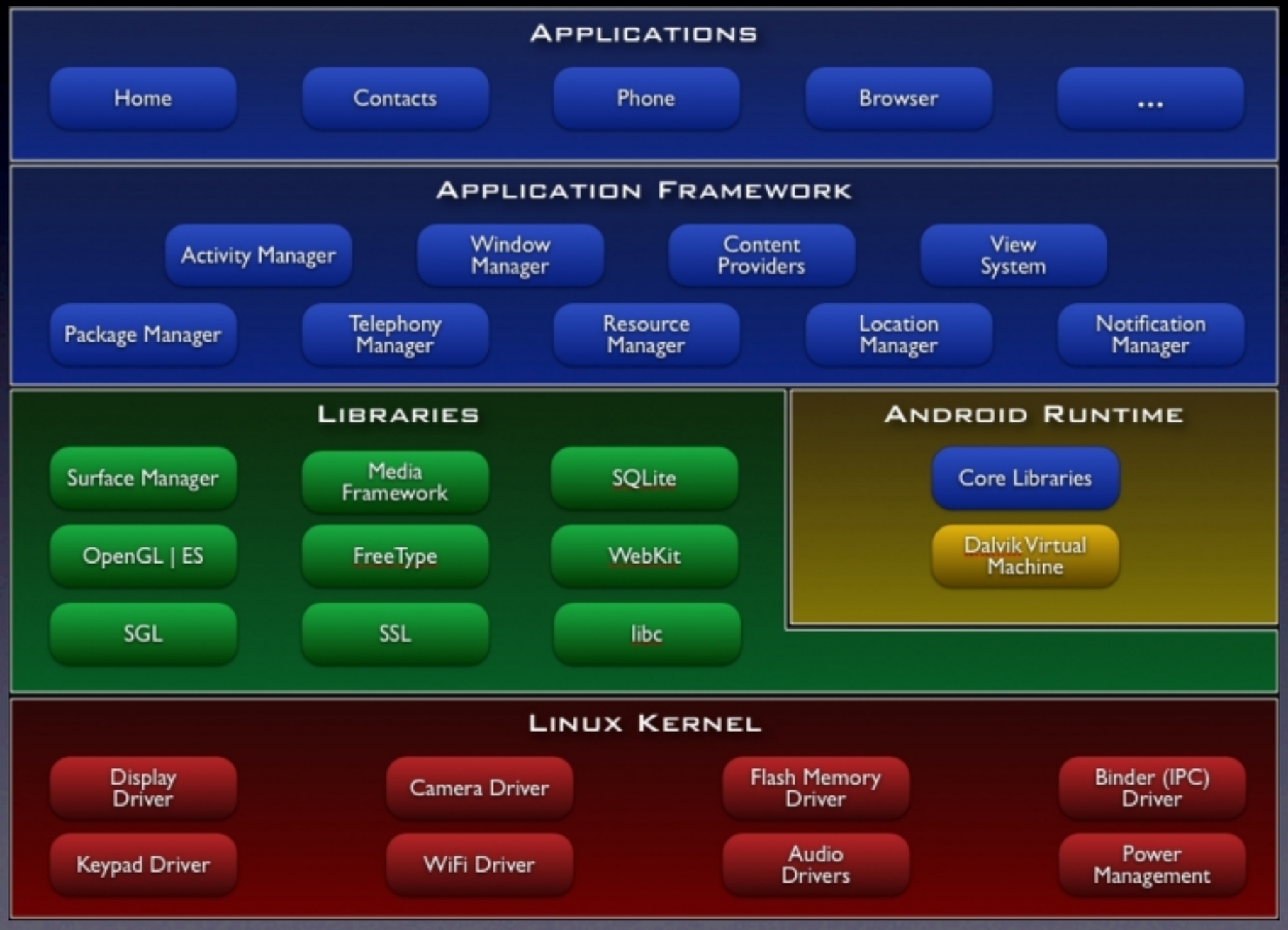
Android Security

Was ist Android

- Betriebssystem (Kernel)
- Middleware
- Kern-Anwendungen
- Android Runtime

Android Architektur

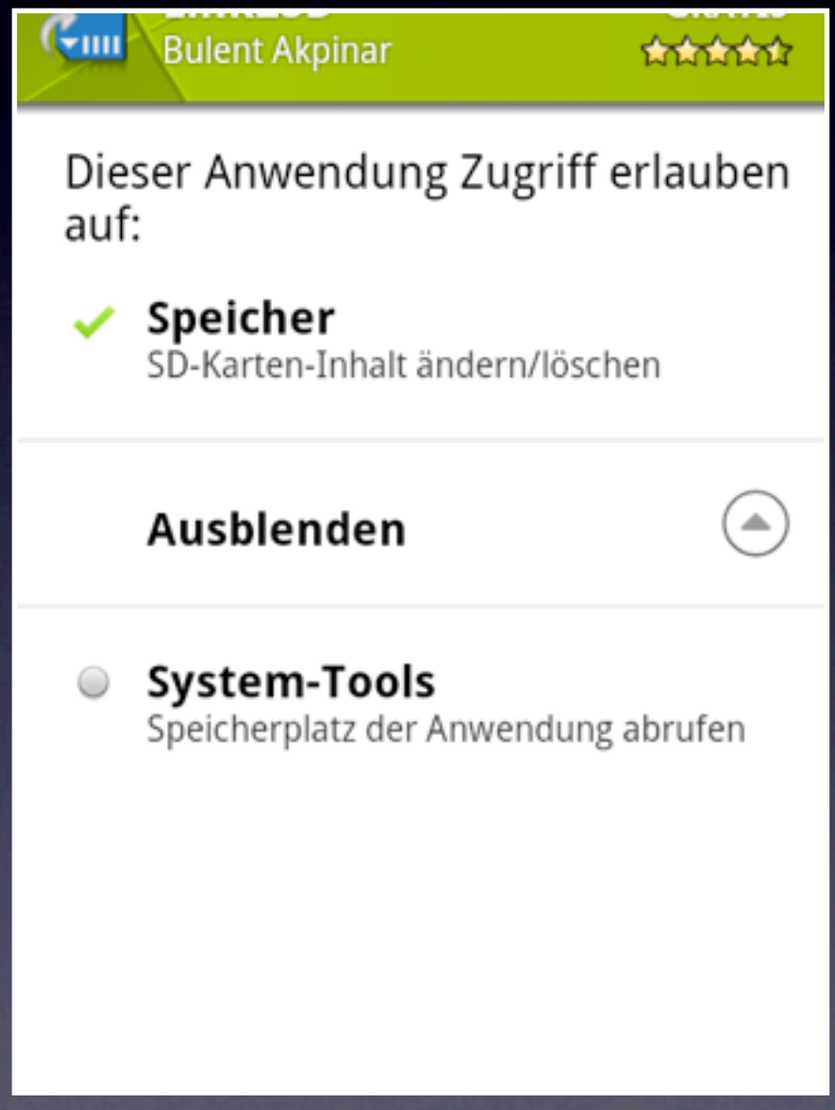


Sicherheitsfeatures

- Jede Anwendung läuft in eigener DalvikVM
- Isolation der Prozesse durch Kernel
- Kommunikation der Prozesse via ContentProvider oder rpc

Rechtevergabe in Android

- Festlegen gewährter Rechte zur Installationszeit
- Gewährung aller Rechte oder keine Installation der Software
- Unterteilung in “kritische” und “erweiterte” Rechte
- Keine erneute Nachfrage zur Laufzeit

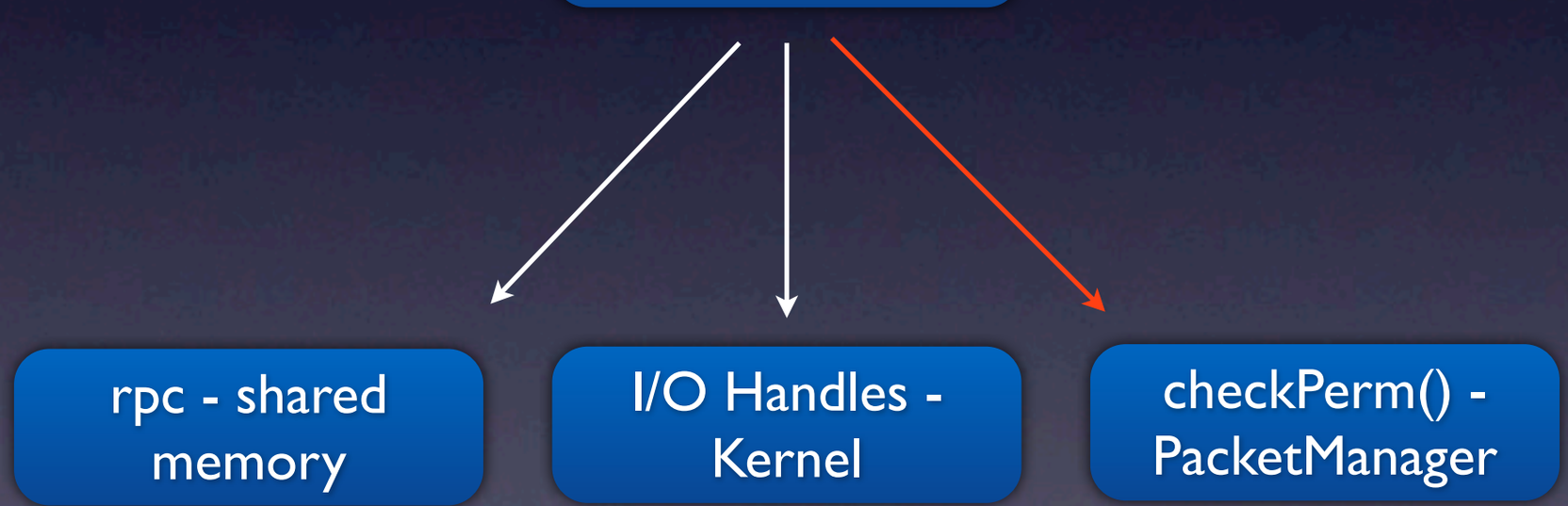


Rechtedefinition in Android

- Android Manifest spezifiziert Zugriff auf Dienste (SMS, Telefon, etc.)
- Zugriff auf Kamera, Bluetooth, etc. realisiert via GID-Zuordnung
- I/O-Zugriffe geprüft im Kernel

Rechteüberprüfung zur Laufzeit

READ_SMS



Umgehen des Rechtesystem

Software gleicher Hersteller

- Software eines Herstellers kann miteinander Daten austauschen
- Aufteilen von “kritischen” Zugriffsrechten
  - Sammeln und Speichern von Daten (Zugriff auf Telefonbuch)
  - Versenden von Daten (Zugriff auf Netzwerkkommunikation)

Intents

- RPC zwischen Applikationen
- Rechte/Methoden anderer Applikationen nutzen
- Mögliches Umgehen von Rechteüberprüfungen
- Browser für Internetzugriff instrumentieren

```
Intent myIntent =
    new Intent(Intent.ACTION_VIEW,
    Uri.parse("http://www.google.com"));
startActivity(myIntent);
```

BroadcastReceiver

- System-Events verteilen
- Ausnutzen des Flags `android:priority`
- Kontrolle über Informationen übernehmen

```
<application>
    ...
    <!-- definition of
        various application
        parameters //-->
    <uses-permission
        android:name=
        "android.permission.RECEIVE_SMS"
    />
    ...
</application>
```

BroadcastReceiver SMS

- Beispiel: SMS
- Keine Benachrichtigung an den Benutzer
- Speichern in Datenbank kann verhindert werden

```
//---get the SMS message passed in---
Bundle bundle = intent.getExtras();
SmsMessage[] msgs = null;
if (bundle != null)
{
    //---retrieve the SMS message
    //   and process---
    Object[] pdus =
        (Object[])bundle.get("pdus");
    msgs = new SmsMessage[pdus.length];
    for (int i=0; i<msgs.length; i++){
        msgs[i] =
            SmsMessage.createFromPdu(
                (byte[])pdus[i]);
    }
}
this.abortBroadcast();
```

Broadcasts & Intents

- Unberechtigter Zugriff auf sensitive Informationen
- Senden der Informationen an entfernte Server
- Senden eigener Informationen / Befehle an Geräte

ContentProvider im Browser

- Aufruf von ContentProvidern direkt aus Adresszeile

- Beispiel: `htmlfileprovider`

```
content://com.android.htmlfileprovider/
    sdcard/info.pdf
```

- Zugriff auf Inhalte des Geräts

Informationen extrahieren

- Bösartig JavaScript Code in Website
- Senden der Information via HTTP-Posts
- Verarbeitung durch serverseitiges Script oder Extraktion aus Server-Logs
- Nur explizites Öffnen von Dateien möglich (kein Verzeichnis listing)

Zusammenfassung

- Rechtesystem nicht konsequent umgesetzt
- Keine Überprüfung des Aufrufs von ContentProvidern und Intents
- Browserupdates nur via Systemupdates

Lösungsansätze

- Rechte zur Installation einzeln vergeben
- Zugriff / Inhalt von ContentProvider kontrollieren
- Update kritischer Softwarekomponenten von Systemupdates entkoppeln

Vielen Dank für die Aufmerksamkeit!

Fragen?