Mobile Security
14-829 – Fall 2013

Patrick Tague

Class #15 – Mobile Operating Systems

Carnegie Mellon University
Silicon Valley

Mobile OSs

- Mobile operating systems are the “glue” that binds all the underlying components, manages them, and represents them to applications
  - The list of Mobile OSs is quite long
  - We'll just talk about the major ones

Symbian

- Developed by Nokia

- Nokia has now mostly migrated to Windows for smartphones
  - Still using Symbian for some feature phones

- Symbian had the highest world-wide unit sales total through 2010, now basically 0

Android

- Android OS was created by the Open Handset Alliance
  - 84 companies including mobile operators, handset manufacturers, semiconductor companies, software developers, commercialization companies
  - “First complete, open, and free mobile platform”
- World-wide unit sales leader since 2011, currently about 4x anyone else

iOS

- Developed by Apple for iPhone, now extended to iPod, Apple TV, and iPad
  - Only licensed for Apple's proprietary hardware / systems
- Originally developed as a web portal, then revolutionized 3rd-party applications on phones
- 2nd world-wide since 2011

Blackberry

- Blackberry is a proprietary OS developed by Research in Motion (RIM)

- Targeted at enterprise use and leader in corporate data management

- 4th in device sales world-wide

Windows Phone

- Windows Phone is the push by Microsoft to become a major mobile OS player
  - Replaced Windows Mobile
- Incorporates many Microsoft services but also heavily integrated with other non-Microsoft ones
- Just passed BB for 3rd place

OS Security Models

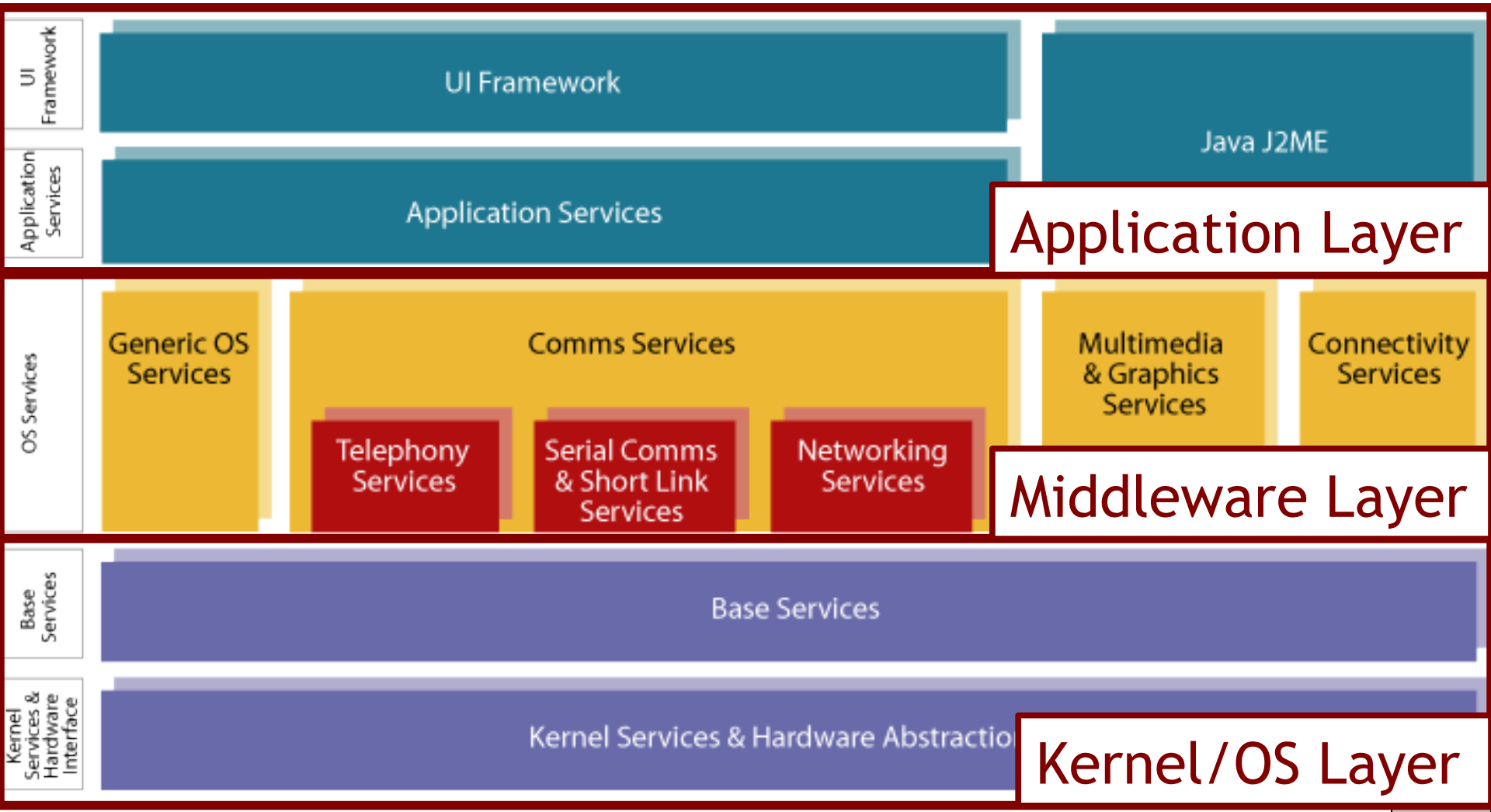
- Every OS is designed around a different user model, security model, and application model

- Security models are primarily built on that provided by the underlying OS / kernel
  - Android and iOS are built primarily on a Linux or Linux-like kernel
  - Windows Phone is built on the Windows NT kernel
  - Others are built on proprietary (micro)kernels

We've already heard a lot about Android, and we'll hear more on Monday.

Today, we'll focus on others.

Symbian Model



Symbian Security Model

- The Symbian security model has three modules:
  - Trusted computing base
  - Data caging
  - Capabilities

Symbian TCB

- Symbian's Trusted Computing Base
  - Collection of software that enforces data caging and capabilities
  - Comprises the kernel, the file system, and the software installer
  - Controlling part of the OS in the security model

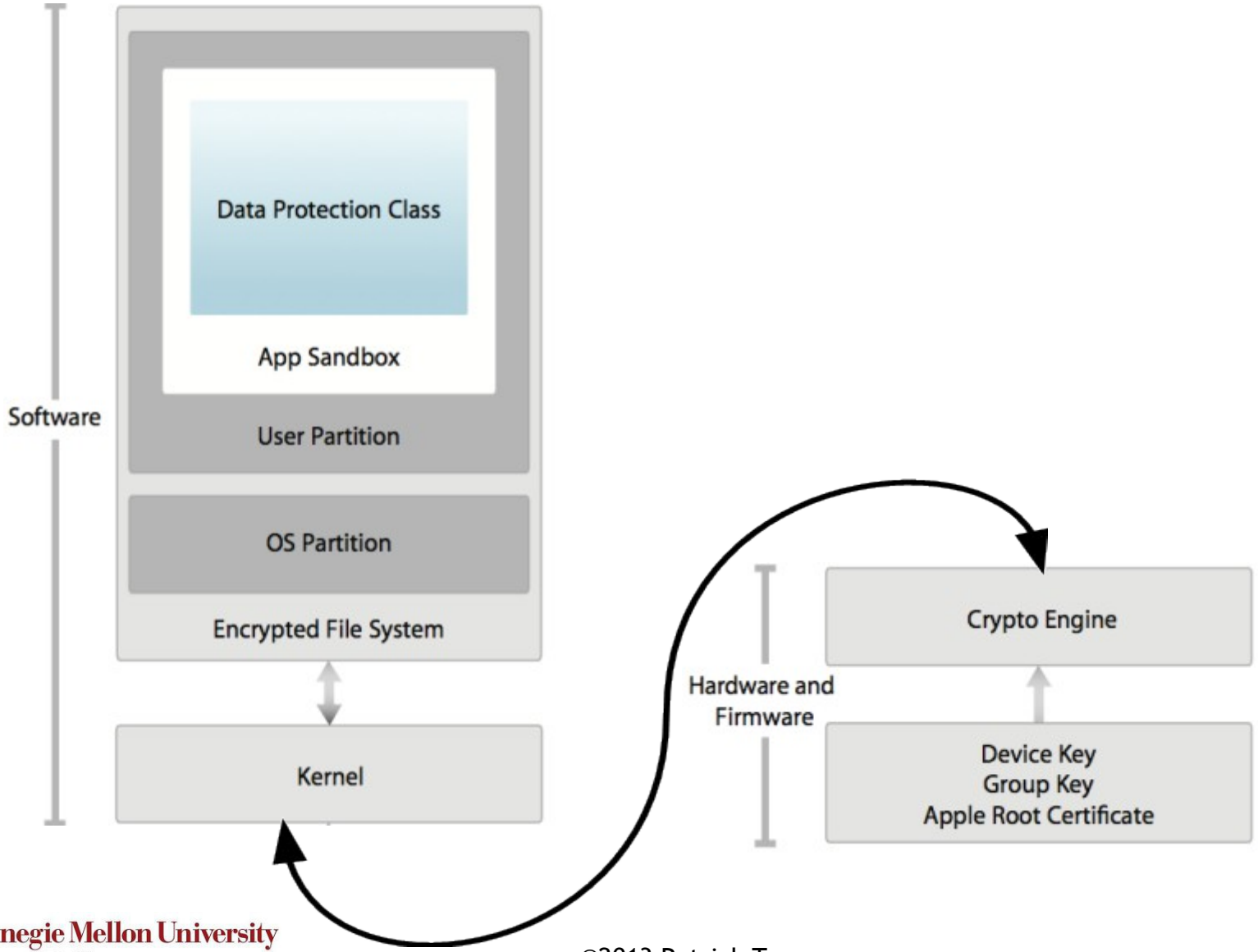
Symbian Data Caging

- Symbian Data Caging
  - Applications and users only have access to certain parts of the file system
  - Apps can access their own private repositories and any public repositories
  - Apps cannot access private content from any others

Symbian Capabilities

- Capabilities
  - A capability grants access to a set of APIs, obtained through certification
  - Types of capabilities:
    - Open to all
    - Granted by user at install time
    - Granted through Symbian Signed
    - Granted by manufacturer
  - Capability enforcement can be bypassed via firmware update (sort of like “jailbreaking”)

iOS Model



iOS Security Model

- iOS's security model is based on OSX, a BSD / Unix-like OS
- Various levels of system, application, resource, network, data / storage, user, and physical protection
- Relies heavily on Unix-like users, processes, permissions, sandboxing, etc.

System Protections

- Sandboxed applications (same as OSX)
  - Protects OS from malware and certain malicious apps
- Code review and signing by Apple
  - Review process is still largely unclear, but rumors are that it provides more of a “warm fuzzy” protection than any sort of guarantee
  - ?
- Multi-user system (“mobile” user, “root” user)
  - Mobile user can run application(s)
  - Root user runs processes
  - Jailbreaking gives root access

Resource Protections

- iOS APIs restrict access to certain resources
  - Bluetooth has no / limited API
  - Proximity sensor is unavailable
  - WiFi connection management is restricted
  - GPS system is carefully managed
    - Apple has a “kill switch” on GPS apps
- Other resources are (mostly) unrestricted
  - Microphone, camera, Internet, accelerometer
  - (This has been changing...)

Jailbreaking & Unlocking

- Jailbreaking
  - Modifying the application processor firmware to allow root access, running unsigned code, bypassing other restrictions, etc.
  - Based on hacking / modifying the bootloader due to a development “flaw”
- Unlocking
  - Removing the carrier-restriction for which network the iPhone can connect to (e.g., leaving AT&T)
  - Not as easy as jailbreaking (no dev “flaws”), but still possible via exploit of baseband software

BlackBerry

- Since 2003 BlackBerry has been touted as the most secure mobile OS
  - Designed on strong foundations of data security, user authentication, etc.
  - Features have causes BB to be embraced by corporations, government agencies, etc.
  - Even to the point that regulators in other countries were annoyed because they couldn't access or monitor user behavior for enforcement purposes

Secure Data Storage

- BlackBerry's security model heavily emphasizes secure data storage
  - All user data is encrypted using AES
  - Access to data requires users authentication
  - Failed authentication 10 times in a row results in all stored data being deleted

BB Password Keeper

- BlackBerry also comes equipped with a password management app
  - Consolidates all of a user's passwords into a single encrypted repository, protected by a “master” password
  - Master password cannot be changed once set
  - Failed password entry 10 times in a row results in deletion of entire repository

Administrator Privileges

- An admin (probably a corporate IT admin) can remotely manage the BB device
  - Delete local data
  - Lock data storage
  - Change the device password

Wireless Data Security

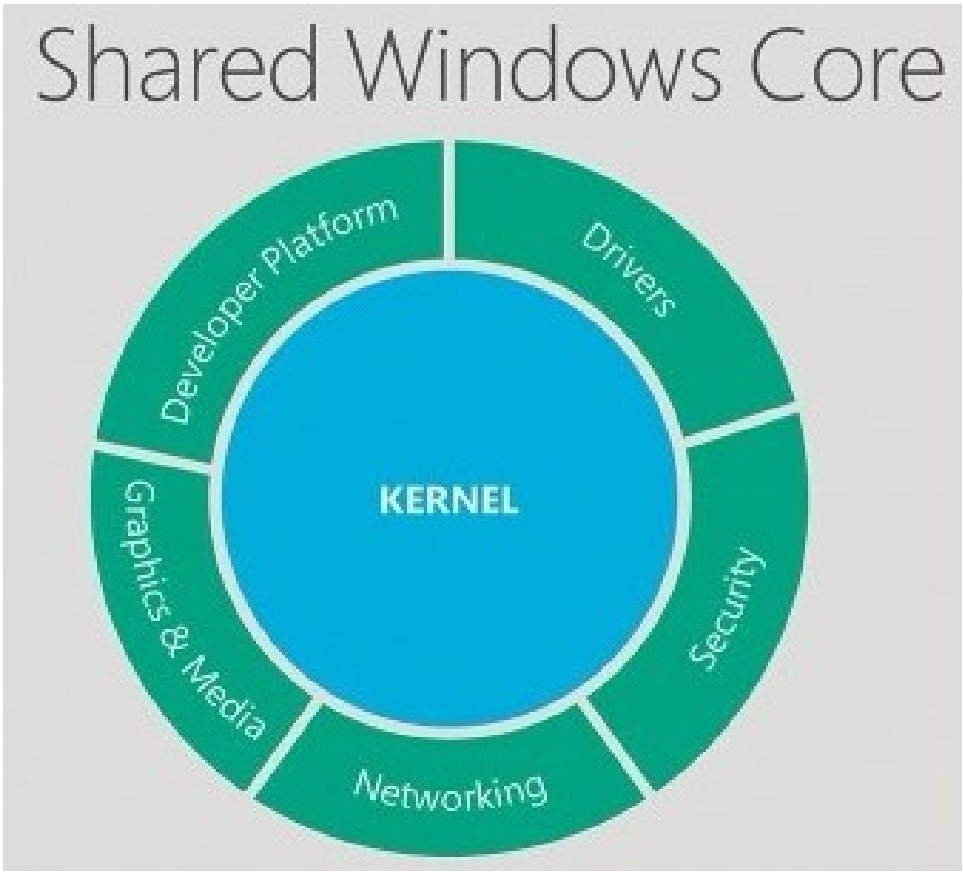
- Data is encrypted from device to BB server using a secret user key
  - Both AES and 3DES are used

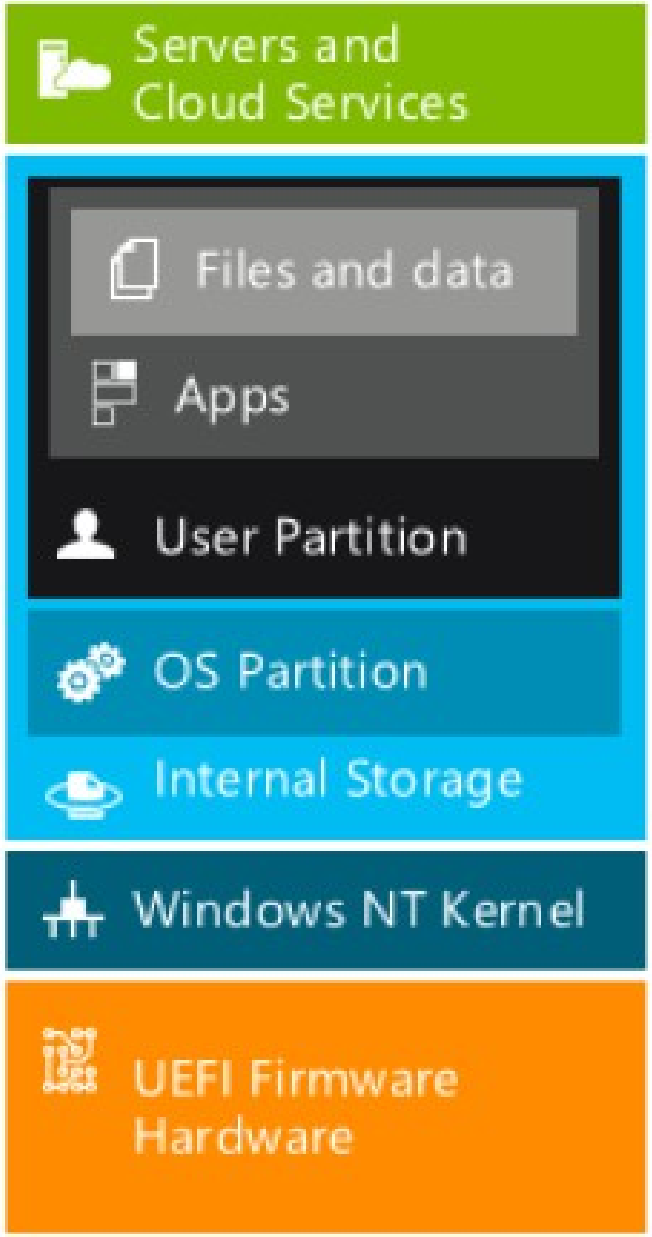
- Access to corporate intranet using RSA SecurID

Advanced Security Features

- Advanced security features can be enabled for government use
  - Certain BlackBerry devices meet DoD standards
    - Federal Information Processing Standard 140-2 validation
    - Secure/Multipurpose Internet Mail Extensions
    - Public key infrastructure
  - Smart Card Reader
  -
- BlackBerry 10 also includes:
  - Permission-based model for 3rd-party apps
  - Heavily restricts root access
  - ASLR

Windows Phone Model





Windows Security Model

- Windows Phone security model is based on years of Windows experience and observations of issues in other platforms

- Considerations:
  - System integrity
  - App platform security
  - Data protection
  - Secure access

System Integrity

- Platform integrity assured through Trusted Boot and code signing
- Trusted Boot validates the firmware image, which is then responsible for validating and loading the OS
- Trusted Boot uses a standardized hardware root of trust (like a TPM)

App Platform Security

- Windows Phone 8 uses **chambers** for isolation
  - Chambers are much like sandboxes that are policy-defined and control interaction between apps
  - Chamber policies are based on capabilities, much like Symbian; app permissions can be expanded using capabilities
- Apps are managed through Windows Phone store
  - Manages: certification/verification of apps, validation of developer, virus scanning, app signing
  - Updates are strictly managed by Microsoft, similar to other products

Data Protection

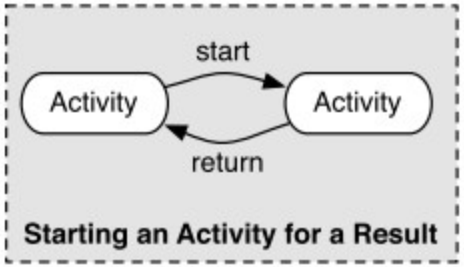
- Every Windows phone, regardless of mfgr, includes common mgmt and security controls
- Relies on Microsoft Exchange ActiveSync for mailbox sync, policy mgmt, password mgmt, and additional security features such as remote wiping of lost devices
- Internal storage is encrypted using BitLocker
- Only media can be stored on SD card, which is unencrypted (and can be disabled)

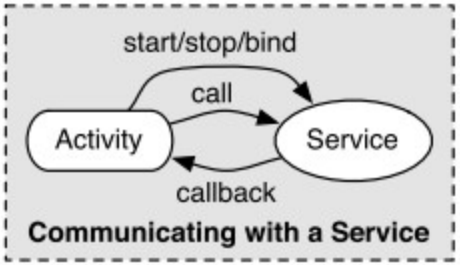
Secure Access

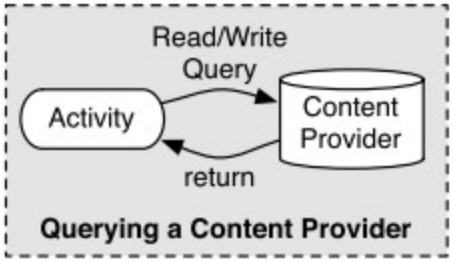
- Windows Phone 8 designed to heavily leverage cloud services
- Data sync between device and most cloud or local services requires SSL connection
- All critical network traffic (including most 3rd-party and custom business apps) is encrypted using 128- or 256-bit AES

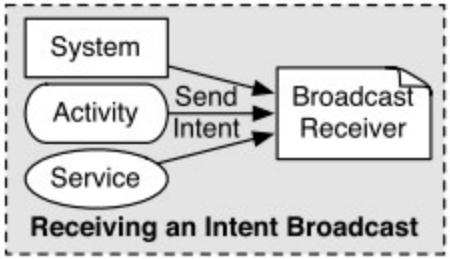
Android Security Model

- Android OS is built on top of Linux
  - Each application has its own uid
- Specialized middleware
- Application interaction between different types of components
  - Activity
  - Service
  - Content provider
  - Broadcast receiver



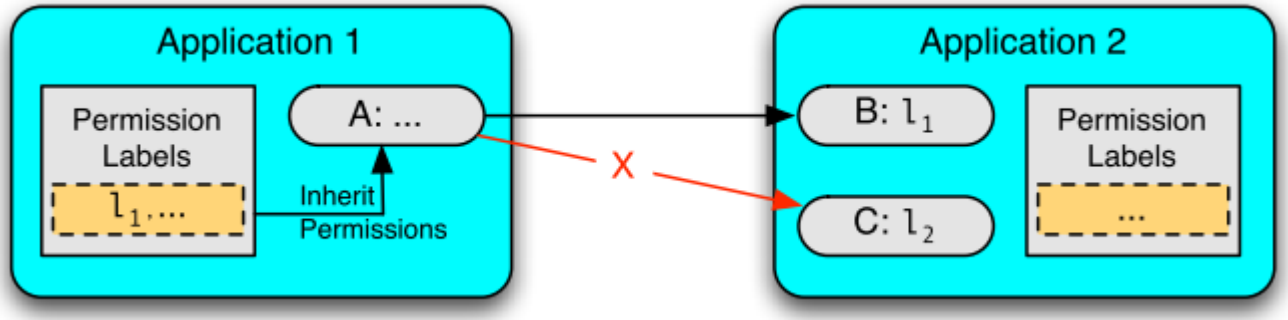






Inter-Component Communication

- Android uses intents for component signaling
- Android manifest file allows developers to define access control policies
  - Each component has an access permission label
  - Each app has a (fixed) list of permission labels



  - With some exceptions...

Slides adapted from [Enck & McDaniel, Penn State, 2009]

Rooting

- Almost all Android phones are root-able
  - Initially due to an exploit of the Android Debug Bridge (adb) tool, but many ways have emerged
- Rooting an Android phone gives arbitrary access to system
  - Modification of drivers, kernel, kernel modules, installed applications
  - Also opens up to system components developed by the "hacker community"
    - Improved/modified bootloaders, better backup utilities, enhanced drivers, apps, and services
    - Many of these have been adopted by the Android community

Summary

- There are many common themes across the mobile OSs
  - Fundamental similarities in OS and security models
  - Many similarities in use of permissions
  - Overall, most of the systems have similar flavors of security vulnerabilities
    - Android is more targeted because it's mostly open, so finding them is easier...for now.

- Despite the similarities, designs and implementations are wildly different

Oct 21:
Guest Lecture: Anmol Misra and Abhishek Dubey, Android Security