

Android Services & Local IPC: Activity & Service Communication

Douglas C. Schmidt

d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt



Professor of Computer Science

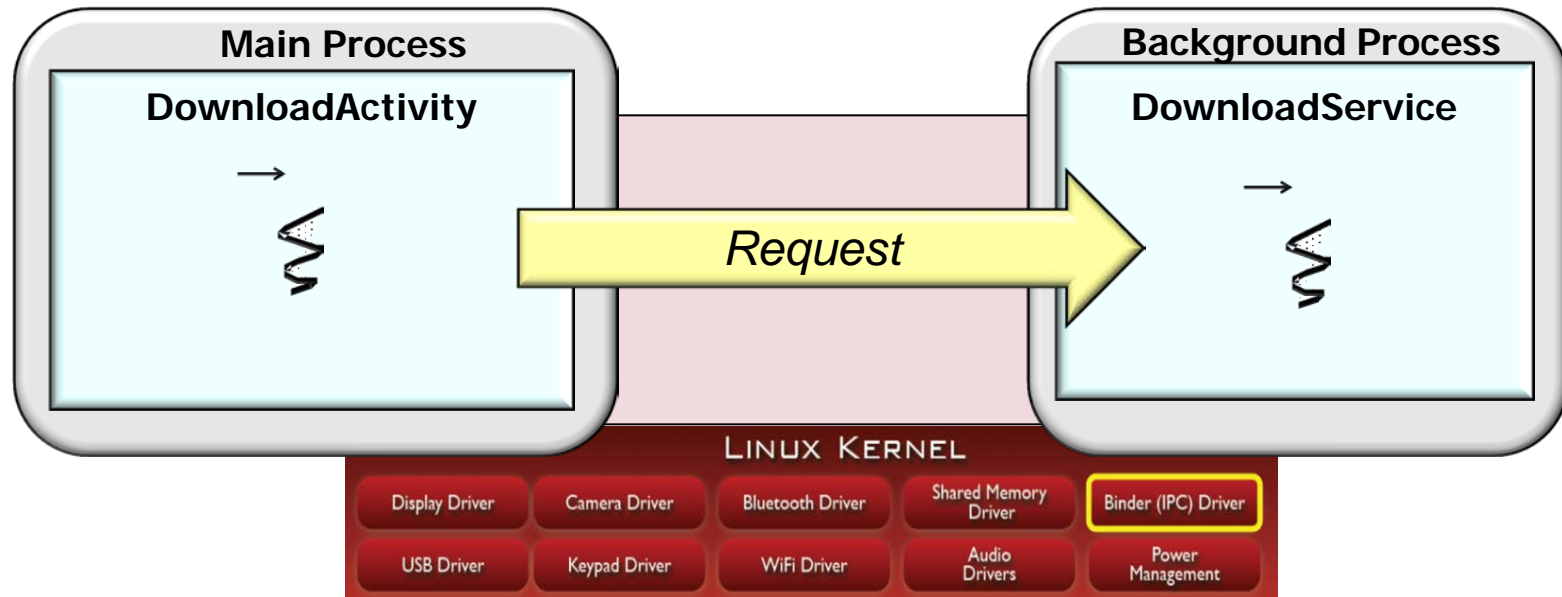
Institute for Software
Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA



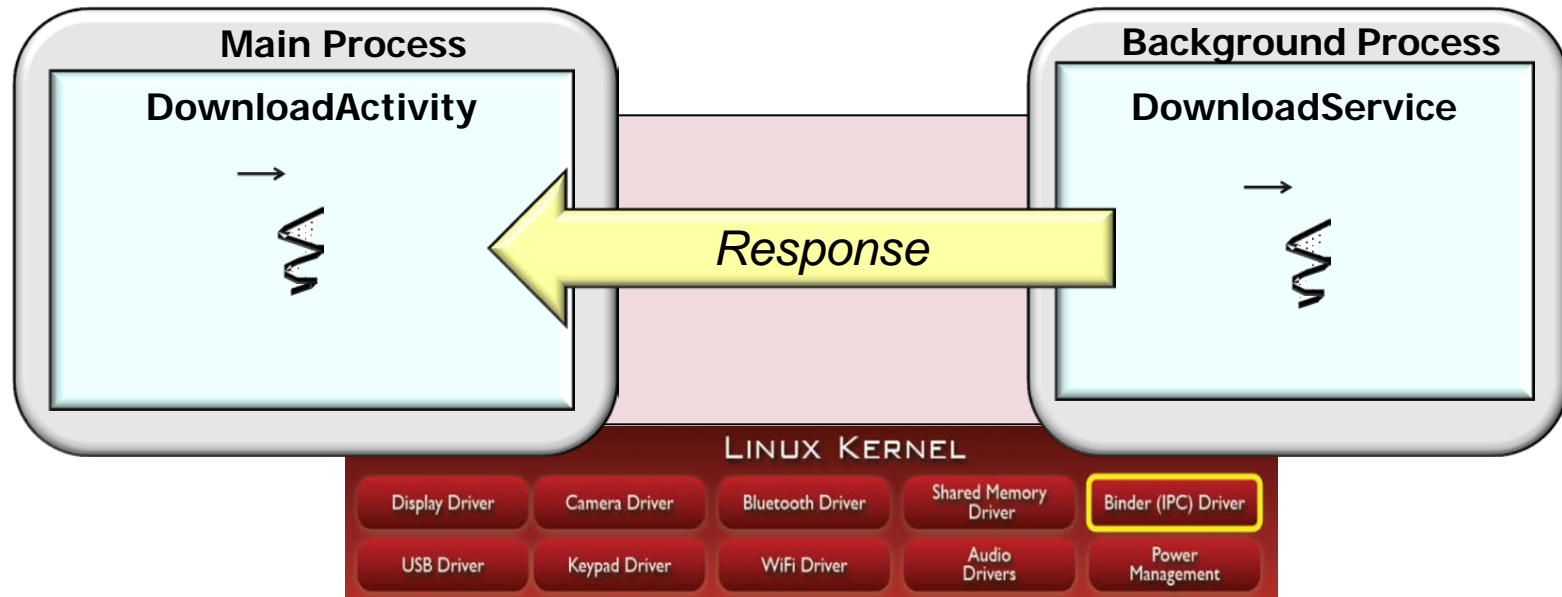
Learning Objectives in this Part of the Module

- Understand the key Android mechanisms, frameworks, & patterns that Activities & Services use to communicate



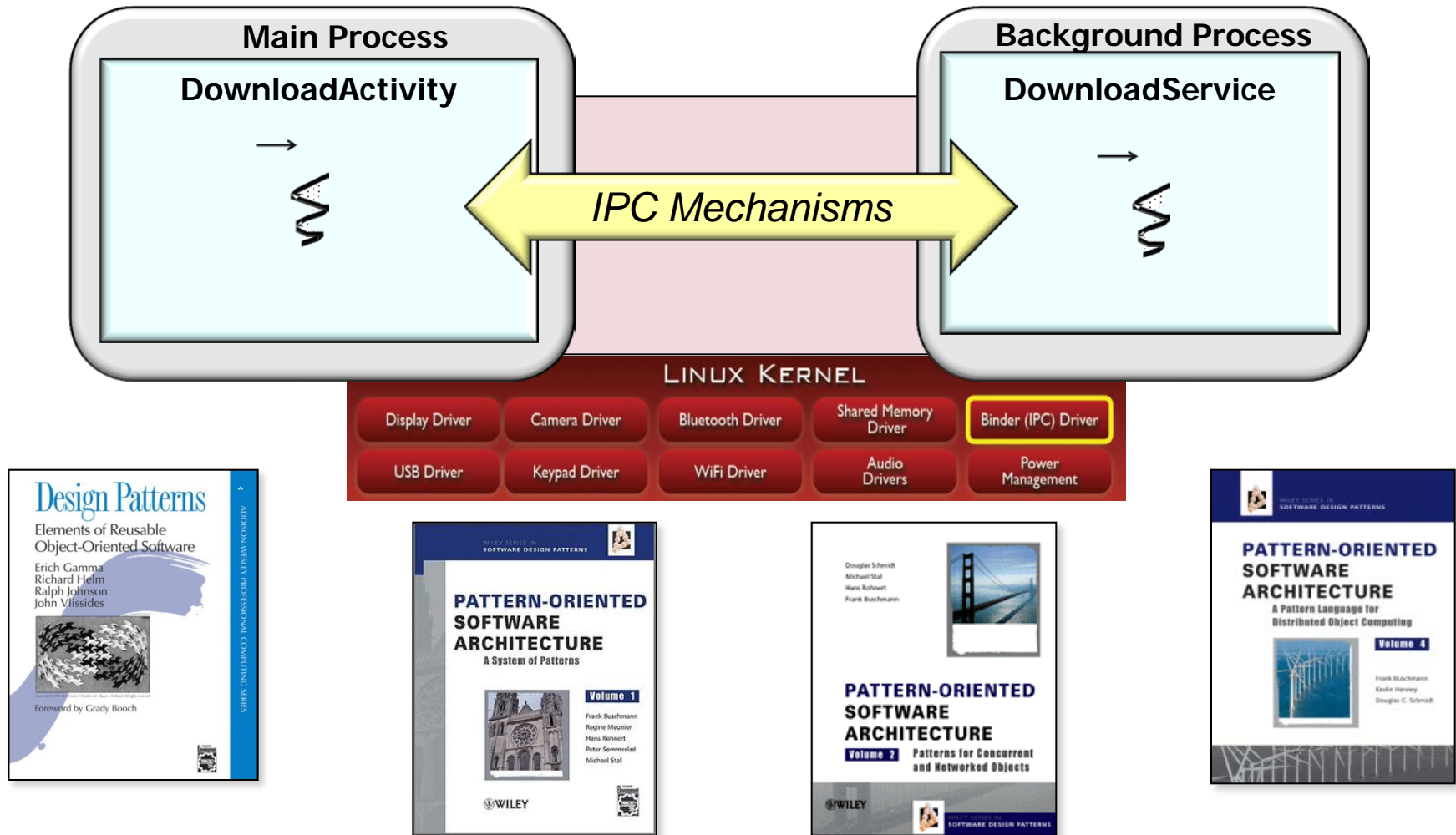
Learning Objectives in this Part of the Module

- Understand the key Android mechanisms, frameworks, & patterns that Activities & Services use to communicate



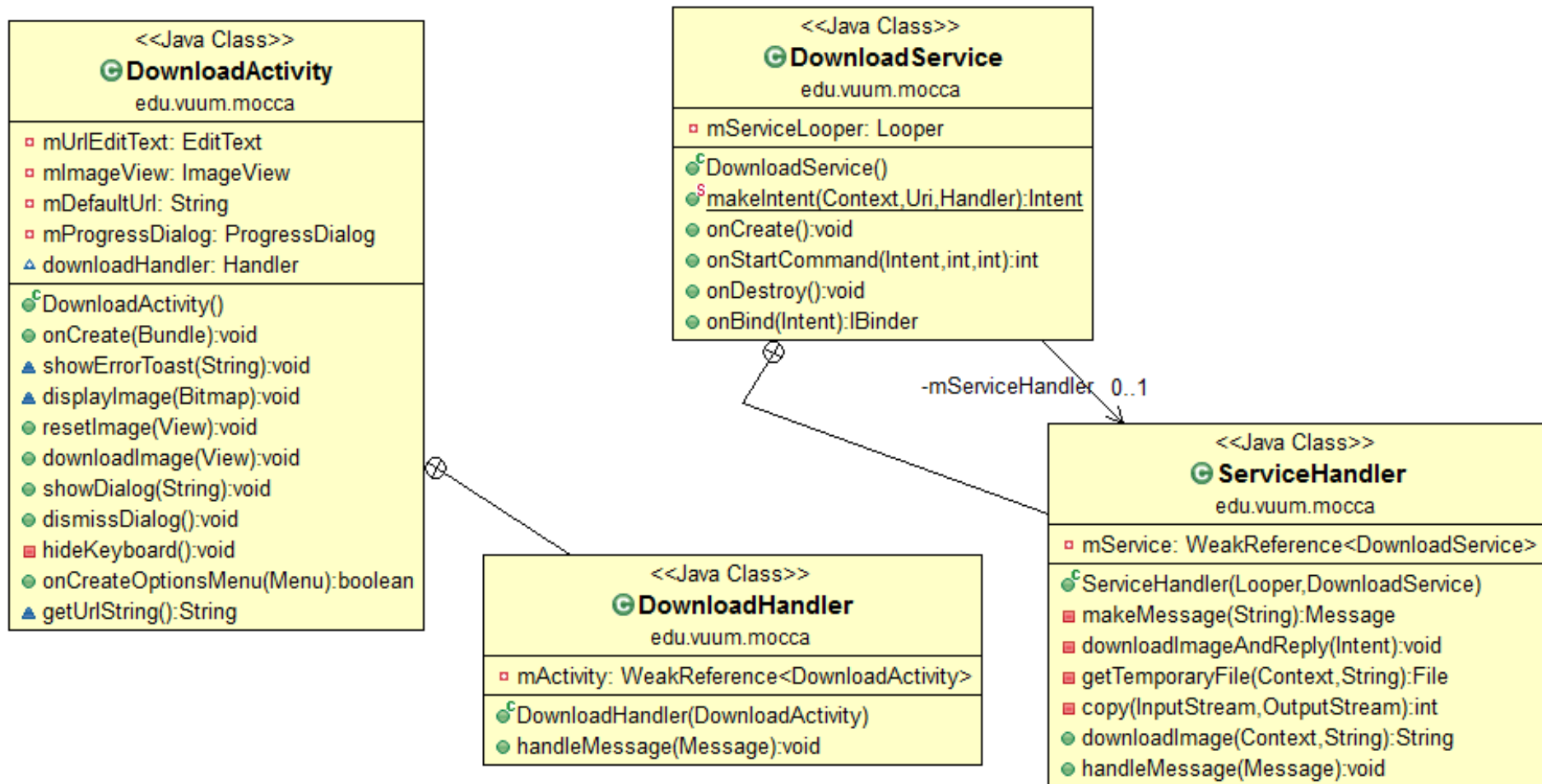
Learning Objectives in this Part of the Module

- Understand the key Android mechanisms, frameworks, & patterns that Activities & Services use to communicate



Overview of the Download Application

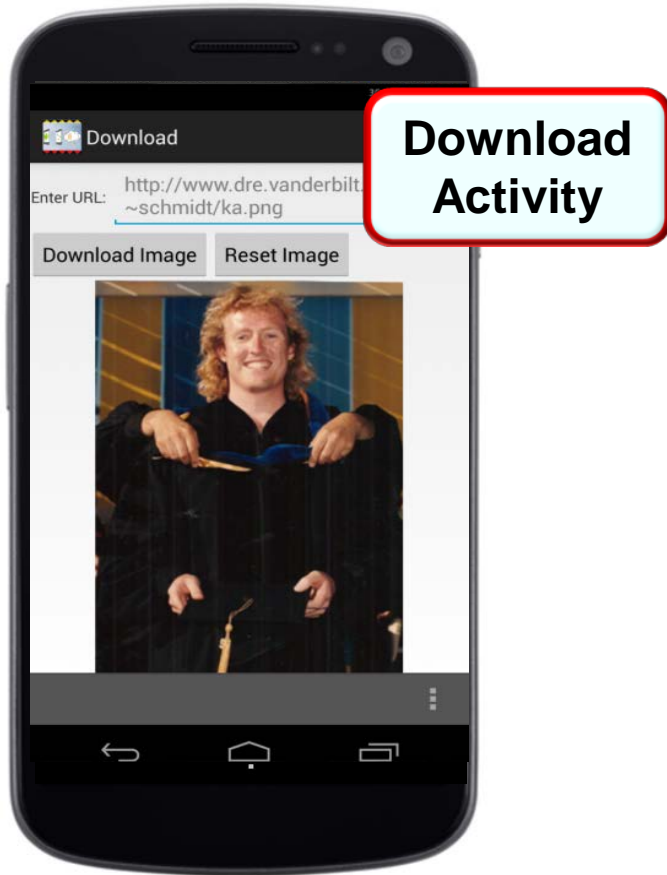
Overview of the Download Application



See github.com/douglasraigschmidt/POSA-14/tree/master/ex/DownloadApplication

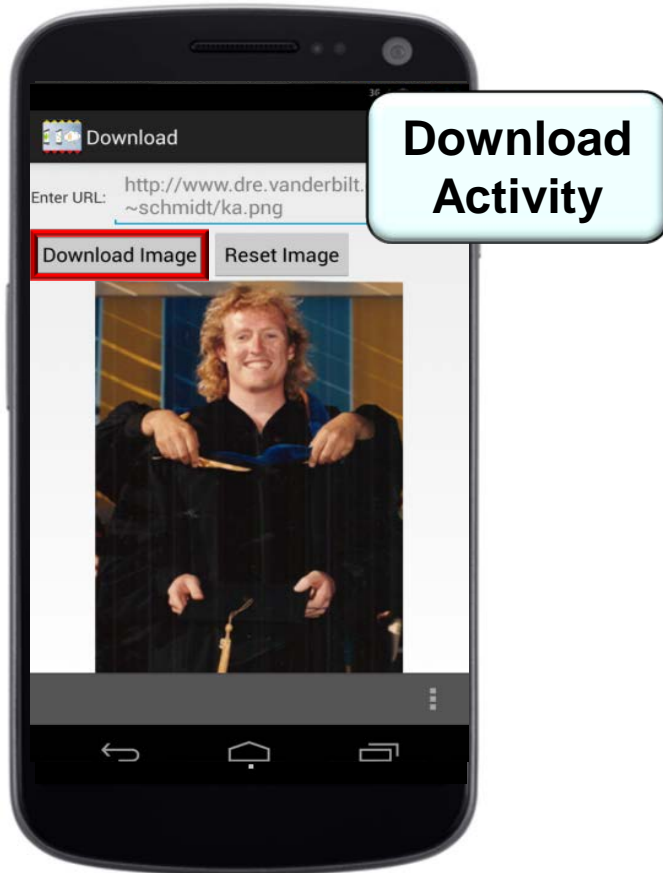
Overview of the Download Application

1. DownloadActivity uses startService() to launch a DownloadService



Overview of the Download Application

1. DownloadActivity uses startService() to launch a DownloadService



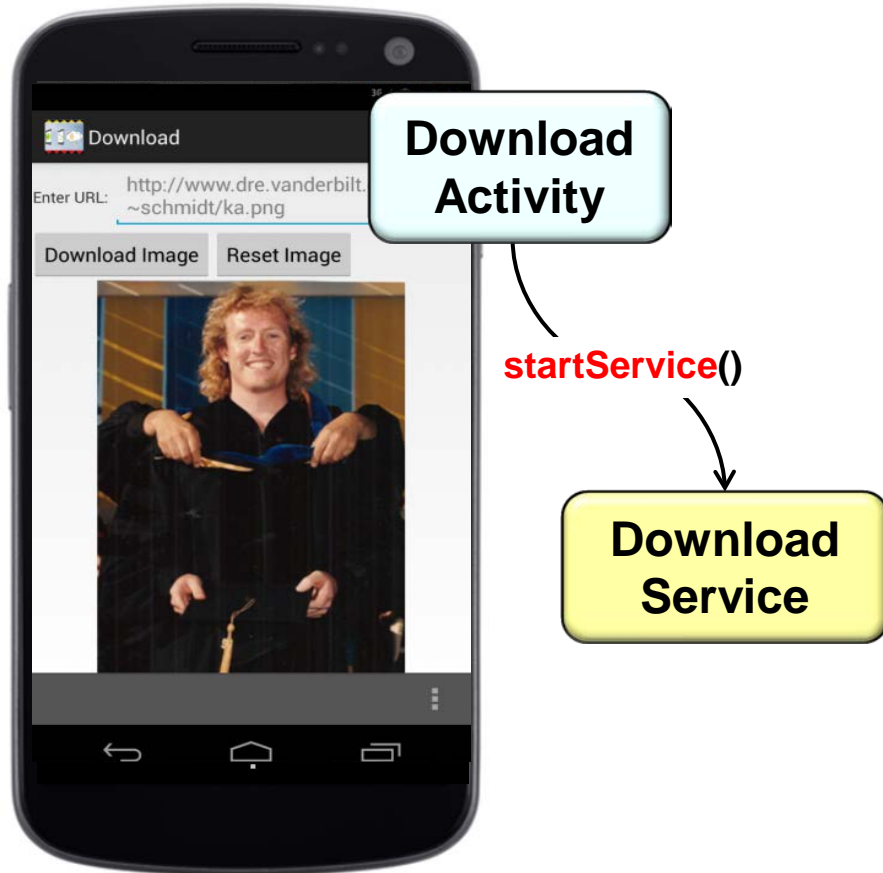
Overview of the Download Application

1. DownloadActivity uses startService() to launch a DownloadService



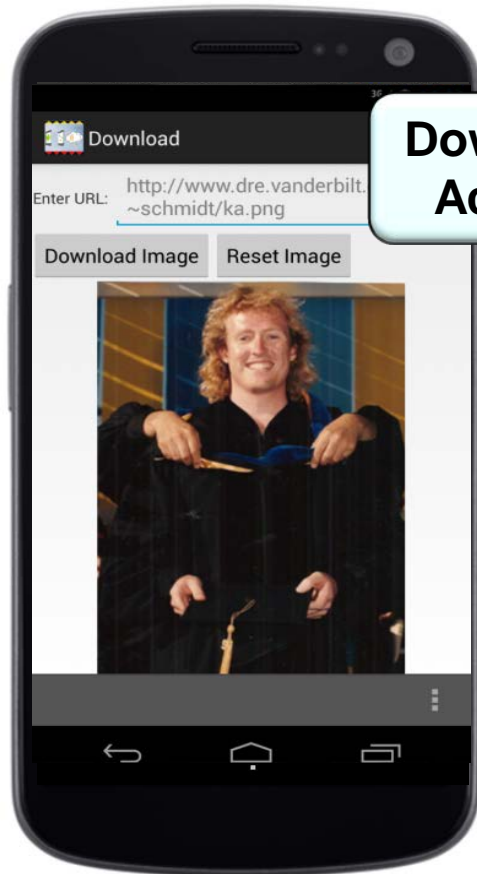
Overview of the Download Application

1. DownloadActivity uses startService() to launch a DownloadService



Overview of the Download Application

1. DownloadActivity uses startService() to launch a DownloadService
2. The DownloadService retrieves the image & stores it in a file on the device



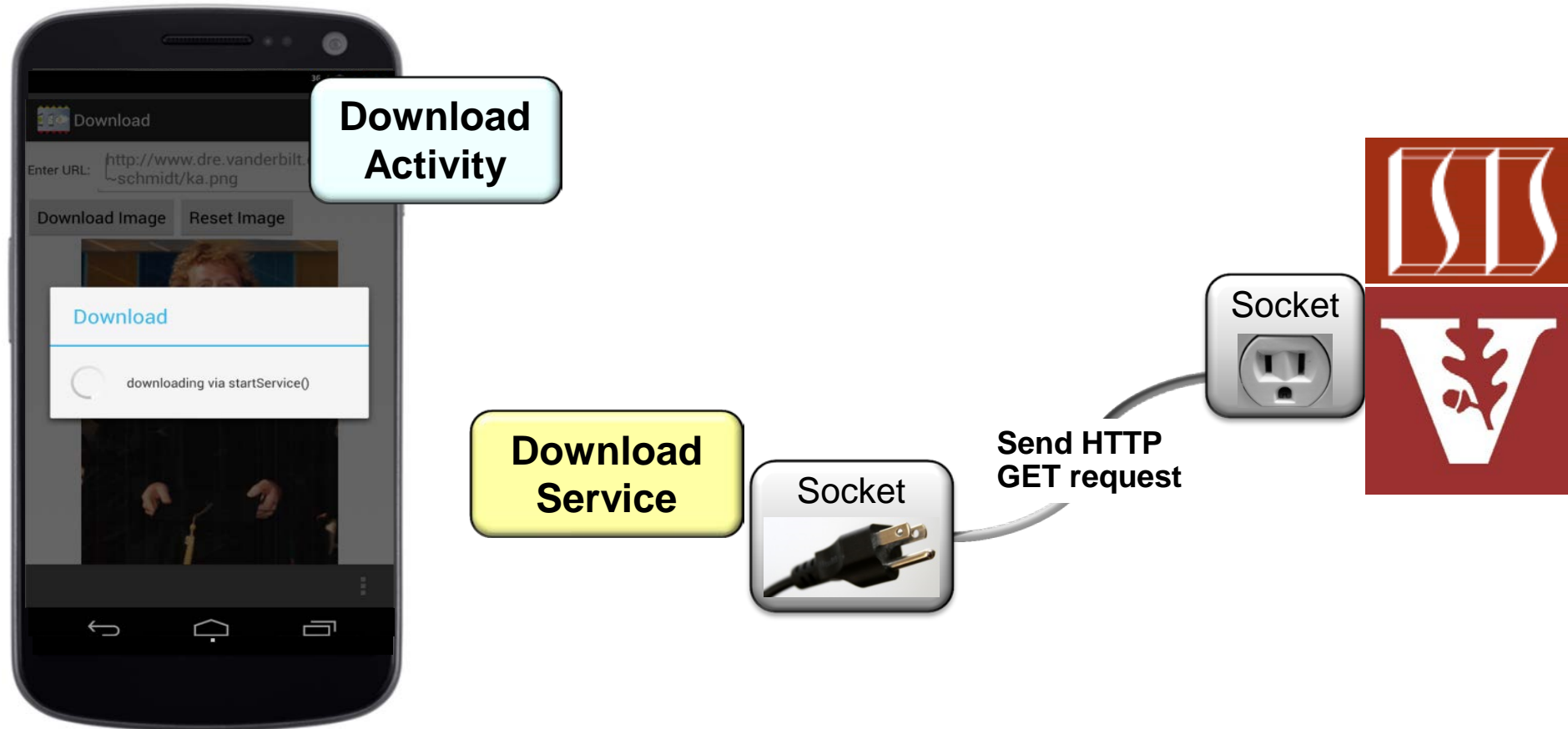
**Download
Activity**

**Download
Service**



Overview of the Download Application

1. DownloadActivity uses startService() to launch a DownloadService
2. The DownloadService retrieves the image & stores it in a file on the device



Overview of the Download Application

1. DownloadActivity uses startService() to launch a DownloadService
2. The DownloadService retrieves the image & stores it in a file on the device
3. DownloadService returns file path to DownloadActivity, which displays image



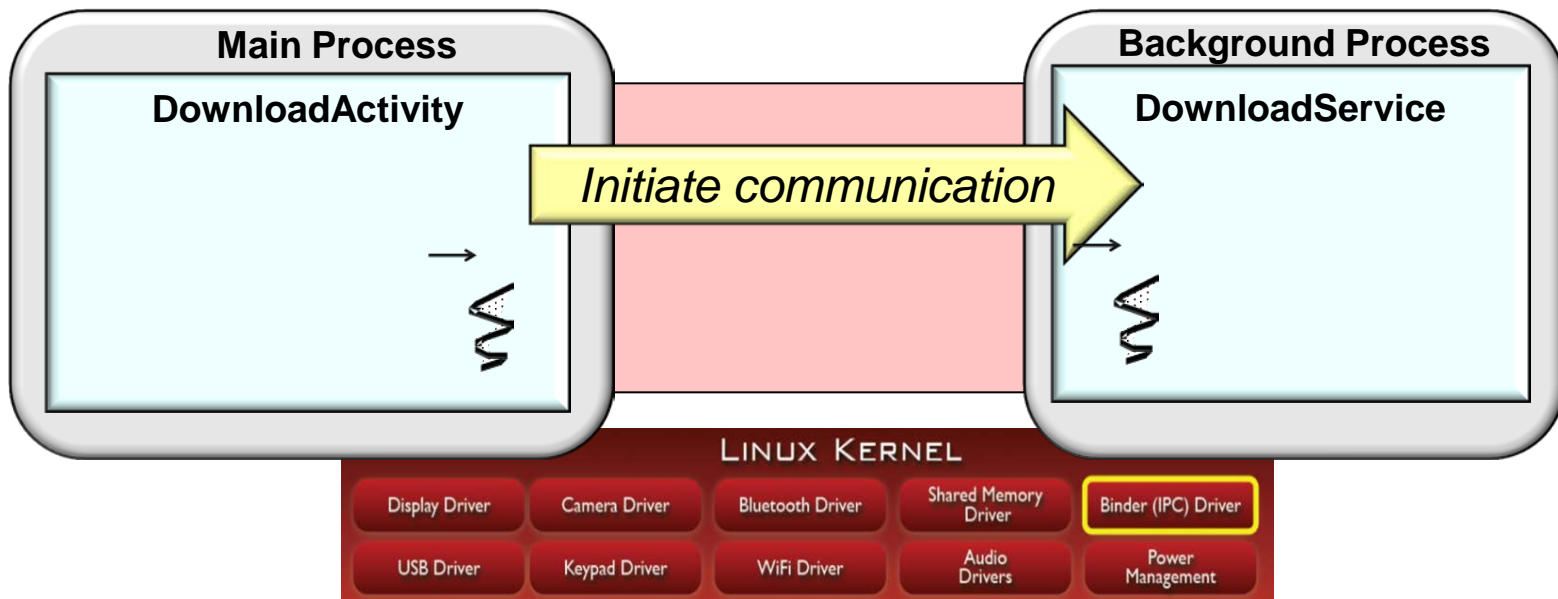
Overview of the Download Application

1. DownloadActivity uses startService() to launch a DownloadService
2. The DownloadService retrieves the image & stores it in a file on the device
3. DownloadService returns file path to DownloadActivity, which displays image



Communicating from Activities to Services

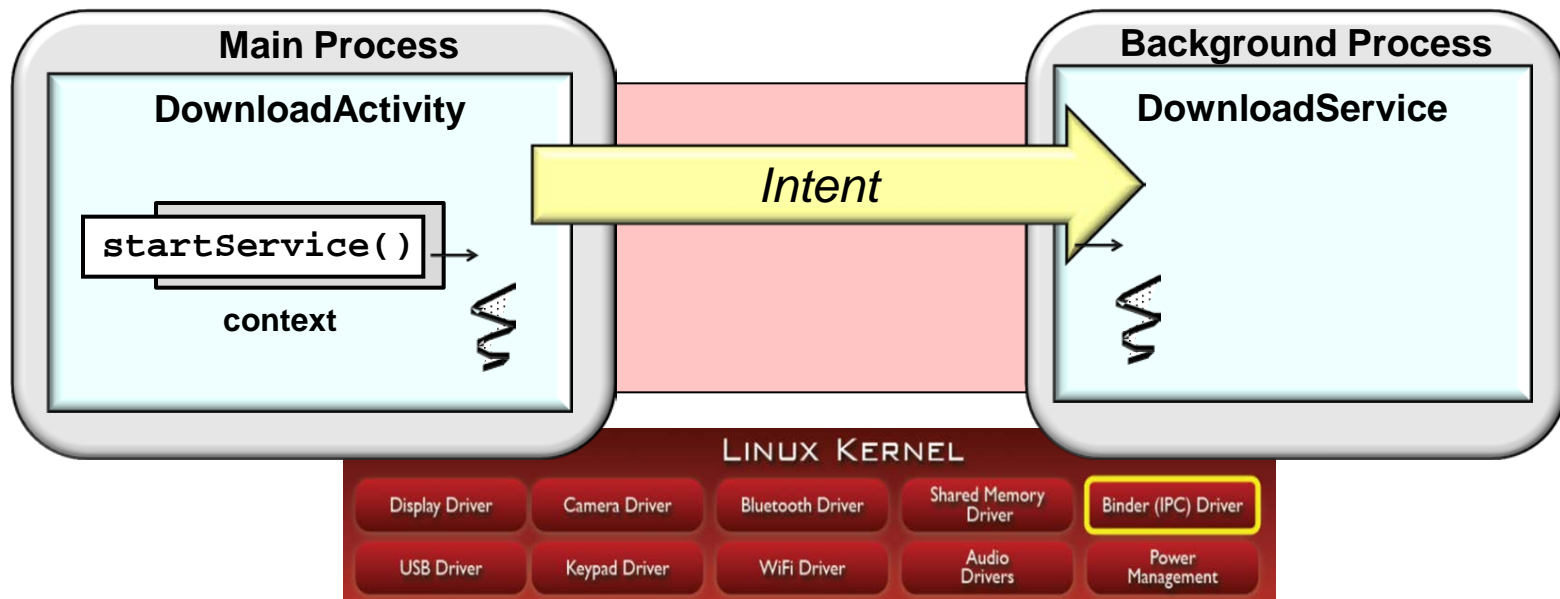
Communicating from Activities to Services



- Activities can use several mechanisms to communicate to Services

Mechanism selection depends on factors like Started vs. Bound Services or message- vs. method-oriented

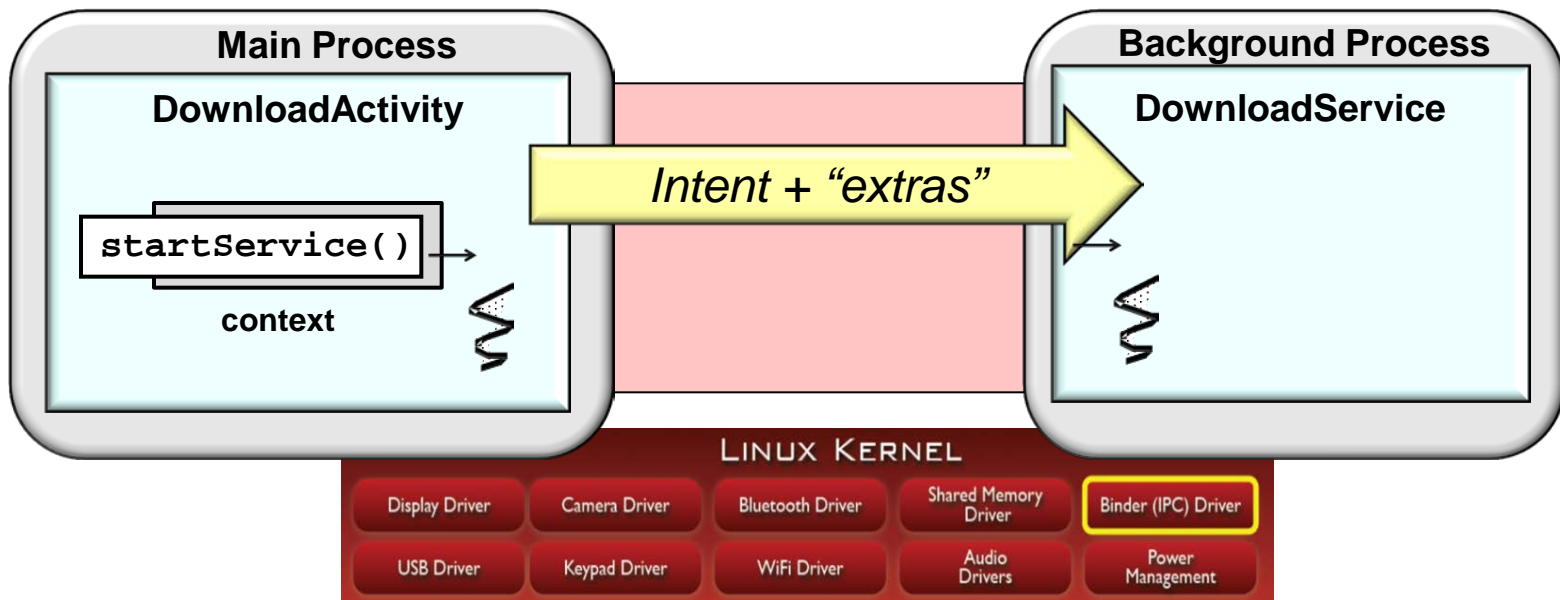
Communicating from Activities to Services



- Activities can use several mechanisms to communicate to Services
 - Send an Intent command to Started Service via `startService()`

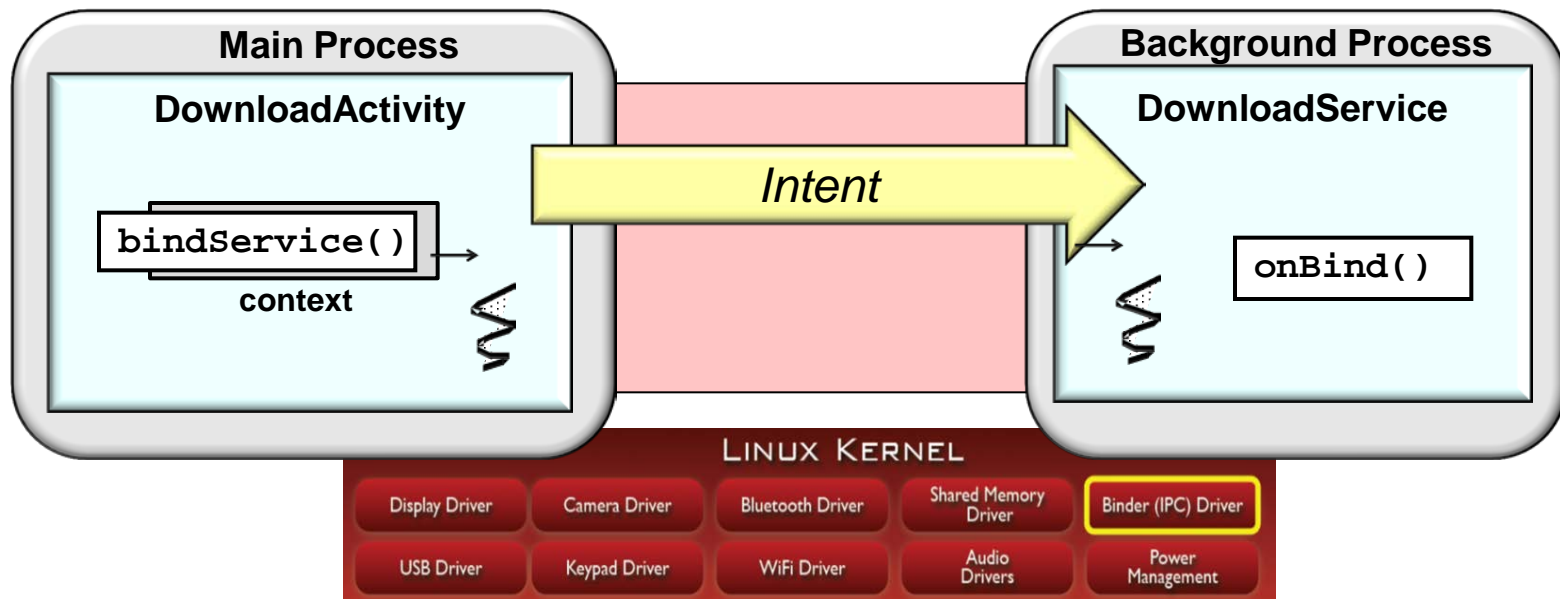
See upcoming parts on "Programming Started Services with Intents & Messengers" & "Android Intent Service"

Communicating from Activities to Services



- Activities can use several mechanisms to communicate to Services
 - Send an Intent command to Started Service via `startService()`
 - Parameters can be added as "extras" to the Intent used to start a Service

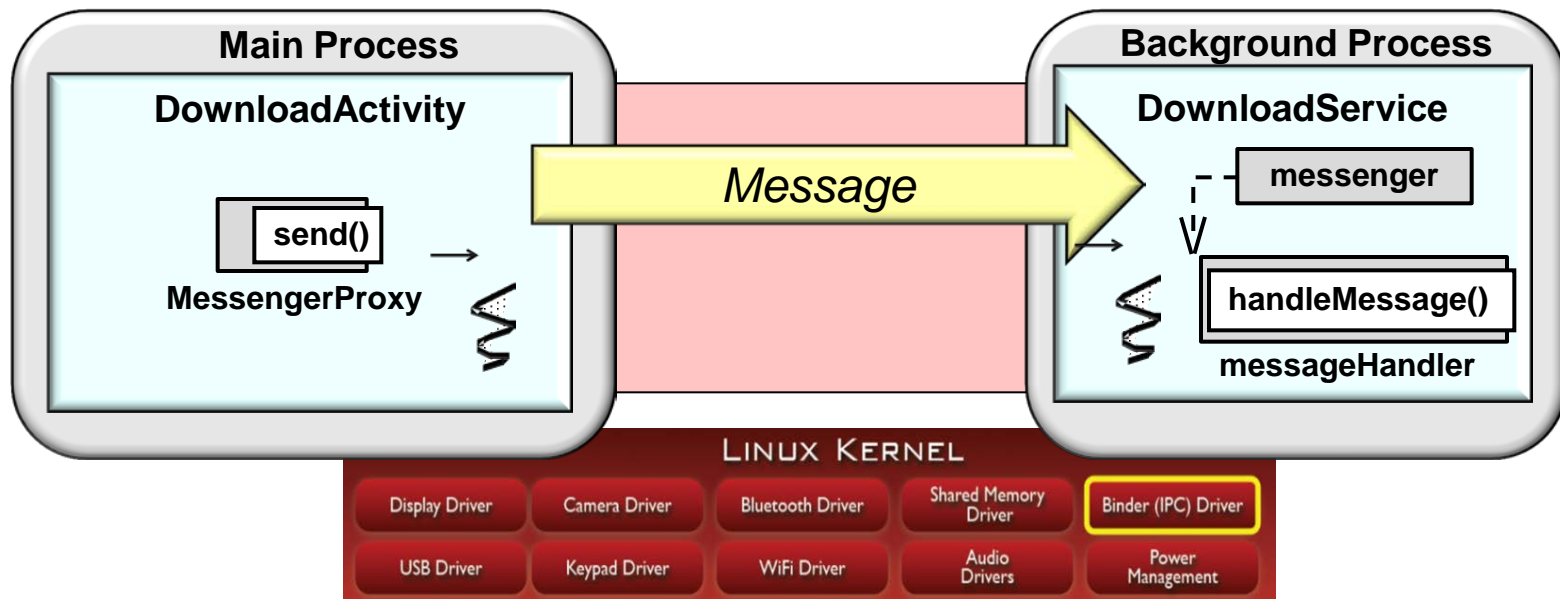
Communicating from Activities to Services



- Activities can use several mechanisms to communicate to Services
 - Send an Intent command to Started Service via `startService()`
 - Bind to a Bound Service via `BindService()`

See earlier part on "Overview of Android Services"

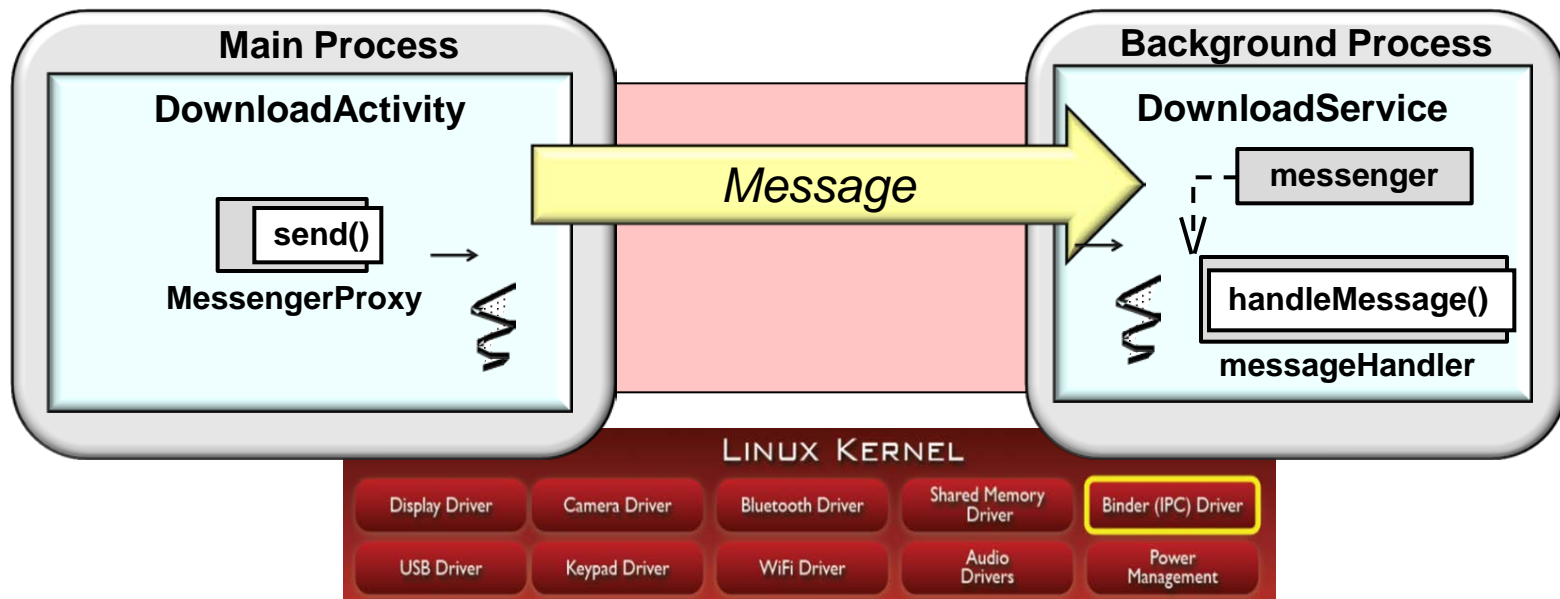
Communicating from Activities to Services



- Activities can use several mechanisms to communicate to Services
 - Send an Intent command to Started Service via `startService()`
 - Bind to a Bound Service via `BindService()`
 - Call `send()` on a reference to a Messenger

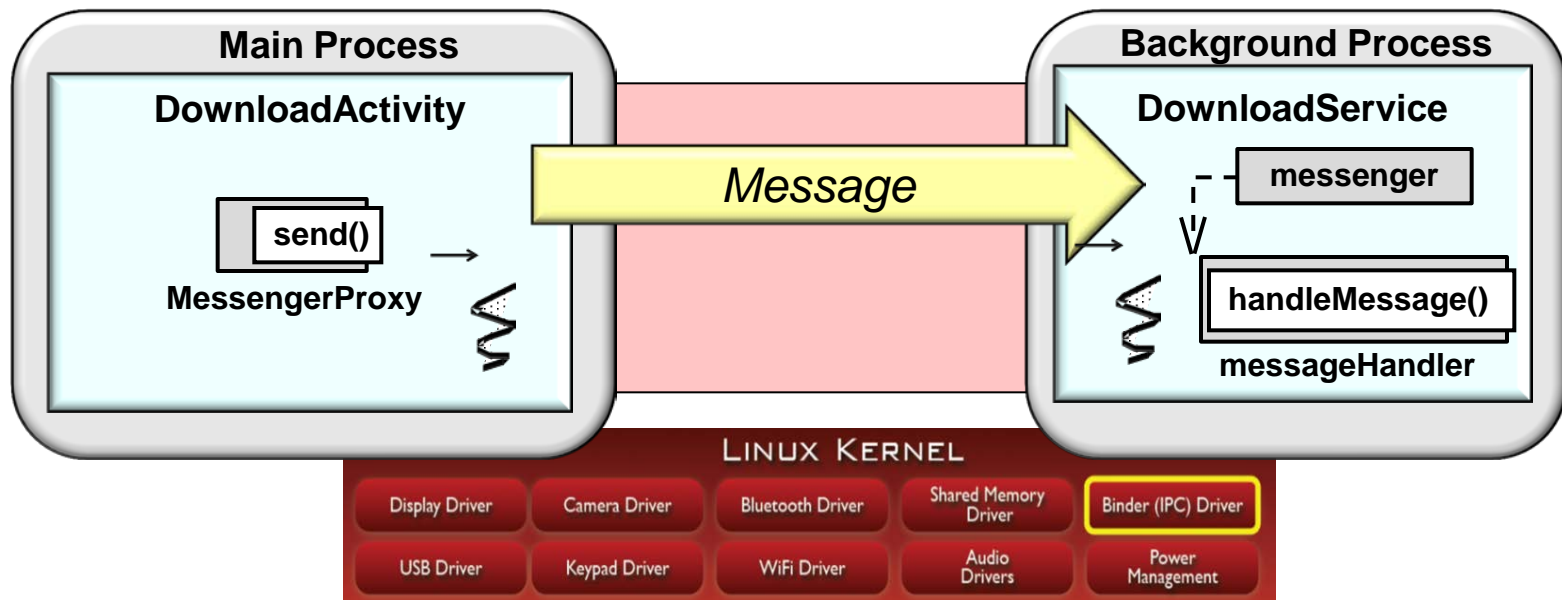
See [developer.android.com/reference/
android/os/Messenger.html](http://developer.android.com/reference/android/os/Messenger.html)

Communicating from Activities to Services



- Activities can use several mechanisms to communicate to Services
 - Send an Intent command to Started Service via `startService()`
 - Bind to a Bound Service via `BindService()`
 - Call `send()` on a reference to a Messenger

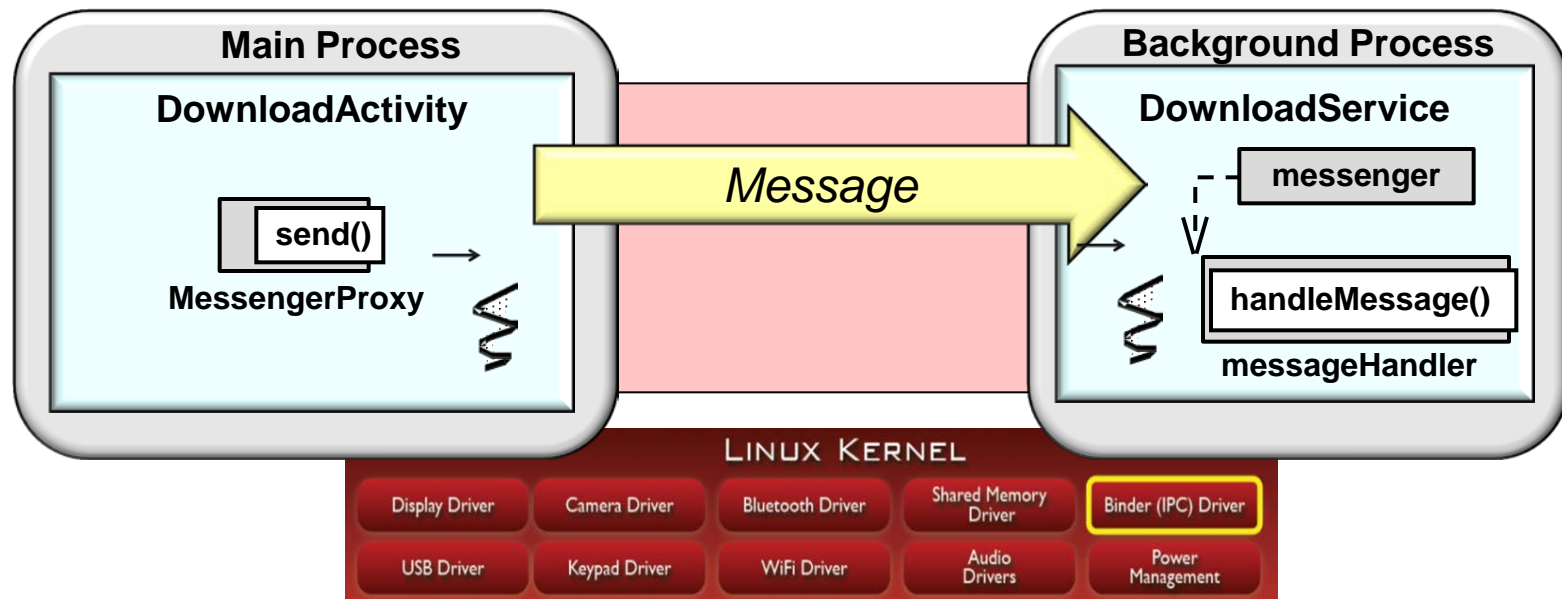
Communicating from Activities to Services



- Activities can use several mechanisms to communicate to Services
 - Send an Intent command to Started Service via `startService()`
 - Bind to a Bound Service via `BindService()`
 - Call `send()` on a reference to a Messenger
 - A Messenger encapsulates a Handler implemented within a Service

See earlier part on "Sending & Handling Messages with Android Handler"

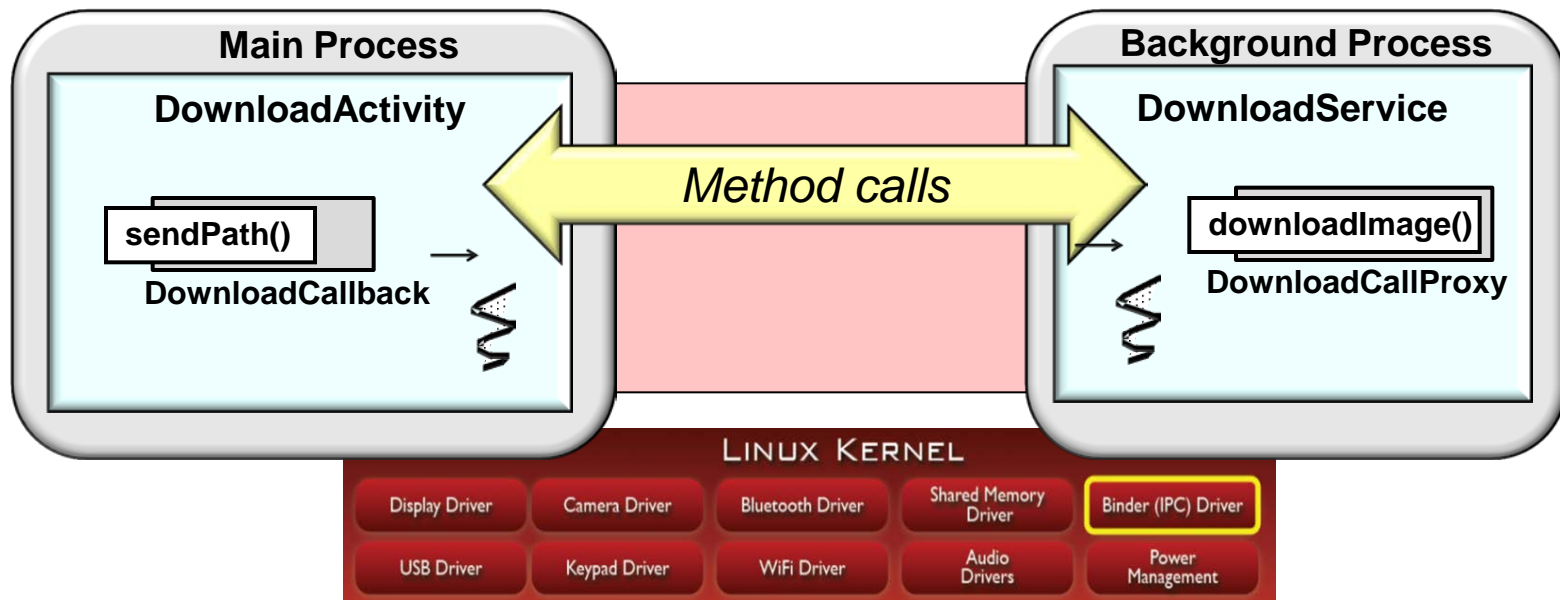
Communicating from Activities to Services



- Activities can use several mechanisms to communicate to Services
 - Send an Intent command to Started Service via `startService()`
 - Bind to a Bound Service via `BindService()`
 - Call `send()` on a reference to a Messenger
 - A Messenger encapsulates a Handler implemented within a Service
 - Enables passing Messages to a Handler across process boundaries

See upcoming part on "Service to Activity Communication via Android Messenger"

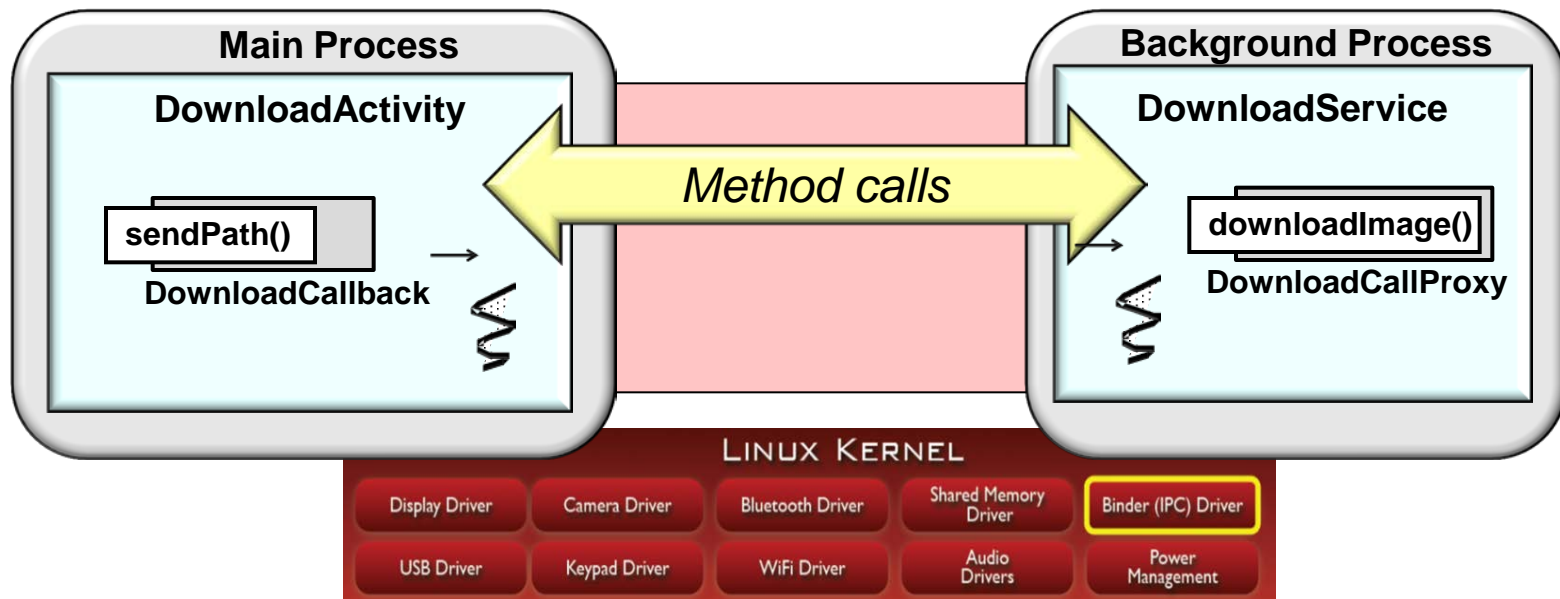
Communicating from Activities to Services



- Activities can use several mechanisms to communicate to Services
 - Send an Intent command to Started Service via `startService()`
 - Bind to a Bound Service via `BindService()`
 - Call `send()` on a reference to a Messenger
 - Invoke method calls
 - Use stubs generated by the AIDL compiler

See developer.android.com/guide/components/aidl.html

Communicating from Activities to Services

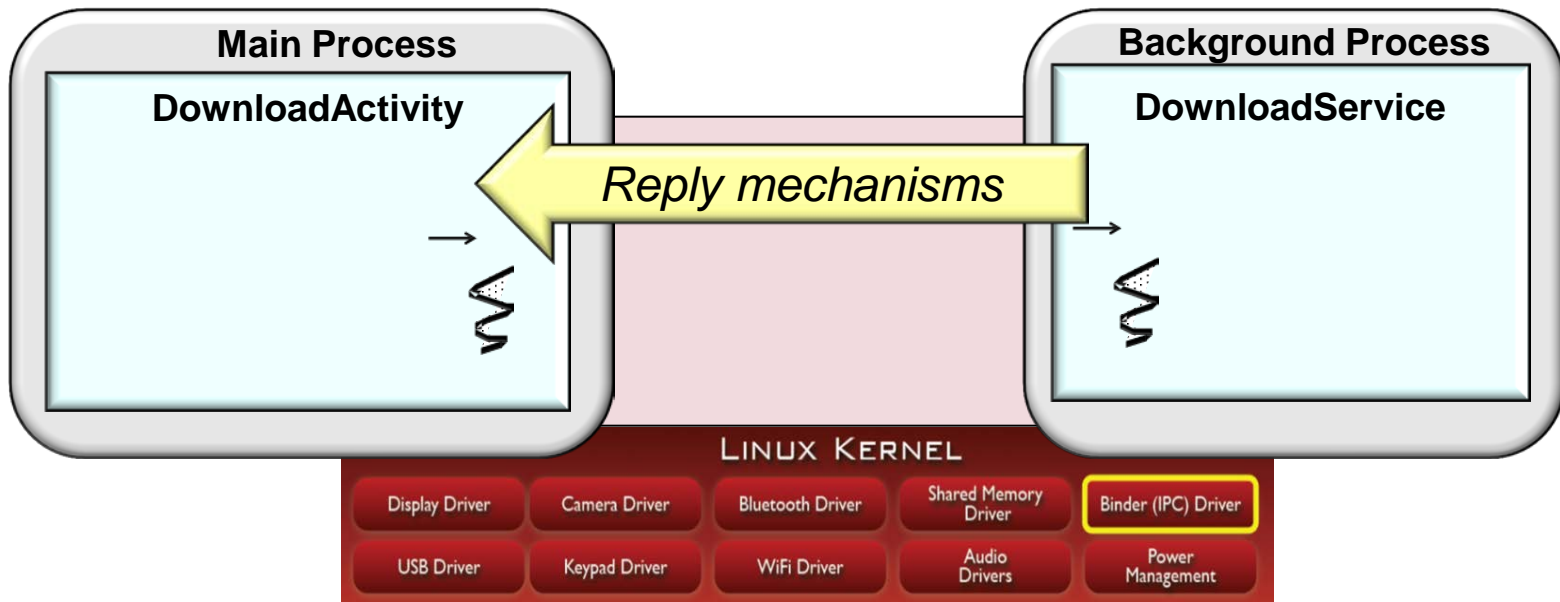


- Activities can use several mechanisms to communicate to Services
 - Send an Intent command to Started Service via `startService()`
 - Bind to a Bound Service via `BindService()`
 - Call `send()` on a reference to a Messenger
 - Invoke method calls
 - Use stubs generated by the AIDL compiler
 - These methods can be programmed to implement various behaviors

See upcoming parts on
"Programming Bound Services"

Communicating from Services to Activities

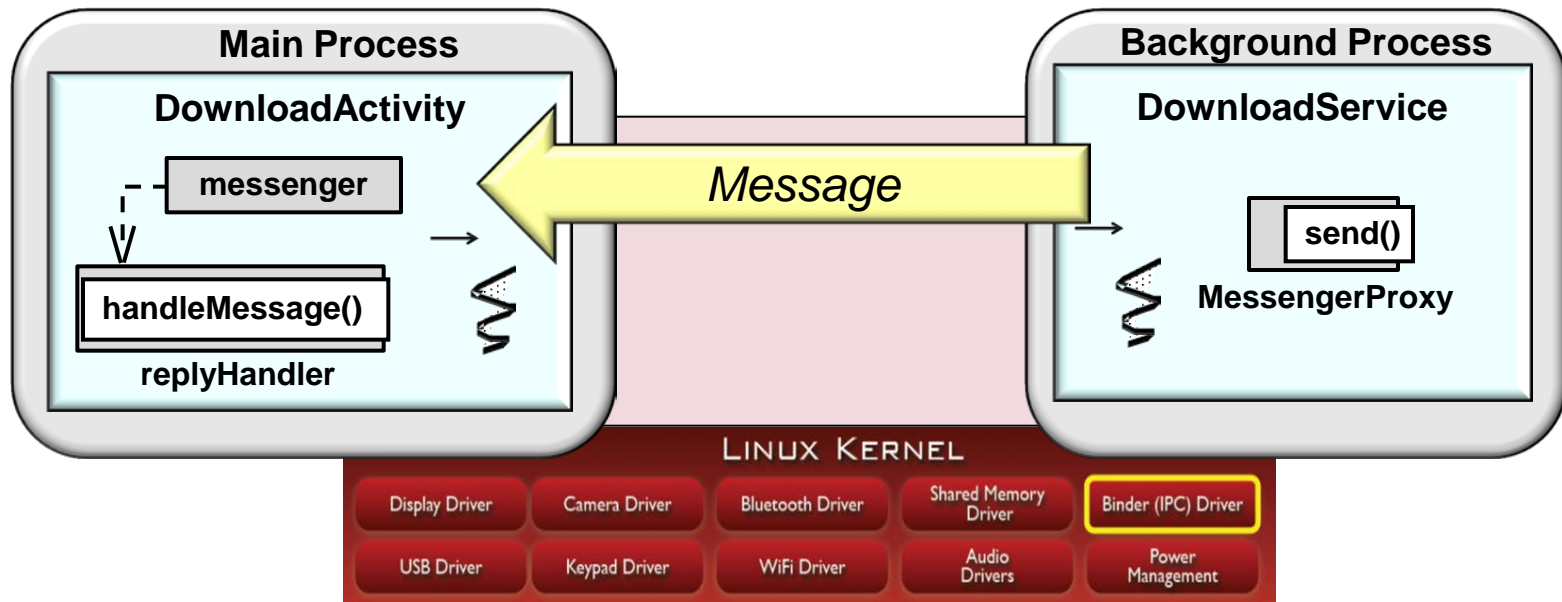
Communicating from Services to Activities



- Services can reply to Activities that initiated communication with them

The Activity initiating the communication typically dictates the reply mechanism

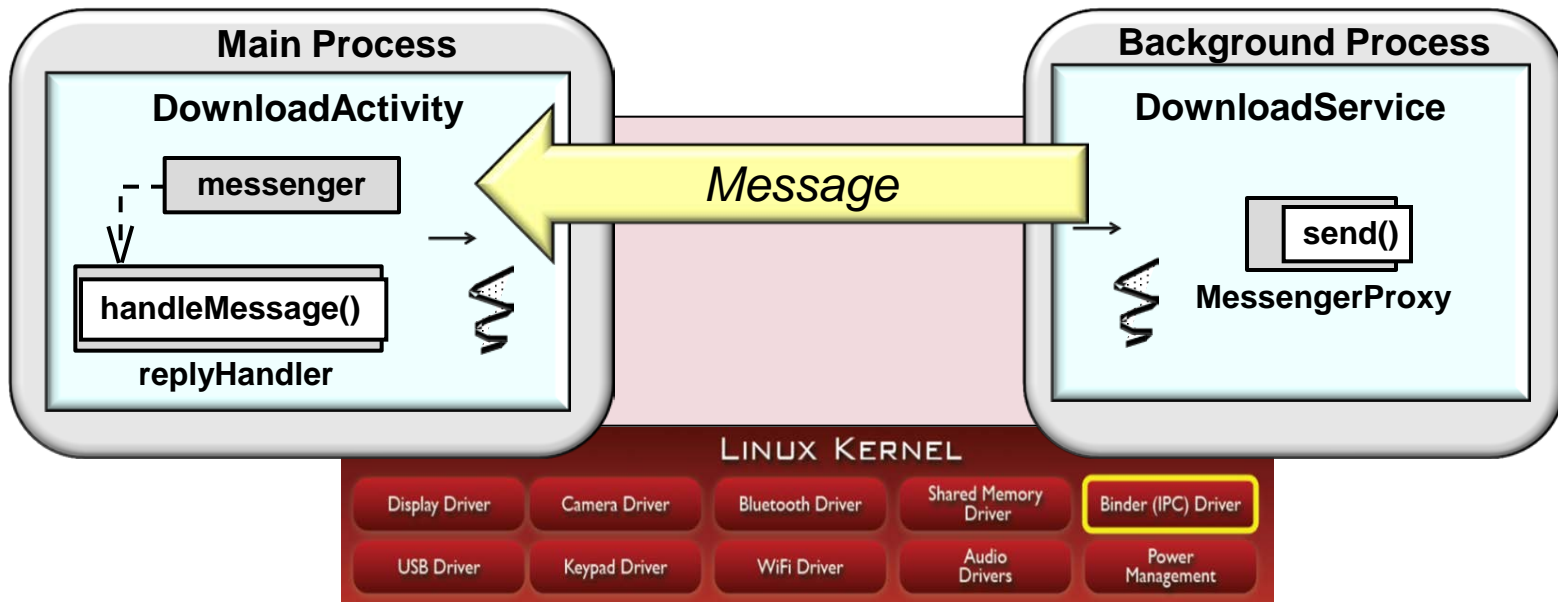
Communicating from Services to Activities



- Services can reply to Activities that initiated communication with them
 - Use a Messenger passed from the Activity to the Service

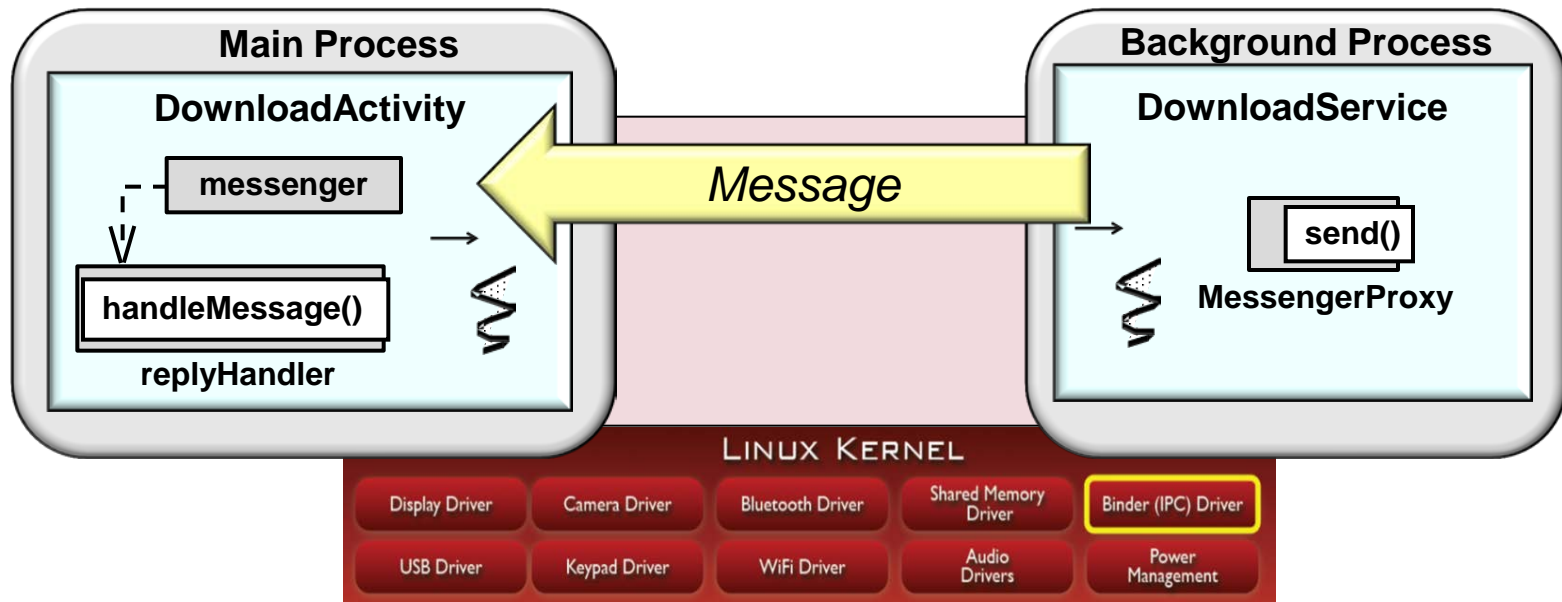
See developer.android.com/reference/android/os/Messenger.html

Communicating from Services to Activities



- Services can reply to Activities that initiated communication with them
 - Use a Messenger passed from the Activity to the Service
 - The Activity creates a Messenger & gives a reference to it to the Service

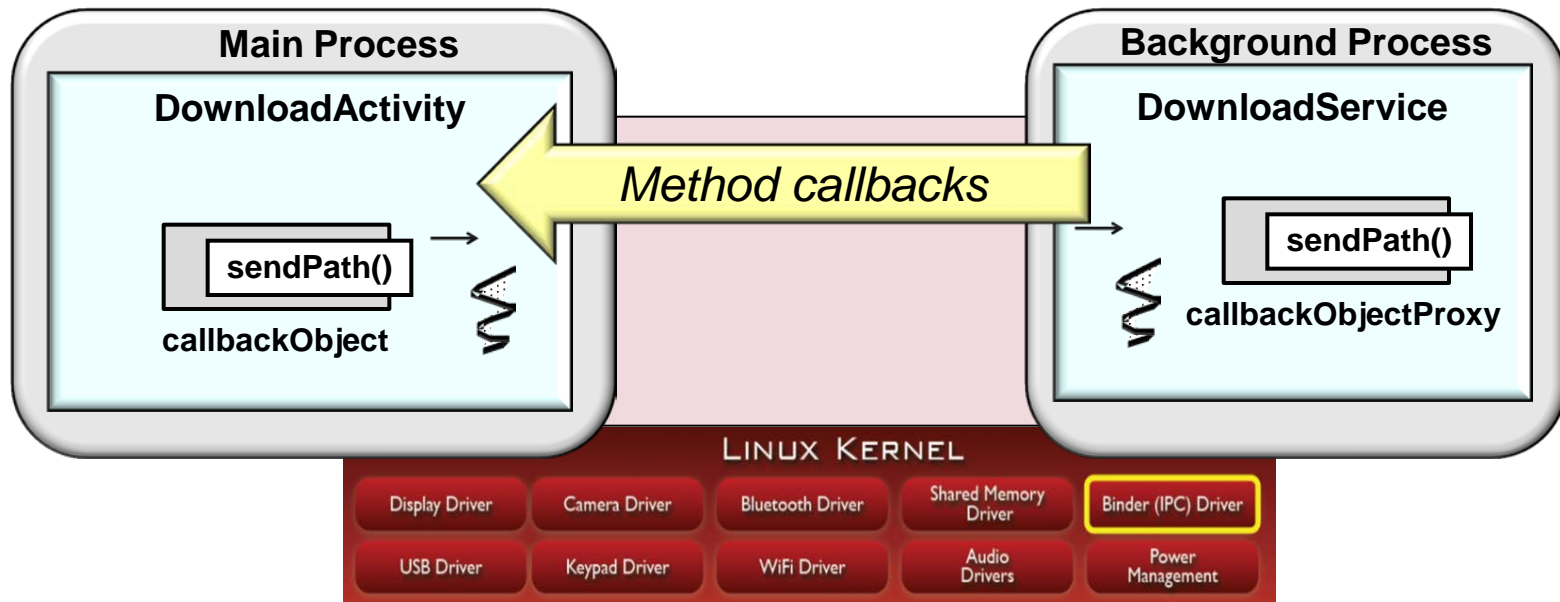
Communicating from Services to Activities



- Services can reply to Activities that initiated communication with them
- Use a Messenger passed from the Activity to the Service
 - The Activity creates a Messenger & gives a reference to it to the Service
 - The Service then uses this Messenger to send reply Messages back to the Activity's Handler

See upcoming part on "Service to Activity Communication via Android Messenger"

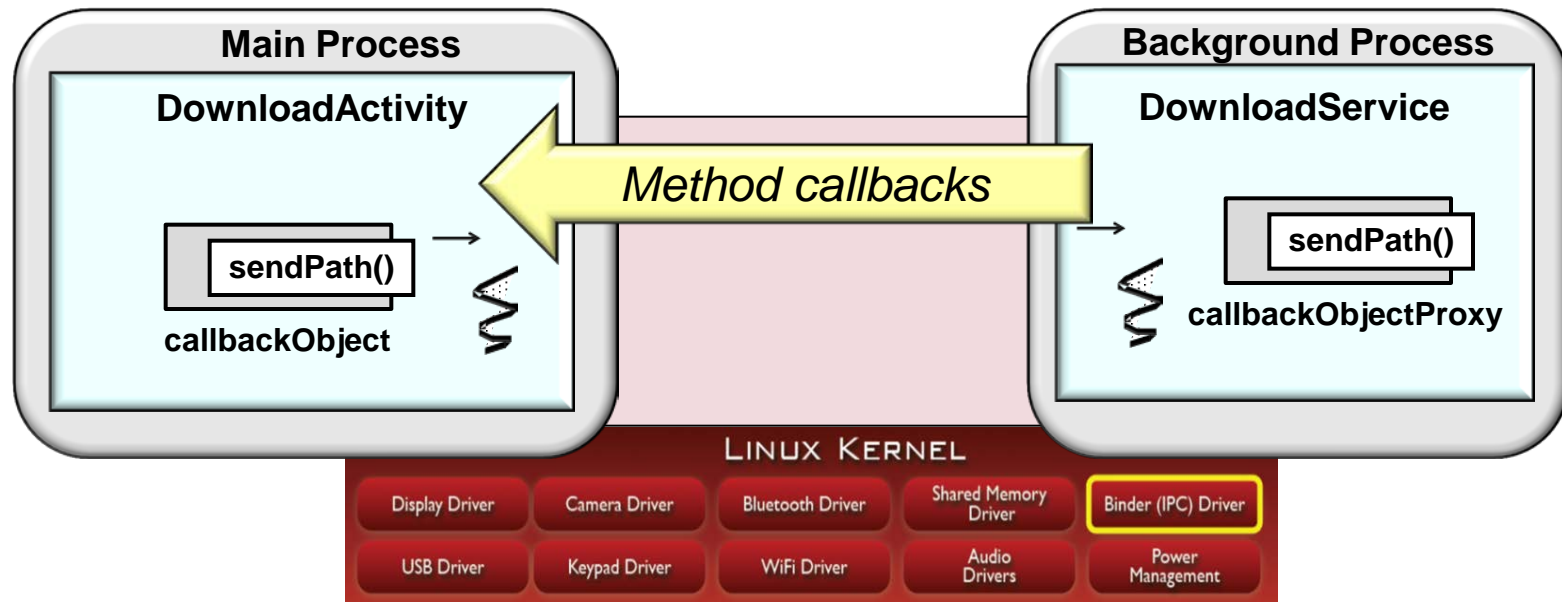
Communicating from Services to Activities



- Services can reply to Activities that initiated communication with them
 - Use a Messenger passed from the Activity to the Service
 - Use an AIDL-based callback object passed from the Activity to the Service

See developer.android.com/guide/components/aidl.html

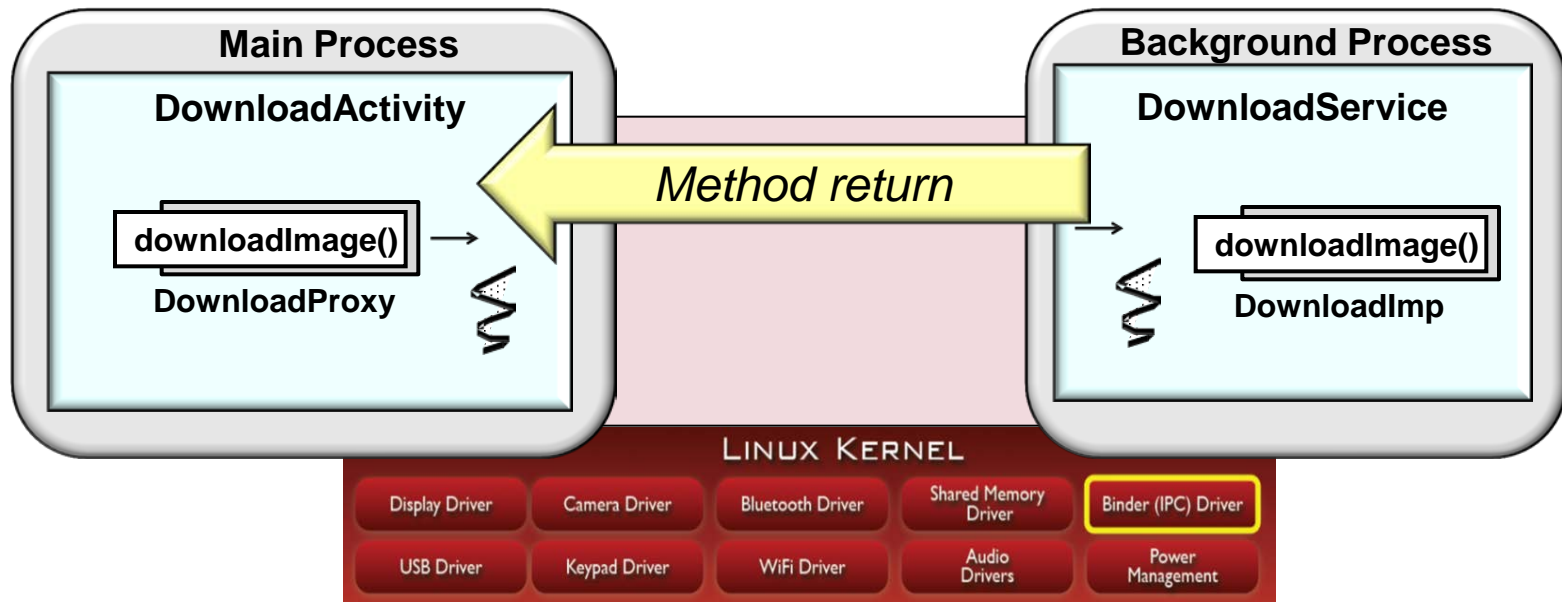
Communicating from Services to Activities



- Services can reply to Activities that initiated communication with them
 - Use a Messenger passed from the Activity to the Service
 - Use an AIDL-based callback object passed from the Activity to the Service
 - Invoke oneway method to return the reply to the Activity

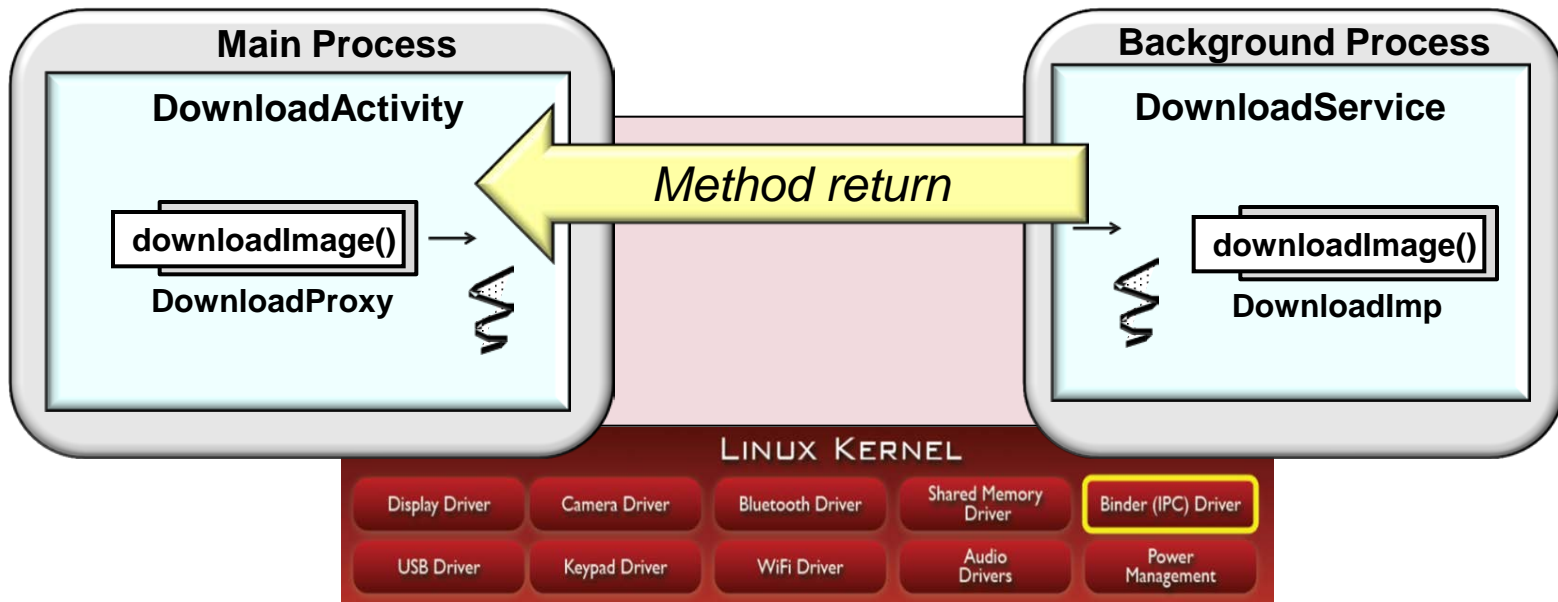
See upcoming part on
"Programming Bound Services"

Communicating from Services to Activities



- Services can reply to Activities that initiated communication with them
 - Use a Messenger passed from the Activity to the Service
 - Use an AIDL-based callback object passed from the Activity to the Service
 - Use an AIDL-based twoway method called from the Activity on the Service

Communicating from Services to Activities



- Services can reply to Activities that initiated communication with them
 - Use a Messenger passed from the Activity to the Service
 - Use an AIDL-based callback object passed from the Activity to the Service
 - Use an AIDL-based twoway method called from the Activity on the Service
 - Out parameter(s) and/or return value twoway method implicitly sends a reply from Service back to Activity

Although twoway method calls seem convenient, they are problematic..