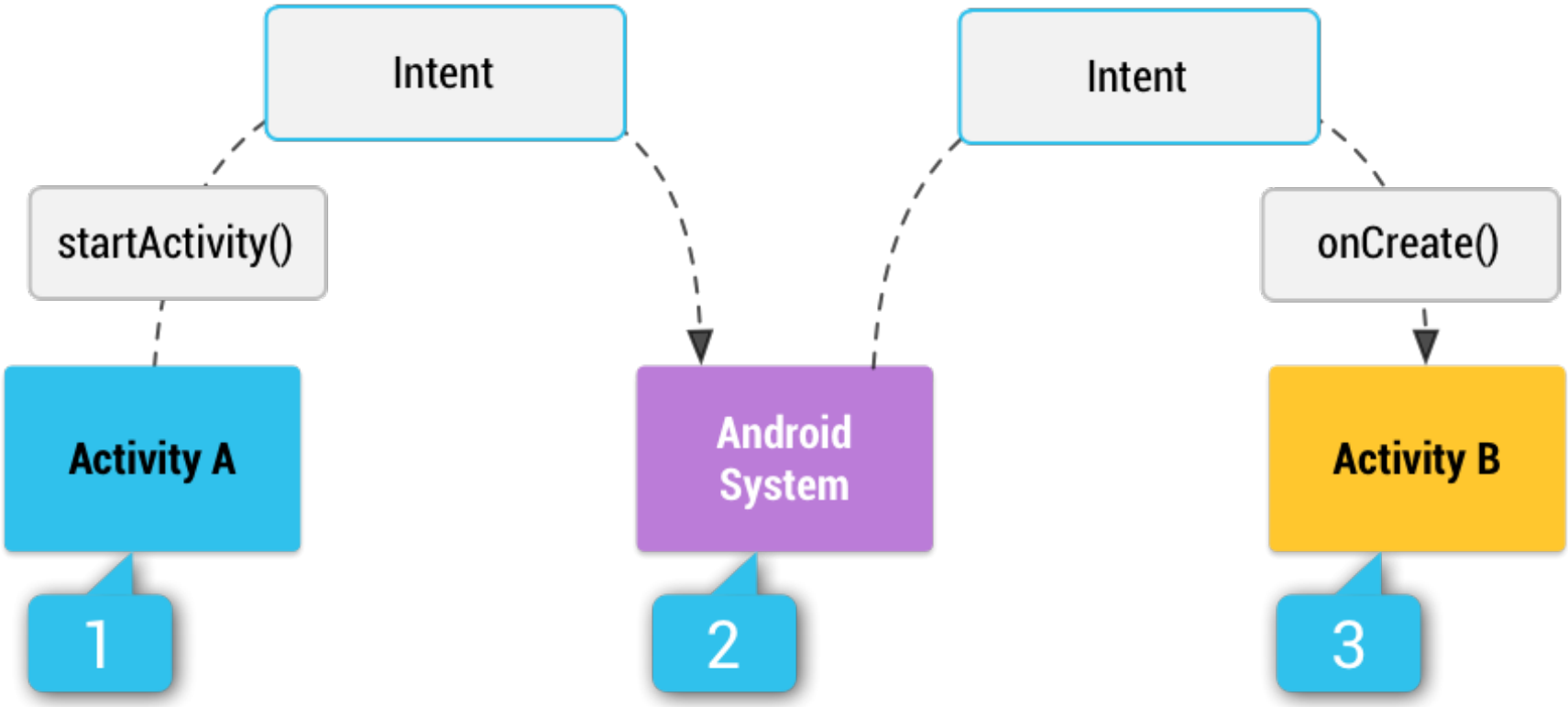
- Essentially three states
  - Active
    - in the foreground
  - Paused
    - still visible, but not top
      - Split screen
  - Stopped
    - obscured by another activity
- If paused or stopped, the system can drop the Activity from memory
  - Stopped activities are suspended in memory
    - Consume 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

Intents

- Don’t just instantiate the Activity sub-class
  - Android works by passing Intent objects around
    - Intent is used to describe an operation
      - Action and the data to operate on (as a URI)
    - Allows for late runtime binding
    - Glue multiple activities together
  - Android figures out how to honour the Intent
    - Inspecting registered **manifests**
- Starting an Activity
  - Create a new Intent object
  - Specify what you want to send it to
    - Either implicitly, or explicitly
  - Pass the Intent object to **startActivity()**
    - New Activity then started by the runtime
- Stopping an Activity
  - The called Activity can return to the original one by destroying itself
    - By calling the method **finish()**
  - Or when the user presses the back button

Starting an Activity



Intents

- Explicit Intent

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

- Implicit Intent
  - Data and action are used to resolve the most appropriate activity

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

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

startActivity(myIntent);
```

Intent Filters

- Manifest specifies *intent filters*
  - Determine which Intents should be handled by the Activity class
  - Match on action, schema of the URI
  - Multiple matches?
    - Allow the user to choose “Application”
    - This could be deliberate

```
<activity android:name="com.example.martinactivities.ForthActivity" >
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:scheme="http" />
    </intent-filter>
</activity>
```

Let’s have a look…

References

- http://developer.android.com/training/basics/activity-lifecycle/index.html
- http://developer.android.com/guide/topics/ui/declaring-layout.html
- http://developer.android.com/guide/topics/ui/controls.html