# Seguridad de los Sistemas Informáticos

## Tema 9: Seguridad en Android

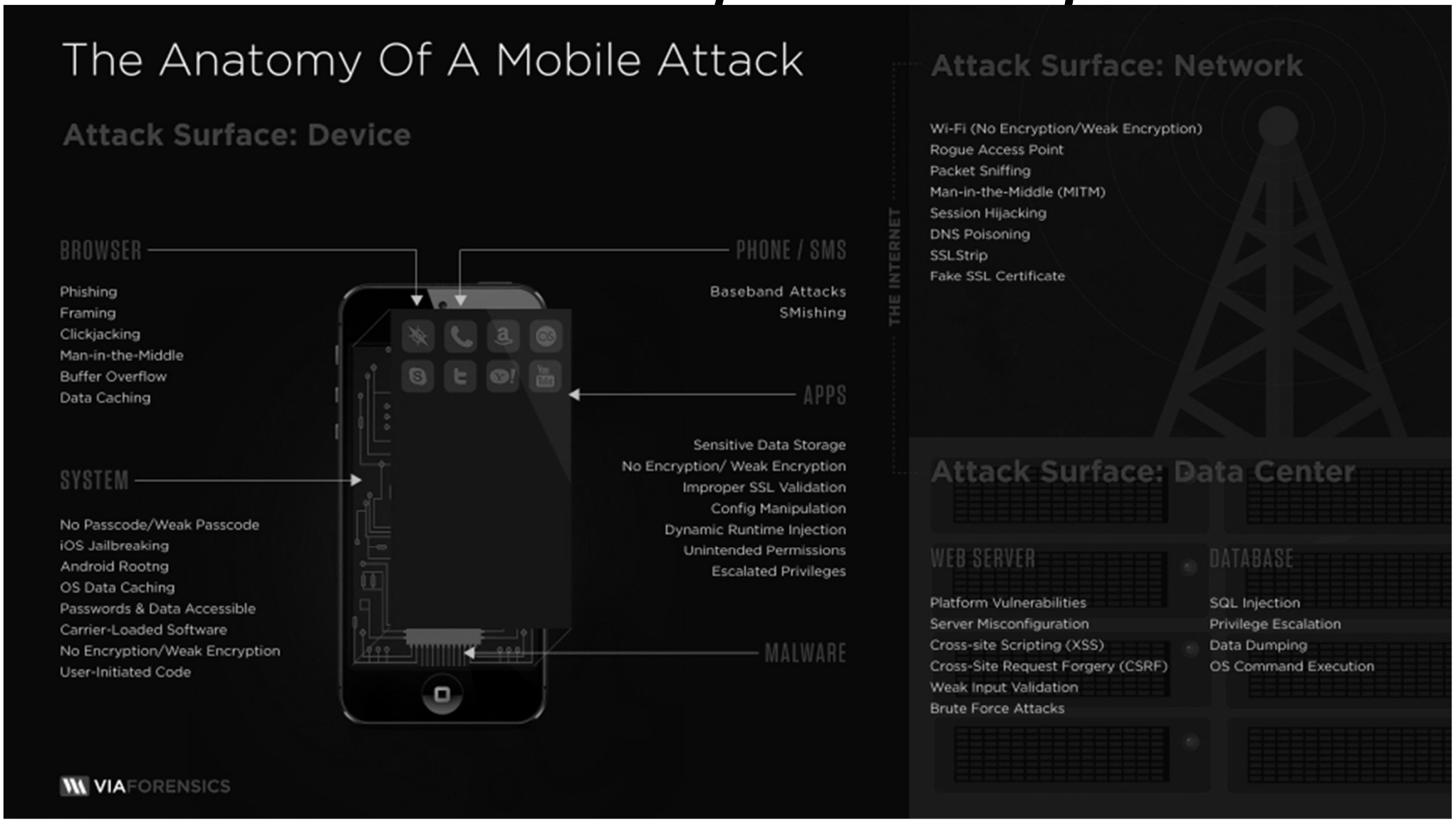
Escuela de Ingeniería Informática
Universidad de Oviedo

# Introducción

- Hoy en día han cobrado muchísima importancia los dispositivos móviles (smartphones, tablets, etc).
- Tanto a nivel particular como profesional o empresarial están ampliamente difundidos existiendo innumerables aplicaciones.
- Las empresas crean aplicaciones tanto para sus clientes finales como para sus propios trabajadores.
- A través de esas aplicaciones se tiene acceso a información muy importante y a prácticamente cualquier tipo de transacción (email, compras, viajes, bancos, redes sociales, etc.).
- Por todo ello es necesario que se tenga en cuenta la seguridad tanto a la hora de instalar aplicaciones en dispositivos móviles como a la hora de desarrollarlas para los mismos.

# Introducción

## Posibles ataques a un dispositivo móvil



# Introducción
## Posibles ataques a un dispositivo móvil

- Existen tres puntos en la cadena de la tecnología móvil que son susceptibles de ataques:
  - El propio dispositivo
  - La red
  - El centro de datos

# Introducción
## Posibles ataques a un dispositivo móvil

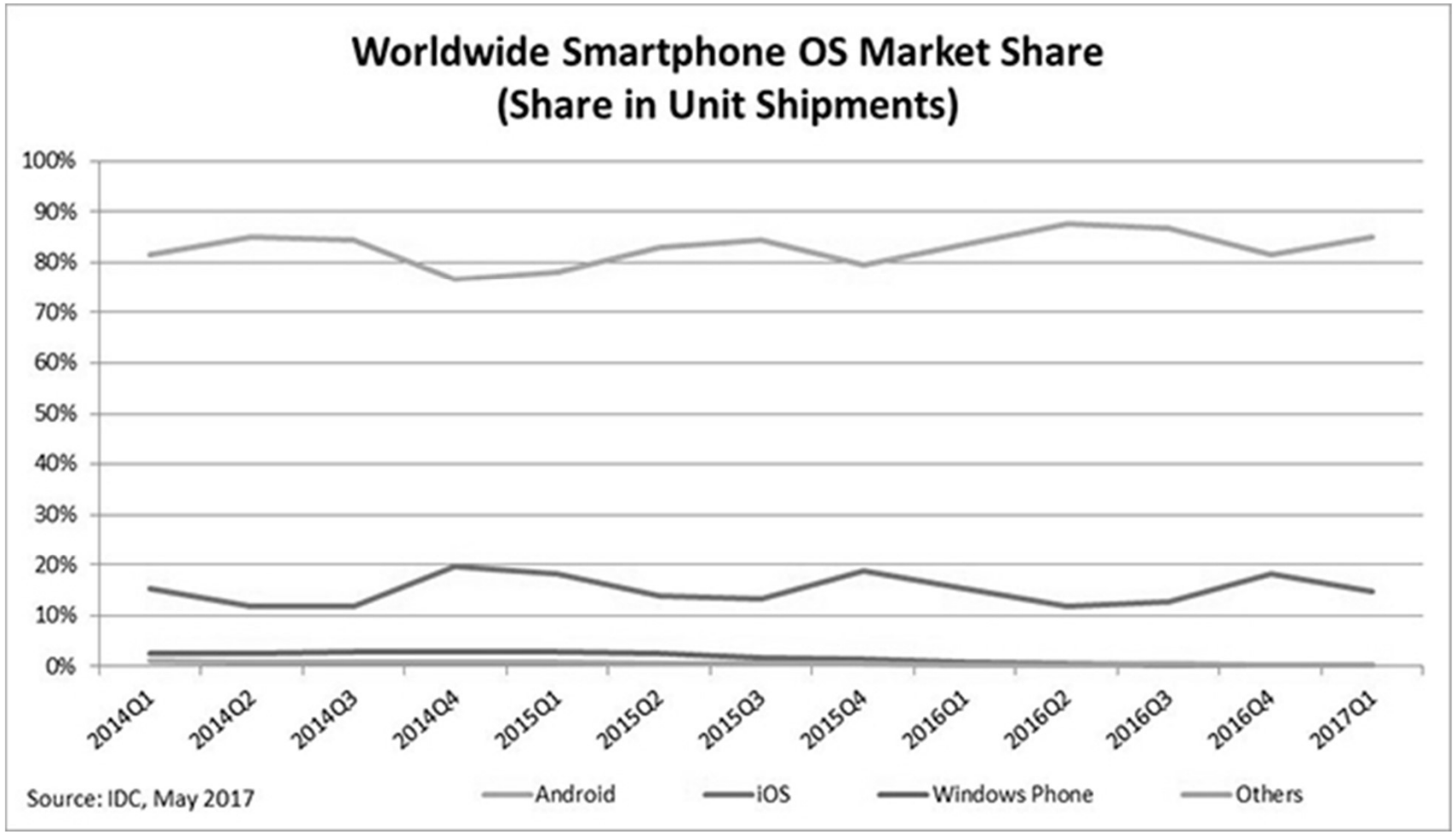
- El dispositivo puede ser atacado:
  - Navegador, mail, aplicaciones precargadas
  - Teléfono/SMS/MMS
  - Aplicaciones de terceros
  - Sistema Operativo
  - Elementos de comunicación como la banda base, Bluetooh u otro tipo de canal de comunicación

# Introducción
## Posibles ataques a un dispositivo móvil

- Los ataques al dispositivo puede ser (entre otros):
  - Phishing
  - Framing
  - Clickjacking
  - Ransomware
  - Drive-by Downloading
  - Man-in-the-Mobile
  - Ataques de la red (GSM, etc.)
  - RF Attacks. Bluetooth, NFC..
  - Acceso a datos sensibles almacenados sin seguridad
  - Encriptación débil o no existente
  - Mala validación de SSL
  - Permisos no intencionados

# Introducción
## Cuota de mercado actual

http://www.idc.com/prodserv/smartphone-os-market-share.jsp



# Introducción

## Cuota de mercado actual

# Android ya es el sistema operativo más usado del mundo

La plataforma de Google supera por primera vez en la historia a Windows, indicando el poder del móvil frente al PC

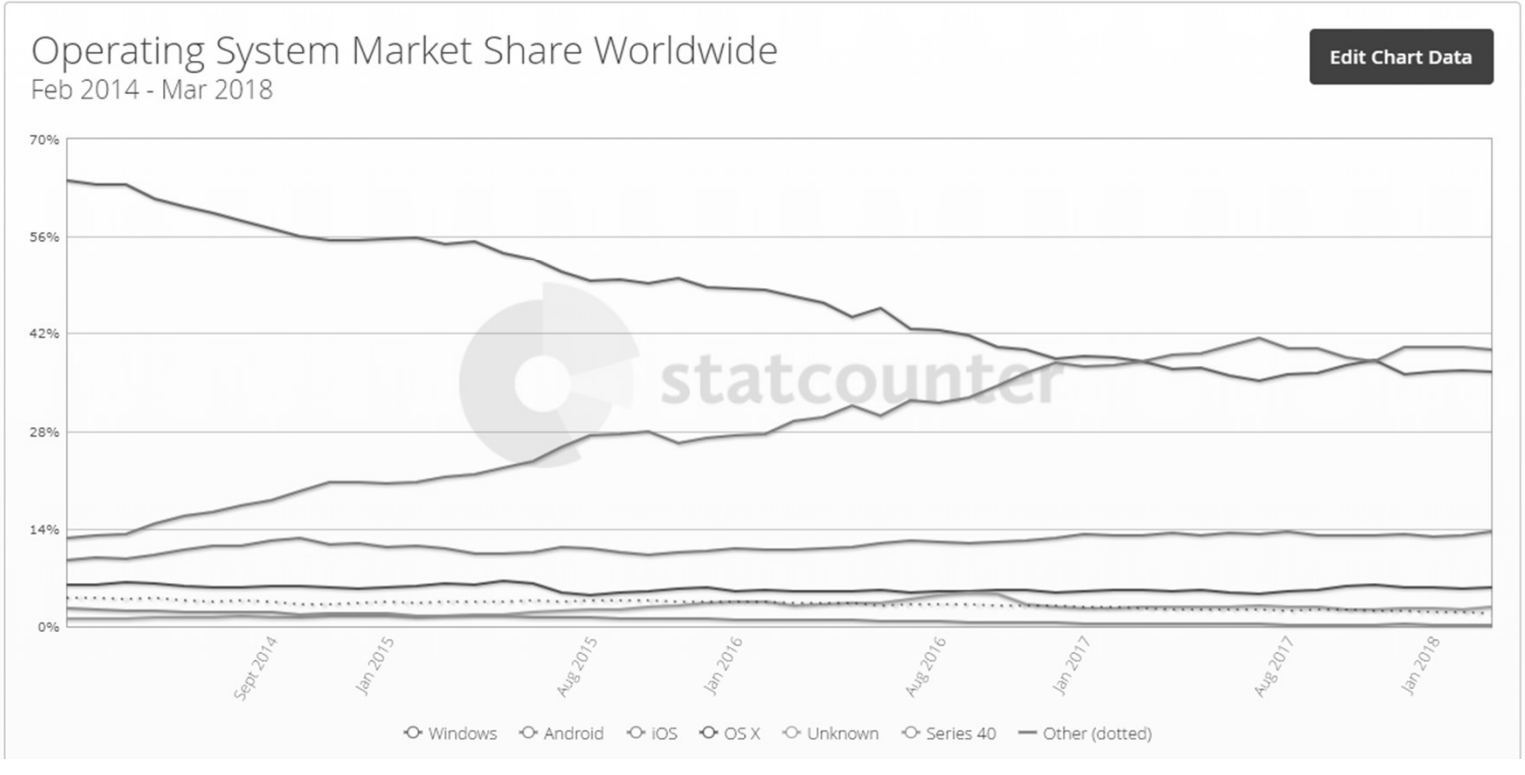
154

JOSÉ MENDIOLA ZURIARRAIN

4 ABR 2017 - 07:59 EDT

| Android | Windows | iOS | OS X | Unknown | Linux |
|---|---|---|---|---|---|
| 39.49% | 36.62% | 13.48% | 5.57% | 2.81% | 0.68% |

Operating System Market Share Worldwide - March 2018



# Introducción
## Android

- En la actualidad Android es el sistema operativo más utilizado en dispositivos móviles (y en global).
- Es un sistema abierto, existiendo un proyecto open source (https://source.android.com/) liderado por Google.
- Es por ello que nos vamos a centrar en él.

# Introducción
## Seguridad

- Si cuando instalamos una aplicación tiene acceso libre al terminal podría ser muy peligroso:
  - Acceso a usuarios/contraseñas.
  - Acceso a la información en memoria (cuentas bancarias, SSL…..).
  - Yo cuantificado (*Quantified Self, Life-Logging, Digital Footprint, Sousveillance…..).*
  - Podría capturar información y enviarla a un tercero sin nuestro consentimiento/conocimiento.
  - Posibilidad de realizar llamadas, enviar SMS….

# Introducción
## Seguridad

"En 2014, cada día fueron detectados 4.500 programas maliciosos dirigidos a aparatos que funcionan a través de la tecnología Android. En total, a lo largo del año, el nuevo *malware* (archivos maliciosos) destinado a atacar el sistema operativo de Google en dispositivos móviles superó la cifra de 1,5 millones, según recoge el informe elaborado por la empresa de seguridad informática G Data. Esto representa un aumento del 30% en los archivos de este tipo respecto al año anterior."

http://www.economiadigital.es/es/notices/2015/04/los-ando-69079.php

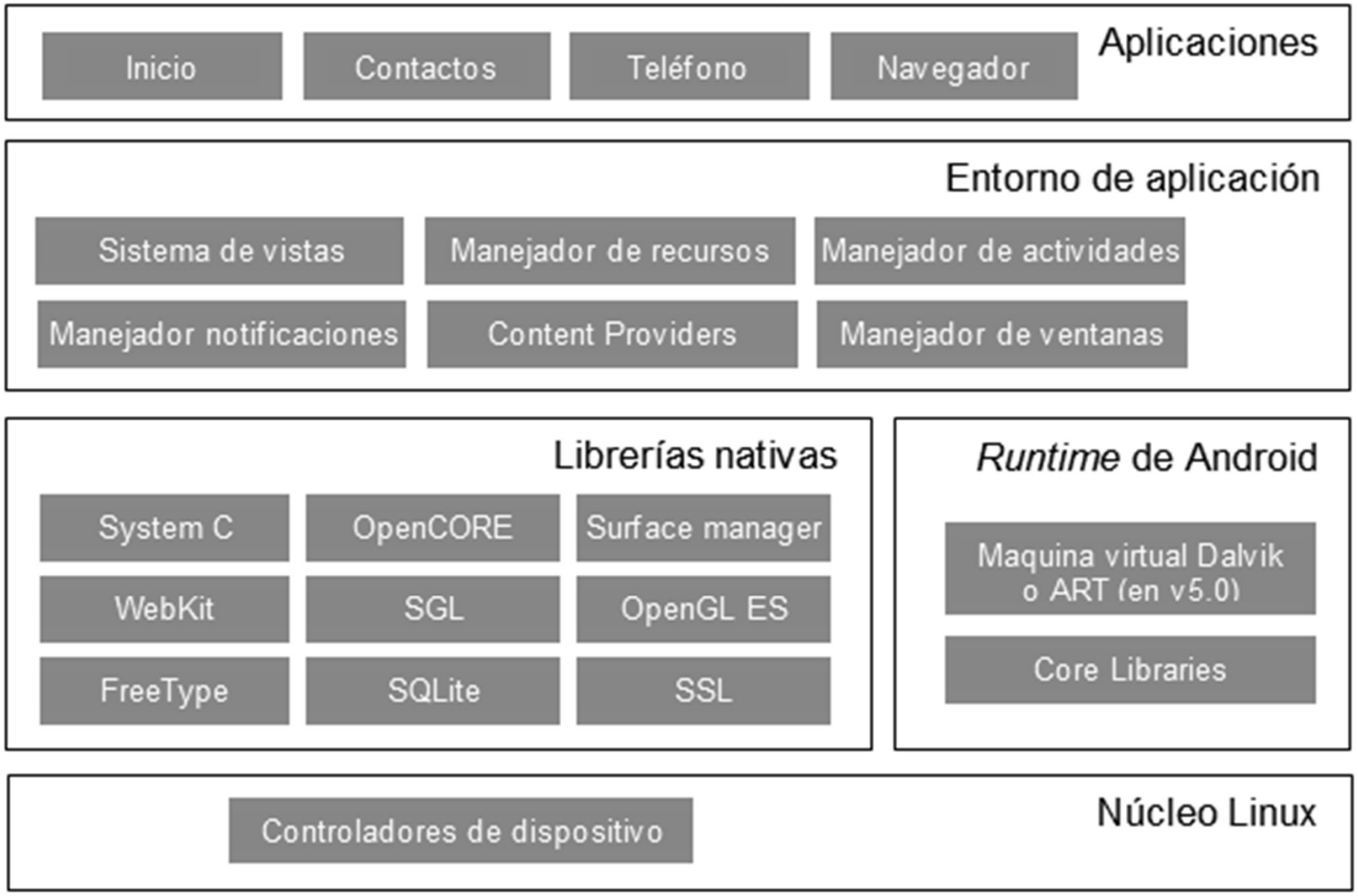
# Introducción
## Seguridad

- Existen diversos planteamientos.
  - Windows Mobile (antiguo)
    - Desprotección total.
  - iOS
    - Todas las aplicaciones pasan validación de Apple.
    - Da mucho poder a Apple que puede decidir qué aplicaciones pueden estar o no disponibles para su uso.
    - Limita a los pequeños programadores y el software libre.

# Seguridad en Android

- Android sigue un modelo que pretende ser más abierto sin perder por ello opciones de seguridad.
- Se basa en tres conceptos fundamentales
  - Ejecución de las aplicaciones en Sandboxes (por lo tanto se consigue aislamiento).
  - Las aplicaciones deben ser firmadas digitalmente.
    - Identifica al autor
    - Confianza (reputación) del autor y la aplicación
  - Un modelo de permisos.

# Arquitectura Android



# Arquitectura Android

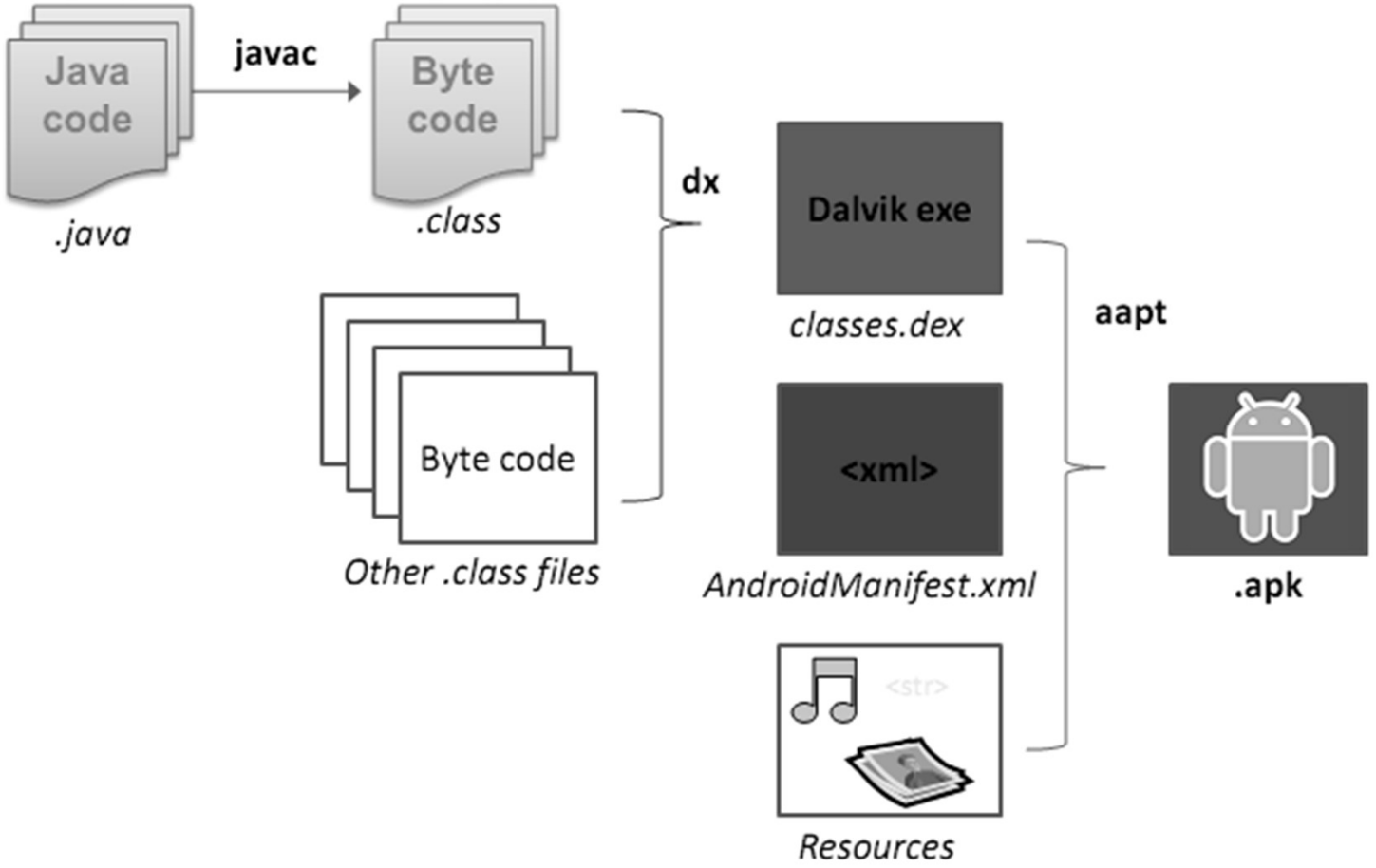
- El sistema se puede ver como una pila con 4 capas.
- En el nivel inferior está el núcleo Linux (basado en un Kernel 2.6) que proporciona seguridad, acceso a memoria, threads, o drivers para los dispositivos. Es la única parte dependiente del hardware.
- Sobre él se encuentra el *Runtime* de Android. Es una MV basada en la JVM pero que tuvo que ser creada de nuevo y específicamente por cuestiones de memoria y procesador.
  - Inicialmente se utiliza Dalvik.
  - A partir de versión 5 se reemplaza por ART.

# Seguridad Android
## Firmado

- Si queremos publicar una aplicación en el *Market* debemos firmarla con un esquema de certificado:
  - El autor firma la aplicación con su clave privada que lo identifica de forma única.
  - Android se basa en la reputación del autor.
  - No es necesario utilizar una autoridad certificadora.
  - No es necesario (pero puede ser conveniente) utilizar el mismo certificado para todas nuestras aplicaciones.
  - Si dos aplicaciones están firmadas con el mismo certificado pueden ejecutarse bajo el mismo proceso o exponer funcionalidad y/o código entre ellas.

# Seguridad Android

- Una aplicación .apk tiene la siguiente estructura



# Seguridad Android

- Cuando se instala una aplicación (apk):
  - Se crea un usuario para la aplicación (user ID). Con ese usuario se ejecutará posteriormente la aplicación. El usuario existe hasta que se desinstale la aplicación del dispositivo.
  - Existe un sistema de permisos muy detallado que da derechos a la aplicación (a través de su usuario) a recursos del sistema. Para ello se usa el AndroidManifest.xml.
  - Originalmente los permisos se asignaban a la hora de la instalación de la aplicación, no pudiendo ser modificados posteriormente (versiones <6)…. En la 6+ sí se pueden modificar y solicitar posteriormente.
  - Es un “sistema Linux”, que mantiene el esquema de usuarios, grupos y permisos.

# Seguridad Android
## Sandboxes

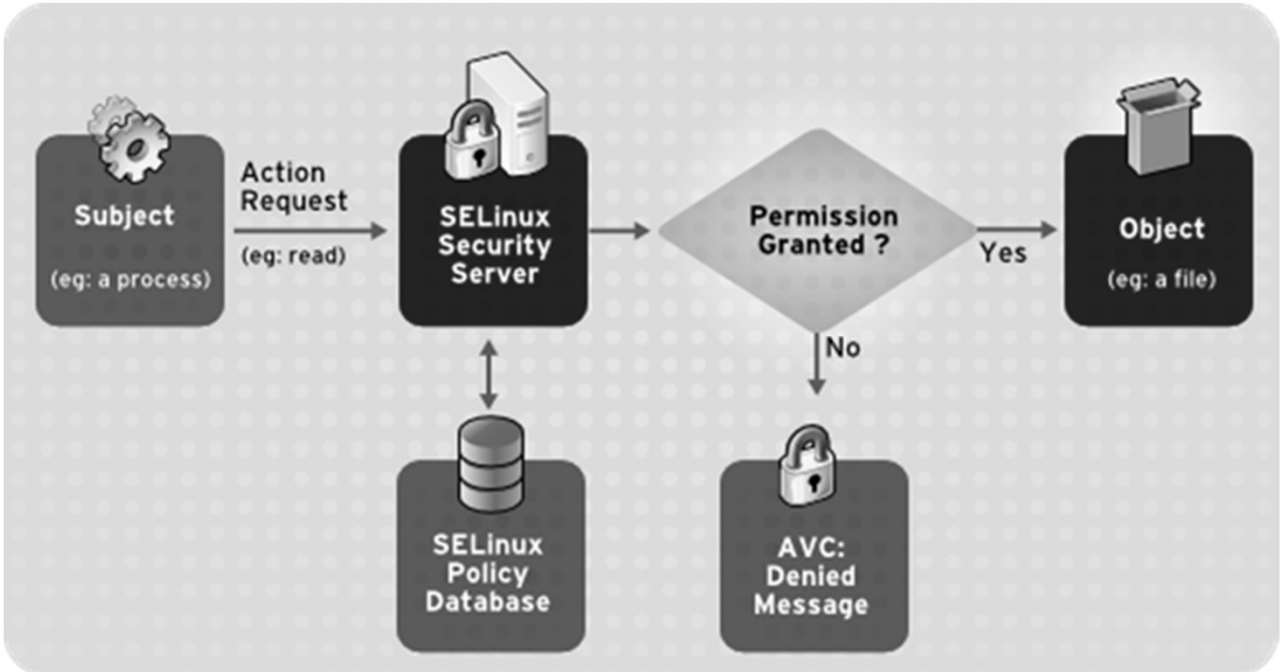
- La aplicación puede contener código java (interpretado por el Runtime) o nativo.
- La aplicación es ejecutada por un único proceso del sistema operativo, ya sea de forma nativa o mediante una MV (una MV para cada aplicación). Ese proceso es la “sandbox” y es el kernel el que se encarga de protegerla e impedir el acceso al exterior y desde el mismo.
- Un proceso no tiene acceso a los recursos de otro (a su información, memoria, etc.). Esto lo garantiza el kernel.
- Existen mecanismos de comunicación entre procesos (IPC) que permitirán intercambiar datos entre procesos (de una forma controlada por ambos procesos).

## Permisos

- Inicialmente el sistema de permisos (recursos a los que puede acceder la sandbox) implementado se basaba en el concepto de usuarios/grupos y permisos sobre recursos. Es decir era un sistema DAC (discretionary access control).
- En DAC existe un propietario de cada recurso el cuál puede otorgar permisos. Es proclive a una escalada de permisos no intencionada.

# Seguridad Android
## Permisos

- A partir de la versión 4.3 se incorpora un sistema MAC (mandatory access control) de forma opcional. El sistema es SELinux.
- En MAC no importa la propiedad del recurso, se consulta a una autoridad central para decidir en todos los intentos de acceso.
- A partir de la versión 5 SELinux viene activado en modo *enforced (obligatorio).*
- Esto afecta incluso a los procesos que se ejecutan como *root* (por lo que no puede ser evitado, en principio).



# Seguridad Android
## Permisos

- Android SELinux funciona con la política de denegar todo por defecto. Cualquier permiso que no se le haya otorgado explícitamente a una aplicación se considera denegado.
- Funciona por encima del sistema DAC original.
- En el modo permisivo (versión 4) cualquier violación de acceso se registraba, pero no se impedía.
- En el modo obligatorio (versión 5+) cualquier intento de violación de acceso se registra y conlleva el abortar la ejecución de la aplicación.

# Seguridad Android
## Almacenamiento

- Android tiene diversas opciones para que las aplicaciones guarden sus datos:
  - Preferencias compartidas: Permite guardar datos primitivos (bool, int, …) en forma de clave – valor.
    - Proporciona un *framework* para guardar y leer pares de este tipo de forma persistente.
    - Estos datos persisten entre sesiones (aunque la aplicación se finalice abruptamente).
  - Almacenamiento interno: Permite guardar datos privados en la memoria del dispositivo, concretamente ficheros.
    - Estos ficheros son en general privados a la aplicación. Se puede hacer que otros procesos tengan permiso de lectura/escritura pero la propiedad siempre es del usuario que lo creó.
    - Cuando la aplicación se desinstala, se borran.

# Seguridad Android
## Almacenamiento

- Almacenamiento externo: Permite guardar datos públicos en un medio de almacenamiento externo removible (tarjeta SD) o bien interno y no removible.
  - Los ficheros grabados en el almacenamiento externo son legibles (y escribibles) para todas las aplicaciones.
  - Si contienen información sensible DEBEN encriptarse.
  - Pueden incluso ser modificados desde un PC si se le conecta el teléfono y se activa el servicio de almacenamiento masivo USB.
- Bases de datos *SQLite*: Permiten grabar datos estructurados en una BBDD privada, con soporte completo de *SQLite.*
  - Puede accederse a la BBDD **privada** desde cualquier clase de la aplicación, pero no desde otras aplicaciones.
- Conexión de red: Permite guardar datos en la web en un servidor externo, incluso vía servicios *web.*

# Seguridad Android
## Encriptación del sistema de ficheros

- En Android 3+ se soporta encriptación a nivel del Sistema de ficheros (por parte del kernel)
- En Android 5+ se soporta encriptación a nivel del disco.
  - Se utiliza una única clave de encriptación para proteger las particiones de datos de usuario.
  - Cuando el usuario entra se le solicitan credenciales para poder acceder a su disco (si no las tiene no se puede acceder)
- En Android 7+ se soporta encriptación a nivel de fichero. Cada fichero puede ser encriptado con una clave única por lo que pueden ser bloqueados y desbloqueados de forma independiente.

# Seguridad Android
## Permisos

- Toda aplicación que acceda a un recurso debe declararlo previamente.
- En versiones <6.0 cuando se instalaba una aplicación se le presentan al usuario (en lenguaje no técnico) los permisos que solicita la aplicación y debe aprobarlos o rechazarlos en conjunto. Si no se aceptan la aplicación no se instala. No se modifican posteriormente.
- En versiones 6.0 y siguientes los permisos se pueden solicitar de forma individual y cuando lo considere oportuno la aplicación (al instalarse, cuando vaya a hacer uso de ese privilegio, etc.). El usuario puede administrar estos permisos en cualquier momento.
- Si una aplicación intenta acceder a un recurso no declarado la ejecución se interrumpe, el error es registrado (y si está activada la verificación de malware Google será informado, de tal manera que pueda tomar medidas).

# Seguridad Android
## Permisos

- Los permisos posibles que podemos solicitar están definidos en http://developer.android.com/reference/android/Manifest.permission.html
- Para solicitarlos basta con incluir la petición en el manifiesto

```
<manifest package="org.example.mi_aplicacion" >
    ...
    <uses-permission android:name="android.permission.RECEIVE_SMS"/>
    <uses-permission android:name="android.permission.SEND_SMS"/>
</manifest>
```

# Seguridad Android
## Permisos

- Algunos ejemplos
  - CALL_PHONE: llamar a números de teléfono directamente.
  - READ_PHONE_STATE: leer ID y estado del teléfono.
  - SEND_SMS
  - RECEIVE_SMS: recibe sms, los procesa y puede eliminarlos.
  - CAMERA: acceso a la cámara para controlarla (sin intervención del usuario).
  - RECORD_AUDIO
  - ACCESS_COARSE_LOCATION/ACCESS_FINE_LOCATION
  - READ_CONTACTS/WRITE_CONTACTS
  - WRITE_EXTERNAL_STORAGE: permite escribir/eliminar cualquier fichero allí
  - INTERNET
  - READ_CALENDAR