Praktikum Entwicklung Mediensysteme

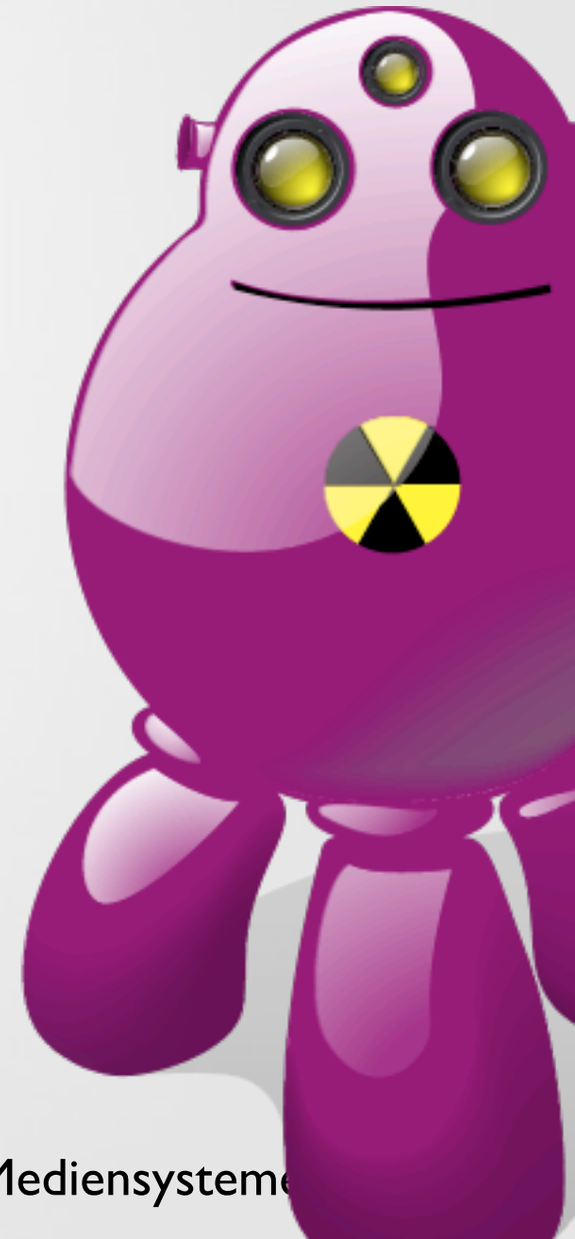
An Introduction to Anroid

Medien Informatik München

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

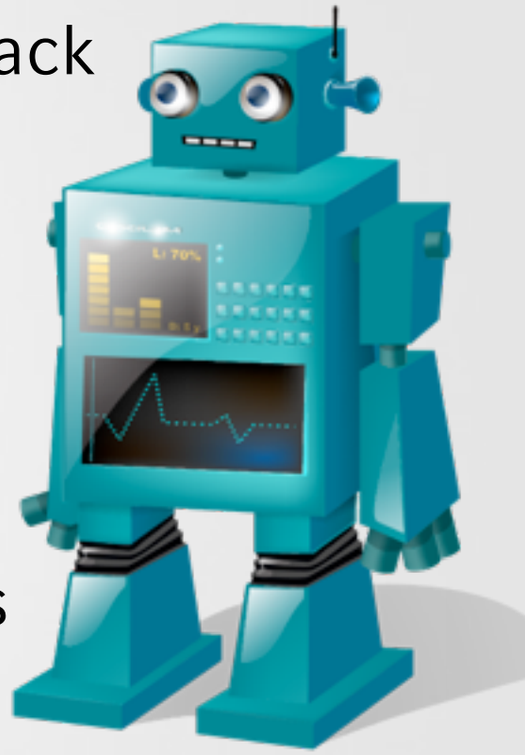
An Introduction to Android

- What is Android?
- Installation
- Getting Started
- Anatomy of an Android Application
- Life Cycle of an Android Application



What is Android?

- Released in Nov. 2007 – rumored to be some kind of GPhone
- Open, free mobile platform with a complete software stack
  - Operating system
  - Middleware
  - Key mobile applications
- Developed by the Open Handset Alliance
- Built on the open Linux kernel
- Custom Dalvik virtual machine for mobile environments
- Applications written in Java
- Open source; Apache v2 open source license
- Applications can access all core functionalities of a mobile device
- (Nearly) no differentiation between core and 3rd party applications
- Can be extended to incorporate new technologies



Open Handset Alliance

- Group of more than 30 technology and mobile companies led by Google
  - Mobile Operators, e.g. China Mobile, KDDI, NTT DoCoMo, TMobile,
  - Sprint Nextelk, Telefonica
  - Semiconductor Companies, e.g. Broadcom, Intel, Nvidia, Qualcomm, SiRF, Texas Instruments
  - Handset Manufactureres, e.g. HTC, LG, Motorola, Samsung
  - Software Companies, e.g. eBay, Google,
- Goal: „to accelerate innovation in mobile and offer consumers a richer, less expensive, and better mobile experience “
- Android as the first project towards an open and free mobile experience, but also commercial deployment
- URL: www.openhandsetalliance.com

Open Handset Alliance

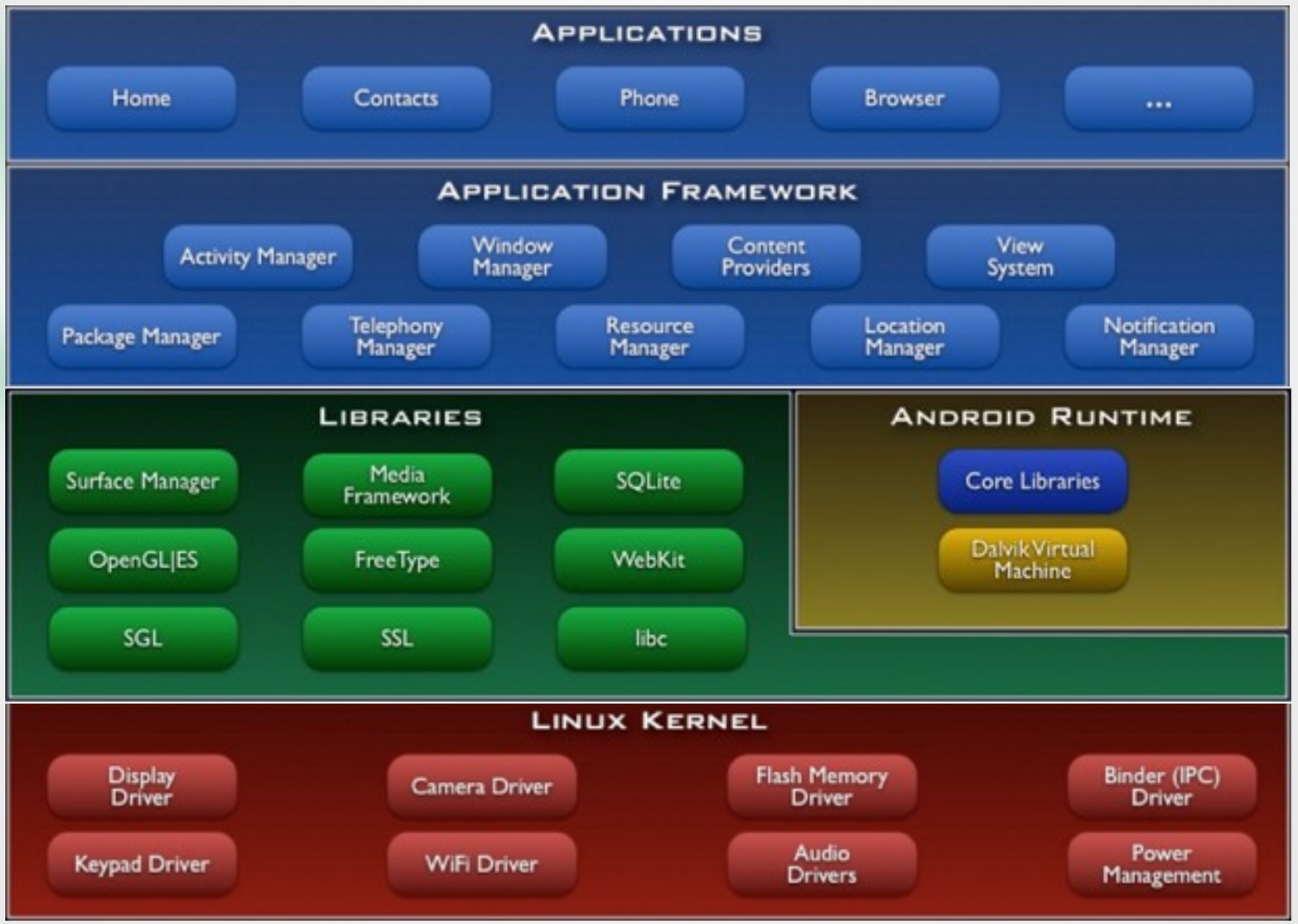
- Group of more than 30 technology and mobile companies led by Google
  - Mobile Operators, e.g. China Mobile, KDDI, NTT DoCoMo, TMobile,
  - Sprint Nextelk, Telefonica
  - Semiconductor Companies, e.g. Broadcom, Intel, Nvidia, Qualcomm, SiRF, Texas Instruments
  - Handset Manufactureres, e.g. HTC, LG, Motorola, Samsung
  - Software Companies, e.g. eBay, Google,
- Goal: „to accelerate innovation in mobile and offer consumers a richer, less expensive, and better mobile experience “
- Android as the first project towards an open and free mobile experience, but also commercial deployment
- URL: www.openhandsetalliance.com

Android Features

- **Application framework** enabling reuse and replacement of components
- **Dalvik virtual machine** optimized for mobile devices (register based)
- **Integrated browser** based on the open source WebKit engine
- **Optimized graphics** powered by a custom 2D graphics library; 3D graphics based on the OpenGL ES specification (hardware acceleration optional)
- **SQLite** for structured data storage
- **Media support** for common audio, video, and still image formats (MPEG4, H. 264, MP3, AAC, AMR, JPG, PNG, GIF)
- **GSM Telephony** (hardware dependent)
- **Bluetooth, EDGE, 3G, WiFi, NFC** (hardware dependent)
- **Camera, GPS, compass, accelerometer, gyroscope** (hardware dependent)
- **Rich development environment** including a device emulator, tools for debugging, memory and performance profiling, and a plugin for the Eclipse IDE

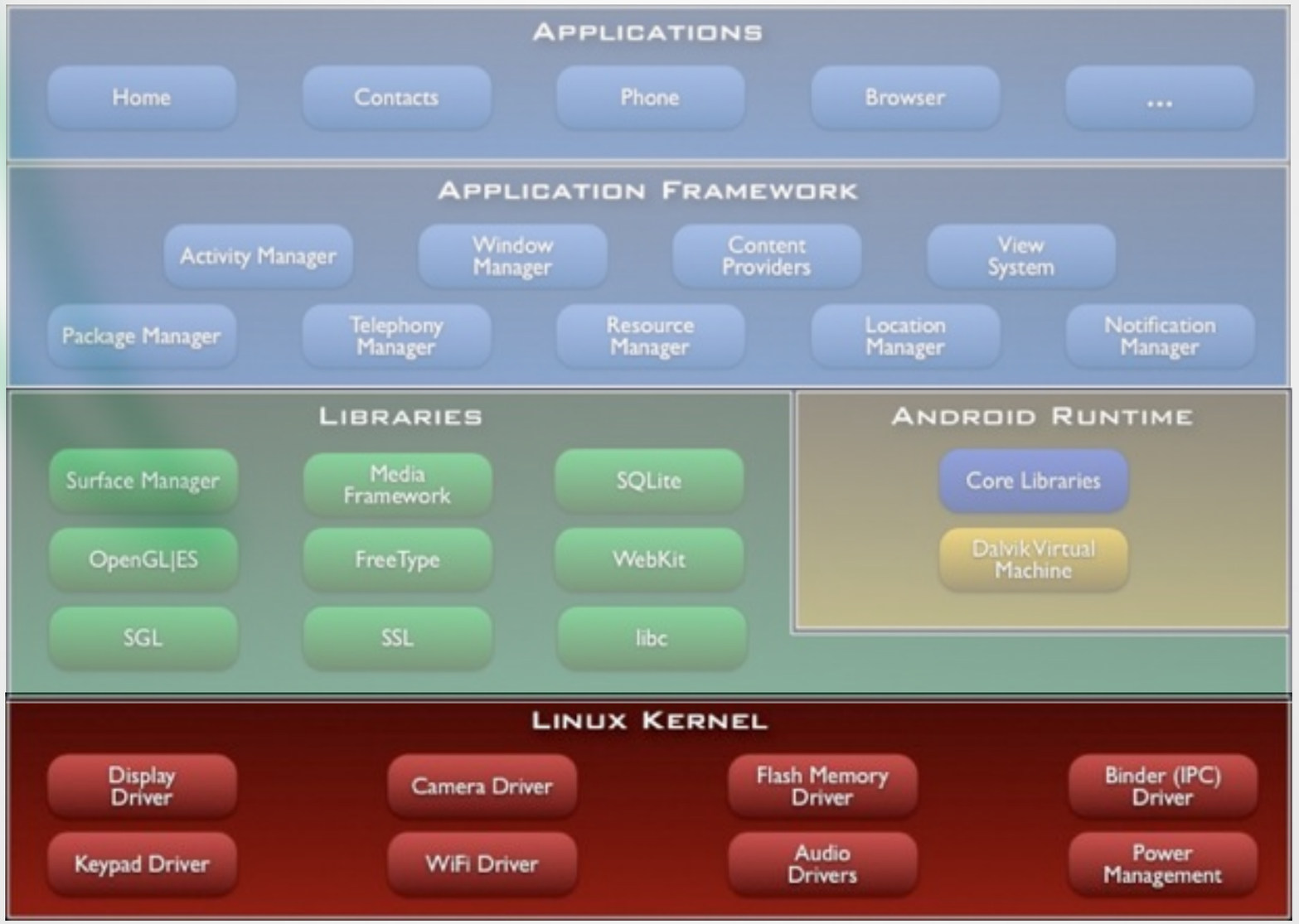
Source: http://code.google.com/android/index.html

Android Architecture



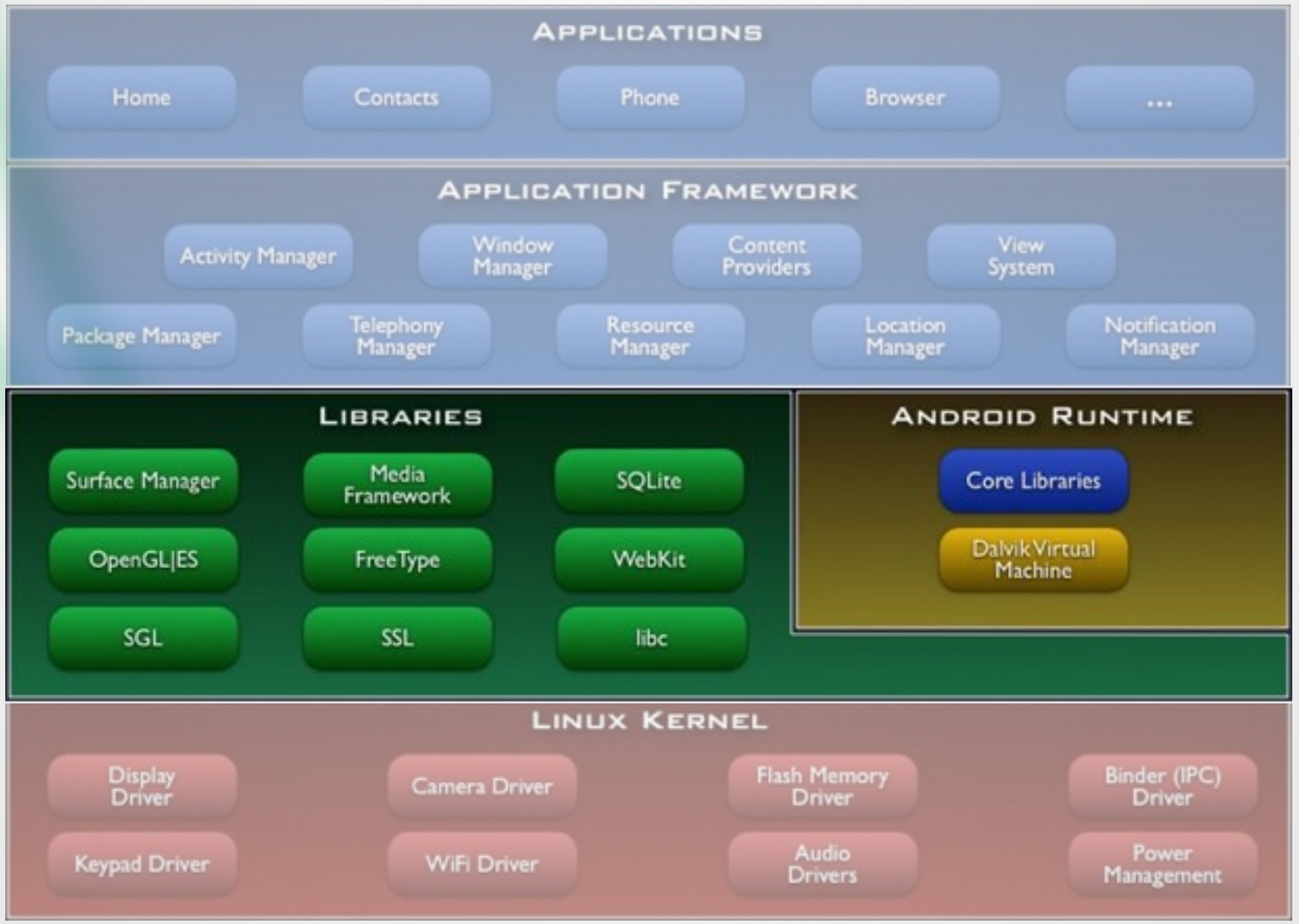
Source: http://code.google.com/android/index.html

Android Architecture



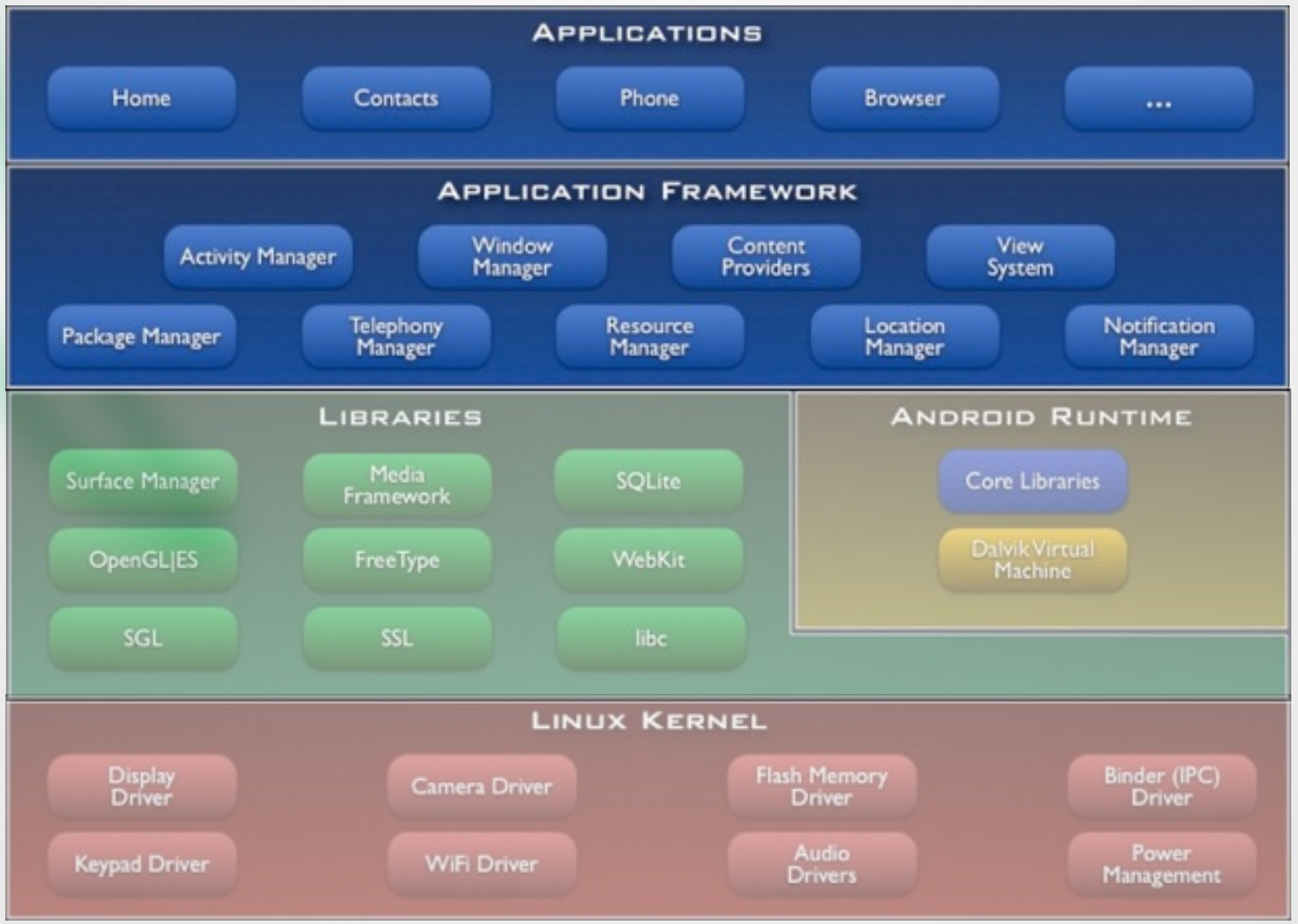
Source: http://code.google.com/android/index.html

Android Architecture



Source: http://code.google.com/android/index.html

Android Architecture



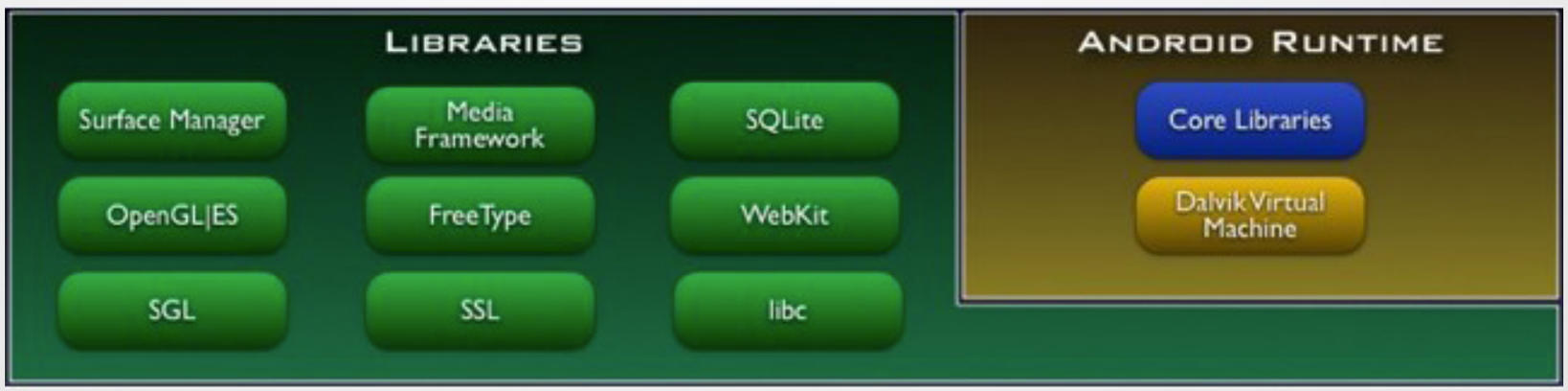
Source: http://code.google.com/android/index.html

Linux Kernel

- Linux kernel version 2.6
- Abstraction layer between hardware and the software stack
- Core services
  - **Security**
  - Memory management
  - Process management
  - Network stack
  - Driver model

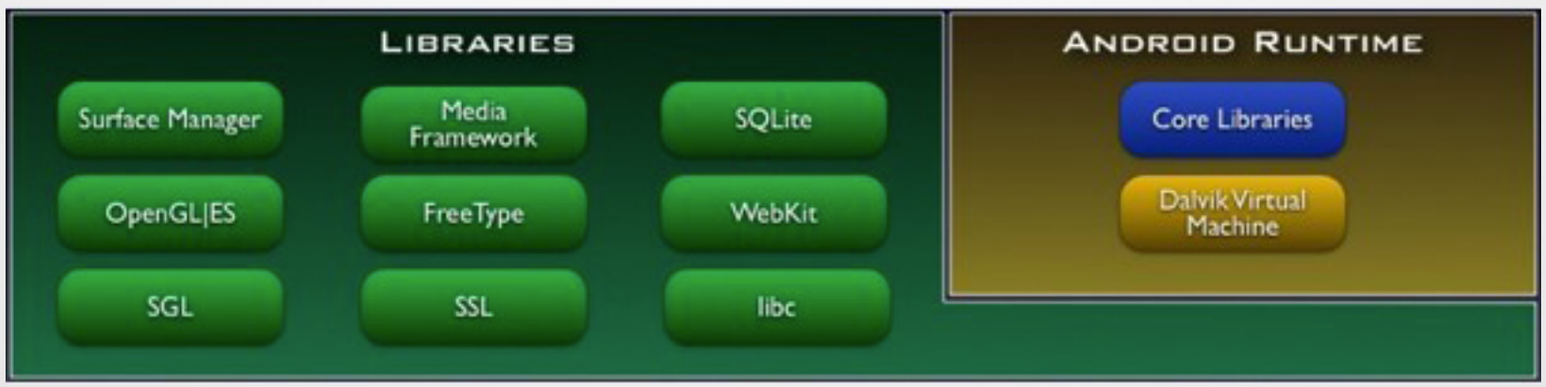
Libraries

- C/C++ libraries used by various Android components
- Developers can use their capabilities through the application framework
- Includes:
  - Media Libraries: includes MPEG4, H.264, MP3, JPG, PNG,
  - WebKit/LibWebCore: web browser engine
  - SQLite: relational database engine
  - Libraries/engines for 2D and 3D graphics



Android Runtime

- Core libraries provide Java functionalities
- Dalvik virtual machine relies on Linux kernel for e.g. threading or low-level memory management
- Devices can run multiple Dalvik VMs, every Android application runs with its own instance of Dalvik VM
- VM executes optimized Dalvik Executable files (.dex)
- Dx-tool transforms compiled Java-files into dex-files



Applications /Application

- Core applications, e.g. contacts, mail, phone, browser, calender, maps, ...
- Full access to all framework APIs for core applications
- Simplified reuse of components
- Applications written in Java

Core Android Packages

- android.util
  - contains various low-level utility classes, such as specialized container classes, XML utilities, etc.
- android.os
  - provides basic operating system services, message passing, and inter-process communication.
- android.graphics
  - is the core rendering package.
- android.text, android.text.method, android.text.style, and android.text.util
  - supply a rich set of text processing tools, supporting rich text, input methods, etc.
- android.database
  - contains low-level APIs for working with databases.
- android.content
  - provides various services for accessing data on the device: applications installed on the device and their associated resources, and content providers for persistent dynamic data.
- android.view
  - is the core user-interface framework.
- android.widget
  - supplies standard user interface elements (lists, buttons, layout managers, etc) built from the view package.
- android.app
  - provides the high-level application model, implemented using Activities.

The Tasteful Android Version History

2008
1.0

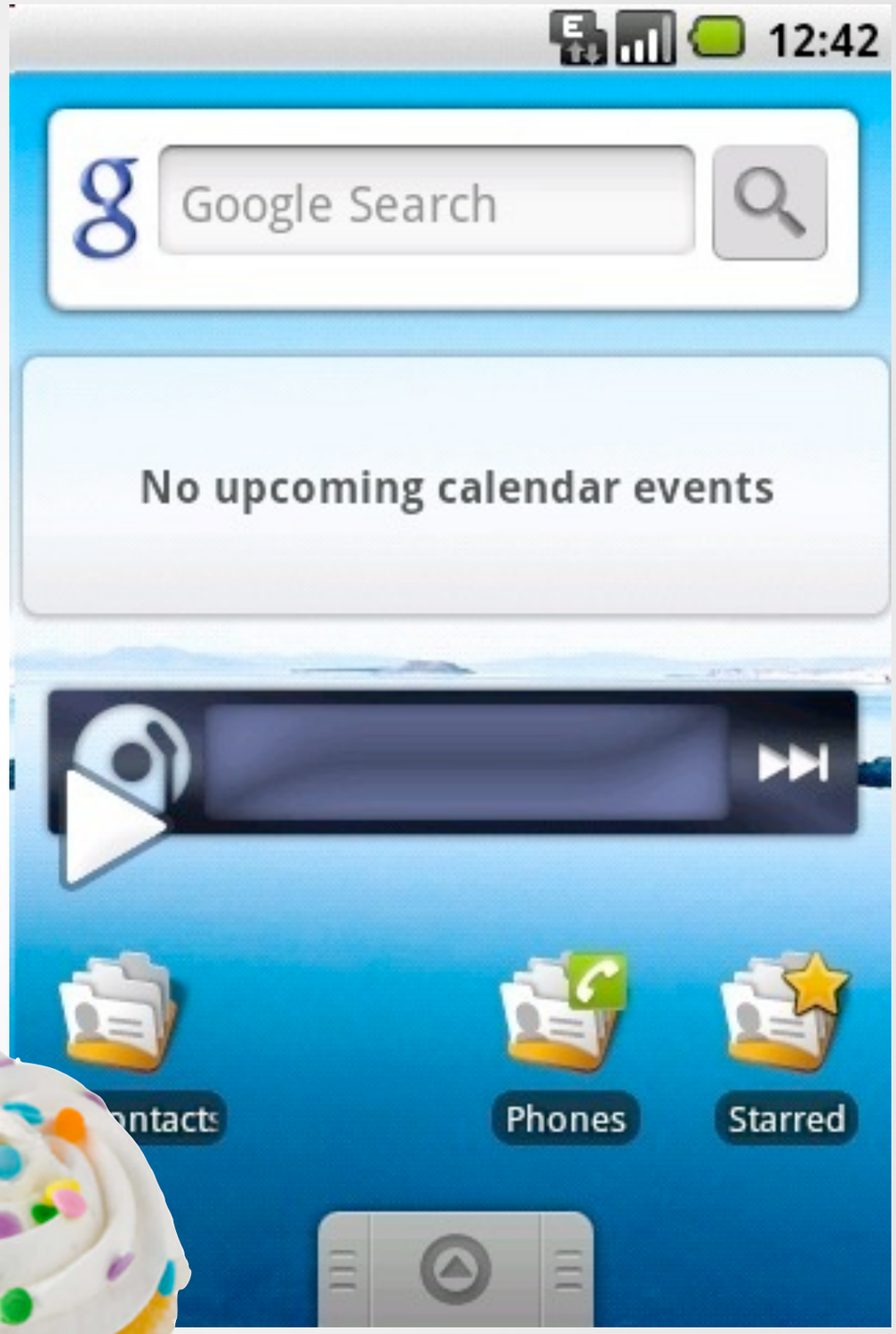
- Many elements where there from the start
- No widgets yet
- Notification Bar
- No tasty name yet ;)

http://www.chip.de/bildergalerie/Android-Timeline-Alle-Versionen-alle-Einfuehrungs-Handys-Galerie_53930690.html

2009

1.5 (Cupcake)

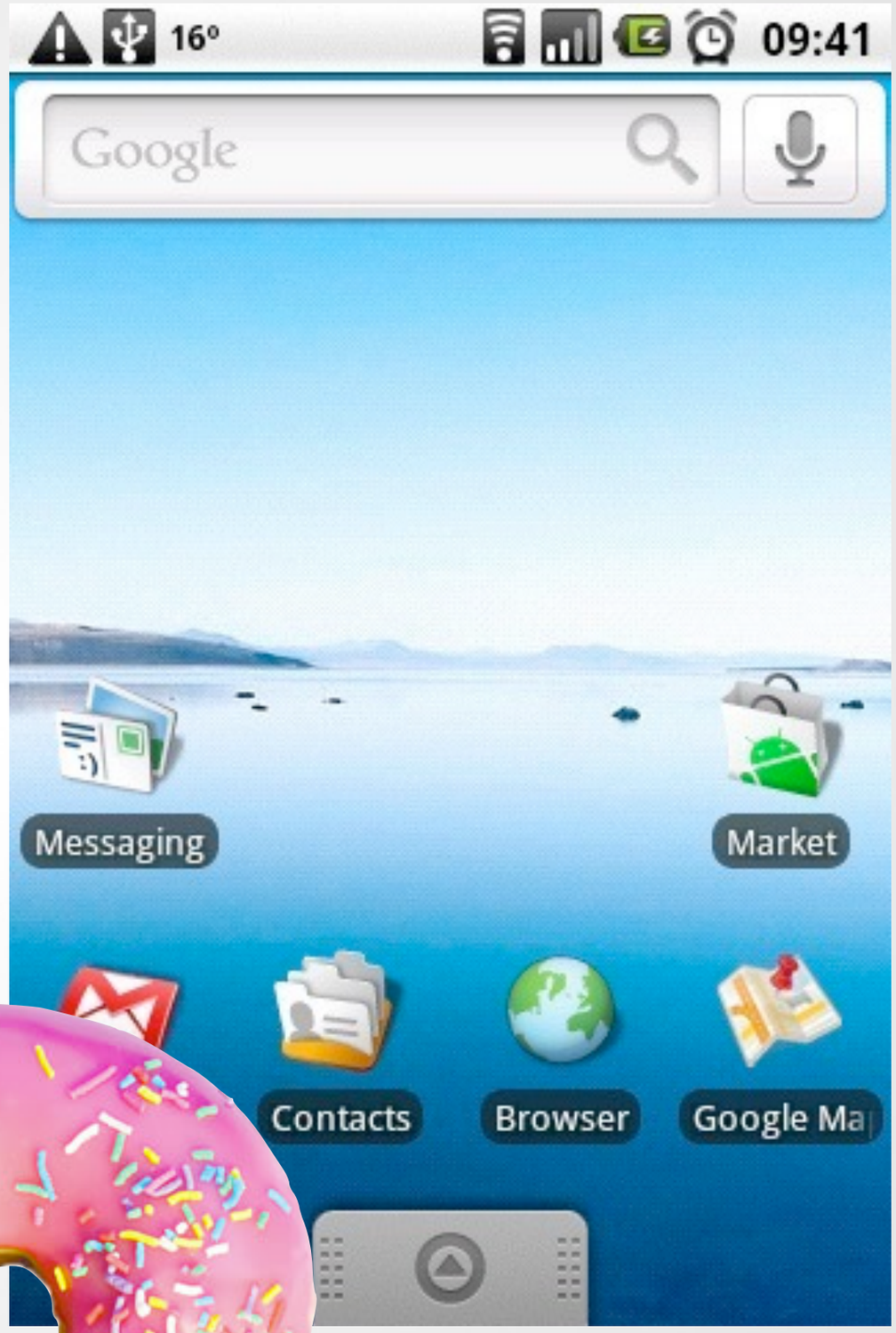
- Onscreen-Keyboard with „Autocomplete“
- Screen switch Animations
- Videoupload



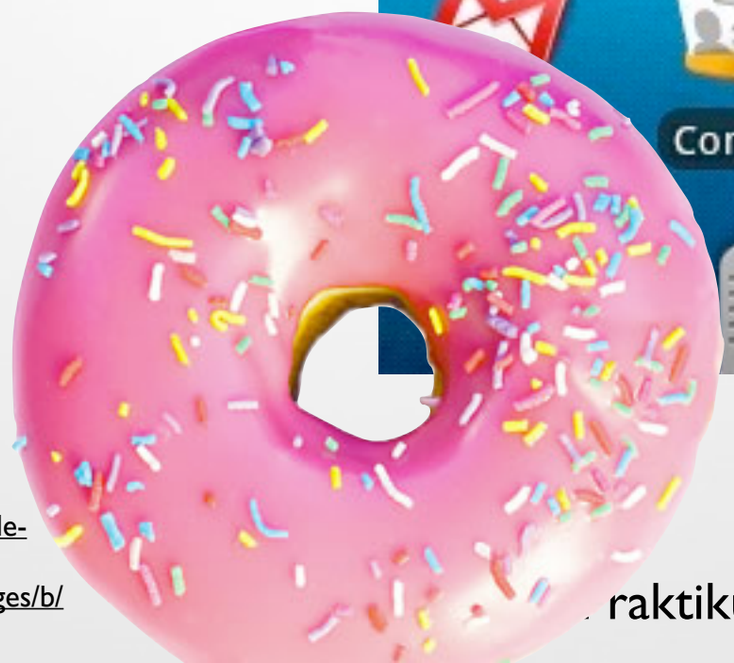
http://www.chip.de/bildergalerie/Android-Timeline-Alle-Versionen-alle-Einfuehrungs-Handys-Galerie_53930690.html
http://images3.wikia.nocookie.net/__cb20110428020146/uncyclopedia/images/4/4b/Cupcake.jpg

2009

1.6 (Donut)



- Screenshots on the android market
- Voice Search
- WVGA resolutions

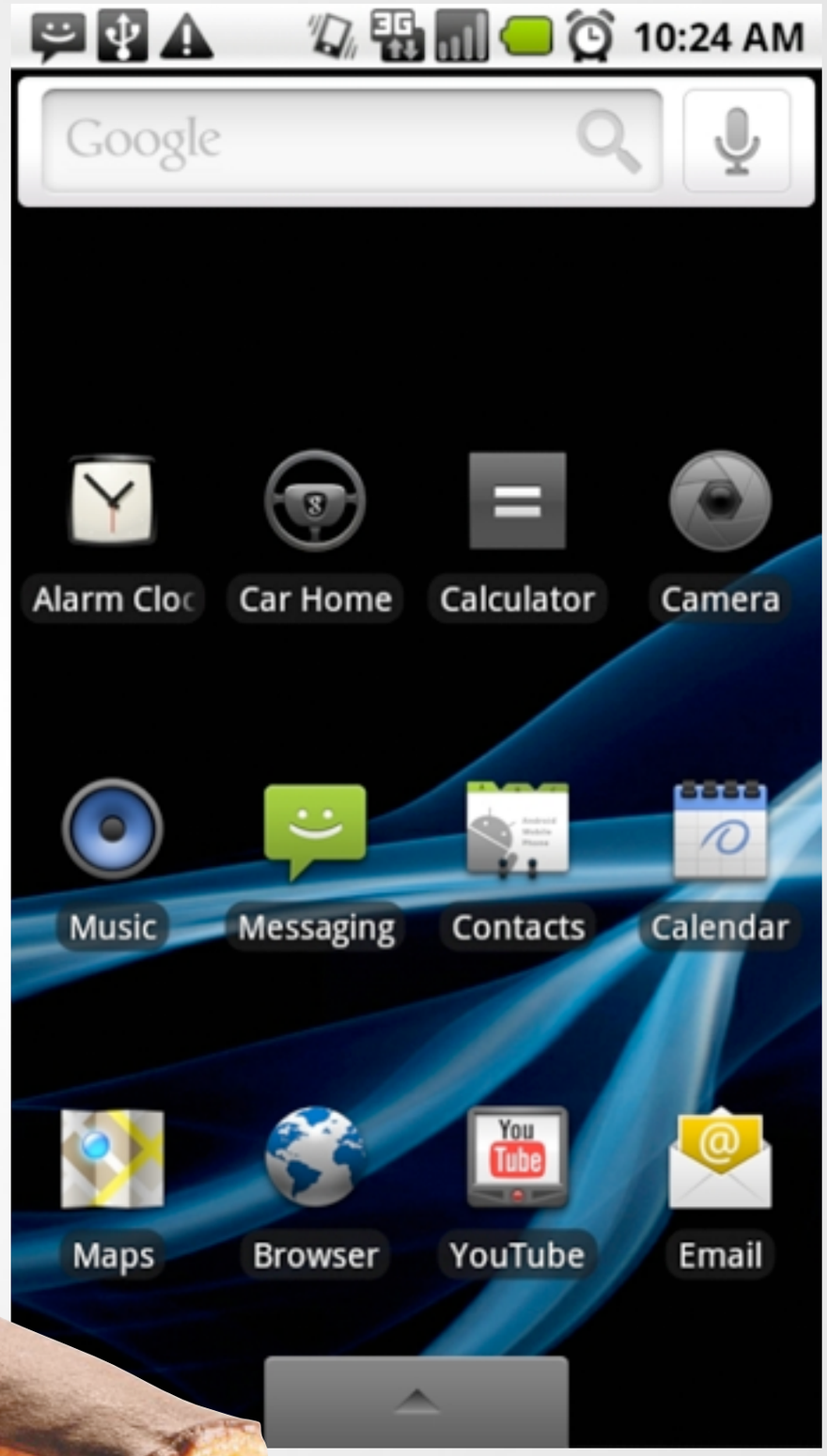


http://www.chip.de/bildergalerie/Android-Timeline-Alle-Versionen-alle-Einfuehrungs-Handys-Galerie_53930690.html
http://images4.wikia.nocookie.net/__cb20120101131538/smosh/images/b/b2/Pink_frosted_sprinkled_donut.jpg

2009

2.0 (Eclair)

- Speed improvements
- More screen resolutions (dip)
- Camera flash support
- Live wallpapers
- Multitouch support



http://www.chip.de/bildergalerie/Android-Timeline-Alle-Versionen-alle-Einfuehrungs-Handys-Galerie_53930690.html
http://gadgetsteria.com/wp-content/uploads/2010/01/eclair.jpg

2010

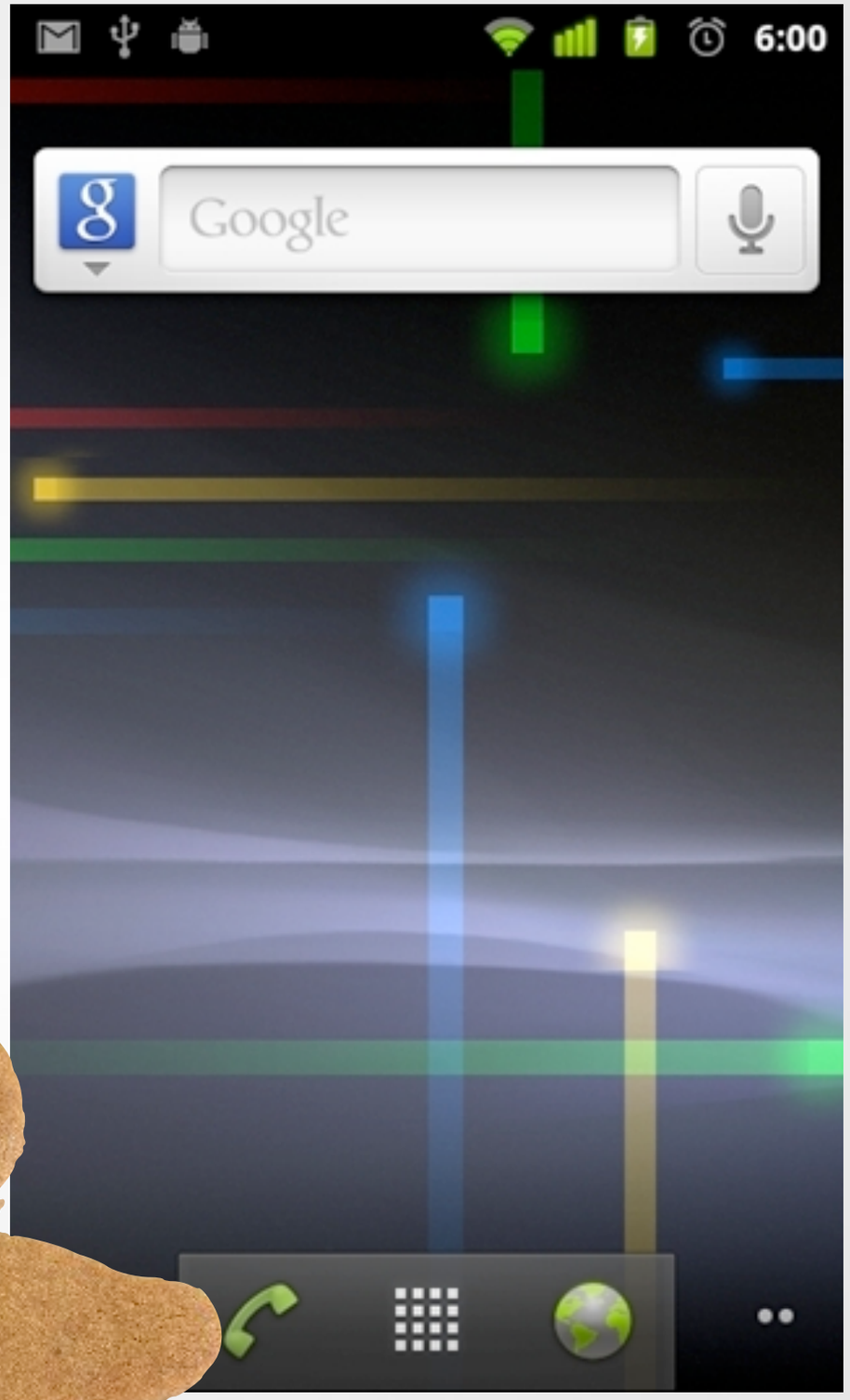
2.2 (FroYo)

- Speed and performance increase
- Flash 10.1 support
- Installing apps on SD-Card
- Tethering

http://www.chip.de/bildergalerie/Android-Timeline-Alle-Versionen-alle-Einfuehrungs-Handys-Galerie_53930690.html
http://www.talkandroid.com/wp-content/uploads/2010/08/FroYo.jpg?3995d3

2010

2.3 (Gingerbread)



- Dual-Core-Unterstützung
- NFC
- HTML5
- bessere Garbage Collection

http://www.chip.de/bildergalerie/Android-Timeline-Alle-Versionen-alle-Einfuehrungs-Handys-Galerie_53930690.html
http://www.exclusivelyfood.com.au/uploaded_images/gingerbread7-744309.JPG

2011

3.X (Honeycomb)

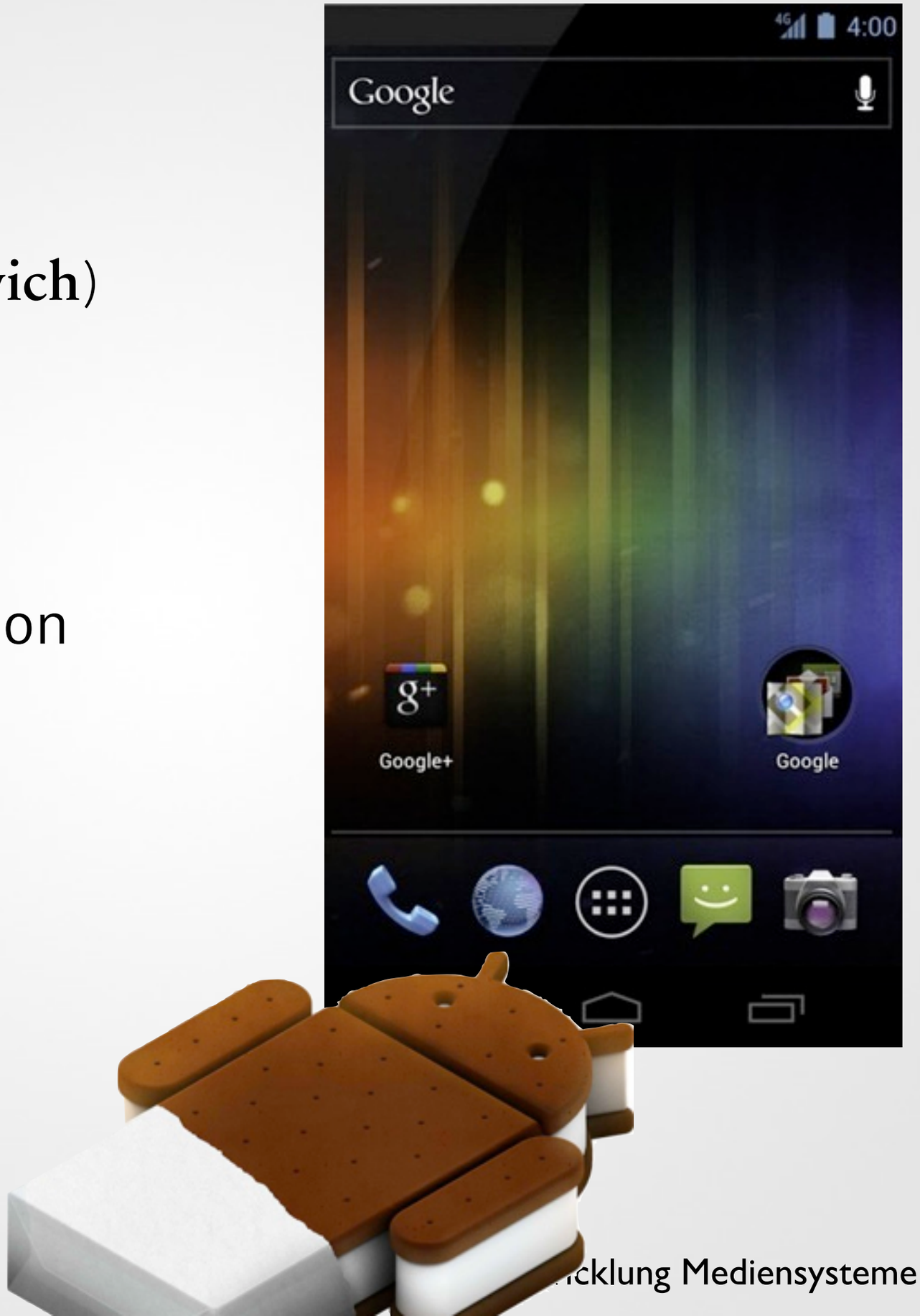
- Tablet Optimized

http://www.chip.de/bildergalerie/Android-Timeline-Alle-Versionen-alle-Einfuehrungs-Handys-Galerie_53930690.html
http://phandroid.s3.amazonaws.com/wp-content/uploads/2010/08/honey-comb-l.jpg

2011

4.X (Ice Cream Sandwich)

- 2.x and 3.x to one version
- Face recognition
- Multitasking



http://www.chip.de/bildergalerie/Android-Timeline-Alle-Versionen-alle-Einfuehrungs-Handys-Galerie_53930690.html
http://www.blogcdn.com/www.engadget.com/media/2011/05/11x05101719.jpg

2012?

5.X (Jelly Bean)

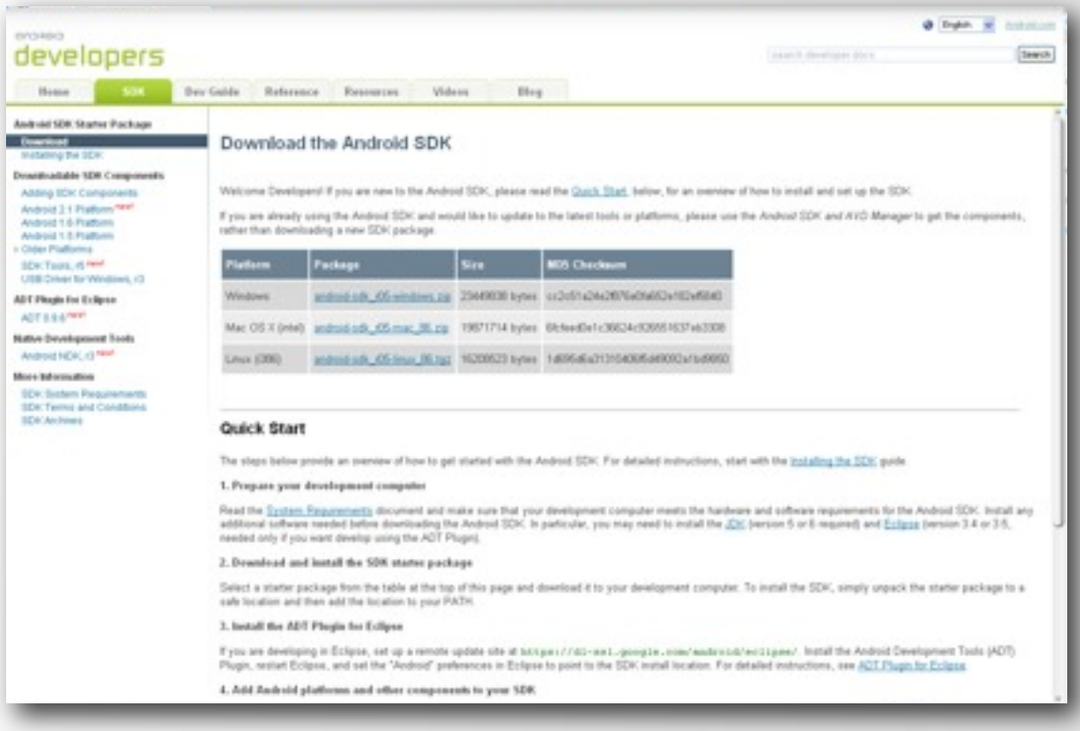
- Assistant (like Siri)
- File Manager
- Malware Protection
- No More Adobe Flash

http://www.chip.de/bildergalerie/Android-Timeline-Alle-Versionen-alle-Einfuehrungs-Handys-Galerie_53930690.html
http://2.bp.blogspot.com/-VYTzIzPtI_o/TbD6WqeBNII/AAAAAAAABfo/sKcKyGozpxg/s1600/jellybelly-flavor-guide.jpeg

Installing The SDK

Installing SDK

- Please follow instructions from the Android doc
- Download and install the Android SDK
- SDK includes documentation, tools and examples
- Set up your IDE; Eclipse (Java EE) recommended
- Install Eclipse Android Development Tools (ADT) plugin, connect it with the Android SDK and Download your Platforms



http://developer.android.com/sdk/index.html

Installing SDK

- Create an Android project
  - Standard Eclipse procedure
  - Automatically creates folders and a Manifest file
  - Can also be used to create a demo project

- Set up a launch configuration
  - Run application from menu or
  - Define settings for run configuration (project, activity, emulator options, …) from Run > Open Run Dialog >

- Run Android application in emulator
  - Be Patient! The emulator takes while to boot up.
  - Keep it open once it was started!

The Nexus One

Source: Wikimedia Commons

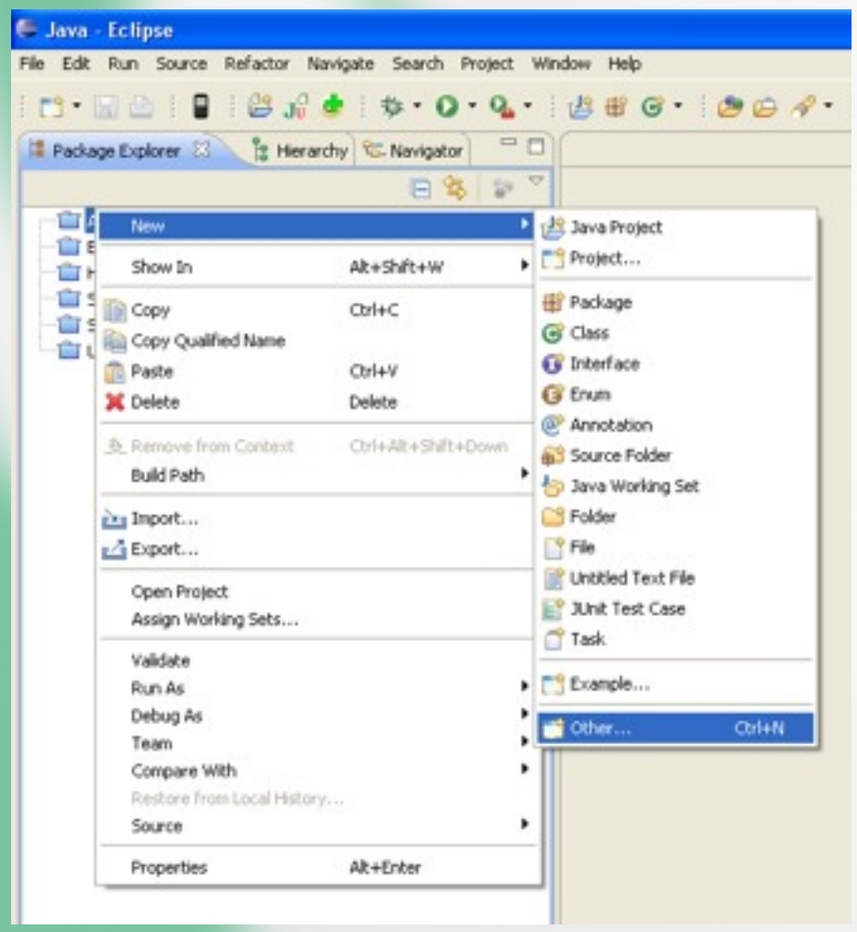
The Nexus One

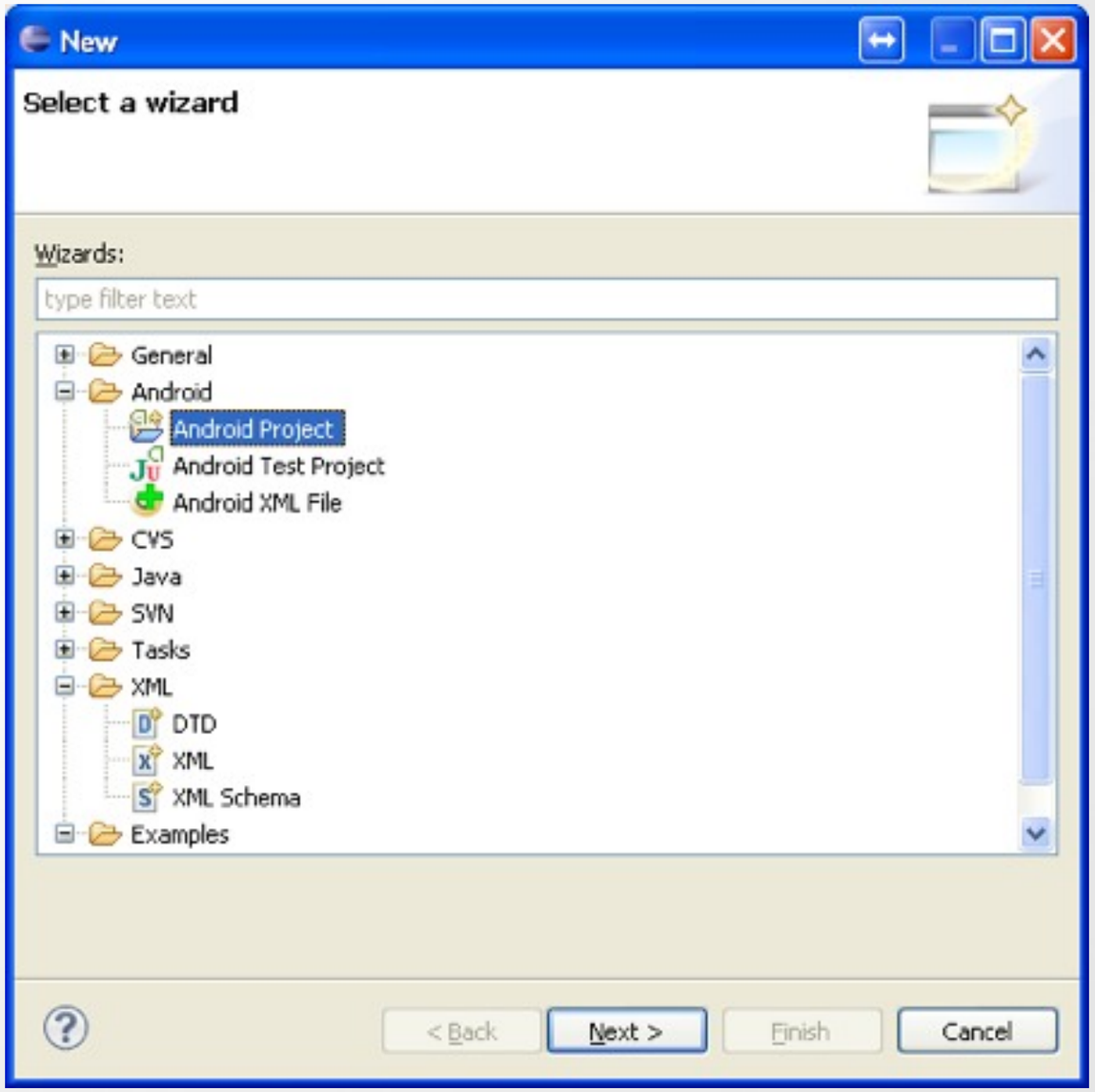
nexus one™

Source: Wikimedia Commons

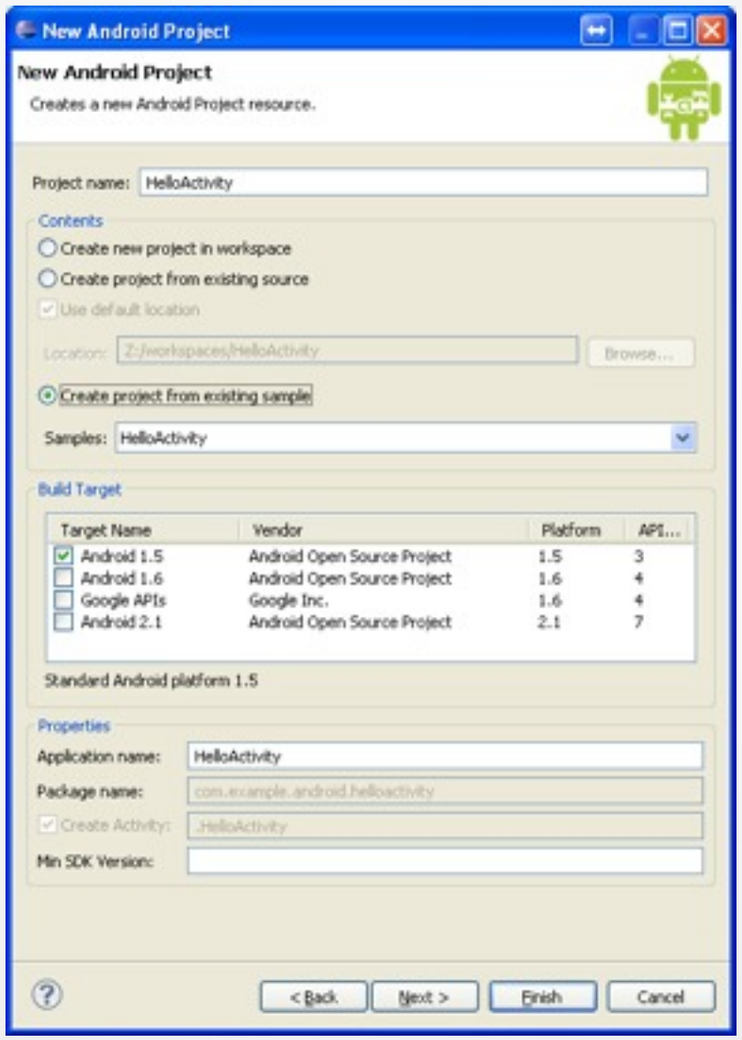
Hello World

Hello Android I





Hello Android II



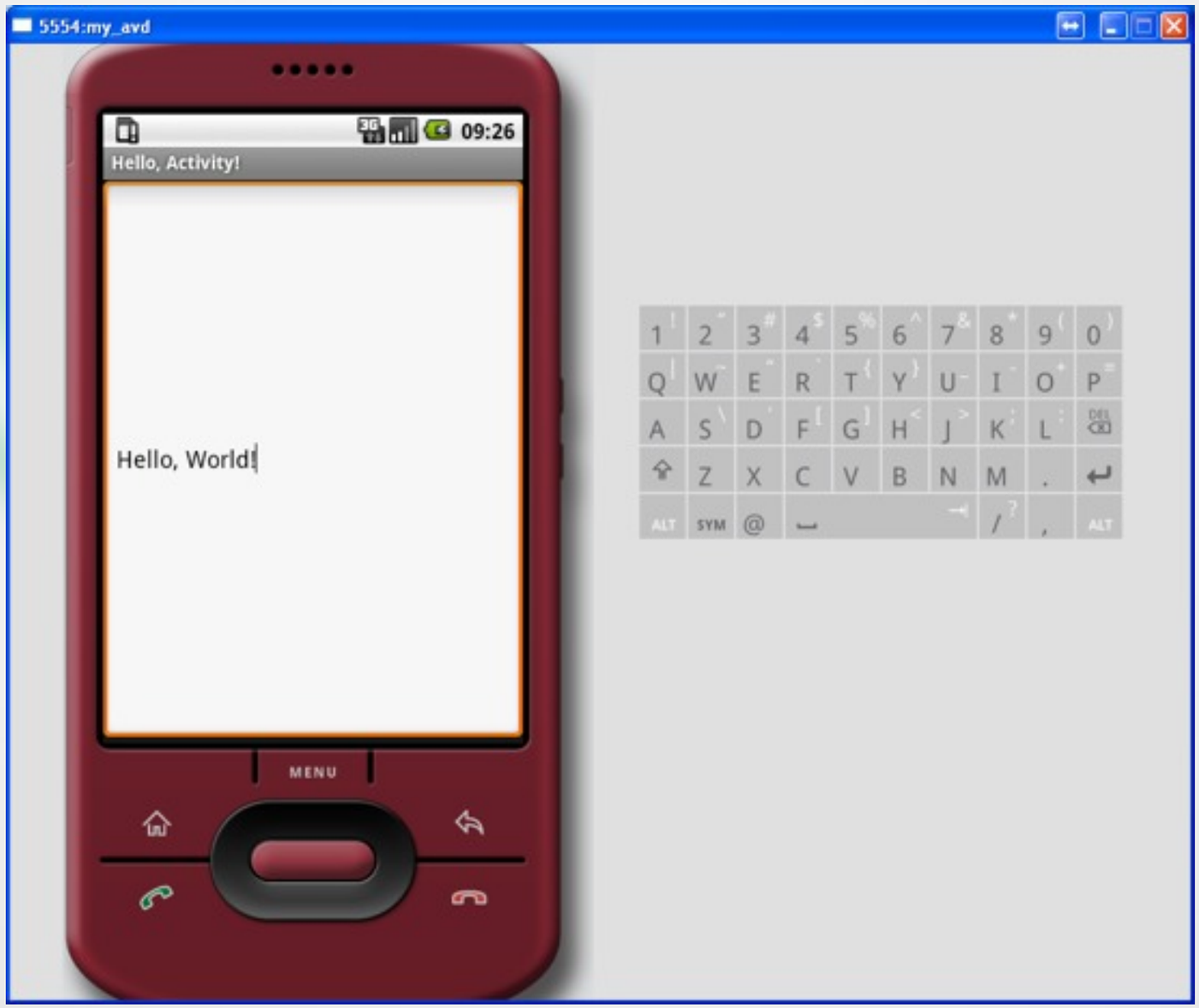
Source: http://code.google.com/android/index.html

Hello Android III

```java
/*
 * Copyright (C) 2007 The Android Open Source Project

package com.example.android.helloactivity;

import android.app.Activity;


/**
 * A minimal "Hello, World!" application.
 */
public class HelloActivity extends Activity {
    public HelloActivity() {
    }

    /**
     * Called with the activity is first created.
     */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // Set the layout for this activity.  You can find it
        // in res/layout/hello_activity.xml
        setContentView(R.layout.hello_activity);
    }
}
```

Source: http://code.google.com/android/index.html

Hello Android IV



Hello World

Anatomy of an Android

- 4 main building blocks for Android applications
  - Activity
  - Intent Receiver
  - Service
  - Content Provider

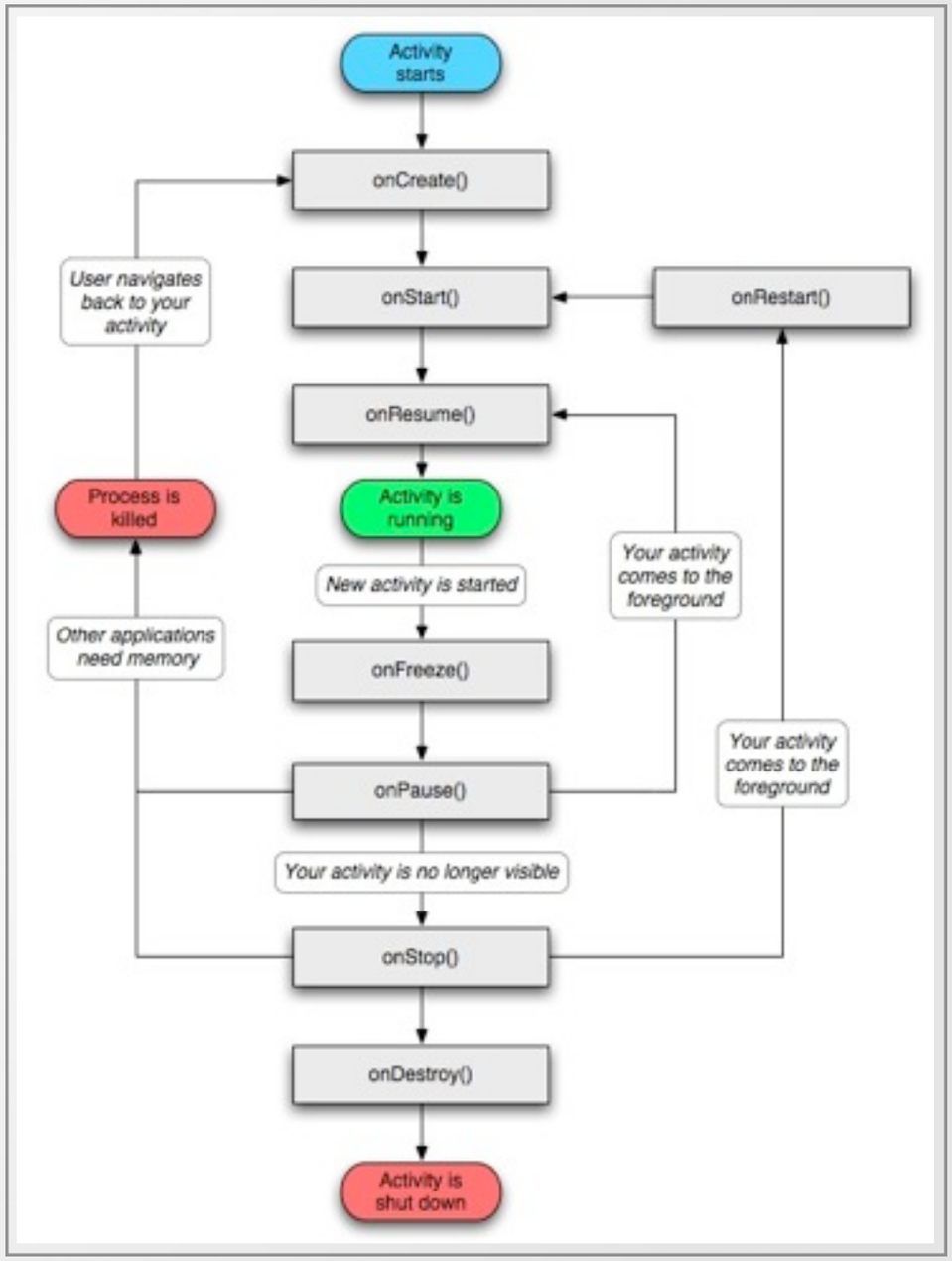
```
<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
        package="com.my_domain.app.helloactivity">

    <application android:label="@string/app_name">

        <activity android:name=".HelloActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>

    </application>

</manifest>
```

- AndroidManifest.xml lists all components of an application, their capabilities and requirements

Source: http://code.google.com/android/index.html

Activity

- Single, focused thing or task
- Extends the Activity base class
- Refers to a single screen in a (multi-screen) application
- Displays a UI, interacts with user, responds to events
- 2 main methods:
  - onCreate(Bundle): initialization of activity, set UI, …
  - onPause(): leaving an activity
- Moving through screens by starting other activities
- Activities managed by activity stack
- New activity put on top of the stack
- 4 states: active/running, paused, stopped, killed/shut down



Source: http://code.google.com/android/index.html

Intents and Intent Filters

- Intent
  - Abstract description of an operation/action to be performed
  - Mostly used for launching activities; “glue between activities”
  - Action: general action to be performed, e.g. VIEW_ACTION, EDIT_ACTION, MAIN_ACTION, …
  - Data: data to operate on, expressed as a URI
  - Example: **VIEW_ACTION content://contacts/1**

- Intent Filter
  - Describes what Intents an activity can handle
  - Activities publish Intent Filters describing their capabilities/ how they can handle certain Intents and their actions
  - Navigating between screens is accomplished by resolving Intents => system matches Intents and Intent Filters
  - Activity calls method startActivity(myIntent)

Intent Receiver, Service, Content Provider

- Intent Receiver
  - Used to execute code upon an external event, e.g. phone rings
  - Usually no UI; may use the NotificationManager
- Service
  - Application component running in the background
  - Runs indefinitely, no UI, no interaction with user
  - E.g. media player
- Content Provider
  - Used to share data with other applications

Life Cycle of an Android Application

- Each Android application runs in its own Linux process
- Process's lifetime not directly controlled by application
- Determined by the system, depending on running applications, their importance, available memory
- Components (Activity, Service, Intent Receiver) impact the lifetime of the application's process
- Importance hierarchy for killing processes based on
  - Components running in them
  - The state of these components

Android's Importance Hierarchy

1. Foreground Process
   - Required for current user activities
   - E.g. running an Activity at the top of the screen
2. Visible Process
   - Activity is visible but not in the foreground (onPause())
   - E.g. previous activity displayed behind a foreground dialog
3. Service Process
   - Holds a Service, not directly visible (e.g. media player, network up/ download)
4. Background Process
   - Holds an Activity that is currently not visible (onStop())
   - Can be killed at any time to reclaim memory
5. Empty Process
   - Holds no active application components

Fragen?