Sicherheit in Android

Peter Salchow

INF-M2 - Anwendungen 1
Sommersemester 2008
Department Informatik
HAW Hamburg

20. Mai 2008

Inhalt

Inhalt

Motivation

- Mobile Geräte sind stark verbreitet
- Bedarf nach mobiler Nutzung des Internets wächst ständig
- Innovative Anwendungen auf mobilen Geräten können komfortable Dienste ermöglichen
- Jedoch: Softwareentwickler haben meist nicht die volle Kontrolle über das mobile Gerät
- Trennung von nativen Anwendungen und anderen Anwendungen
- Android ist vielversprechender Kandidat für Offenheit von mobilen Geräten
- Bei verstärkter Nutzung von mobilen Anwendungen steigt auch Relevanz der Daten auf dem Gerät
- Thema Sicherheit rückt weiter in den Vordergrund
- Was bietet die Architektur von Android an Sicherheit?

Inhalt

Was ist Android?

- Software-Stack für mobile Geräte
- Besteht aus:
  - Betriebssystem
  - Middleware
  - Anwendungen
- Umgebung zum Ausführen von Anwendungen auf mobilen Geräten
- Stellt Schnittstellen zum Zugriff auf das Gerät zur Verfügung

Was ist Android?

- Kann vorerst nur im Emulator ausgeführt werden - Erste Geräte im zweiten Halbjahr 2008 erwartet
- Wird von der „Open Handset Alliance“ entwickelt
- Zusammenschluss von 34 Mitgliedern (z.B. T-Mobile, Samsung, ebay) und von Google initiiert
- Ziele:
  - Kostenlos
  - Open Source (Apache-Lizenz v2)
  - Geräteunabhängig
  - Einfache und schnelle Anwendungsentwicklung

Architektur

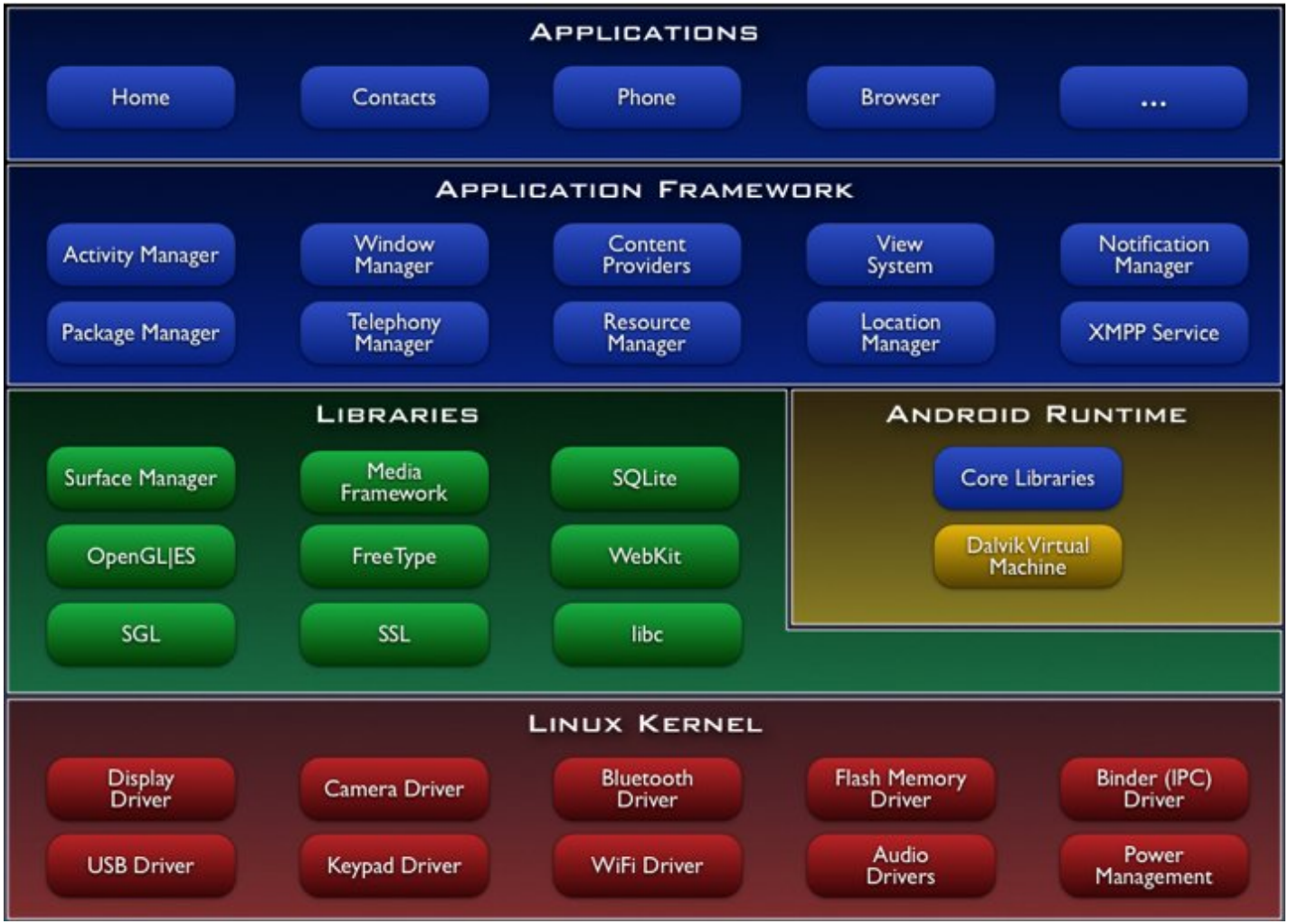
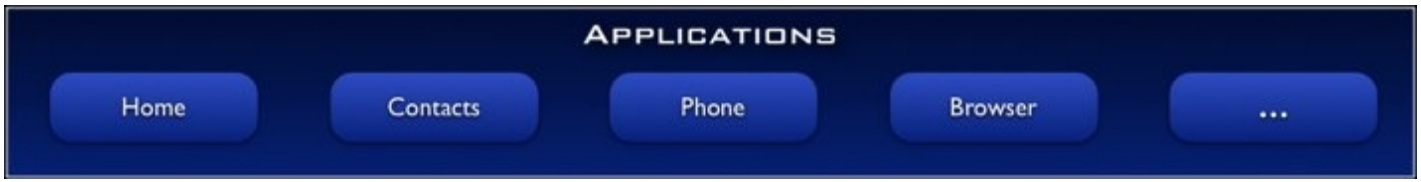


Abbildung: Architektur von Android [Google Android]

Applications



- Oberste Schicht der Architektur
- Enthält die Core-Anwendungen (Kalender, E-Mail-Client etc.)
- Neu entwickelte Anwendungen werden hier ausgeführt
- Alle Anwendungen in dieser Schicht sind in Java geschrieben

Framework

- Alle Anwendungen bauen auf Framework auf
- Stellt Views für Anwendungen zur Verfügung
- Übernimmt Lifecycle-Kontrolle der Applikationen
- Ermöglicht Wiederverwendung von Komponenten
- Anwendungen können ihre Ressourcen für andere zur Verfügung stellen
- Regelt die Berechtigungen für den Zugriff auf Anwendungen

Libraries

- Android enthält C/C++ Bibliotheken, die von vielen anderen Komponenten genutzt werden
- Entwickler erhalten nur über das Framework Zugriff auf Funktionen der Bibliotheken
- Beispiele:
  - SQLite - relationales DBMS
  - libc - Standard C-Library
  - Surface Manager - Darstellung von 2D / 3D Grafiken

Android Runtime

- Runtime besteht aus der Dalvik VM und den entsprechenden Java Klassenbibliotheken
- Dalvik VM führt die Anwendungen aus
  - Spezielle VM - Optimiert für mobile Geräte
  - Effizienter für kleinere CPUs, geringer Stromverbrauch
  - Basiert auf Registerautomat (Java-VM auf Kellerautomat)
- class-Dateien werden über Cross-Compiler in Dalvik-Executables umgewandelt
- Jede Anwendung wird in einer eigenen VM ausgeführt
- Threading und Low-Level Memory Management wird dem Linux-Kernel überlassen

Kernel

- Basiert auf dem Linux-Kernel 2.6
- Abstrahiert zwischen der Hardware und dem restlichen Software-Stack
- Stellt Systemtreiber zur Verfügung
- Funktionen:
  - Speicherverwaltung
  - Prozessverwaltung / Scheduling
  - Network Stack

Inhalt

Sicherheitsfeatures in Android

Jede Anwendung wird in eigener VM ausgeführt:

- Anwendungen während der Ausführung gegeneinander isoliert
- Ein Prozess kann einen anderen nicht stoppen
- Fehlerhafte Anwendungen können nicht das gesamte System blockieren

Sicherheitsfeatures in Android

Dateizugriff und UserID:

- Alle Anwendungsdaten (auch Dateien) sind private
- Jede Anwendung erhält eigene UserID
  - Wird bei Installation zugewiesen und bleibt konstant
  - Unter Linux wird jeder Datei diese UserID zugewiesen
  - Zugriff auf Datei erhält nur der Prozess mit entsprechender UserID
- Anwendungen können bei Bedarf gleiche UserID erhalten
  - Zuweisung über *sharedUserId*-Tag in der *AndroidManifest.xml*
  - Anwendungen werden dann als gleiche Prozesse mit gleichen Rechten behandelt
  - Anwendungen müssen dazu mit gleicher Signatur signiert sein
- Freigabe (r/w) von Dateien über bestimmte Flags möglich

Sicherheitsfeatures in Android

Rechtevergabe:

- Pessimistischer Ansatz - Vorerst keine Rechte für Anwendungen
- Benötigte Rechte auf Dienste (SMS, Telefon, etc.) müssen in *AndroidManifest.xml* beantragt werden
- Bei Installation kann Benutzer die einzelnen Rechte vergeben / entziehen
- Während der Ausführung wird Benutzer nicht wieder gefragt
- Jede Anwendung kann ihre eigenen Rechte definieren

Sicherheitsfeatures in Android

Verschlüsselung von Daten:

- Verschlüsselte Datenübertragung über SSL - Verwendung von *javax.net.ssl*
- Android beinhaltet *javax.security* - Verschlüsseln / Signieren von Daten auf Anwendungsebene

Schwachstellen in Android

Speichern von Daten:

- Speicherung von Daten entweder über Dateisystem oder über SQLite
- Dateisystem:
  - Keine Verschlüsselung von Daten im Dateisystem
  - Bei Verlust / Diebstahl des Geräts kann auf Daten zugegriffen werden (lowlevel)
  - Mit Root-Rechten können Dateien von jeder Anwendung gelesen werden
- SQLite:
  - Bislang kein Krypto-Modul für SQLite in Android enthalten
  - Existierende Module werden vorerst nicht eingebunden, da diese nicht frei zugänglich sind
- Lösungsansatz: Verschlüsselung der Daten in Anwendung - Sehr aufwändig

Schwachstellen in Android

- Keine Unterstützung von PKCS
  - PKCS12 - Format zur Speicherung von privaten Schlüsseln und zugehörigen Zertifikaten
  - Zur Zeit können diese Formate nicht eingelesen und verwendet werden (z.B. bei SSL/TLS)
  - Bislang noch keine Reaktion der Entwickler auf das Problem
- Android erlaubt JNI
  - Möglicherweise Zugriff auf Core-Libraries
  - Zugriffe könnten Sicherheitskonzept von Android umgehen

Schwachstellen in Android

- Apache-Lizenz
  - Eigentliches Vorhaben: Verbesserung der Sicherheit durch OpenSource
  - Jedoch: Drittanbieter können Komponenten durch eigene Implementierungen ersetzen
  - Durch Apache-Lizenz müssen diese Komponenten nicht freigegeben werden
  - Problematisch bei sicherheitsrelevanten Komponenten
- Zertifizierung von Anwendungen
  - Anwendungen für Android sollen zertifiziert werden
  - Bislang hat Google noch keinen Zertifizierungsprozess vorgestellt

Bewertung der Sicherheit

- Gute Konzepte für Rechtevergabe und Datenzugriff
- Jedoch: Alle Anwendungen haben potenziell Zugriff auf alle Dienste des Geräts
- Nach erfolgreicher Installation einer Anwendung kann diese z.B. Daten empfangen und versenden
- Anwendung kann sich auf bestimmte Ereignisse registrieren (z.B. Tastendruck, Anruf etc.) und Daten weiterverarbeiten
- Architektur birgt viele Risiken für unerfahrene Benutzer

Inhalt

Ausblick

- Ausgewählte Sicherheitsaspekte müssen auf konkrete Anwendungsbereiche projeziert werden
- z.B. Sicherheit von Spielen für mobile Geräte untersuchen
  - Großer Markt - Es kann viel Geld verdient werden
  - Spiele werden sich schnell verbreiten
  - Wie müssen Spiele konzipiert werden, damit diese sicher sind

Quellen

- Google Android: http://code.google.com/android/
- Open Handset Alliance: http://www.openhandsetalliance.com/
- Vortrag von Rich Miner beim Computer Systems Colloquium der Stanford University, Herbst 2007: http://deimos3.apple.com/WebObjects/Core.woa/Browse/itunes.stanford.edu.1294764018.01488377969.1489155906?i=1822422778
- Brodkin, Jon: Google: Android won't suffer from incompatibility. In: Network World vom 06.03.2008

Vielen Dank für die Aufmerksamkeit!
Fragen?