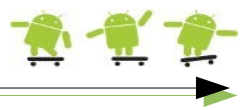




Google's Betriebssystem für mobile Plattformen



- Einleitung
- Softwarearchitektur
- Softwareentwicklung für Android
- Unterschied zu anderen mobilen Plattformen
- Wirtschaftliche Aspekte
- Fazit



Was ist Android?

- OS von Google
- Für Mobiltelefone / Smartphones
- Für Tablet-PCs / Netbooks

- Ursprünglich 2003 von Andy Rubin entwickelt
- 2005 von Google aufgekauft
- Erstes Gerät: (2008)

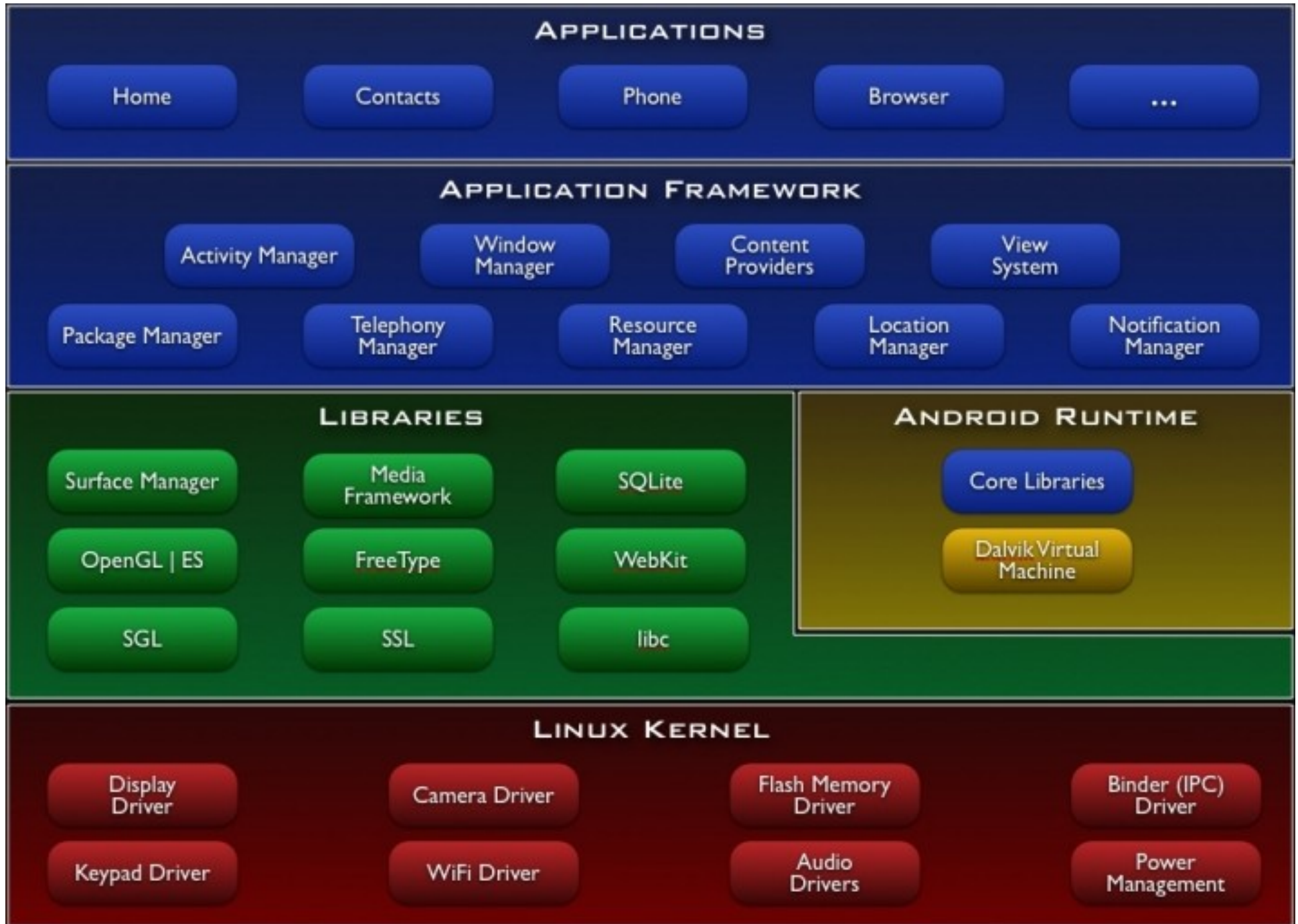
HTC Dream

„T-Mobile G1“



Softwarearchitektur





Dalvik Virtual Machine

- Entwickelt von Google Mitarbeiter Dan Bornstein unter der Apache Lizenz 2.0
- Virtuelle Registermaschine für mobile Plattformen
- Unterstützt moderne Prozessoren wie den ARM
- Kann Code für eine JVM in DVM Code umwandeln (mit dx: *.class's → *.dex)
- Ausführung dann sehr schnell und ressourcenschonend

→ Ausführung mehrerer DVMs möglich!

→ Jeder Prozess hat eigene DVM

► Multitasking



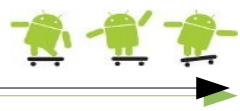
Dalvík: isländische Stadt

ANDROID



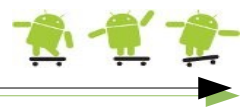
Die Applikation

- in Java geschrieben
- wird in eine *.apk-Datei kompiliert und wird zur Installation und Ausführung genutzt
- wird innerhalb einer Sicherheits-Sandbox ausgeführt
- Mehrbenutzerfähigkeit Linux → ID pro Applikation anstatt Benutzer
- Dalvik → DVM pro Prozess / Applikation
- „principle of least privilege“ → sperrt unerwünschte Zugriffe auf andere Applikationen
- erwünschter Austausch zwischen Applikationen durch:
 - Zuweisung der gleichen „Benutzer“-ID an zwei Applikationen
 - Einstellung von Privilegien pro Applikation bei Installation durch Benutzer



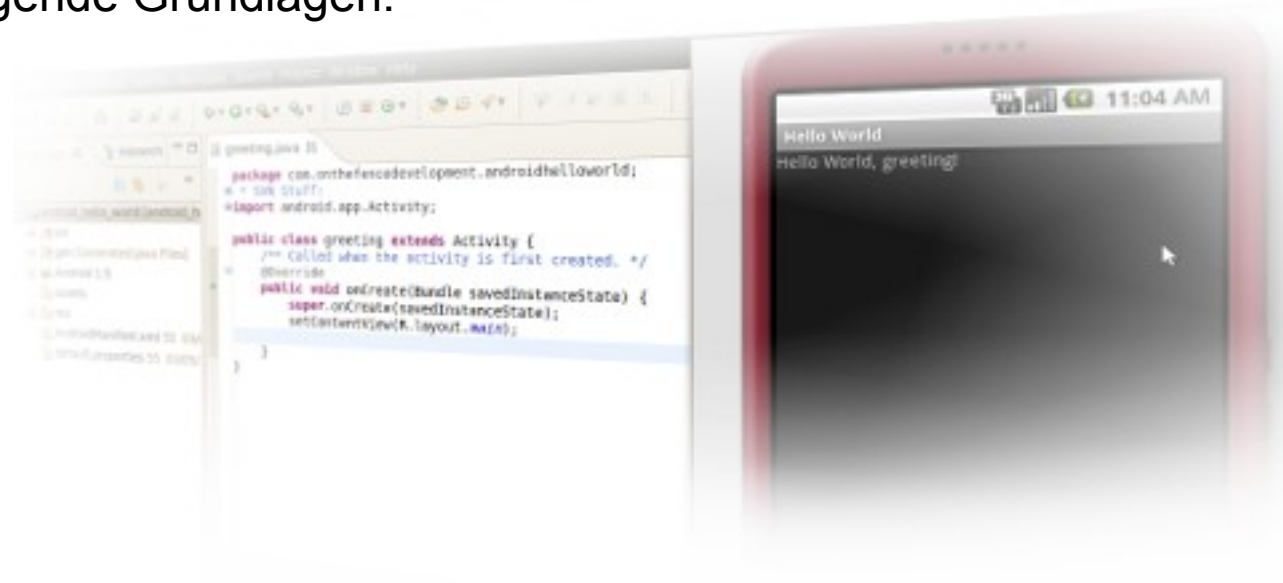
Softwareentwicklung für Android

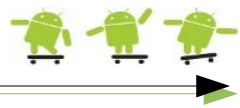




Erfordert Kenntnis über folgende Grundlagen:

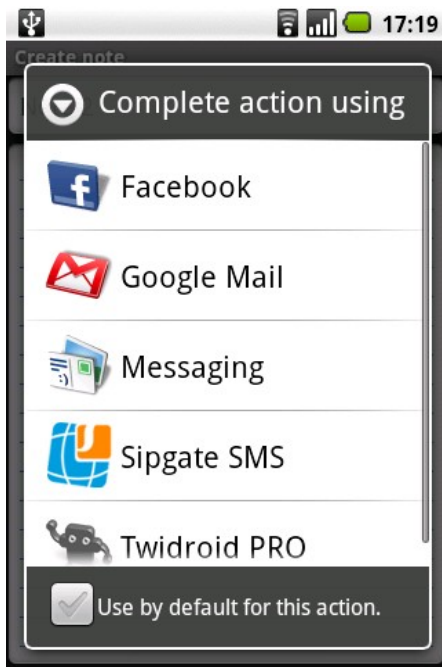
- Aktivitäten
- Services
- Content Provider
- Broadcast Recievers
- Manifest-Datei
- Applikations-Ressourcen





Aktivitäten

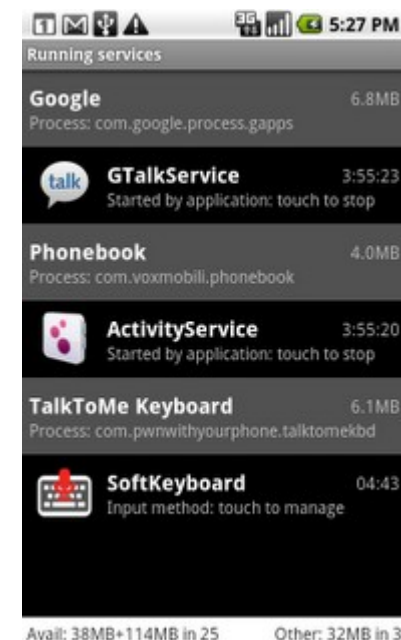
- User-Interfaces für bestimmte Aufgaben oder Anzeigen
- Können unabhängig von ihrer Applikation aufgerufen werden
- Stehen, sofern Sicherheitsprivilegien es zulassen, anderen Applikationen zur Verfügung
- Beispiel:
 - Gerade ein mit der „camera“-Applikation erstelltes Foto über die allgemein verfügbare „send“-Aktivität versenden





Services

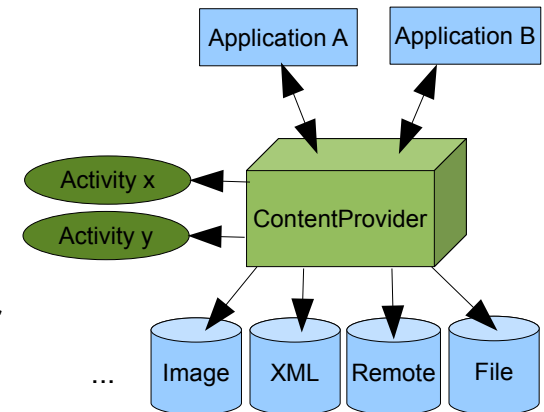
- Zum Ausführen langwieriger oder aktualisierender Operationen im Hintergrund
- ... im Hintergrund → blockiert den Benutzer nicht!
- Keine Benutzeroberfläche
- Können vom Benutzer auch deaktiviert werden
- Applikationen können Services starten und sich an sie „binden“, um eine Interaktion mit diesen herzustellen
- Beispiel:
 - Abspielen von Musik (als Service) während Benutzer surft
 - Chat-Aktualisierung (als Service), ohne dass Applikation laufen muss (ressourcenschonend)

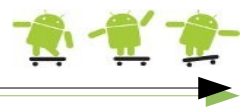




Content Provider

- Verwaltet gemeinsamen Zugriff auf Applikations-Daten
- Unerheblich, wo die Daten liegen
- Leitet nur Informationen an Applikationen weiter, die über Zugriffsprivilegien verfügen
- Beispiel:
 - Alle Kontakte dürfen sichtbar sein für die Telefon-Applikation
 - Benutzer will aber nicht, dass ein Spiel Zugriff auf Kontakte hat

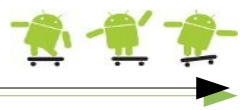




Broadcast Recievers

- Komponente, die auf systemweite Ankündigungen reagiert
 - Antwort kann eine Statusnachricht sein
 - Antwort kann in Form eines Service-Starts geschehen
 - Systemweite Ankündigungen meist vom OS, können aber auch von Applikationen initiiert werden
-
- Beispiel:
 - Systemweite Meldung bezüglich des Akku-Stands
→ Dialog / Tasbaranzeige: „niedriger Akkustand“





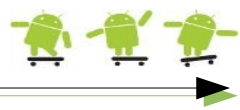
Die Manifest-Datei

- XML-Konfigurationsdatei
- Beschreibt Komponenten einer Applikation
- Definiert benötigte Zugriffsrechte
- Definiert minimales API-Level (Android-Version)
- Definiert zusätzliche Libraries (z.B. Google Maps)
- Definiert benötigte Hard- und Software-Features
- u.v.m.

- Beispiel:
 - Kamera oder Bluetooth Service wird für eine Applikation vorausgesetzt und das Gerät verfügt über diese nicht

- die Applikation im AndroidMarket für das Gerät nicht angezeigt.
 - **Filterung** im AndroidMarket über Manifest-Datei

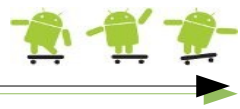
```
<manifest . . . >
  <permission android:name="com.example...
  <uses-permission android:name="com.example...
  . . .
  <application . . . >
    <activity android:name="com.example.pr
      android:permission="com.example...
    . . . >
  </activity>
</application>
</manifest>
```



Applikations-Ressourcen

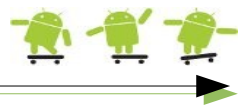
- XML-Templates, Bilder, Konfigurations- und Übersetzungsdateien werden in vordefinierte Ordnerstrukturen abgelegt werden
- Anhand des Ordners werden die Ressourcen richtig geladen
- Jeder Resource bekommt eine ID und ist darüber im Programmcode ansprechbar
- Keine Pfad-Anpassungen innerhalb des Programmcodes notwendig

- Beispiel:
 - Bilder: `./drawable/`
 - Franz. Übersetzung: `./values-fr/`



Unterschied zu anderen mobilen Plattformen

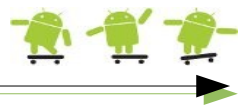




Vorteile

- Data-Sharing über ContentProvider → hohe Benutzerfreundlichkeit und Wiederverwendbarkeit
- Mehrere Einstiegspunkte für Applikation (durch Ansteuerung über Aktivitäten)
- Isolation von Applikationen in separate Prozesse → kein unerwünschter Datenaustausch
- Android und SDK-Tools (inkl. Debugger, Emulator) sind OpenSource
- Es existieren kostenlose Tools /IDEs. Herausragend ADT Plugin für Eclipse.
- Jeder kann Applikationen in den Market stellen, anders als bei Apple wegen Filterung
- Entwicklung von Android Applikationen ist kostenlos
- Für Entwicklung ist ein Android-Gerät nicht zwingend erforderlich

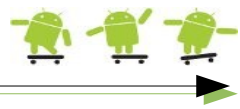




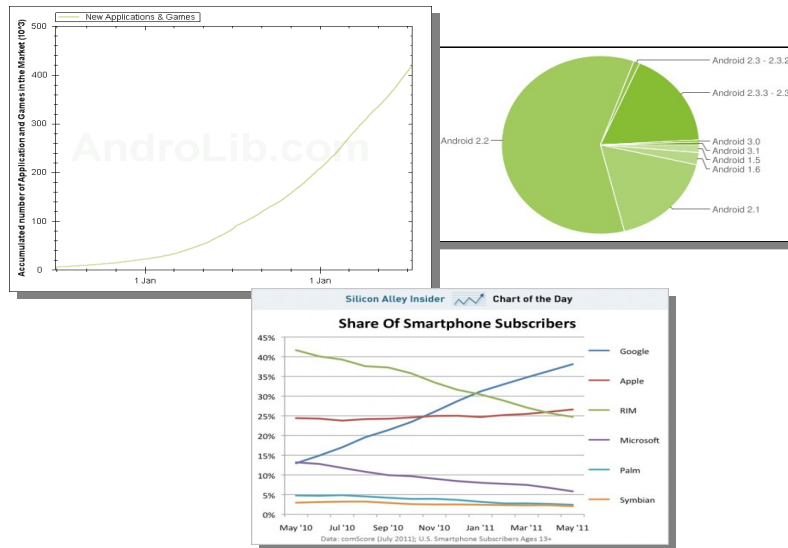
Nachteile

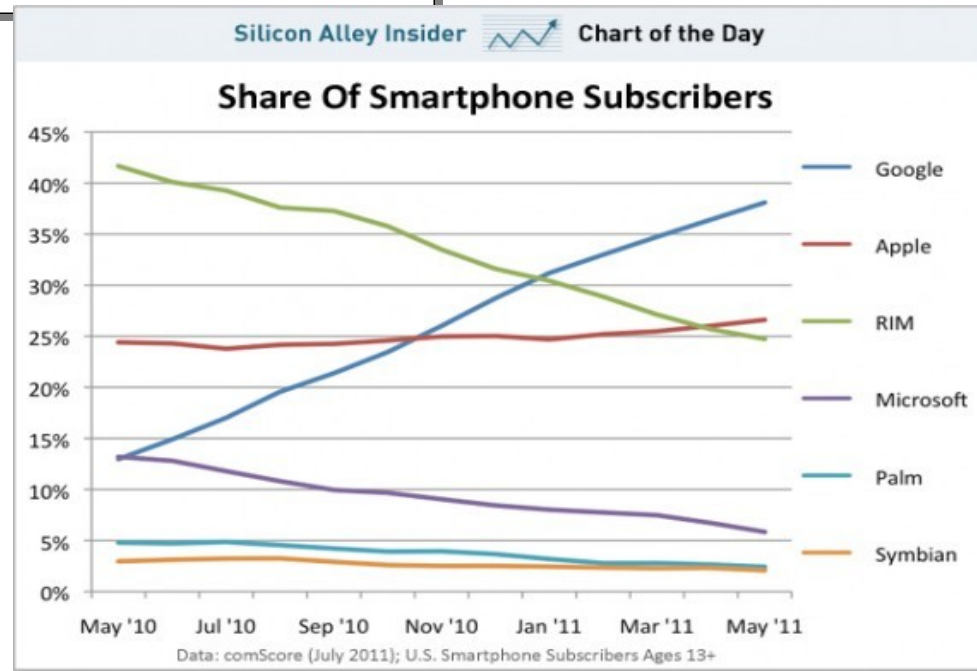
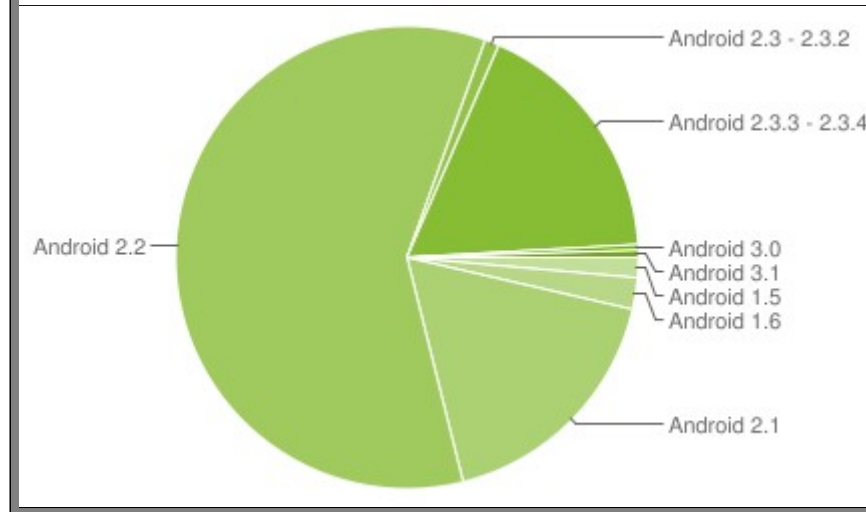
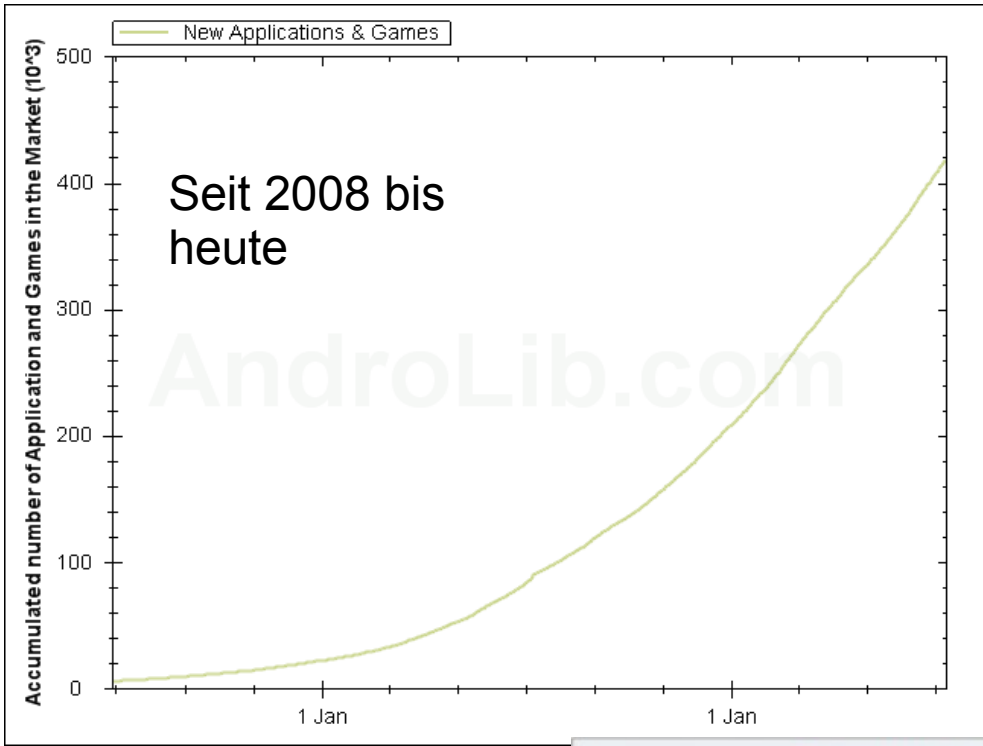
- Freier, nicht zensierter Market kann und wurde schon zum Verbreiten schädlicher Software genutzt
 - Google greift dann per entfernter Löschung der Applikation auf betroffenen Geräten zu. Dies geschieht zeitlich verzögert.
 - Es wird sich sehr auf Meldungen aus der Community verlassen
- Smartphones mit älteren Android-Versionen können zum Teil neuere Applikationen nutzen. → Entwickler müssen sich über die Unterschiede in den Android-Versionen bewusst sein und ggf. ihre Applikationen anpassen
 - Mittlerweile aber nicht mehr so viele Unterschiede wie von 1.6 auf 2.0
 - Mittlerweile sind fast alle aktuellen Smartphones leistungsstark genug, zumindest 2.1 zu nutzen.
- Erteilung von Zugriffsrechten über den Benutzer kann Nachteile haben
 - es wird sich hier auf den Verstand des Benutzers verlassen

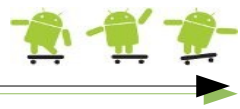




Wirtschaftliche Aspekte bezüglich Android







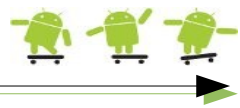
OHA (Open Handset Alliance)

- ~ 80 Firmen haben sich mit Google zusammengeschlossen, um offene Standards zu fördern und zu entwickeln

Darunter z.B.:



- viele namenhafte Hersteller verwenden Android: z.B. Samsung, LG, HTC, Motorola, Nvidia
- es werden täglich ~ 500.000 Android Smartphones aktiviert
- im Android Market sind derzeit ~ 420.000 Applikationen



Gegenstück zur OHA: CPTM-Konsortium

- CPTM-Konsortium besteht aus:

Microsoft



ORACLE

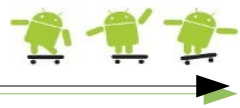
EMC²
where information lives

CPTN vs. OHA, möglicher Einfluss auf die Android-Entwicklung?

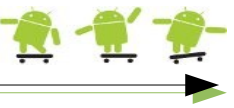
- 2010 Oracle kaufte SUN auf
- Es folgte eine Klage von Oracle gegen Google, diversen Code kopiert und für Dalvik verwendet zu haben
- Klageerwiderung Googles: Oracle habe Beweise gefälscht und Google selbst keine Urheberrechtsverletzungen begangen
- November 2011: voraussichtlicher Prozess, falls keine Einigung



Fazit



- Android ist ein stabiles, umfangreiches und erweiterbares System
- Bietet viele Funktionalitäten und noch mehr durch seine Vielfalt an Applikationen
- Spricht ein großes Entwickler-Publikum an, da kostenfrei
- Attraktiv für StartUp-Unternehmen: da keine Lizenzgebühren
- Dokumentation für Entwickler absolut empfehlenswert: Klar, strukturiert, verständlich, aktuell!
- Vielfalt von Applikation nicht kontraproduktiv, sondern produktiv: motiviert Entwickler, ihre Applikationen zu verbessern, wenn sie denn die Nr. 1 bleiben wollen → Qualitätssteigerung
- Sicherheitsaspekt bezüglich isolierter Applikationen gut, könnte mit Überwachung des Markets seitens Google verbessert werden
- Endverbraucher zieht Vorteil aus dem Wettrüsten zwischen Apple, Google, Microsoft, RIM etc.



Danke für eure Aufmerksamkeit

;-)