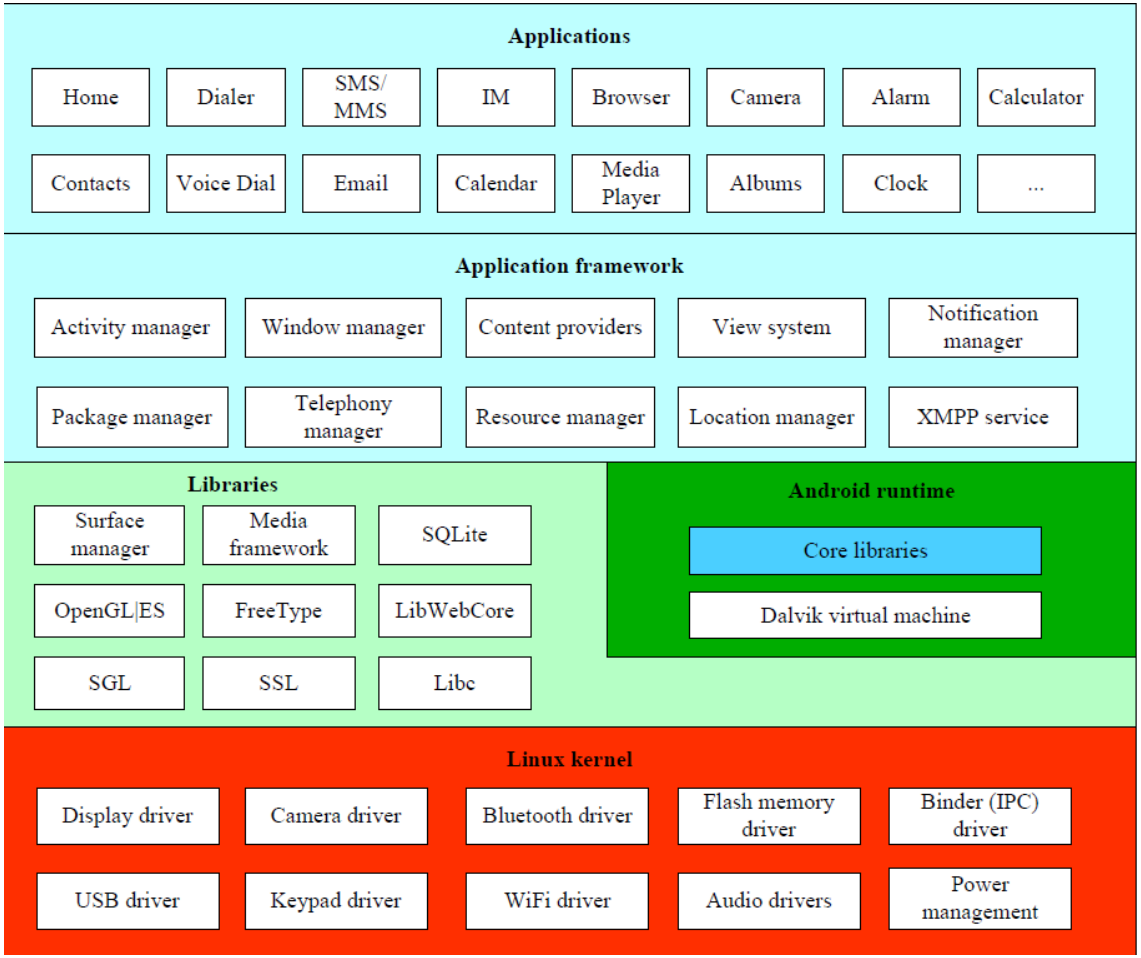
Android安全研究进展

Android软件栈


Applications
Home
Dialer
SMS/ MMS
IM
Browser
Camera
Alarm
Calculator
Contacts
Voice Dial
Email
Calendar
Media Player
Albums
Clock
...
Application framework
Activity manager
Window manager
Content providers
View system
Notification manager
Package manager
Telephony manager
Resource manager
Location manager
XMPP service
Libraries
Surface manager
Media framework
SQLite
OpenGL|ES
FreeType
LibWebCore
SGL
SSL
Libc
Android runtime
Core libraries
Dalvik virtual machine
Linux kernel
Display driver
Camera driver
Bluetooth driver
Flash memory driver
Binder (IPC) driver
USB driver
Keypad driver
WiFi driver
Audio drivers
Power management

Android 组件的通信

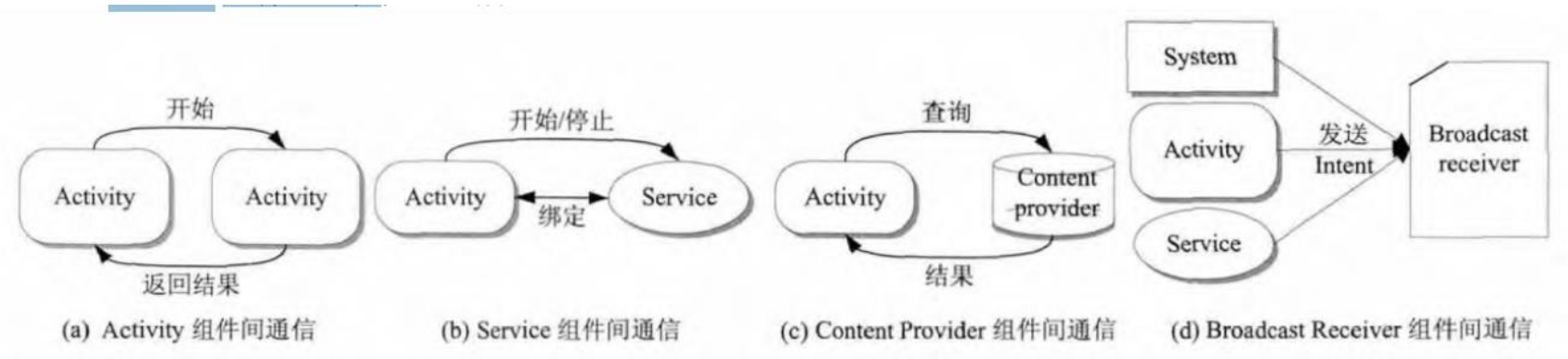


Fig.4 Android inter-component communication

Android与Linux对比

- alarm 驱动器、ashmem(Android 共享内存驱动器)、binder 驱动器(Android 特有的轻量级进程间通信机制)、电源管理、低内存管理器(low memory killer,简称LMK)和内核调试器(kernel debugger)

Android安全机制

- 继承Linux

（1）POSIX(portable operating system interface of unix)用户：赋予每个.apk唯一的ID，使始终运行在自己的进程中，固定的权限

（2）文件访问控制：每个文件都绑UID(用户ID)、GID(用户组ID)和rwx 权限

- Android 本地库及运行环境安全

（1）内存管理单元(memory management unit,简称MMU).为不同的进程分配不同虚拟内存

（2）强制类型安全：Android 使用强类型Java 语言

（3）移动设备安全：电话系统的基本属性集来自识别用户、监督使用和收费的需求，SIM卡保存使用者的密钥

- Android 特定安全机制

（1）权限机制：最小特权原则 => AndroidManifest.xml (手机所有者、Root、应用程序权限)

（2）组件封装：exported：false只能被本身或UID相同调用；true为公开组件

（3）签名机制：应用程序的作者对该应用负责

（4）Dalvik 虚拟机：每个应用程序都作为一个Dalvik 虚拟机实例在自己的进程中运行

Android权限机制的安全缺陷

- 粗粒度的授权机制 => 授权或者拒绝，一般的都会授权
- 粗粒度授权 => 比如Internet权限，程序可以发送所有的Http(s)请求，连接任意的目标地址和端口
- 不充分的权限文档
- 溢权问题 => 恶意溢权和非故意溢权
- 未提供“权限-API”对应关系映射集 => 比较应用申请的权限和API 调用,检查应用是否溢权

Android安全分析

- 静态分析：从应用的AndroidManifest.xml 文件中提取权限,然后自动检测应用的数据流是否与这些权限一致
- 动态分析：在原生Android 系统中加入监视器,实时监视数据的流向;在危险函数调用时,检测所需权限等
- 机器学习分析方法：提取什么特征、如何提取特征以及提取多少特征等问题还没有完美解决