UNIVERSITY OF ONTARIO
INSTITUTE OF TECHNOLOGY

Mobile Devices

Hardware Architecture, Limitations, and Software Architecture

Randy J. Fortier
@randy_fortier
randy.fortier@uoit.ca

Outline

- Mobile devices
  - Capabilities
  - Limitations
- Platform survey
- Android platform/architecture introduction

UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

Mobile Devices

Mobile Hardware and its Limitations

Mobile Devices

- Of course phones, but what else?

Mobile Devices

- Of course phones, but what else?
- Tablets
- Portable media players
- Not really mobile:
  - BluRay players
  - Smart TVs

UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

Mobile Device Capabilities

- Touch screen
- Sensors
  - GPS
  - Accelerometer
  - Gyroscope
- Quad core CPUs
- (Integrated) multi-code GPUs (supports 3D)
- Audio, video playback, decoding
- Camera(s)
- Network (WiFi, LTE, etc.)
- High memory capacity

UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

Mobile Device Capabilities

- Touch screen
- Sensors
  - GPS
  - Accelerometer
  - Gyroscope
- Quad core CPUs
- (Integrated) multi-code GPUs (supports 3D)
- Audio, video playback, decoding
- Camera(s)
- Network (WiFi, LTE, etc.)
- High memory capacity
- Oh yeah: phone, SMS, and E-Mail!

UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

Mobile Device Limitations

- Small screens
- Input devices
- Limited processing capabilities
- Limited storage
- Limited network speed/download capacity
- Limited battery life

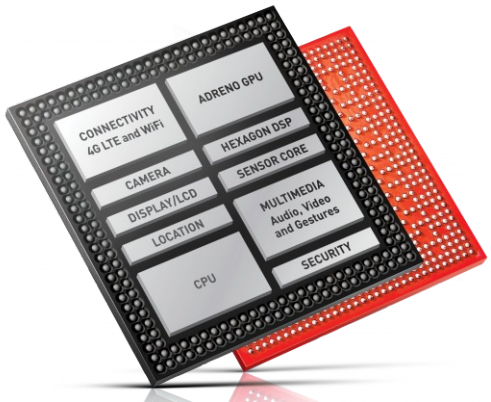
UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

Mobile Device Architecture

- At the heart of a mobile device is a SoC
  - System on a Chip
  - Almost a complete device on a single chip:
    - Processor/CPU (often, ARM)
    - Graphics processor/GPU
    - Digital and analog input/output
    - Radio communications (LTE, WiFi, BlueTooth)
    - Hardware communications (Ethernet, USB)
    - Memory (RAM, ROM, Flash)

Example: Qualcomm Snapdragon 805

- CPU: Qualcomm Krait 450 (quad core, 64-bit, 2.7Ghz)
- GPU: Qualcomm Adreno 420 (128 pipeline, DirectX/OpenGLES, 600Mhz)
- DSP: Hexagon V50 (supports 4K displays
- Memory controller: 800Mhz
- Radio/Modem: 4G LTE, WiFi n/ac, NFC
- Connectivity: USB 2.0/3.0,BlueTooth 4.1
- Sensors: GPS, image processor



UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

Mobile Devices

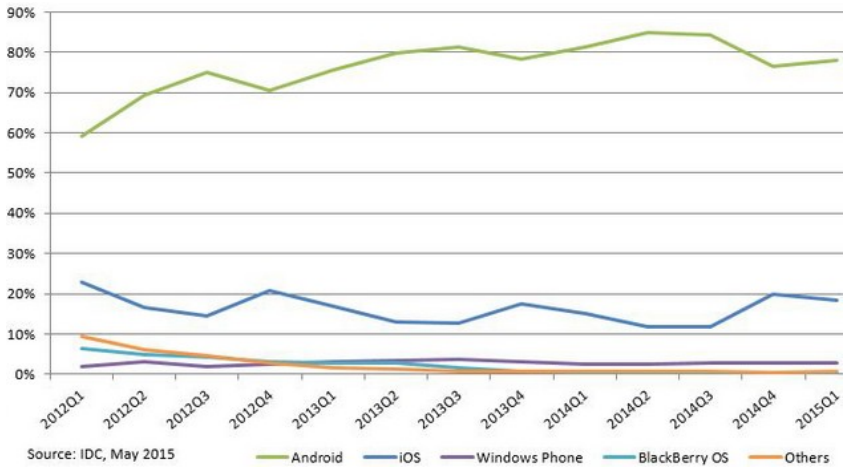
Mobile Platforms

Mobile Platforms

- **Android**
- iOS
- Windows Phone
- Blackberry

Mobile Platforms

- **Android**
- iOS
- Windows Phone
- Blackberry



Phone OS	Share (%)
Android	78.0%
iOS	18.3%
Windows Phone	2.7%
Blackberry OS	0.3%

Tablet OS	Share (%)
Android	51.1%
iOS	41.6%
Windows Phone	2.5%
Others	4.8%

UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

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

Mobile Platforms

- **Android**
- iOS
- Windows Phone
- Blackberry
- HTML5

Android

- Currently the dominant platform for mobile devices
- Developed by the Open Handset Alliance
- Largely open source
- Multitasking operating system
- Support for OpenGL ES
- Apps are written in Java
- Google Play

UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

iOS

- Developed by Apple
- Proprietary and closed source
- Multitasking operating system
- Support for OpenGL ES
- Apps are written in Swift (on Macs only)
- App Store

UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

Windows Phone

- Developed by Microsoft
- Proprietary and closed source
- Multitasking operating system
- Support for XNA (somewhat compatible with Windows, Xbox)
- Apps are written in many languages (e.g. C#)
- Windows Phone Marketplace

UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

Blackberry OS

- Developed by Research in Motion (RIM)
- Proprietary (with some open source)
- Multitasking operating system
- Support for OpenGL ES
- Apps are written in Java ME
- Blackberry App World

UNIVERSITY OF ONTARIO
INSTITUTE OF TECHNOLOGY

HTML5

- Using HTML5, CSS3, and JavaScript to create platform-independent applications
  - Companies like Google have been developing APIs to make web applications more capable
  - Also work on desktops
- Hybrid applications
  - In mobile OS, use a web component that displays a web page with other non-web UI elements
- Games
  - Use HTML5's canvas element for drawing, WebGL for 3D graphics, etc.

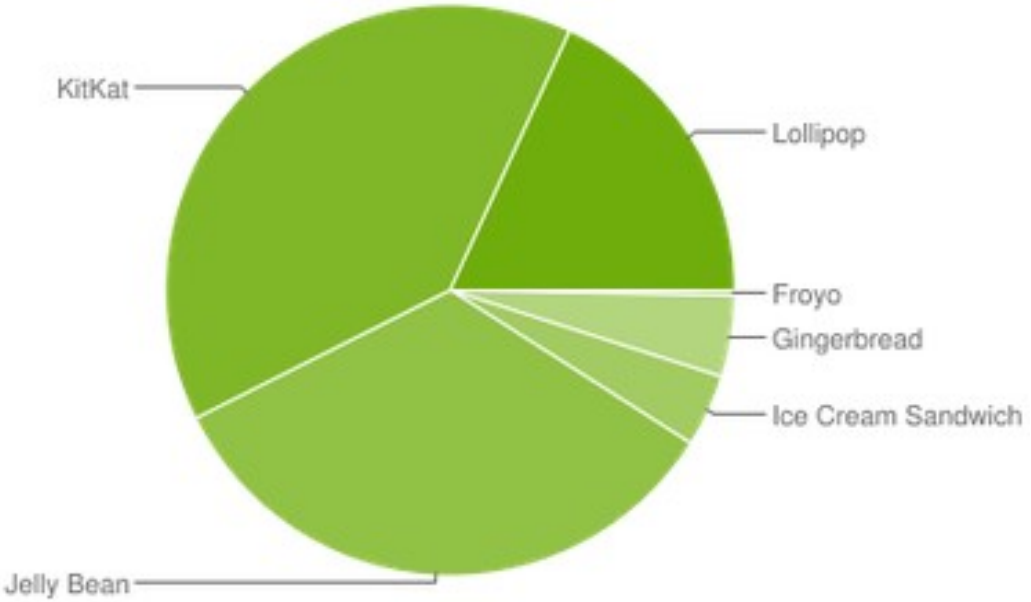
UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

Android

MVC Architecture

Android: Versions

- 1.5 – Cupcake
- 1.6 – Donut
- 2.1 – Éclair
- 2.2 – Froyo
- 2.3 – Gingerbread
- 3.x – Honeycomb
- 4.0 – Ice cream sandwich
- 4.1-4.3 – Jelly bean
- 4.4 – KitKat
- 5.0-5.1 – Lollipop
- 6.0 – Marshmallow



UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

https://developer.android.com/about/dashboards/index.html - August 2015

Android: Kernel

- The AndroidOS uses the Linux Kernel
- The kernel is modified by Google developers
- The kernel provides all low-level functionality:
  - Memory management and protection
  - File access
  - Device interface
  - Process management
  - Security

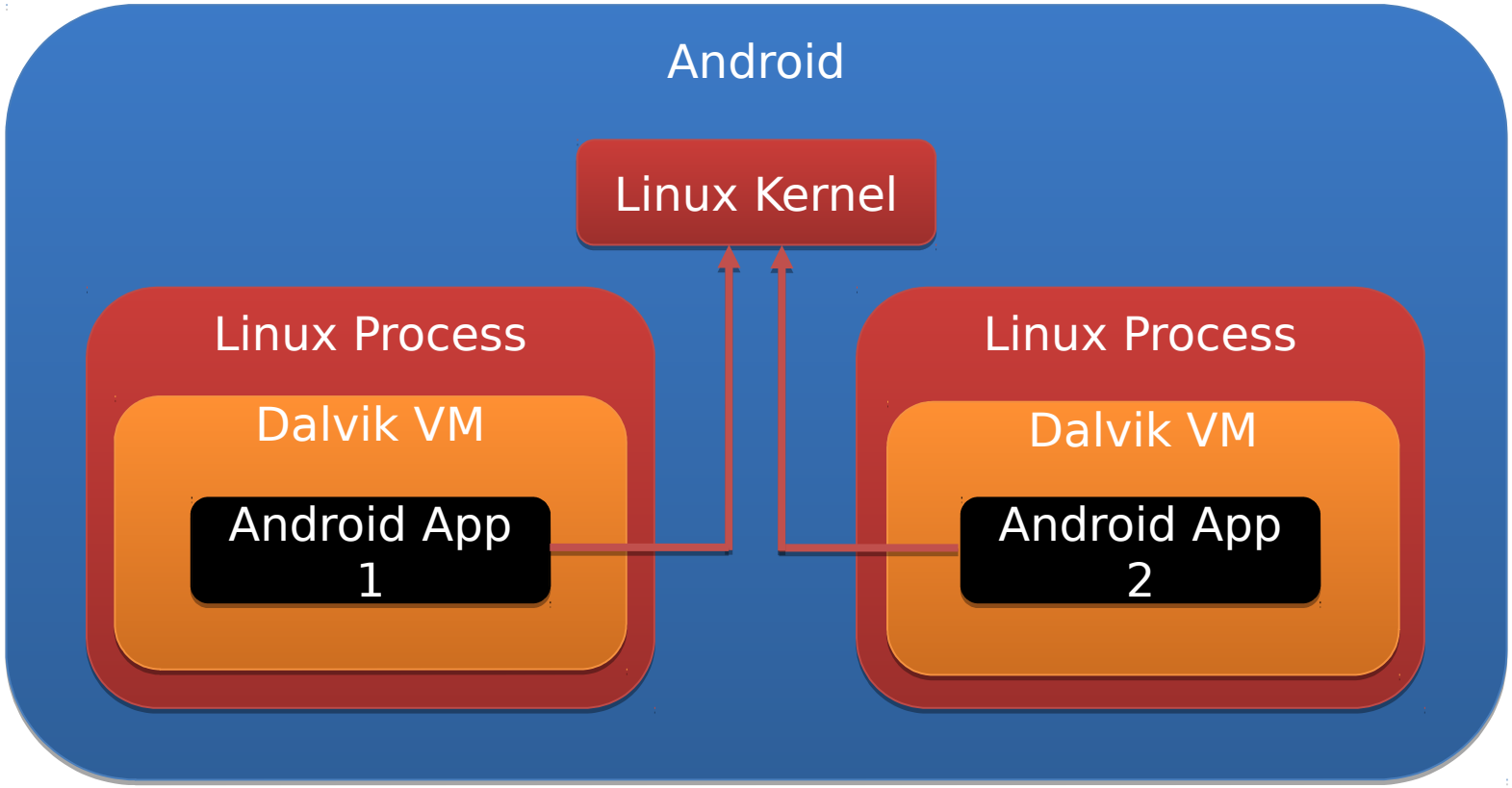
Android: Virtual Machine

- What is a virtual machine (VM)?
  - A sandbox is a constrained environment where programs can execute with little fear that they will (be able to) do anything malicious
  - A VM is a sandbox where uncompiled/interpreted code can be executed
  - The code is often turned into native machine language code as it executes (called just-in-time compilation)

Android: Virtual Machine

- Android uses the Dalvik virtual machine
  - A customized Java VM designed for performance, stability, and security
  - Each app runs in its own VM
  - Each VM is its own process (similar to tabs in Chrome)
    - The idea is that apps can't easily interfere with each other
    - Dalvik has been optimized to allow multiple VMs to be running simultaneously, while minimizing memory requirements

Android: Visualized

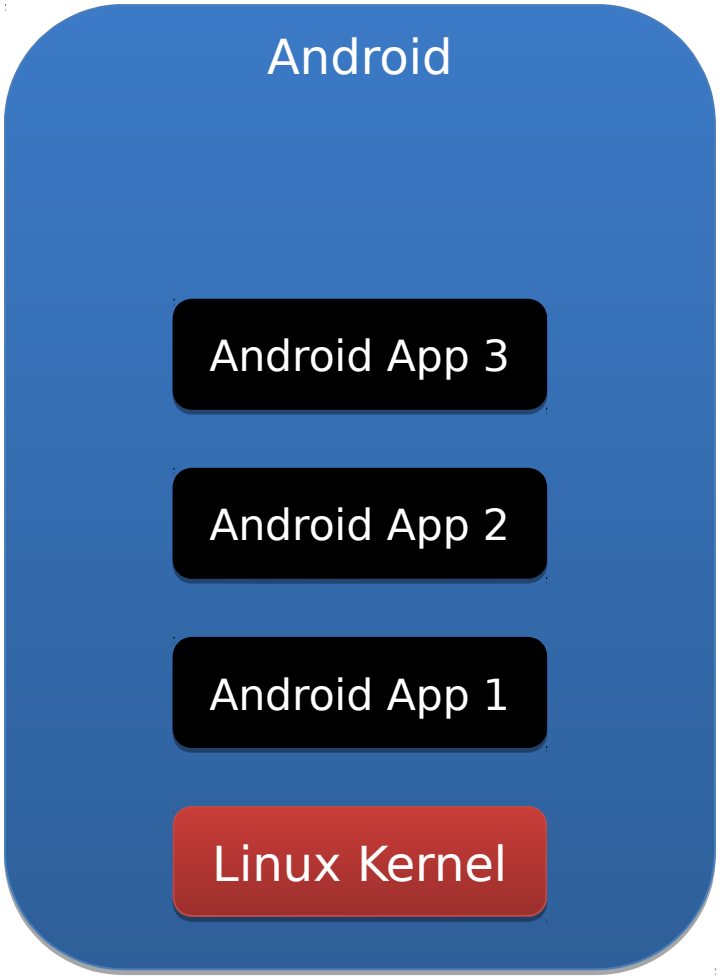


UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

Android: Memory Management

- Android apps (or even parts of the same app) can be run simultaneously
  - However, mobile devices do not have unlimited memory (typically 1-4GB)
  - When a foreground application needs memory, background applications can be removed from memory
    - This is important for understanding the life cycle of Activity objects (discussed later)
    - When pausing an Activity, you must store their state (data) so that the Activity can be re-created later, if necessary

Android: Memory Management

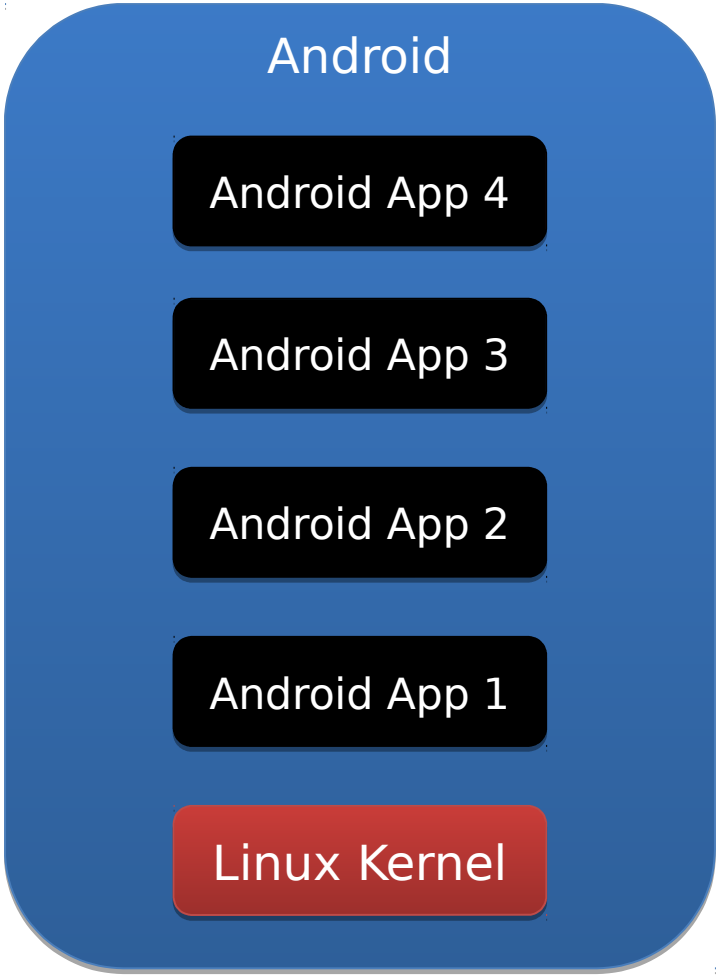


To be loaded:

Android App 4

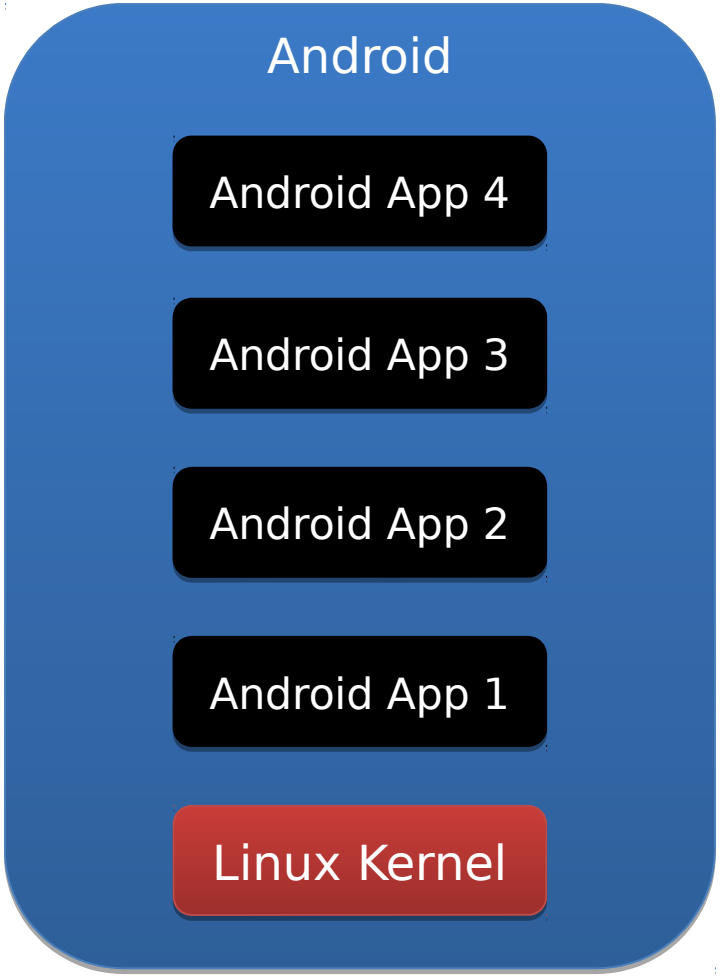
UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

Android: Memory Management



UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

Android: Memory Management

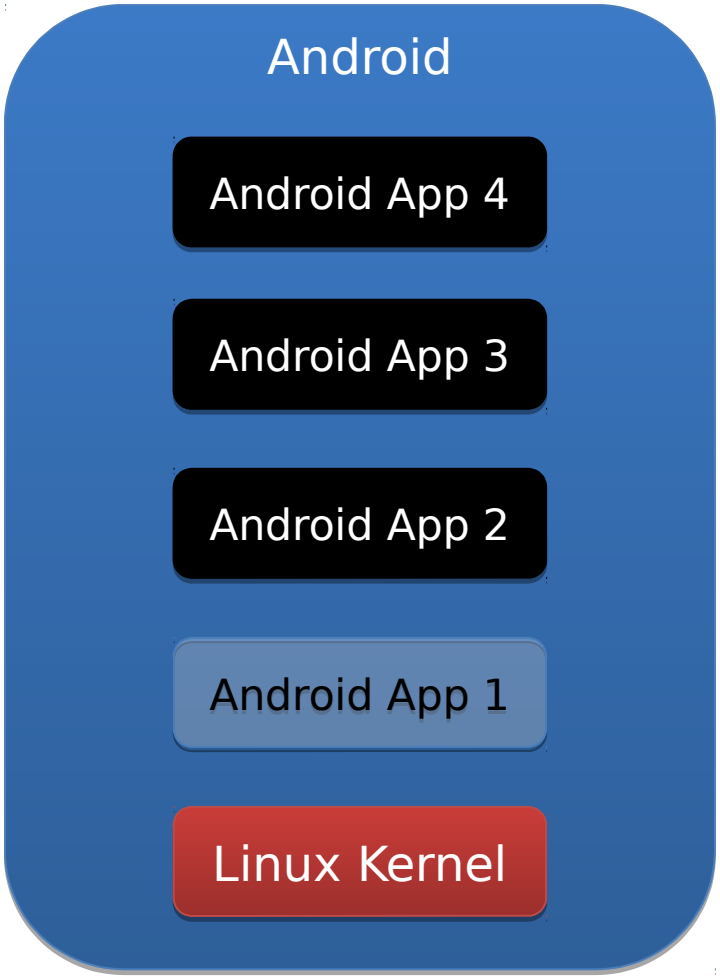


To be loaded:

Android App 5

UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

Android: Memory Management

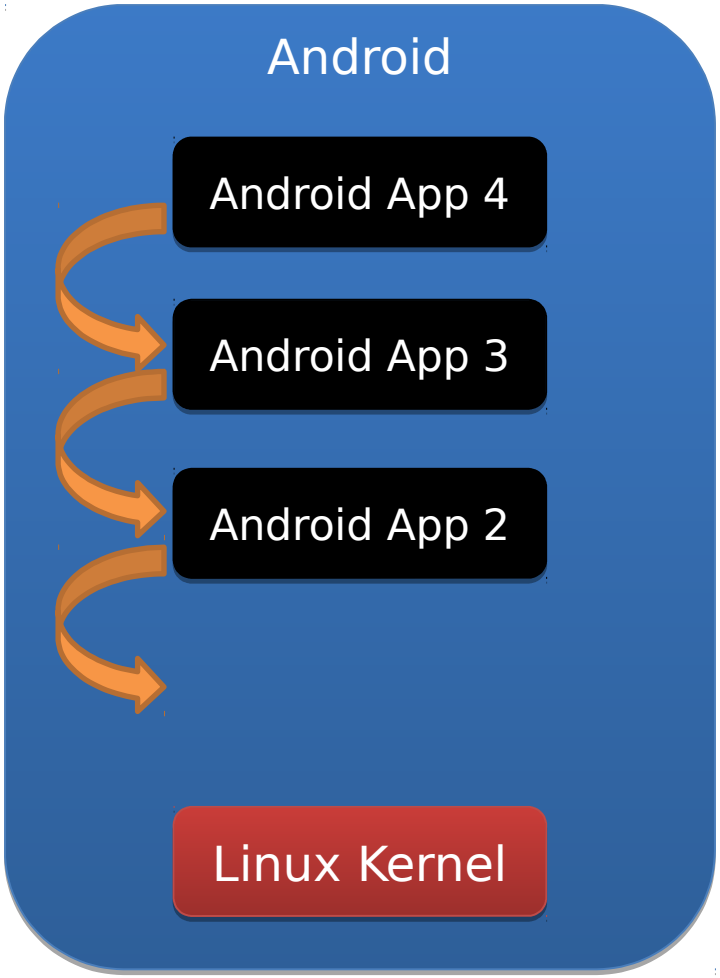


To be loaded:

Android App 5

UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

Android: Memory Management

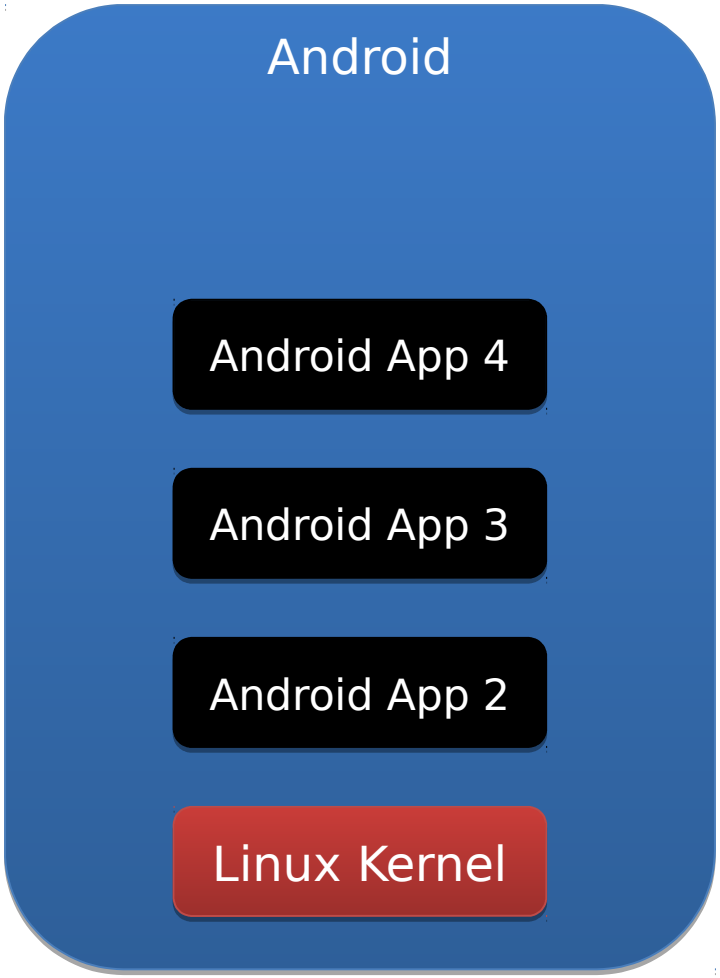


To be loaded:

Android App 5

UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

Android: Memory Management

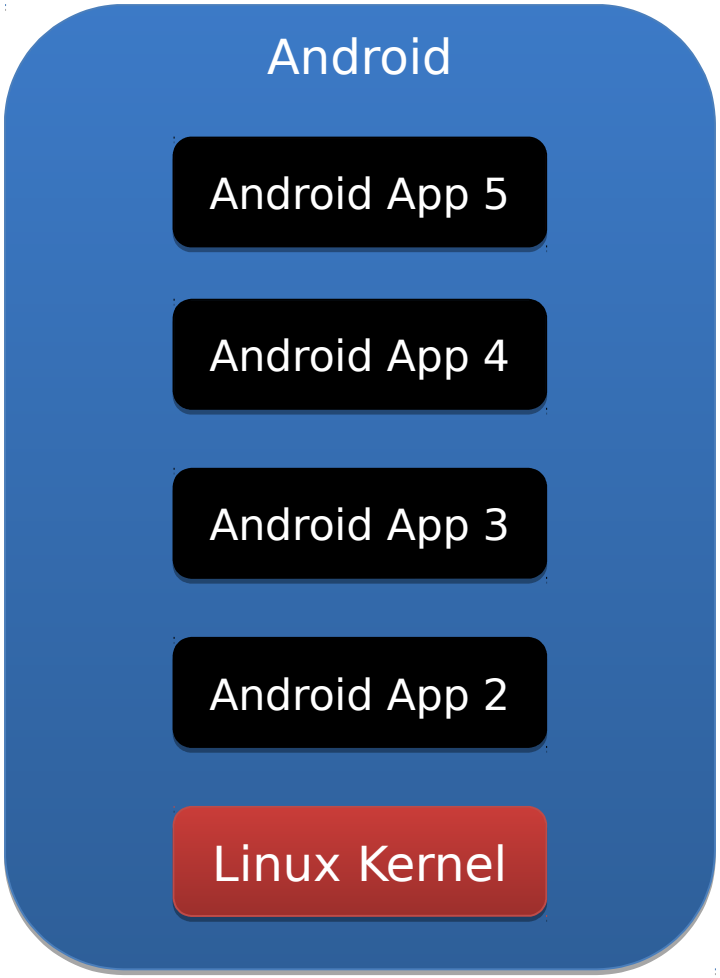


To be loaded:

Android App 5

UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

Android: Memory Management



UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

Architecture: MVC

- Model/View/Controller (MVC)
  - Model: Represents the data and behavior of the application (high level)
  - View: Represents the user interface elements that display information to the user
  - Controller: Collects information from the user and passes data between the Model and View

Architecture: MVC

- There are many variations of MVC
  - e.g. Multi-MVC, MVVM, MVP
- All variations generally have this in common:
  - Views and Model should not directly communicate
  - The Controller often knows about the Model and Views
  - The Views and Model often don't know about the Controller, or each other

Architecture: MVC

- MVC rationale:
  - If the Model knows about user input, then its behaviour can only operate where that type of input is used
  - If the Model knows about user output, then its behavior can only operate where that type of output is used
  - If the Views know about storage, then they can only display data stored that way

Architecture: MVC

- MVC rationale:
  - If the Model knows about user input, then its behaviour can only operate where that type of input is used
    - e.g. Gamepad versus keyboard/mouse versus accelerometer
  - If the Model knows about user output, then its behavior can only operate where that type of output is used
  - If the Views know about storage, then they can only display data stored that way

Architecture: MVC

- MVC rationale:
  - If the Model knows about user input, then its behaviour can only operate where that type of input is used
  - If the Model knows about user output, then its behavior can only operate where that type of output is used
    - e.g. Graphical UI, 3D graphics, text, HTML
  - If the Views know about storage, then they can only display data stored that way

Architecture: MVC

- MVC rationale:
  - If the Model knows about user input, then its behaviour can only operate where that type of input is used
  - If the Model knows about user output, then its behavior can only operate where that type of output is used
  - If the Views know about storage, then they can only display data stored that way
    - e.g. Database, cloud storage, files

MVC: Example

- Application: A social networking app
  - Model: Users, stream, status update, comment, like
  - View: Comment boxes, ads, image uploader, profile view, login form
  - Controller:
    - Collects data from submitted forms (e.g. submit comment)
    - Sends that data to the Model (e.g. add new comment)
    - Calls upon the correct View to display the results, such as an HTML page showing the new comment

UNIVERSITY OF ONTARIO
INSTITUTE OF TECHNOLOGY

Android

Android Application Architecture

Activities

- An application can have multiple activities
  - E.g. A navigation app
    - Main activity: Handles control over the map view
    - Search activity: Allows the user to search for points of interest
    - Preferences activity: Allows the user to control the settings of the application
  - Each activity typically shows one screen UI
    - The UI itself is a View
    - The activity that controls it is a Controller

Layouts

- A layout is a View
  - e.g. a screen, window, or dialog
  - They are defined in XML
  - You can combine any layout (or other view) together
  - The layout describes the relative position of the View components
- e.g. A search layout
  - A text field at top left, search button top right, and result table at the bottom

Basic Views

- Basic views are also Views
- Basic views represent basic user interface elements available to Android apps:
  - Buttons
  - Text fields
  - Checkboxes
  - Dropdown lists
  - Scrollable panel

Resources

- i18n

Resources

- i18n: internationalization
  - Your application is becoming popular
  - Statistics show that enough Koreans want to use it
    - Old way: Copy the program, convert all text/graphics to Korean, maintain both versions
    - New way: Keep strings and other resources separate, and let Android choose the correct versions
      - Text is put into XML files, which can be easily translated

Services

- Services are also Controllers, similar to activities
  - They handle outside input
  - They decide what to show to the user
- Primary differences:
  - Services run in the background
  - Services have a very minimal user interface
    - e.g. Notifications

Services: Example

- A music-playing service might handle playing music in the background
  - It doesn't require much of a user interface
  - A service could create a user interface with next track, pause, play, etc. in a notification control

Content Providers

- Content providers are Models that handle the data of an Android application
  - Accessing data from your application from another application
  - Accessing data for your whole application from multiple activities
- Content providers are optional
  - If you want to share data between applications, they are recommended

Content Providers: Example

- Google Fit records your activity throughout the day
  - Google Fit exposes this data to other applications
  - e.g. An application that keeps track of your calorie intake could access your Google Fit data to see if you burned more calories than you consumed

Intents

- An intent is a message
  - Intents are often sent between activities
  - Intents are usually broadcast
    - Senders don’t know anything about the recipient
- You can use an intent to invoke another activity
  - One activity of an app may show another
  - The operating system (or something else) may launch your application

Intents: Examples

- Send an intent to launch a different activity of the same application
  - e.g. show the Preferences activity when the menu item is selected
- Send an intent to launch an activity of another application
  - e.g. your app could open a browser to a specific page
  - e.g. another app may open your app to display an incoming SMS
  - e.g. Android may launch your application on a timer

Broadcast Receivers

- A broadcast receiver listens for intents
  - Similar to a web server listening for incoming web requests from browsers
  - Broadcast receivers tell Android what kind of intents they'd like to handle
    - e.g. Generic, e.g. Open Browser
    - e.g. App-specific, e.g. Display order details
  - Android waits for a matching intent to be broadcast, and if so it forwards it to your activity
    - If the activity is not running, it is started

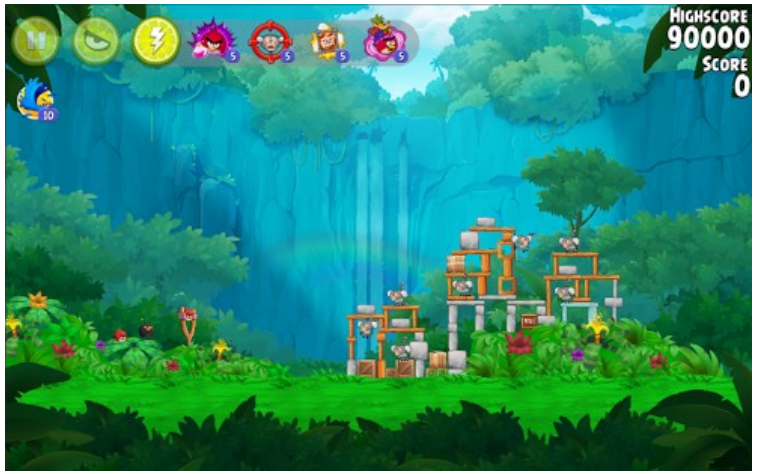
Broadcast Receivers: Example

- Create a broadcast receiver to handle incoming SMS messages
  - In this way, you can create your own app to handle what is built-in on other mobile operating systems (e.g. iOS)
  - Nearly all non-kernel features can be replaced, making Android very flexible

Manifest

- Every Android app must have a manifest file
  - The manifest contains the app's meta-data
  - It describes what the application needs:
    - What permissions does it require? (e.g. file read)
    - What platform does it require? (e.g. Lollipop)
    - What hardware capabilities are necessary? (e.g. GPS)
    - What events (intents) does it want to receive? (e.g. show map)

Example: An Angry-Birds Game



- Layout:
  - Displays the game board
- View:
  - Canvas (drawing game board)
  - ImageButton (pause)
  - TextView (high score)
- Activity:
  - Handles touch events on the game board, shows high scores, animated birds
- Model:
  - Maintains the score, block and bird positions

UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

Image courtesy of Rovio Entertainment Ltd

Example: A Navigation App

- Layout:
  - Displays the map
- View:
  - Map (Google API view)
  - ImageButton (zoom in/out)
  - EditText (search keywords)
- Activity:
  - Handles touch gestures on the game board, handles button presses
- Model:
  - Maintains the coordinates, points of interest, start, destination, other waypoints, directions, and preferences

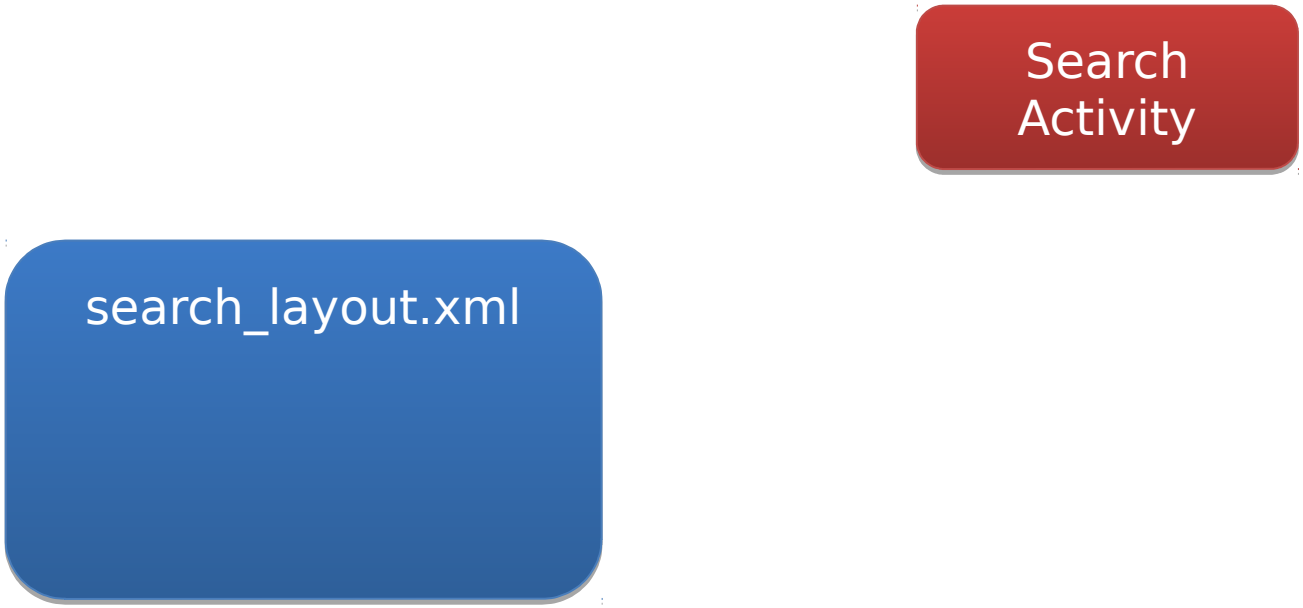
UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

Image courtesy of Alphabet Inc.

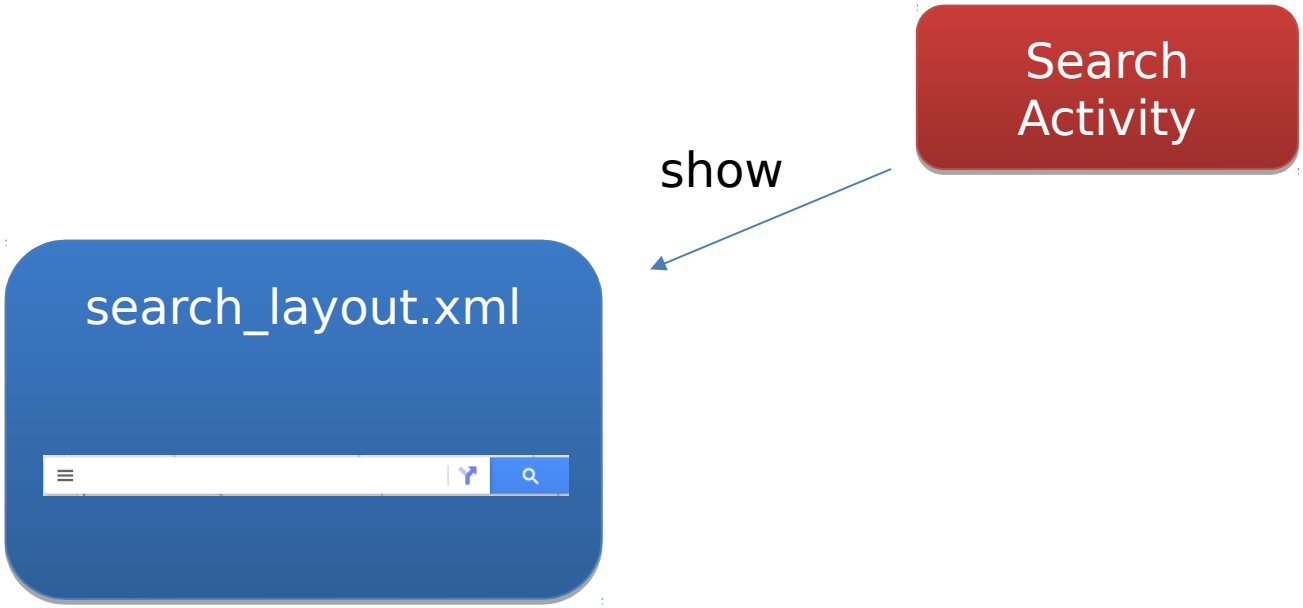
Example: A Navigation App

- Application is launched by the OS sending an intent that SearchActivity is set to receive
- SearchActivity displays search_layout.xml
- search_layout.xml contains:
  - A text field (EditText) for entering a waypoint name, address
  - A search button
- When the search button is pressed, SearchActivity collects the search text from the text field
- SearchActivity finds the coordinates, and creates a new intent, meant for the MapActivity (passing the coordinates)

Example: A Navigation App

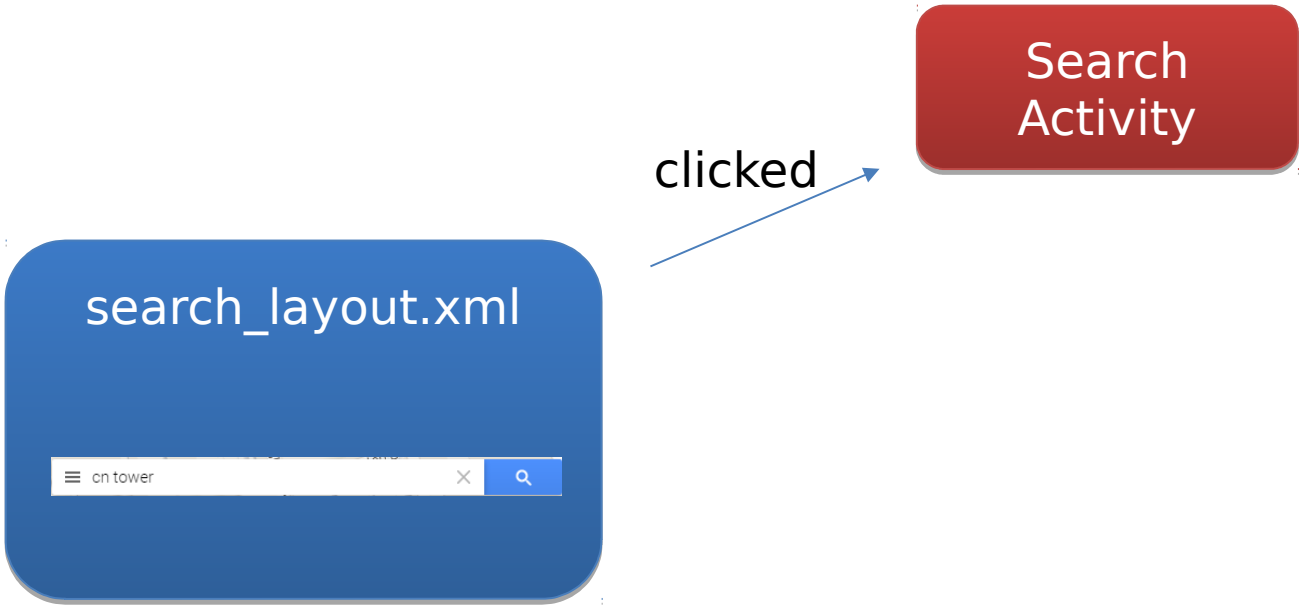


Example: A Navigation App



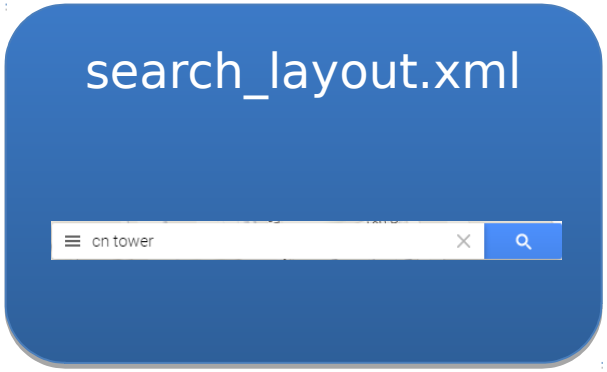
UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

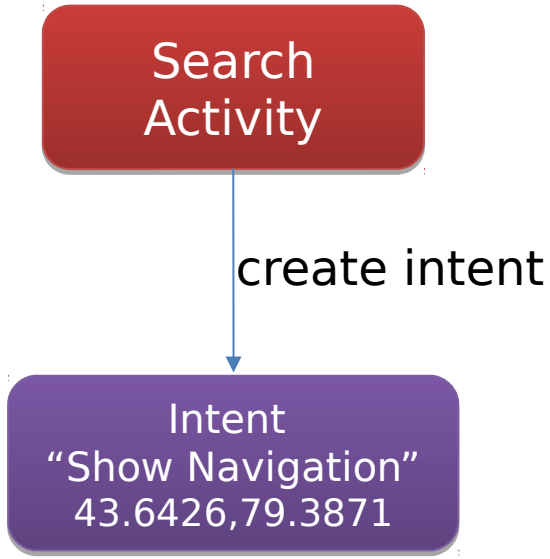
Example: A Navigation App



UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

Example: A Navigation App





UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

Example: A Navigation App

Search Activity

search_layout.xml

Intent
“Show Navigation”
43.6426,79.3871

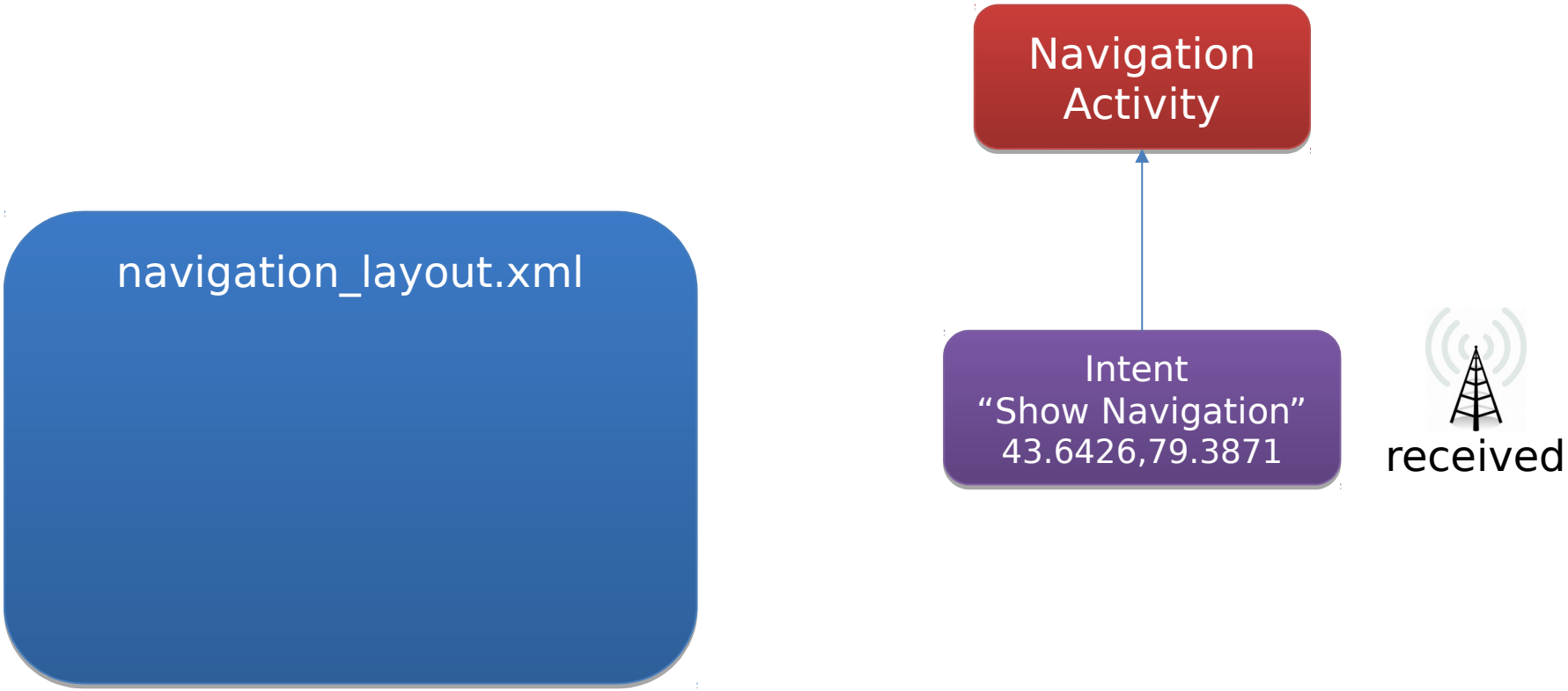
broadcast

UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

Example: A Navigation App

- NavigationActivity receives the intent and the coordinates
- NavigationActivity shows navigation_layout.xml
- navigation_layout.xml contains a MapView
- NavigationActivity extracts the coordinates from the intent
- NavigationActivity tells the MapView to centre on those coordinates
- MapView displays the map to the user

Example: A Navigation App



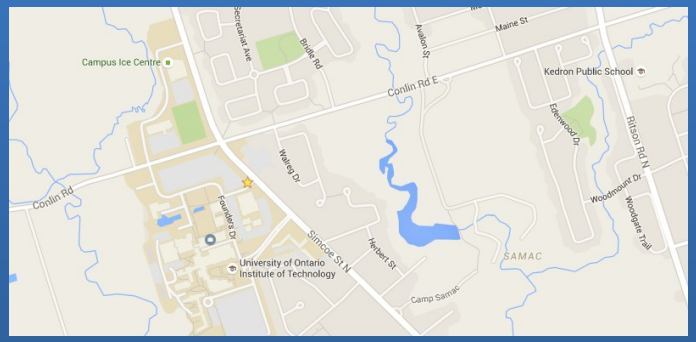
UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

Example: A Navigation App

Navigation Activity

show

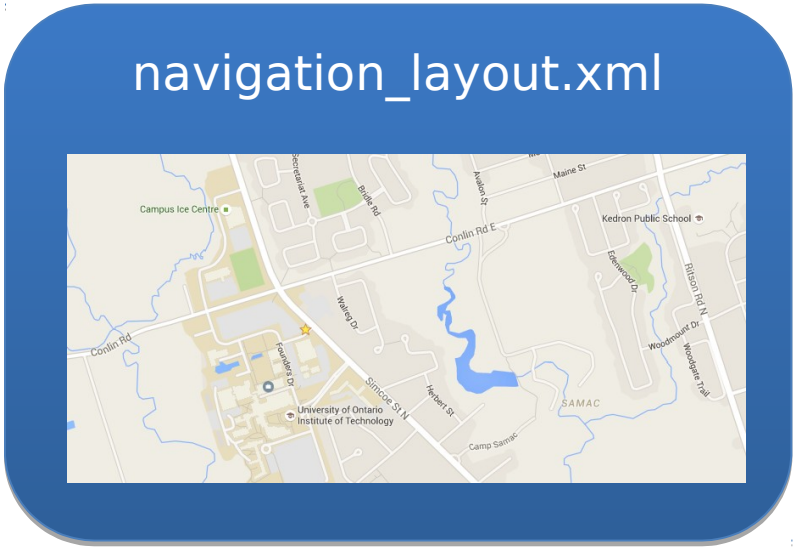
navigation_layout.xml

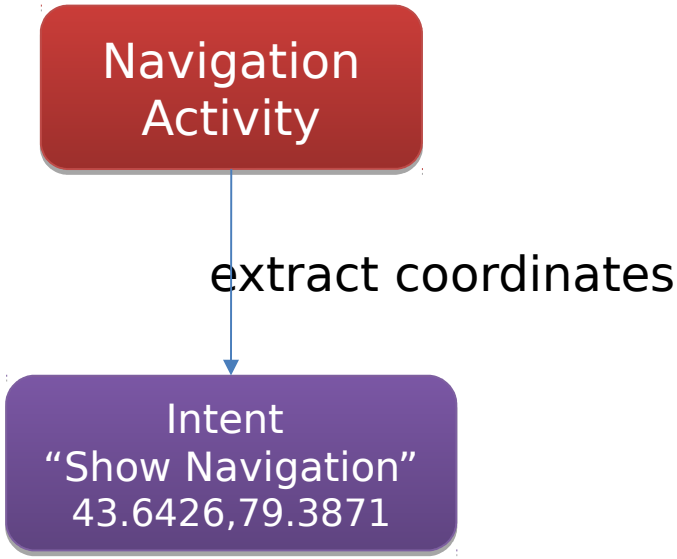


Intent
“Show Navigation”
43.6426,79.3871

UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

Example: A Navigation App





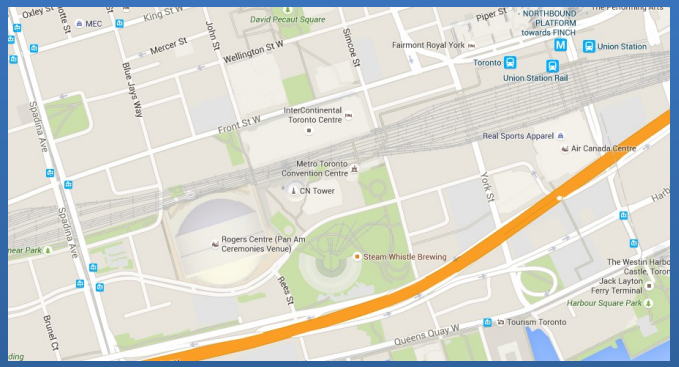
UNIVERSITY OF ONTARIO
INSTITUTE OF TECHNOLOGY

Example: A Navigation App

show coordinates:
43.6426,79.3871

Navigation Activity

navigation_layout.xml



Intent
“Show Navigation”
43.6426,79.3871

UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

Wrap-Up

- Mobile devices
  - Capabilities
  - Limitations
- Mobile platforms
- Introduction to the Android platform