
Android Internals

- De ce bazat pe Linux?
 - Drivere
 - Rețea, servicii
 - Gestiune procese, memorie
- Imens
- Se schimbă repede
- Schimbări imprevizibile
 - Schimbări interne, dar API relativ stabile

Au fost adăugate

- Biblioteci C proprii (bionic)
- Mediu de rulare JAVA (dalvik, ART)
- Framework aplicații
- Binder, ashmem
- **Gestiune consum**
 - wakelocks

“Click” →



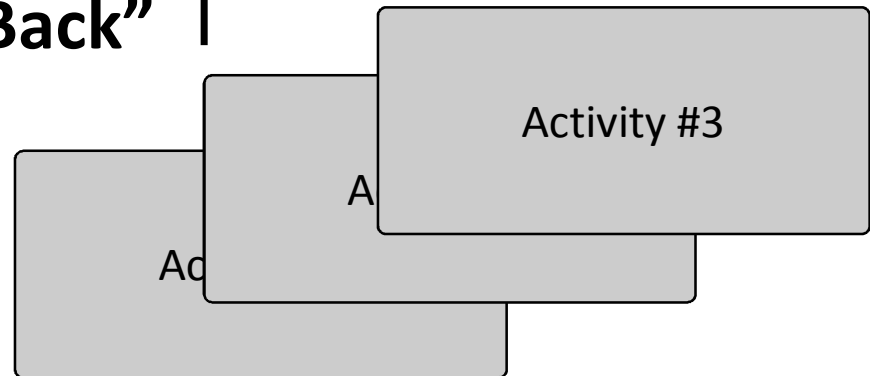
“Back” ↑

“Click” ↓

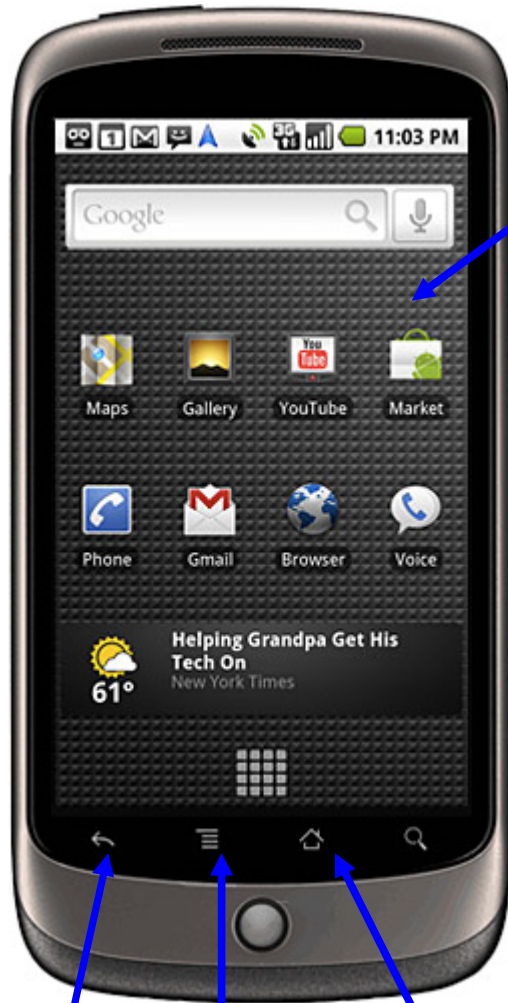


“Back” ↑

“Click” ↓



Activity



“Home” ↖

Back

Menu

Home

- Filozofie diferită de aplicații desktop (Unix, Windows)
- Fără punct de intrare unic... fără main() !?
- Procesele și aplicațiile terminate aleator
 - Developerul trebuie să țină cont
- UI deconectat de “creierul” aplicației
- Aplicațiile sunt foarte izolate
- **Comportarea controlată de**
 - **Memorie redusă**
 - **Baterie redusă**

Concepte Linux



- Processes (fork() și prietenii)
- Signals (kill() ... or be killed)
- Sockets / Pipes / Fifos / SysV IPC
- Hardware devices ca fișiere(/dev)
- Daemons
- Shell / scripts
- Useri (root vs. ceilalți-- # vs. \$)
- ELF files
- GNU toolchain
- ... 40 de ani de Unix

Concepte Android



- Components
- Intents
- Manifest file
- Component lifecycle
- Processes and threads
- Remote procedure calls
- Permissions
- Storage
- Native development

Componente



- 1 Aplicație = N Componente
- aplicațiile pot partaja componente
- Procesele aplicațiilor sunt pornite automat când o componentă e apelată
- **N puncte de intrare, !1, !main()**
- Componente:
 - Activities
 - Services
 - Broadcast Receivers
 - Content Providers

Intents



- Intent = mesaj asincron cu/fără target
- Ca un signal polimorfic de Unix signal, fără target
- Intent Filters specificate în Manifest file

Manifest file



- Informează sistemul despre componentele aplicației
- Numit mereu AndroidManifest.xml
- Activity = <activity> ... static
- Service = <service> ... static
- Broadcast Receiver:
 - Static = <receiver>
 - Dynamic = Context.registerReceiver()
- Content Provider = <provider> ... static

Ciclul de viață al componentelor



- .Sistem pornește/oprește/ompoară procesele automat
- .Toată comportarea sistemului ține cont de**
 - .Memorie redusă**
 - .Baterie redusă**
- .Sistemul
 - .Apelează callback-uricând e necesar
 - .Gestionează ciclul de viață
- .Unele comopnente sunt mai complex de gestionat

□ Procese

- default: \forall callback către \forall componentă \rightarrow main thread
 - nu se fac operații blocante în thread-ul principal:
 - threaduri suplimentare
 - terminare/restartare proces – la discreția sistemului
- Sistemul gestionează ciclul de viață**

□ Threaduri

- create cu obiectul Java Thread
- nu GUI în thread-ul de networking
- nu networking în thread-ul de GUI

Remote procedure calls



- .Nu există IPC de UNIX (semafoare , cozi de mesaje, pipe)
 - .Dificilă portarea programelor UNIX
- .Android RPC = Binder
- .Binder in kernelul de Linux 3.19 (2015)

- .Securitate la nivel de proces: UID, GID
- .Permisii implementează restricții la:
 - .Operații per-process
 - .Acces per-URI
- .Aplicațiile rulate în sandbox
- .Permisii speciale pentru a ieși din sandbox
- .Se dă accesul bazat pe:
 - .Certificate
 - .User prompt
- .Toate permisiunile declarate static



Stock Android Apps

Launcher2	Phone	AlarmClock
Email	Settings	Camera
Gallery	Mms	DeskClock
Calendar	Browser	Bluetooth
Calculator	Contacts	...

Your Apps / Market Apps

App
API

android.*

Binder

System Services

Power Manager	Mount Service	Status Bar Manager
Activity Manager	Notification Manager	Sensor Service
Package Manager	Location Manager	Window Manager
Battery Service	Surface Flinger	...

**java.*
(Apache Harmony)**

Zygote (ART, libandroid_runtime.so, Preloaded Classes)

JNI

Libraries
Bionic / zlib / OpenSSL / ...

**Hardware
Abstraction Layer**

Native Daemons

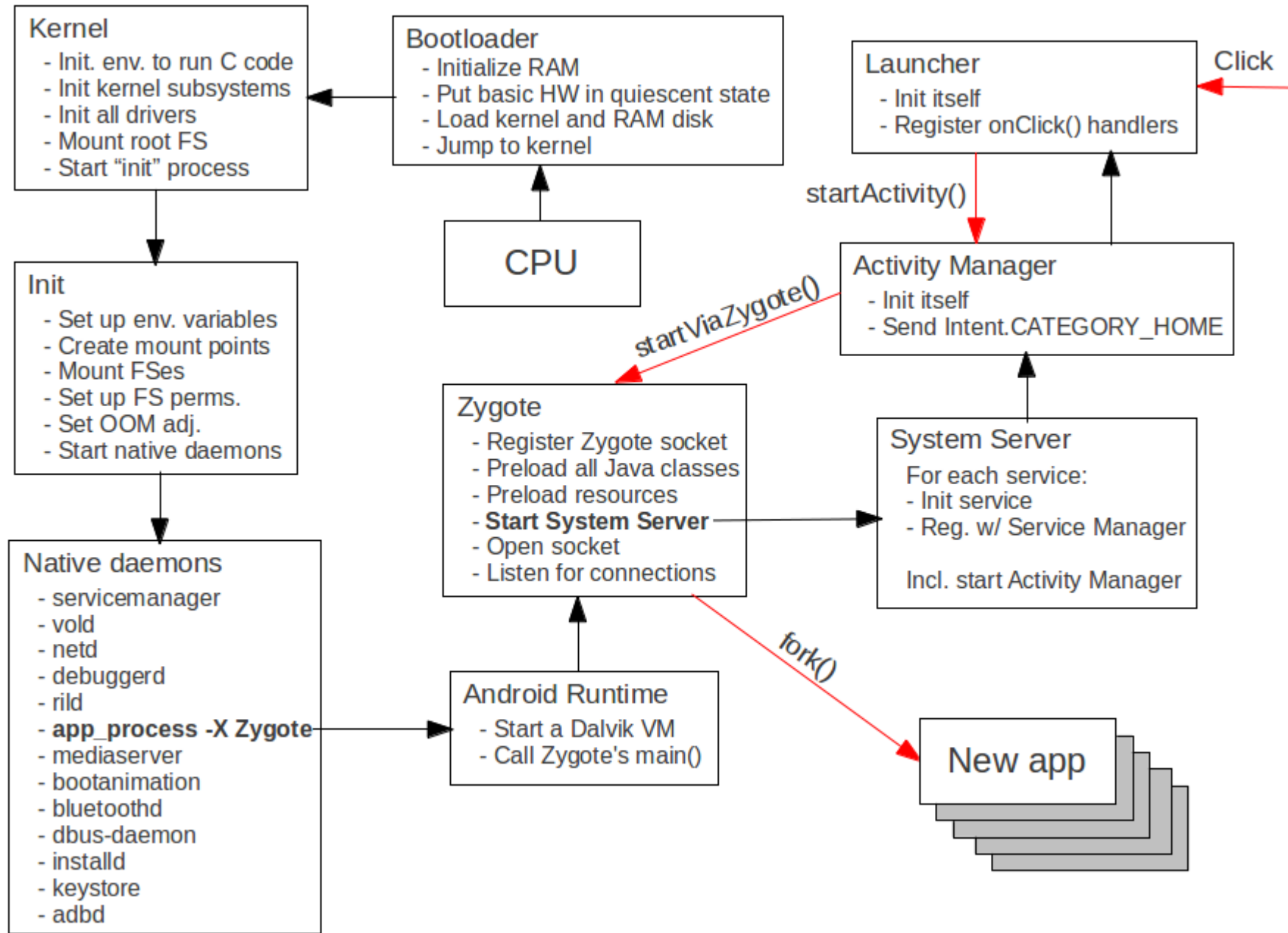
Init / Toolbox

Linux Kernel
Wakelocks / Lowmem / Binder / Ashmem / ION / ...

Secvența de startup



- Bootloader
- Kernel
- Init
- Zygote
- System Server
- Activity Manager
- Launcher (Home)



Zygote



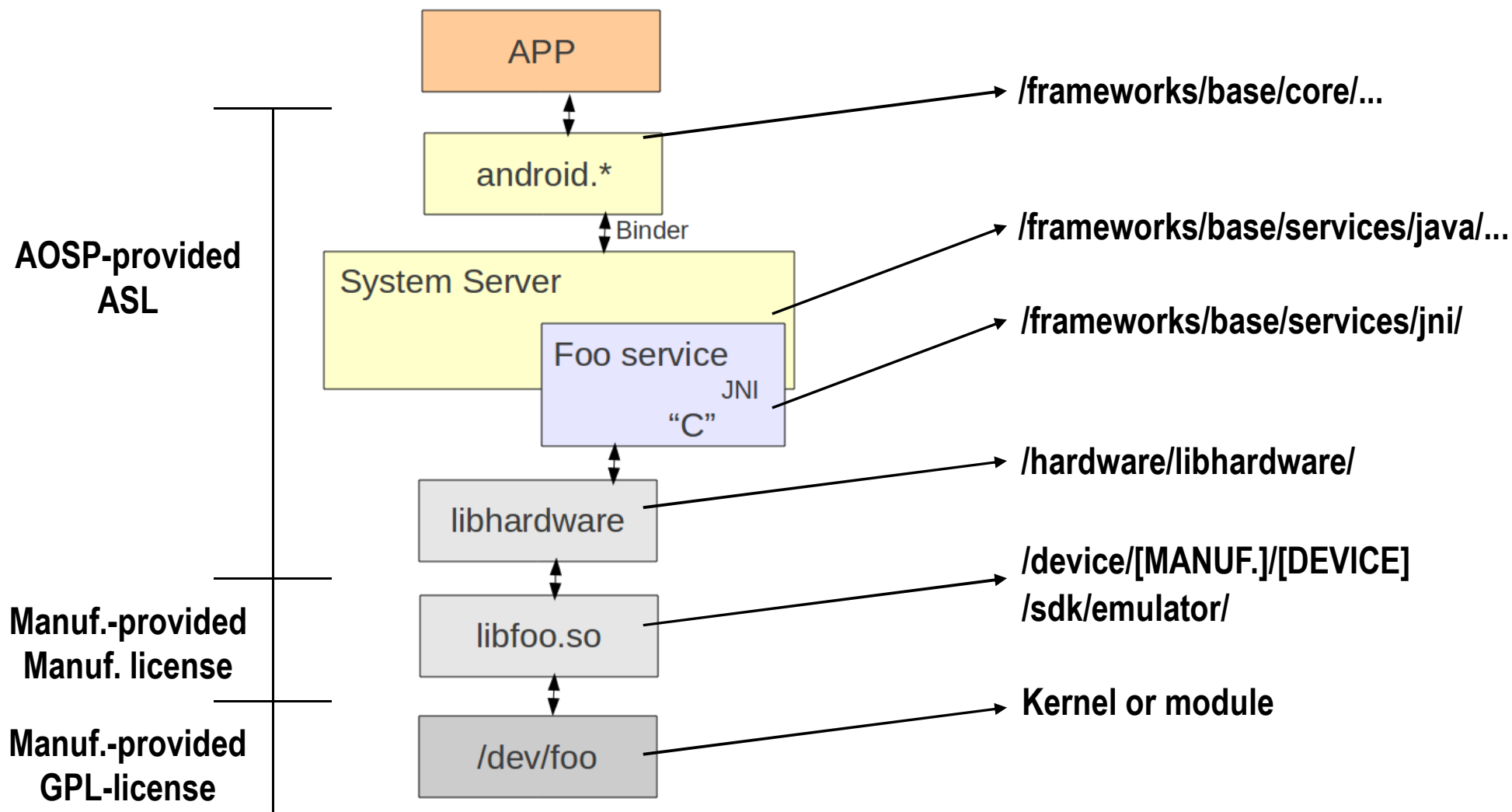
- Pornește după startup-ul tipic de Linux
- Are toate bibliotecile încărcate
- Conține mașina virtuală Java
 - Dalvik < 4
 - ART (Android Runtime) ≥ 4
- Fiecare proces nou este fork de zygote

Hardware Support



- hardware not accessed directly
 - HAL = hardware abstraction layer
 - HAL “modules” are .so files
-
- Activity Recognition
 - Audio
 - Bluetooth
 - Camera
 - ConsumerIr
 - Framebuffer
 - Fingerprint
 - Fused Location
 - GPS
 - Gralloc
 - HWcomposer
 - Keymaster
 - Lights
 - NFS
 - Power
 - Sensors

Hardware Abstraction Layer



- Principale:
 - /data => User data
 - /system => System components
 - /cache => Cache (& OTA update)
- Sistem Linux
 - /dev
 - /proc
 - /sys
 - /sbin
 - /mnt
- Multedirectoare Linux lipsesc...

Java din Android



- Oracle (Sun)
 - Java = Java language + JVM + JDK libs

- Android < 5 (Lollipop)
 - Java = Java language + Dalvik

- Android < 7 (Nougat)
 - Java = Java language + ART + Apache Harmony

- Android >= 7
 - Java = Java language + ART + OpenJDK

Mașina virtuală java



- Aplicațiile Android rulează propriul proces
 - Mașina virtuală java
- ART = Android RunTime
 - 64 bit
 - Multi-core
 - AOT instead of JIT
- Dalvik

.Sistemul Android

. ~100 servicii (Lollipop)

. 5 - 6 servicii noi la fiecare release

. adb shell "service list" pentru listare

- WifiService
- ActivityManagerService
- PowerService
- PackageManagerService
- LocationManagerService

System Services

System Server

Java-built Services

Power Manager	Mount Service
Activity Manager	Notification Manager
Package Manager	Location Manager
Battery Service	Search Service
Window Manager	Wallpaper Service
Status Bar	Headset Observer
Clipboard Service	...

C-built Services

Sensor Service

Surface Flinger

Media Service

Audio Flinger
Media Player Service
Camera Service
Audio Policy Service

Includes:

- StageFright
- Audio effects
- DRM framework

Phone App

JNI

Native Methods for
Java-built Services

Hardware Abstraction Layer

ActivityManager



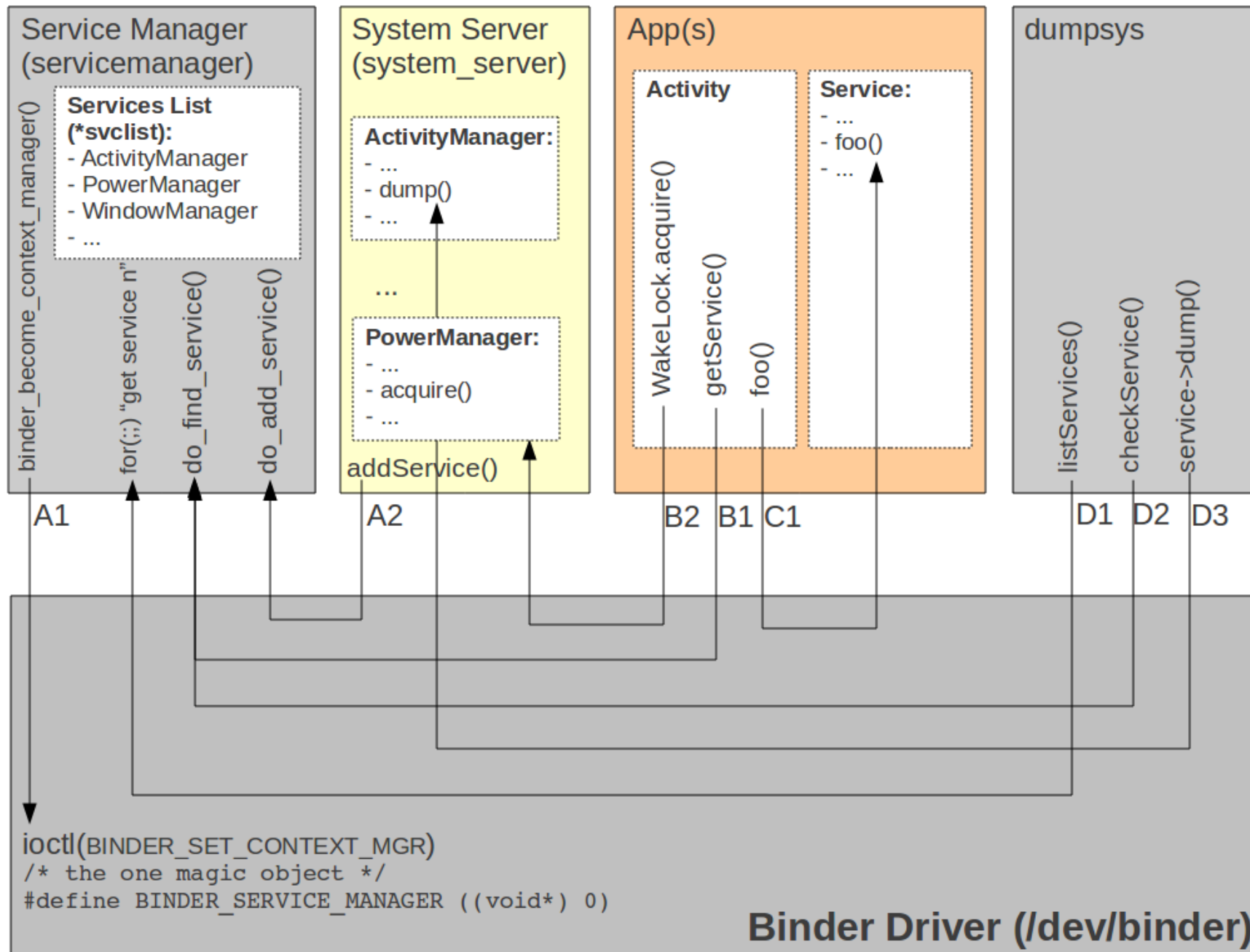
- Pornește Activities, Services
- “găsește” Content Providers
- Intent broadcasting
- Application Not Responding
- Permișiuni
- Task management
- Lifecycle management

- **Exemplu:**
- Startăm app din Launcher:
- `onClick(Launcher)`
- `startActivity(Activity.java)`
- *<Binder>*
- `ActivityManagerService`
- `startViaZygote(Process.java)`
- *<Socket>*
- `Zygote`

Binder



- Datele trimise prin “parcels”, folosind “transactions”
- Implementat în kernel
- /dev/binder
- /sys/kernel/debug/binder/*
- android.* API se conectează la System Server prin binder



Gestiunea bateriei



- Bazat pe “Suspend to RAM” din Linux
- modulul PowerManager
 - Display, backlight, keyboard
 - Monitorizează bateria
 - Coordonează circuitul de încărcare
 - Necesită permisiunea WAKE_LOCK *
- Kernel Wake Locks: wlan_wake, alarm_rtc, suspend_backoff
- Partial Wake Locks

Flag	CPU	Screen	Keyboard
PARTIAL_WAKE_LOCK	On	Off	Off
SCREEN_DIM_WAKE_LOCK	On	Dim	Off
SCREEN_BRIGHT_WAKE_LOCK	On	Bright	Off
FULL_WAKE_LOCK	On	Bright	Bright

* <https://forum.xda-developers.com/showthread.php?t=1827676>