ANDROID APP MODEL CONT.

Lecture 8

COMPSCI 702

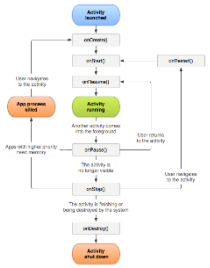
Security for Smart-Devices

Muhammad **Rizwan** Asghar

March 17, 2021

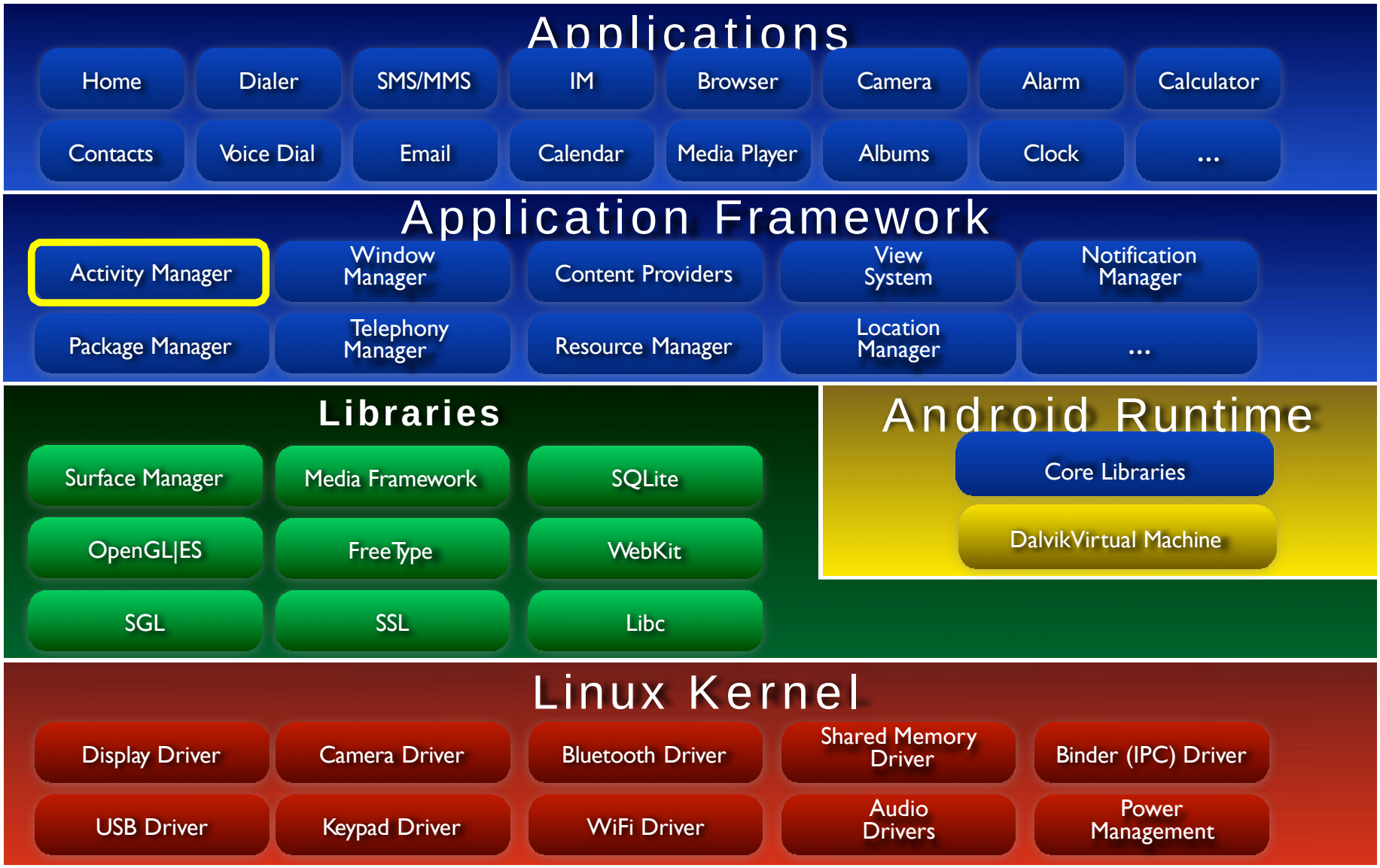
THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

ACTIVITY MANAGER



- Activity Manager is responsible for creating, destroying, and managing activities
- When the user starts an application for the first time, the Activity Manager will create its activity and put it onto the screen
- Later, when the user switches screens, the Activity Manager will move that previous activity to a holding place
- This way, if the user wants to go back to an older activity, it can be started more quickly
- Older activities that the user has not used in a while will be destroyed in order to free more space for the currently active one
- This mechanism is designed to help improve the speed

ACTIVITY MANAGER

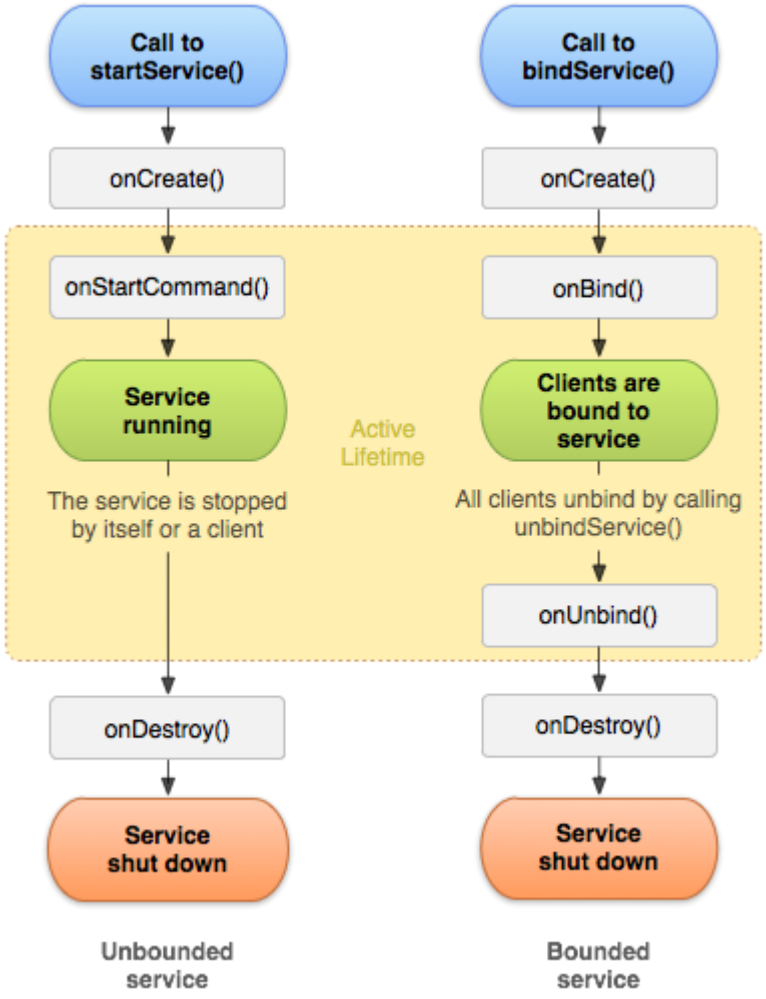


SERVICE

- A background process that has no user interface
- Typically used to perform some long-running operation
- Examples
  - Downloading files or fetching emails from a server
  - Playing music
- Can be local to the app or remote (provided by other apps)
- Services can define a remote interface using the Android Interface Definition Language (AIDL)
- AIDL compiler creates skeleton for implementation of the service (stub)
- Services are started and stopped on demand

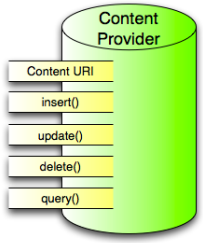
SERVICE TYPES

- App components start services
- An unbounded service is stopped by itself or a client
- A bounded service acts as a server: The app component (the client), logs in (binds) to the server, consumes the service, and then logs out (unbinds)
- Usage
  - Use an unbounded service to do work if the app components do not require interaction with the service again
  - Use a bounded service if the app components require interaction with the service



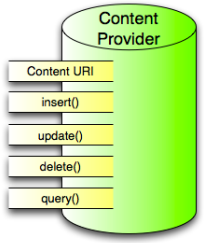
Source: android.com

CONTENT PROVIDERS



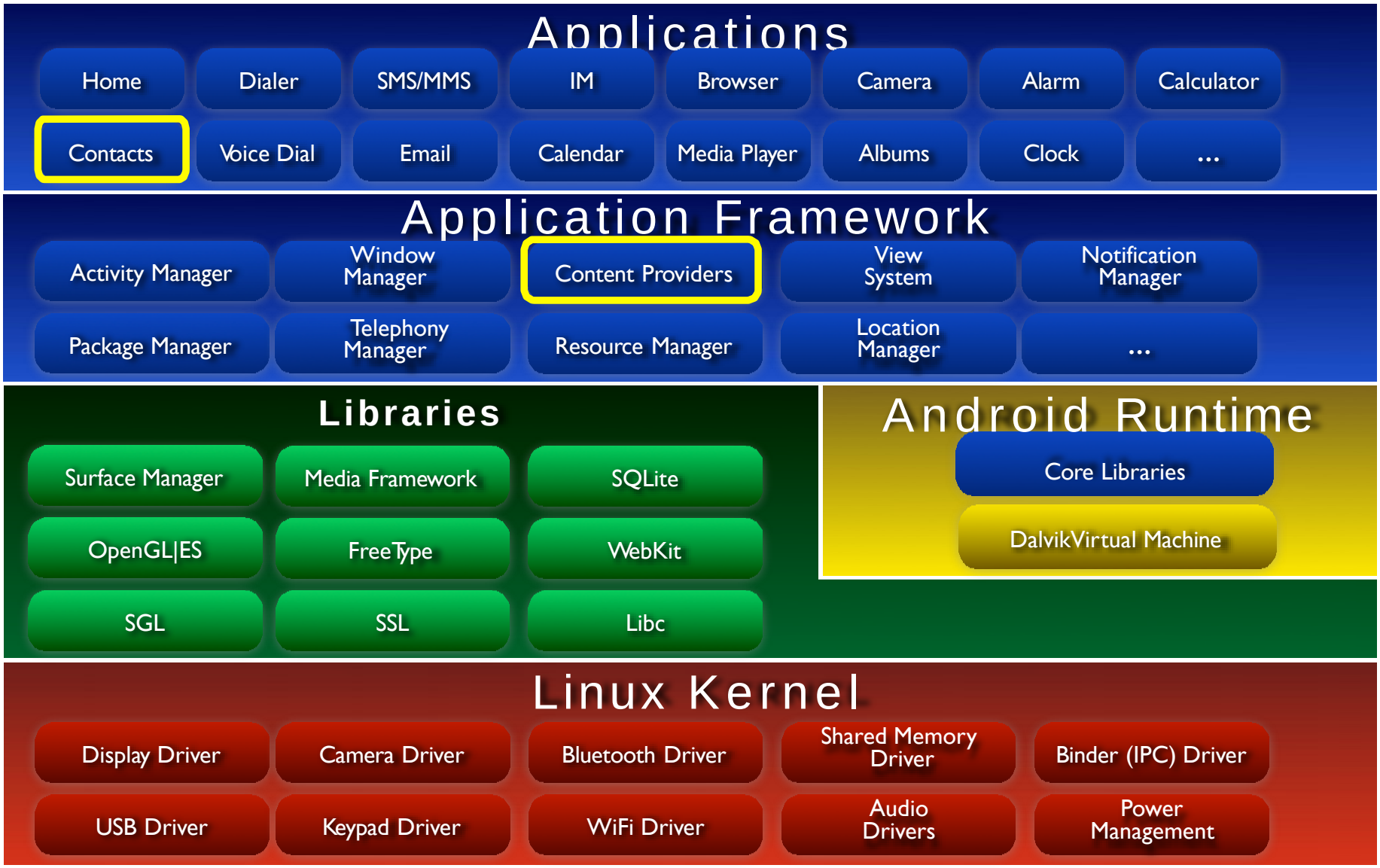
- Content providers are interfaces for sharing data between apps
- Relatively simple interfaces, with the standard select(), insert(), update(), and delete()
- Content providers must be declared in the manifest file using the <provider> tag
- Content providers are accessed by the URI
  - content://<authority>/<resource>

CONTENT PROVIDER: EXAMPLES



- **Contacts** provider is a content provider that exposes all user contact data to various applications
- **Settings** provider exposes system settings to various applications, including the built-in Settings application
- **Media** store is responsible for storing and sharing various media, such as photos and music, across various applications

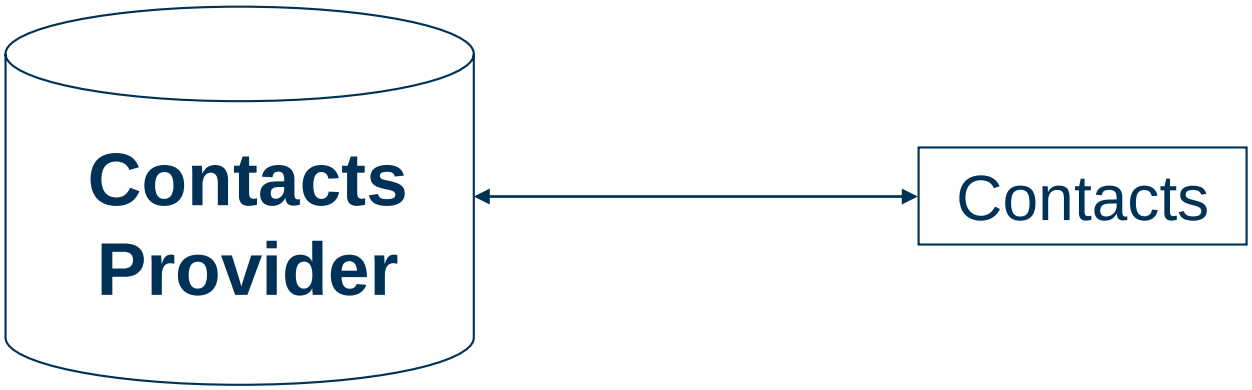
CONTENT PROVIDERS AND CONTACT



CONTACTS PROVIDER

- The Contacts app uses contacts provider, a totally separate application, to retrieve data about users' contacts

- The Contacts app itself does not have any contacts data



BROADCAST RECEIVER: OVERVIEW

- A broadcast receiver is a component that responds to system-wide events
- A mailbox for broadcast intent messages
  - Define intent filters to indicate what kinds of messages to receive
  - An intent includes an action string and a category
- Events can originate from the system
  - E.g., low battery
  - SMS arrival or
  - Change in network connectivity
- Events can also originate from a user application
  - E.g., announcing that background data update has completed

BROADCAST RECEIVER: DETAILS

- Broadcast receivers are Android's implementation of a system-wide publish/subscribe mechanism
  - Publishers are user apps or the system
  - Typically, subscribers are user apps
- A subscribing application can subscribe by indicating intent filters in the application manifest or registering dynamically
- The receiver will receive a triggered event if there is a subscription for it
- Generally, all events are broadcasted to a number of receivers that subscribe for the event

BROADCAST RECEIVER: REGISTRATION

- A broadcast receiver has to register with the Activity Manager and the Package Manager

- Registration can be done through
  - The manifest file
  - Programmatically

REGISTRATION USING MANIFEST

```
<receiver android:name="‘MsgListener" >
   <intent-filter>
         <action
           android:name="‘compsci702.intent.action.BROADCAST" />
   </intent-filter>
</receiver>
```

REGISTRATION USING MANIFEST

```
<receiver android:name="“MsgListener" >
   <intent-filter>
        <action
          android:name="“compsci702.intent.action.BROADCAST" />
   </intent-filter>
</receiver>
```

Class responsible for processing the intent

REGISTRATION USING MANIFEST

```
<receiver android:name="“MsgListener" >
   <intent-filter>
         <action
           android:name="“compsci702.intent.action.BROADCAST" />
   </intent-filter>
</receiver>
```

Filter specifying intents to be received

REGISTRATION THROUGH PROGRAM

```
IntentFilter filter = new IntentFilter();
filter.addAction(``compsci702.intent.action
.BROADCAST'');
receiver = new BroadcastReceiver();
//@Override public void onReceive(Context
context, Intent intent)
{
System.out.println(``message received'');
}
};
registerReceiver(receiver, filter);
```

REGISTRATION THROUGH PROGRAM

```
IntentFilter filter = new IntentFilter();
filter.addAction(``compsci702.intent.action
.BROADCAST'');
receiver = new BroadcastReceiver();
//@Override public void onReceive(Context
context, Intent intent)
{
System.out.println(``message received'');
}
};
registerReceiver(receiver, filter);
```

Filter specifying intents to be received

REGISTRATION THROUGH PROGRAM

```
IntentFilter filter = new IntentFilter();
filter.addAction(``compsci702.intent.action
.BROADCAST'');
receiver = new BroadcastReceiver();
//@Override public void onReceive(Context
context, Intent intent)
{
System.out.println(``message received'');
}
};
registerReceiver(receiver, filter);
```

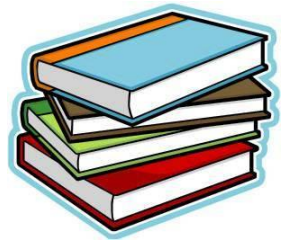
Action performed when the intent is received

REGISTRATION THROUGH PROGRAM

```
IntentFilter filter = new IntentFilter();
filter.addAction(``compsci702.intent.action
.BROADCAST’’);
receiver = new BroadcastReceiver();
public void onReceive(Context context,
Intent intent)
{
System.out.println(``message received'');
}
};
registerReceiver(receiver, filter);
```

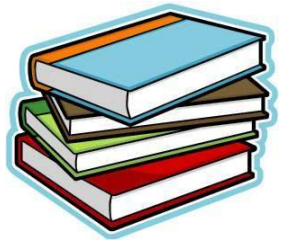
Registering broadcast receiver and the filter

RESOURCES



- **Chapter 1** of
  Android Security Internals: An In-Depth Guide to Android's Security Architecture
  Elenkov, Nikolay
  First Edition
  No Starch Press 2014
  ISBN:1593275811 9781593275815

RESOURCES (2)



- Enck, William, Machigar Ongtang, and Patrick McDaniel
  Understanding Android Security
  IEEE Security & Privacy 1 (2009): 50-57

- Intents and intent filters: https://developer.android.com/guide/components/intents-filters

ACKNOWLEDGEMENT

- Some slides on registration of broadcast receiver are based on the lecture delivered by Giovanni Russello, thanks to him!

Questions?

Thanks for your attention!