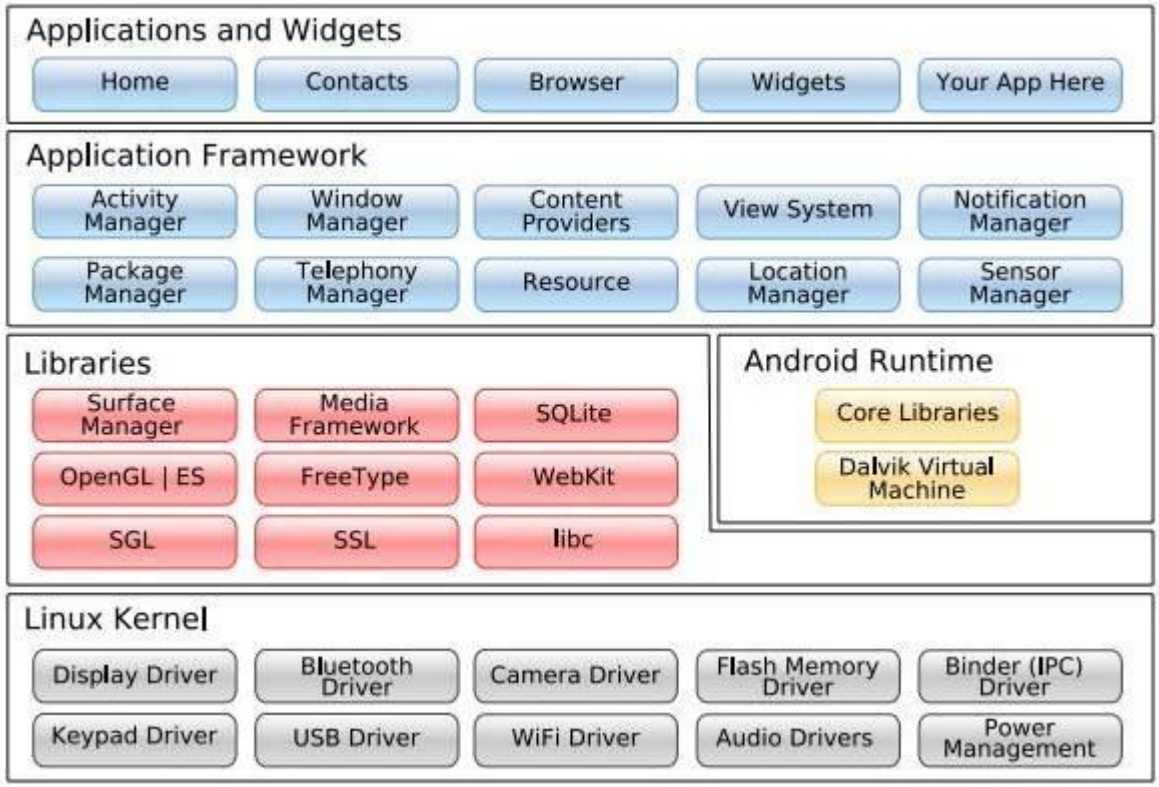
Andraoid platforma

Android arhitektura

- Android je razvijen prvenstveno za procesorsku arhitekturu **ARM**, ali otvoreni kod dozvoljava implementaciju i na druge platforme (Android i86);
- **ARM arhitektura** , **Advanced RISC Machine** i **Acorn RISC Machine** 32-bitna arhitektura razvijena od strane britanske firme ARM holding , najrasprostranjenija familija procesora u svetu. Zbog svoje specifične arhitekture, ARM licencirani čipovi imaju izuzetno malu potrošnju energije za razliku od drugih setova arhitekture, pa su kao takvi najbolje rešenje za mobilne uređaje;
- **ARM Holding** ne proizvodi čipove, već svoje Licence daje proizvođačima poluprovodnika koji koristeći setove instrukcija dodaju svoja sopstvena rešenja, gde trenutno u proizvodnji prednjače Quallcom, Apple (Ax), i Exynos (Samsung), Texas instruments, Nvidia …

Android Software Stack

- Linux kernel i kolekcija c/c++ biblioteka izloženih kroz application framework

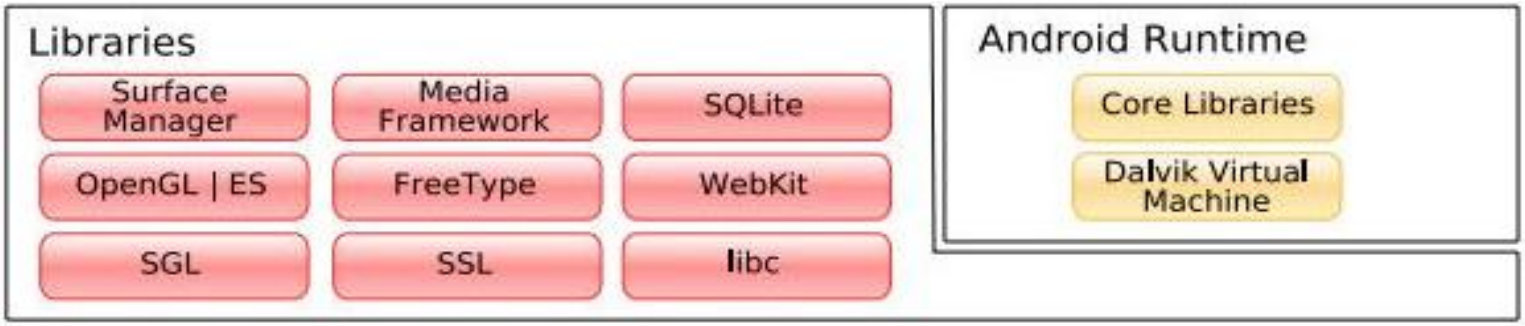


Linux Kernel

- Drajveri uređaja definisani za različite hardverske komponente
  - driver za međuprocesnu komunikaciju (IPC - Inter-process communication) koji služi za razmenu podataka između različitih procesa ili niti unutar istog procesa
  - driver za upravljanje napajanjem (Power Managment).

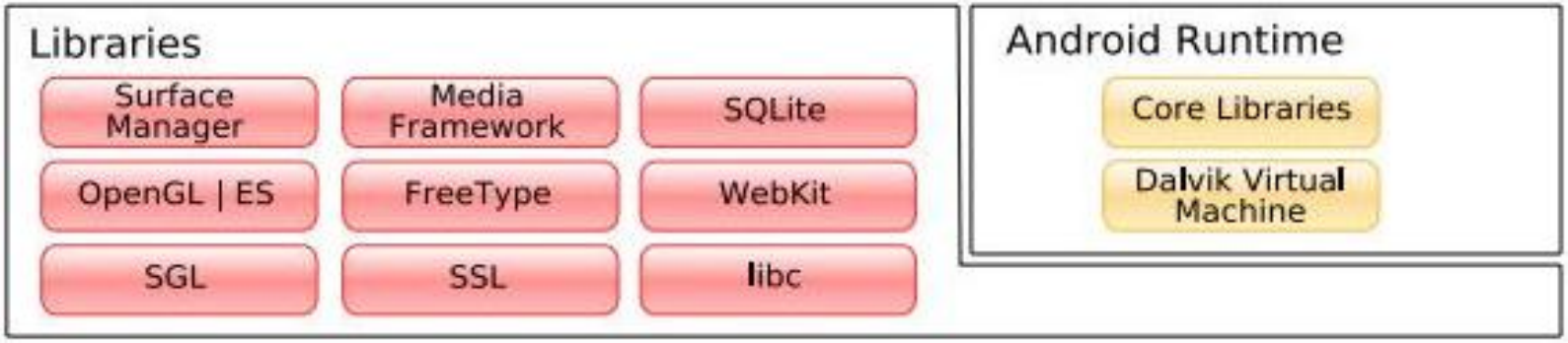
Native Libraries

- biblioteke pisane u C/C++ programskom jeziku, kod koji obezbeđuje osnovne funkcije Android operativnog sistema
  - Surface Manager – nadzire iscrtavanje grafičkog interfejsa
  - OpenGL | ES (Open Graphic Library) – API za 2D i 3D grafiku
  - Media Framework – podržava snimanje i reprodukciju poznatih audio/video formata
  - SQLite – podrška za korišćenje baza podataka
  - WebKit – native web browser engine (i u drugim browser-ima Google Chrome, Safari)



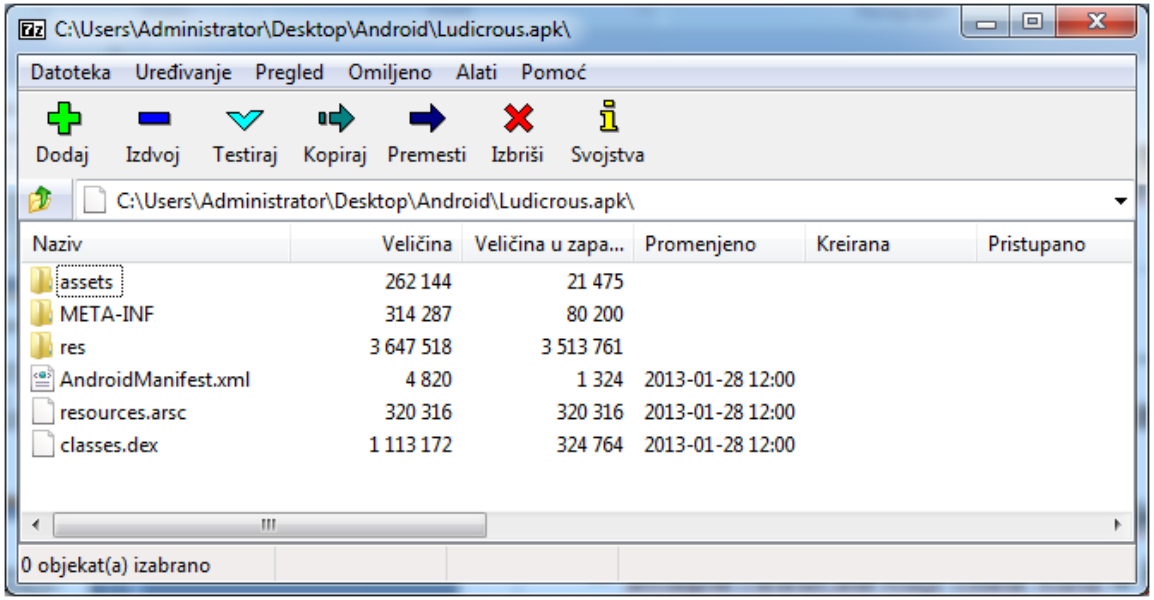
Android Runtime

- Core libraries – skup osnovnih biblioteka programskog jezika Java. Omogućavaju pisanje aplikacija u Java programskom jeziku
- Dalvik Virtual Machine (Dan Bornstein i Google tim) – virtuelna mašina projektovana za Android i optimizovana za mobilne uređaje koji koriste bateriju pri radu i imaju ograničenja memorijskih resursa i brzine CPU
- Pokreće aplikacije kao posebne procese odnosno kao instance virtualnog uređaja, svaki proces na posebnoj Dalvik VM

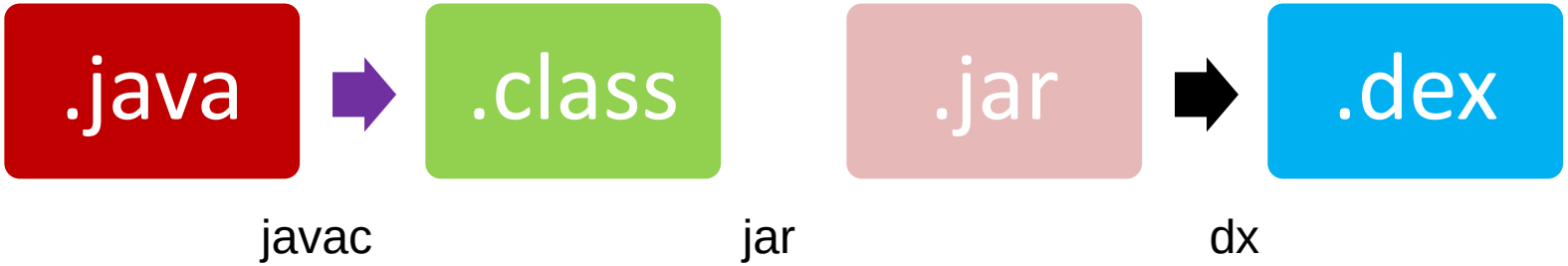


Struktura apk file-a

- apk fajl (otvara 7-ZIP) koji se sastoji od :
  - egzekutabilnog koda za Dalvik VM (.dex)
  - skupa resursa koji nisu kod, a koje koristi program (ikonice, slike, xml fajlovi, tekst u aplikaciji...)
  - assets
  - certificate
  - manifest fajl



Kompajliranje



- Java source code se prvo kompajlira u Java bytecode (.class);
- Java bytecode (.jar) se zatim kompajlira u Dalvik bytecode (.dex), koristeći dx tool koji je deo Android SDK;
- **Dalvik JIT (Just In Time)** kompajler uveden u verziji Android 2.2 što je ubrzalo izvršenje koda za 2-5 puta. Kompajlira runtime bytecode u mašinski kod.

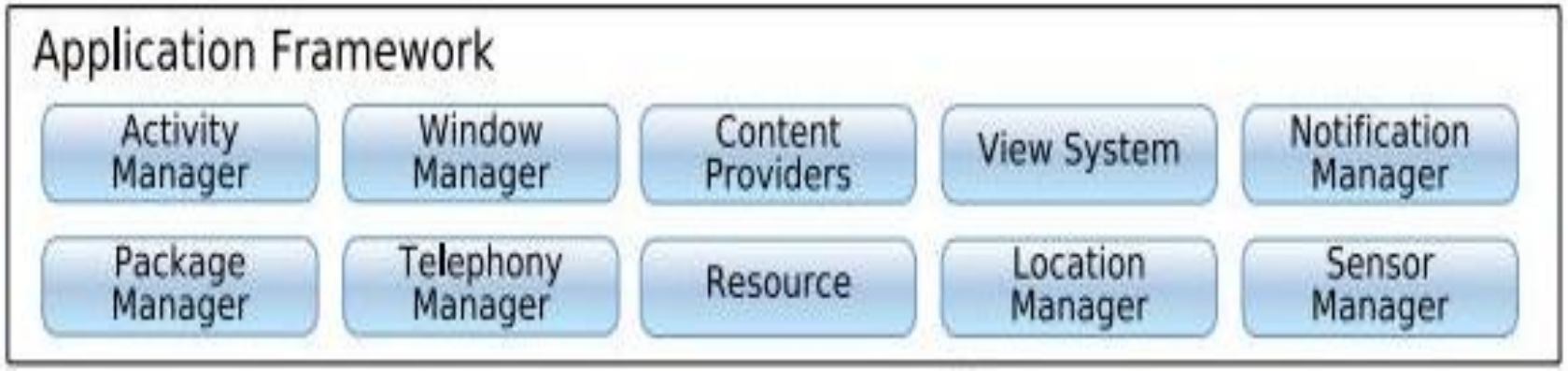
Neke karakteristike

- Register based VM (brže), minimalna potrošnja memorije;
- Memorija troši bateriju kad se u nju upisuje, čita i briše;
  Svaka aplikacija radi u svojoj posebnoj Dalvik virtuelnoj memoriji;
- Izlazak iz programa (dugme back), ostavlja program još neko vreme u keš memoriji;
- **OOM (**Out Of Memory) **killer** -oslobađanje memorije u slučaju kad neki od procesa javi da mu nedostaje memorije. Koristi poseban heuristički algoritam da odredi koji proces minimalno utiče na preformanse **(Foreground programi , Visibile, Secondary Server** , **Hidden applications** , **Content Providers , Cached applications).**

Application Framework

Skup sistemskih servisa Java API (Application Programming Interface)

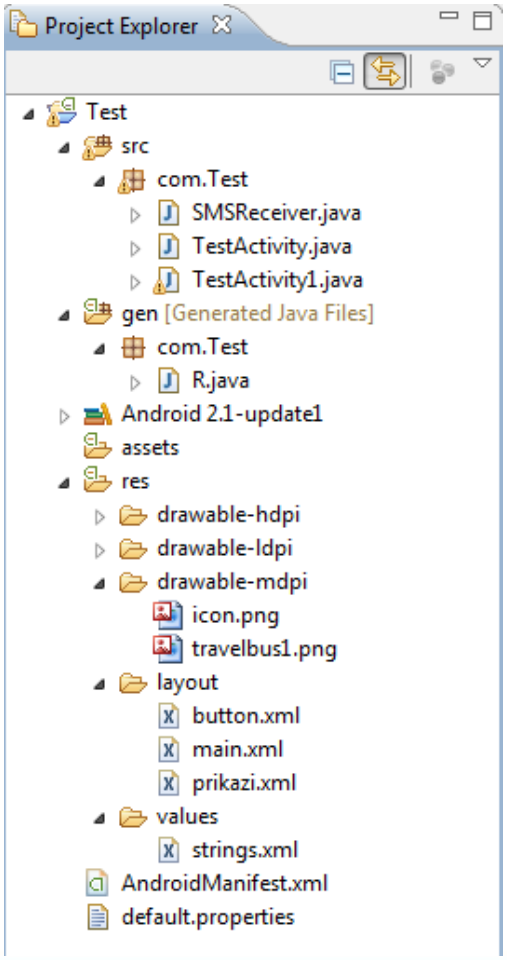
- Prezentacija usluga aplikativnom sloju;
- Upravljanje programskim paketima, aktivnostima aplikacije (životni ciklus aplikacije), pozivima, prozorima, resursima, omogućava aplikacijama da dele resurse, lokacijski servisi…



Applications and Widgets

- Aplikacije vidljive krajnjem korisniku
  - sastoji se kako od osnovnih, ugrađenih aplikacija poput e-mail klijenta, SMS programa, kalendara, web browser-a pa sve do third-party aplikacija koje se preuzimaju sa Google Play marketa;

Struktura aplikacije



- Src - source fajlovi (java fajlovi)
- gen – R.java fajl koji predstavlja vezu java sveta i resursa
- res – resursi (više poddirektorijuma)
- drawable (ikone, slike, animacije)
- layout (grafički prikaz ekrana, statički kreiran)
- values (strings, color, dimension, drawable…)
- AndroidManifest – metapodaci o svim klasama, dozvolama, podržanoj verziji platforme

Struktura aplikacije

- **Src** i **Gen** direktorijumi pripadaju Java svetu;
- **Res** direktorijum pripada svetu resursa;
- **AndroidManifest.xml** predstavlja metapodatke o celom programu, dakle podatke o svim klasama, parametrima, resursima, permisionima;
- **R.java** predstavlja **lepak** između Java sveta i sveta resursa... automatski se generiše i update-uje prilikom svake promene u resursima, pomaže java kodu da pronađe i referencira željeni resurs;
- **main.xml** predstavlja vizuelni prikaz za aktivnost/klasu main;

Struktura aplikacije

- **Src direktorijum**
- U **Src** direktorijumu se nalaze source fajlovi odnosno java fajlovi od kojih se sastoji aplikacija;
- Prilikom kompajliranja svaki .java fajl se kompajlira u .class fajl koji se smešta dublje u strukturu bin direktorijuma tog projekta u workspace-u;
- **Gen direktorijum**
- U **Gen** direktorijumu se nalazi R.java fajl koji predstavlja vezu src direktorijuma sa res direktorijumom, odnosno java sveta sa svetom resursa;
- Ovaj fajl ne bi trebalo menjati ručno.

Struktura aplikacije

- **Res direktorijum**
- Ima vise poddirektorijuma
- **Drawable** direktorijum sadrži slike (ikone,slike i neke animacije);
- Za više rezolucija ekrana priprema se više verzija slika koje se smeštaju u različite direktorijume označene na tačno određeni specifičan način. Frame animacije se nalaze takođe ovde.
- Pristup preko **R.drawable** klase
- **Layout** direktorijum sadrži grafički prikaz ekrana koji je statički kreiran;
- U formi xml-a je : **nesto.xml;**
- layout se može i dinamički kreirati u runtime-u i tada se ne nalazi kao poseban fajl na ovom mestu;
- Pristup preko **R.layout** klase

Struktura aplikacije

- **Res direktorijum**
- **Values** poddirektorijum sadrži xml fajlove u kojima su date specifikacije različitih tipova resursa koje su specificirane unapred;
- **Strings.xml** je fajl u kome su navedene vrednosti svih resursa tipa string koji se javljaju u kodu;
- Mogući posebni fajlovi s parametrima koji se primenjuju u zavisnosti od različitih parametara npr. jezika, zone, doba dana i sl.
- Omogućena laku lokalizacija aplikacija;
- Pristup preko **R.string** klase;
- **nesto.xml** je fajl gde su definisani izgled i format za UI parametre;
- Pristup preko **R.style** klase;

Struktura aplikacije

- **Res direktorijum**
- U V**alues** poddirektorijumu specificiraju se i resursi tipa **boolean, integer, dimensions...**

- **Color** poddirektorijum
- U **color.xml** fajlu cuvaju se podaci o bojama;
- Pristup preko **R.color** klase;

- **Menu** poddirektorijum
- U **nesto.xml** fajlu čuvaju se podaci o menijima;
- Pristup preko **R.menu** klase;

Struktura aplikacije

AndroidManifest.xml

- Opisuje ceo projekat u smislu da su tu dati metapodaci o svim klasama, dozvolama, podržanoj verziji Android platforme i još mnogo toga...
- Jedan <manifest>
- Jedan <application> tag
- Više <activity> tagova
- Više tagova za različite posebne dozvole...

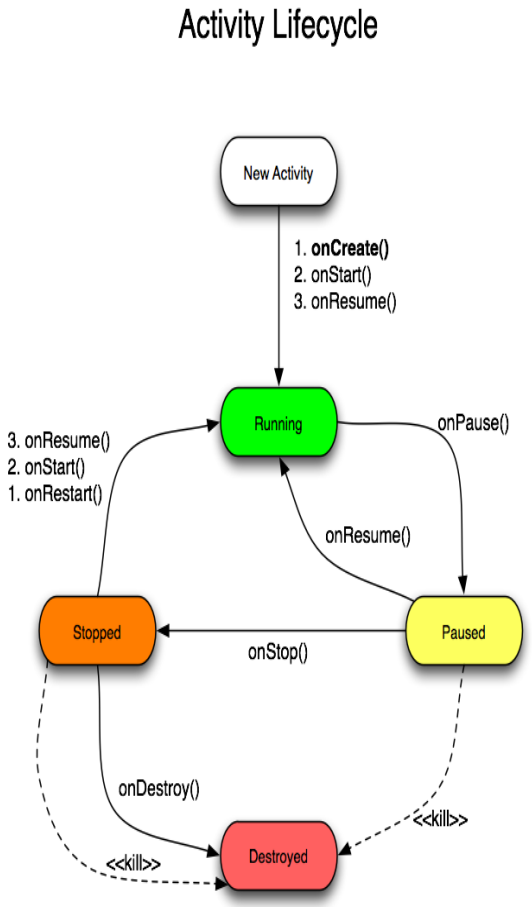
Osnovne kockice

- Activities - “aktivnosti”
- Intents - “namere”
- Services - “servisi”
- Content Providers - “skladišta”
- Broadcast Receivers - “primaoci poruka”

Osnovne kockice

Activities – aktivnosti

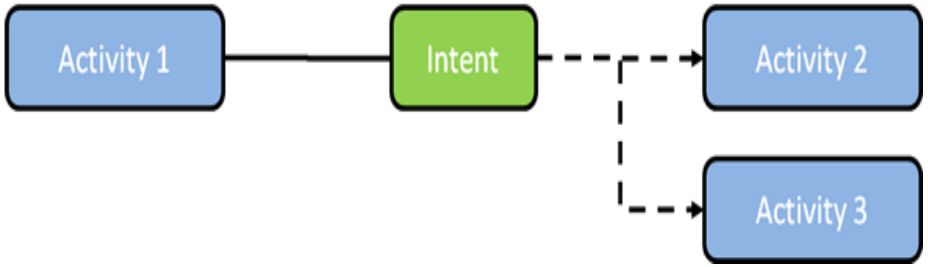
- Predstavljaju ekran odnosno prozor u windows terminologiji;
- Jedna aplikacija može imati više activity-ja;
- Njihov životni ciklus je striktno određen OS-om (running, stopped, paused, destroyed);
- Android OS menja stanja u kojima se aktivity nalazi dok mi obezbeđujemo ostalo;



Osnovne kockice

***Intents* - "namere"**

- Poruka između dva dela sistema, **task;**
- Možemo poslati i podatke drugom activity-ju ili programu (**bundle** objekat mehanizam; poput URL-a);
- Asinhroni mehanizam;
- Implicitni intenti – kažemo sta hoćemo da uradimo, ali ne i ko treba da opsluži nas zahtev;
- Eksplicitni intenti – kažemo sta hoćemo da uradimo, ali i ko treba da opsluži nas zahtev;



Osnovne kockice

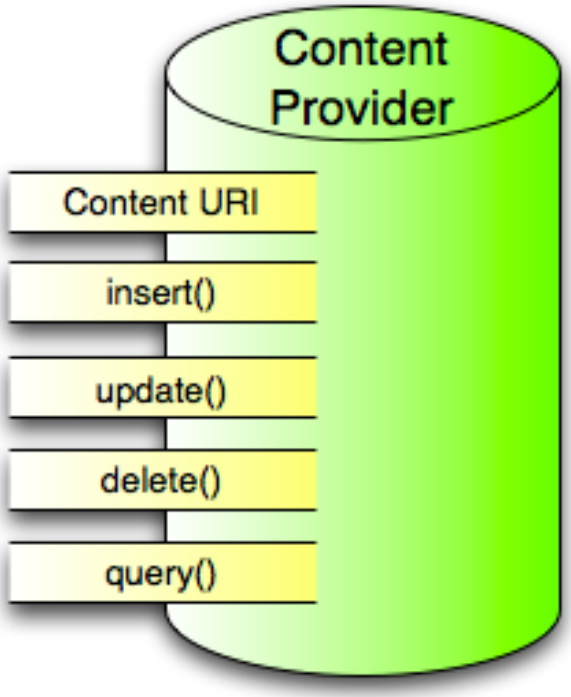
***Services* - "servisi"**

- Kod koji je aktivan u background-u;
- Nemaju UI;
- Imaju lifecycle jednostavniji od activity-ja;
- Bound servisi– lifecycle je isti kao i kod njegovog pozivaoca;
- Unbound servisi– lifecycle odvojen od pozivaoca;

Osnovne kockice

Content Providers - "skladišta"

- Čuvaju u sebi sadržaje/podatke koje međusobno deli više aplikacija;
- API jako sličan DB API-ju;
- Različite implementacije: DB, file system, memory, cloud...
- primeri
  - Contacts,
  - MediaStore,
  - Settings.



Osnovne kockice

Broadcast Receivers - "primaoci poruka"

- Prihvataju i hendluju Broadcast Intent-e;
- Receiver se registruje na sistemu da prihvata neki Intent i dobija obaveštenja da se neki intent pojavio.