

# Android for Java Developers



Dr. Markus Schmall

Jochen Hiller



# Who we are



Dr. Markus Schmall  
[m.schmall@telekom.de](mailto:m.schmall@telekom.de)  
Deutsche Telekom AG



Jochen Hiller  
[j.hiller@telekom.de](mailto:j.hiller@telekom.de)  
Deutsche Telekom AG



# Agenda

- Introduction to Android
- Android Security model
- Sample Application / Concepts
- Demo

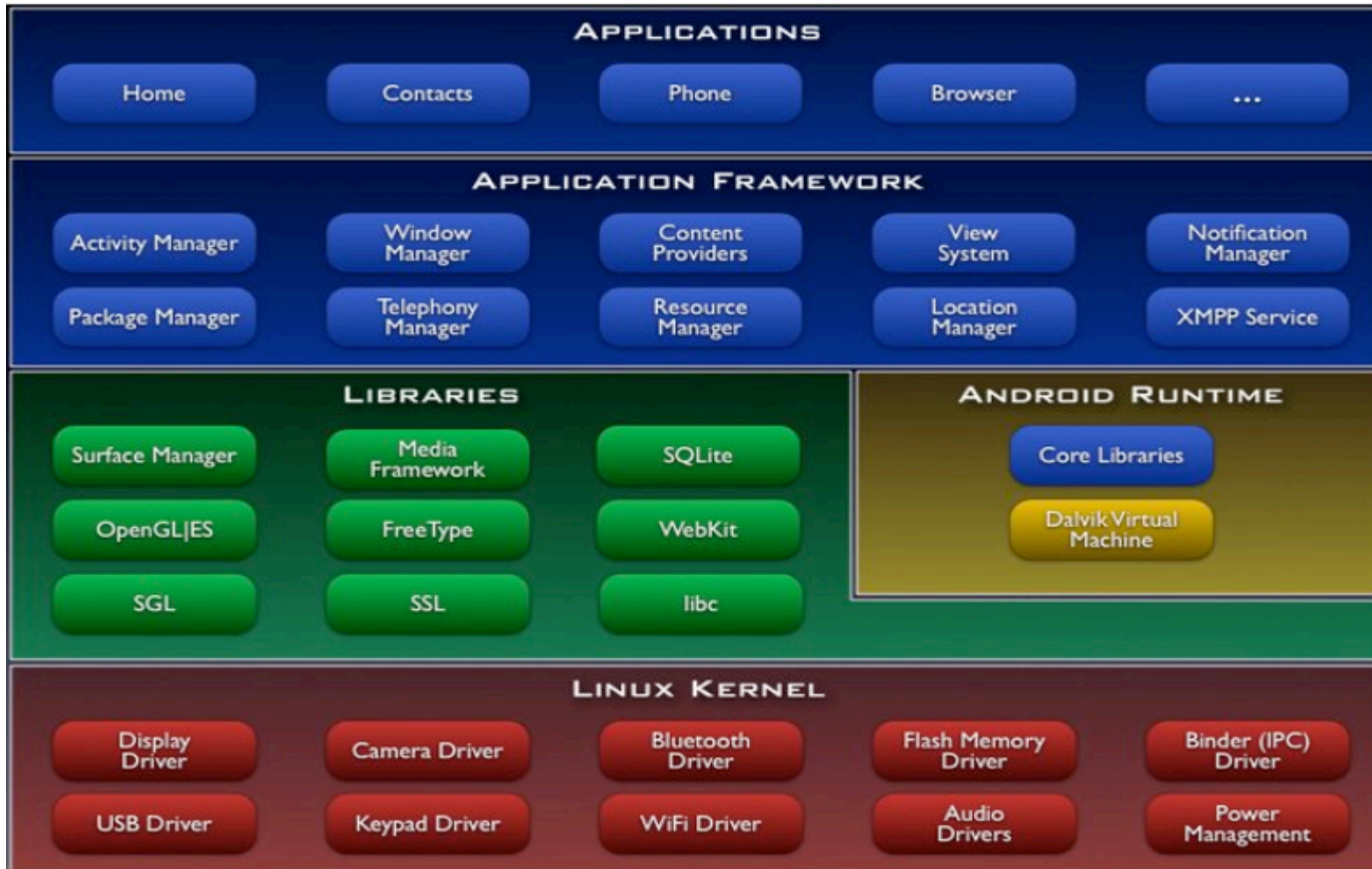


# What is Android?

- The first *complete*, *open*, and *free* mobile platform
  - Initiated by the Open Handset Alliance [1] project
  - Main development done by Google
  - Pushed through
    - Early available Developer SDK
    - Easy startup: no registration, no hardware, software for free
    - 10 Mio\$ Android Developer Challenge
- July 2008: no Telco offers out-of-the-box Android mobile devices
- First devices planned for end of year 2008
- Competitive platforms
  - Windows Mobile, Symbian OS (will be OpenSource too, [2]), iPhone



# The Android architecture



# The Android architecture

- based on Linux 2.6.23 system \*)
- Mobile Hardware support (GSM, WiFi, GPS, Camera, Bluetooth, USB, ...)
- Integrated Browser (WebKit Engine)
- Graphics (OpenGL/ES), DB, Media, ... support
- Custom Java environment: Dalvik VM
  - Bytecode is NOT compatible to the standard Java VM
  - derived from Java 1.5 \*)
  - Dalvik VM offers optimized functionality for mobile devices, small footprint, no swap, no JIT (at least for v1.0 time frame)
- Java based Application Framework
- Apache 2.0 License

\*) as of Android SDK m5-rc15



# The Android Class Library

- Bundled in `android.jar`
- Android platform library: `android.*`
  - XML Parser implementations included (DOM, SAX, XMLPullParser)
- Apache Harmony (Standard Java Library implementation)
- Several popular OpenSource projects available out-of-the-box
  - Apache Commons (HttpClient 3/4, Codec)
  - BouncyCastle JCE provider
  - SQLite, JUnit 3.8.x
- GData APIs partly included as wireless version
  - `com.google.wireless.gdata.*`
- GTalk (XMPP-Client) supported



# Agenda

- Introduction to Android
- Android Security model
- Sample Application / Concepts
- Demo





# Android Security Concepts

Relevant aspects:

- File system Security
  - Linux based permissions
- Encapsulation of single installed applications
  - own rights on a per application basis (approved by the user)
  - every application has its own user id
  - applications can share files and databases
- Security of installed applications
  - restricted access to file system
  - restricted access to cache, manifest files, databases



# Android Security Concepts

- Example AndroidManifest.xml file:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="org.ms.android.ASipGrab" android:sharedUserId="flake">
    <application android:icon="@drawable/icon">
        <activity android:name=".ASipGrab"
            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>
    <uses-permission android:name="android.permission.WRITE_SETTINGS">
</manifest>
```

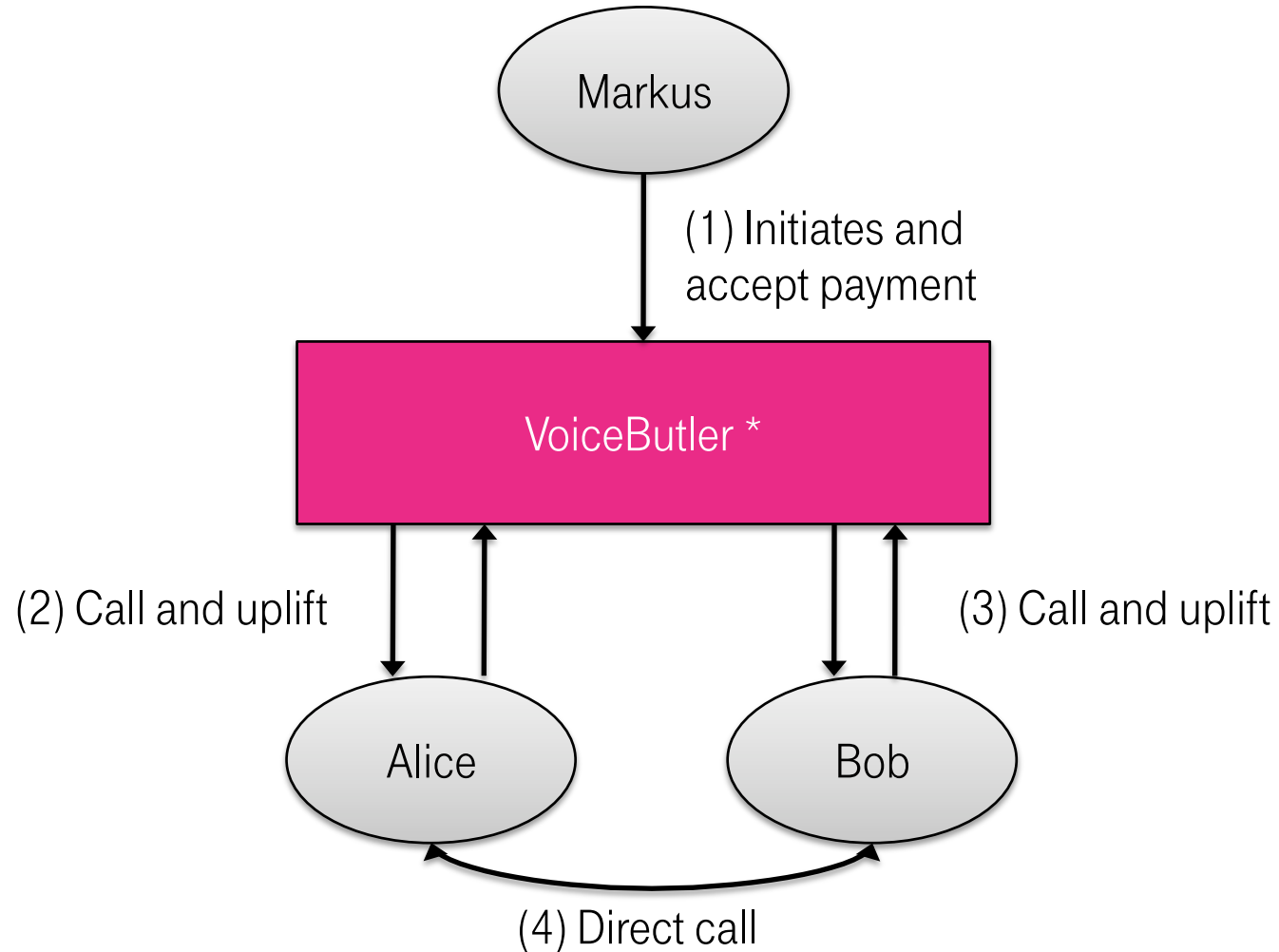


# Agenda

- Introduction to Android
- Android Security model
- Sample Application / Concepts
- Demo



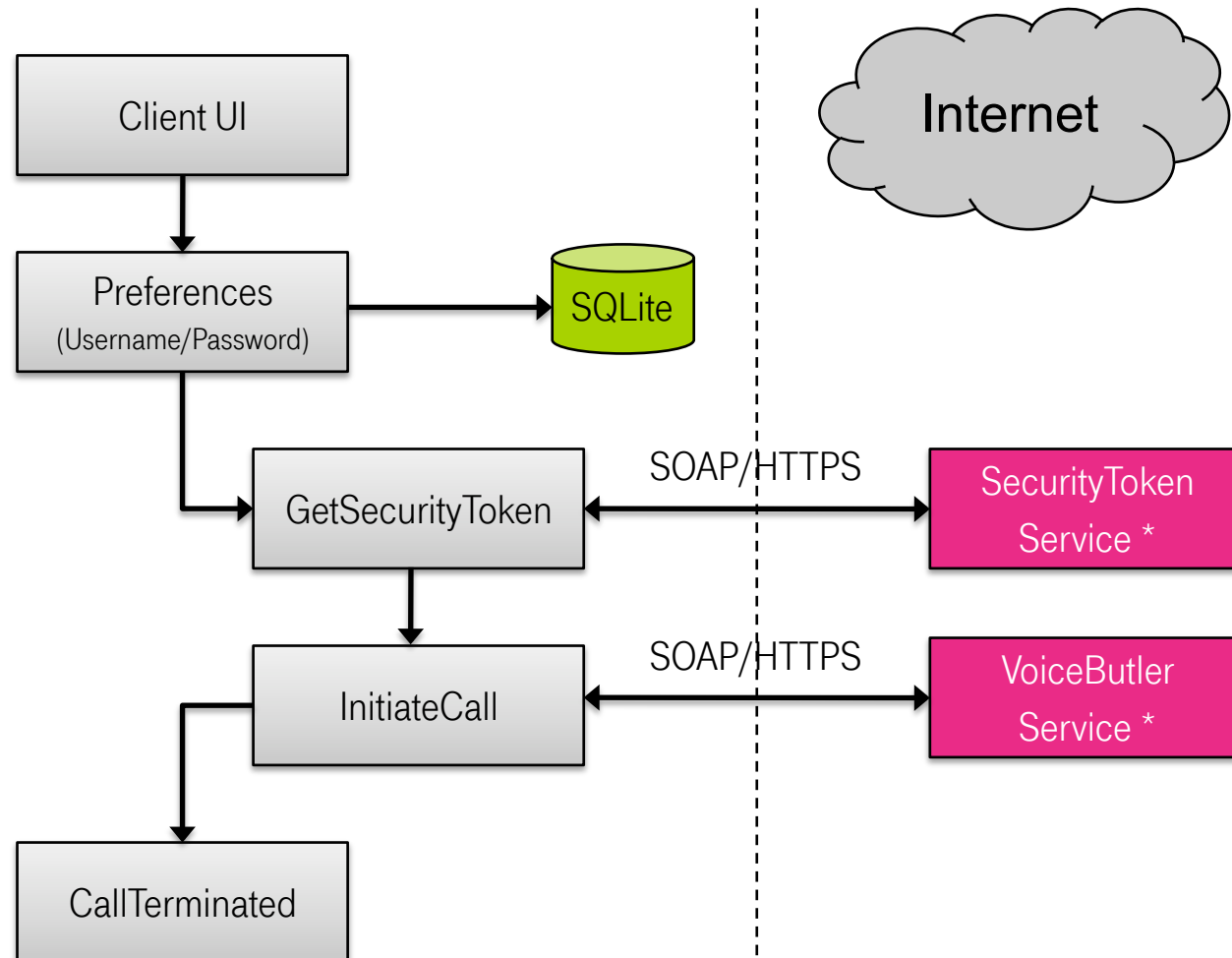
# Sample application – VoiceButler



\* based on services from <http://developer.telekom.com>



# VoiceButler – architecture

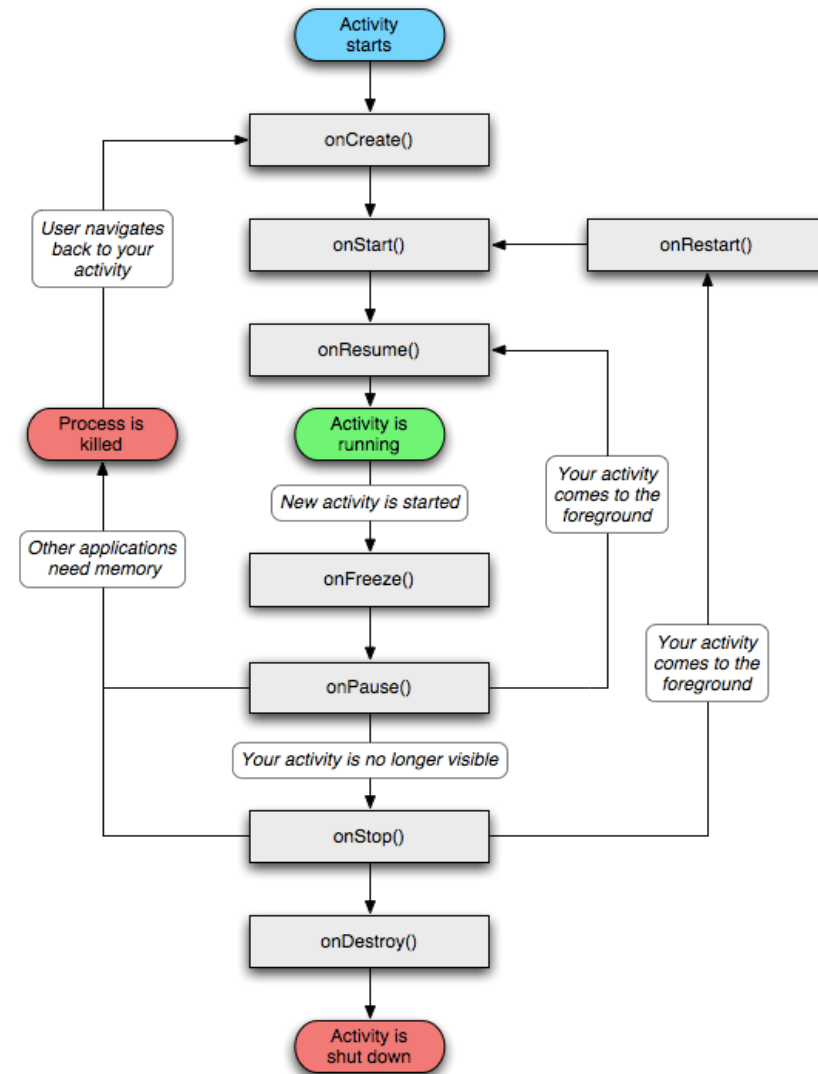


\* based on services from <http://developer.telekom.com>



# Concept: *Activity*

- Activities are basic objects, describing things a user can see or do
- Activities can be swapped out (killed)
  - Has to care about its state, when re-created
- Activities can subscribe for Intents
- Activities are loosely coupled



# Concept: *Activity* – Sample

```
package helloandroid;

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

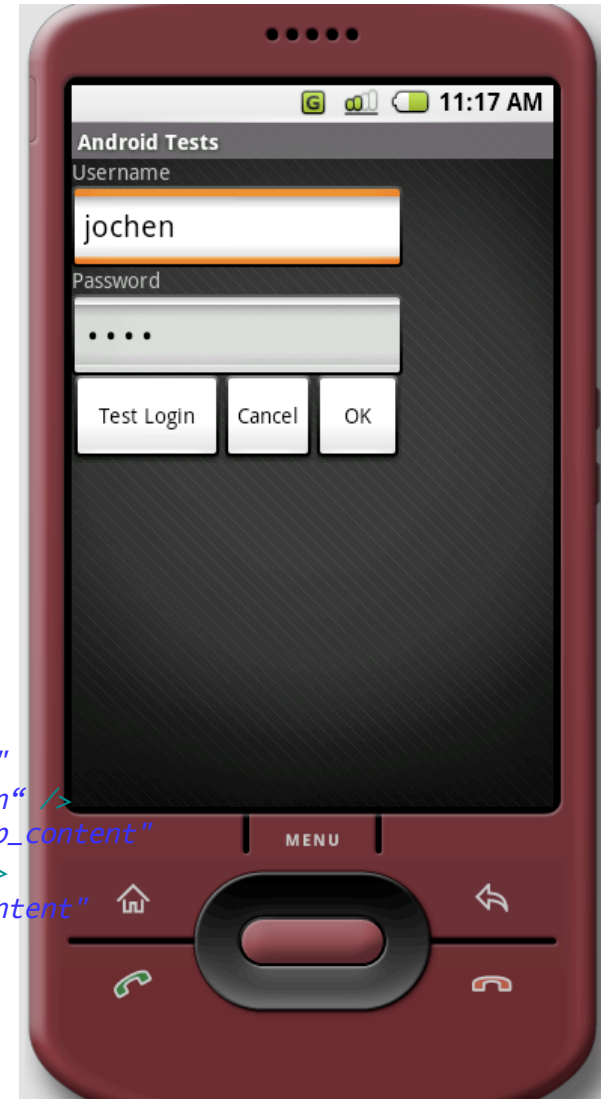
public class HelloAndroid extends Activity {

    public void onCreate(Bundle icle) {
        super.onCreate(icle);
        TextView view = new TextView(this);
        view.setText("Hello Android\n");
        setContentView(view);
    }
}
```



# Concept: *Layout / Declarative UIs*

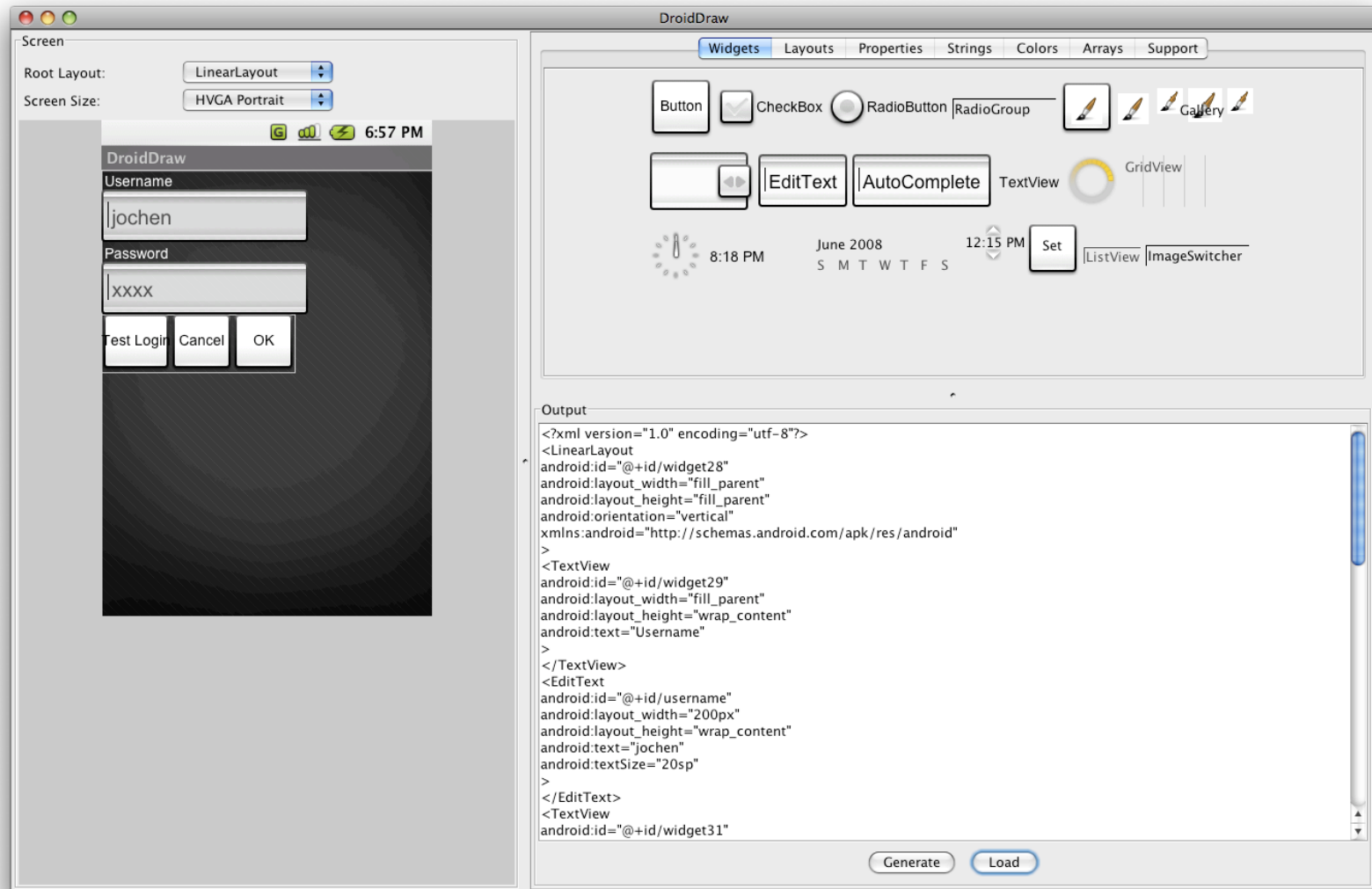
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout android:layout_width="fill_parent"
  android:layout_height="fill_parent" android:orientation="vertical"
  xmlns:android="http://schemas.android.com/apk/res/android">
  <TextView android:layout_width="fill_parent"
    android:layout_height="wrap_content" android:text="Username" />
  <EditText android:id="@+id/username" android:layout_width="220px"
    android:layout_height="wrap_content" android:text="jochen"
    android:textSize="20sp" />
  <TextView android:layout_width="fill_parent"
    android:layout_height="wrap_content" android:text="Password" />
  <EditText android:id="@+id/password" android:layout_width="220px"
    android:layout_height="wrap_content" android:text="xxxx"
    android:textSize="20sp" android:password="true" />
  <LinearLayout android:layout_width="wrap_content"
    android:layout_height="wrap_content">
    <Button android:id="@+id/buttonTest" android:layout_width="100px"
      android:layout_height="wrap_content" android:text="Test Login" />
    <Button android:id="@+id/buttonCancel" android:layout_width="wrap_content"
      android:layout_height="wrap_content" android:text="Cancel" />
    <Button android:id="@+id/buttonOK" android:layout_width="wrap_content"
      android:layout_height="wrap_content" android:text="OK" />
  </LinearLayout>
</LinearLayout>
</LinearLayout>
```





# DroidDraw: Interactive UI designer

- See [3]



# Android and Webservice Toolkits

- Standard WS-toolkits (e.g. AXIS) does not work out-of-the-box.
  - `java.rmi.*` is missing. See: Java vs. Android APIs [4]
- Why?
  - RMI is based on Serialization ! Dalvik VM Serialiation != Java VM Serialization !
  - Android provides its own (shared memory based) IPC: Binder (IPC), Android Interface Definition Language (AIDL)
- Google states [5]:

For Android there is no official policy or APIs about using web services. However, it is better to stick with the light weight options, one of which would be to use REST+JSON.

Android includes the Apache HttpClient, using which you can easily implement client-server communication.
- Options:
  - Use HttpClient and XML parser
  - Use lightweight SOAP toolkit (kSOAP2, see [6], JavaME based SOAP toolkit):
  - Downsize an existing WS-toolkit for Doc/Lit encoding only



# Concept: *HttpClient* and *XML Parser*

```
// invoke a SOAP call
String request = "<soapenv:Envelope ...";
HttpClient c = new HttpClient();
PostMethod m = new PostMethod("https://../SAMClientTokenProvider");
try {
    m.setRequestHeader("SOAPAction", "http://../getTokenForPwd");
    m.setRequestHeader("Content-Type", "text/xml; charset=utf-8");
    m.setRequestHeader("Accept", "application/soap+xml, text/*");
    m.setRequestEntity(new StringRequestEntity(request));
    c.executeMethod(m);
    InputStream is = m.getResponseBodyAsStream();
    DocumentBuilder parser = DocumentBuilderFactory.newInstance()
        .newDocumentBuilder();
    Document xmlDoc = parser.parse(is);
    // get results from xmlDoc
} finally {
    m.releaseConnection();
}
```



# Android and HTTPS / Digest Authentication

There are some problems with HTTPS:

- SSL not working: <http://code.google.com/p/android/issues/detail?id=50>
- HttpClient does not work with HTTPS:  
<http://groups.google.com/group/android-developers/browsethread/thread/4e92297be5b6adcd>
- We identified problems with HttpClient, HTTPS and Digest Authentication
- HTTPS seems to be rather slow on emulator
  - e.g. >3 sec overhead for first SSL connection
- MD2 hash algorithm missing in Apache Harmony



# Property files vs. Android Resources

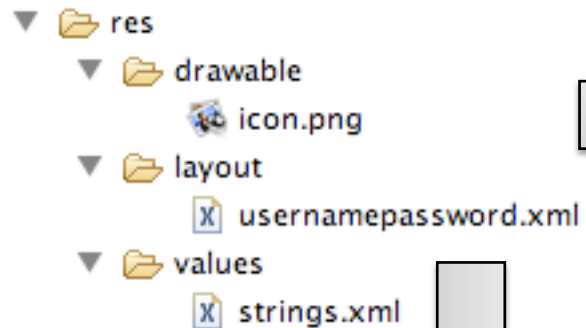
```
// try to get resources, does return null
URL url = this.getClassLoader().getResource("my.properties");
InputStream is = this.getClassLoader().
    getResourceAsStream("/root.properties");
```

- Does NOT work ! Resources can NOT be loaded via `ClassLoader` !
- Therefore: 3rdParty libraries requires often adaption to Android
- Android supports an optimized resource handling concept:
  - Resources will be referred through numeric constants → reduce Strings
  - `R.java`: Symbolic names of all resources → Refactoring !
  - `R.java` will be maintained by tooling → Zero effort
  - Resources can be referred via symbolic names in views
  - Optimized packaging of resources
  - Resources will be resolved device specific and supporting I18N



# Concept: *Resources*

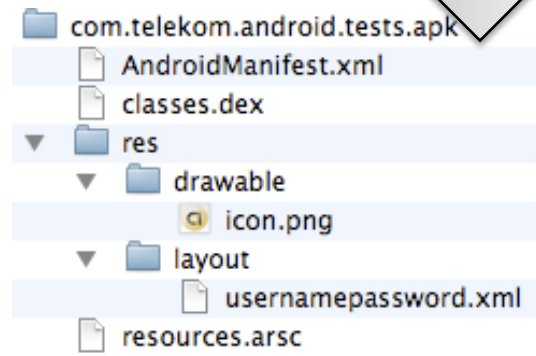
/res Folder



ADT generates

aapt

Android Package



Resolved at runtime

R.java

```
R.java
/* AUTO-GENERATED FILE. DO NOT MODIFY.
 *
 * This class was automatically generated by the
 * aapt tool from the resource data it found. It
 * should not be modified by hand.
 */
package com.telekom.android.tests;

public final class R {
    public static final class attr {
    }
    public static final class drawable {
        public static final int icon=0x7f020000;
    }
    public static final class id {
        public static final int buttonCancel=0x7f050003;
        public static final int buttonOK=0x7f050004;
        public static final int buttonTest=0x7f050002;
        public static final int password=0x7f050001;
        public static final int username=0x7f050000;
    }
    public static final class layout {
        public static final int usernamepassword=0x7f030000;
    }
    public static final class string {
        public static final int app_name=0x7f040000;
    }
}
```

Refer constants

Code referring to resources

```
public void onCreate(Bundle icle) {
    super.onCreate(icle);
    setContentView(R.layout.usernamepassword);
}
```



# Concept: *Responsiveness UIs*

- Handlers are offered as solution for handling messages between threads.
- See [7]: Responsive applications

```
Handler handler = new Handler() {  
    /** Gets called on every message that is received */  
    public void handleMessage(Message msg) {  
        switch (msg.what) {  
            case 0xdeadbeef: {  
                // do some thing, e.g. update UI  
                break;  
            }  
        }  
        super.handleMessage(msg);  
    }  
}
```

```
// create message from child  
// thread  
Message m = new Message();  
m.what = 0xdeadbeef;  
X.this.handler.sendMessage(m);
```



# Agenda

- Introduction to Android
- Android Security model
- Sample Application / Concepts
- Demo





# Demo – VoiceButler

- Give me one number !
- Features:
  - Calling webservices using SOAP
  - SQLite
  - Responsive UIs



# When will be Android devices available?

- Devices shown in Barcelona, Google I/O [7]
  - See: Android Dream has built-in compass for Google StreetView, <http://www.youtube.com/watch?v=4PRfVKzuUJ4>
- Expected HW vendors: Motorola, Samsung, HTC
- Yet planned for end of 2008



Thank you very much for your attention!

Any questions ?



# References

- [1] Open Handset Alliance: <http://www.openhandsetalliance.com/>
- [2] Nokia kauft Symbian: <http://www.heise.de/newsticker/Nokia-kauft-Symbian-Update-/meldung/109873>
- [3] DroidDraw, UI Designer: <http://www.droiddraw.org/>
- [4] Java vs. Android APIs: <http://blogs.zdnet.com/Burnette/?p=504>
- [5] Web Service Deployment: [http://groups.google.com/group/android-developers/browse\\_thread/thread/2b073fb5a263c717/](http://groups.google.com/group/android-developers/browse_thread/thread/2b073fb5a263c717/)
- [6] Calling SOAP 1.1. Web Services from Android (ksoap2 + apache http):  
[http://groups.google.com/group/android-developers/browse\\_thread/thread/9b246db7d612a6b5](http://groups.google.com/group/android-developers/browse_thread/thread/9b246db7d612a6b5)
- [7] Developing Responsive Applications: <http://code.google.com/android/toolbox/responsiveness.html>
- [8] Access the web behind a proxy:  
[http://groups.google.com/group/android-developers/browse\\_thread/thread/e855c4998b25fc9c/](http://groups.google.com/group/android-developers/browse_thread/thread/e855c4998b25fc9c/)
- [9] Android Development Community: <http://www.anddev.org/index.php>
- [10] Google I/O 2008 - Dalvik Virtual Machine Internals: <http://www.youtube.com/watch?v=ptjedOZEXPM>
- [11] DEX File Format: <http://www.retrodev.com/android/dexformat.html>

