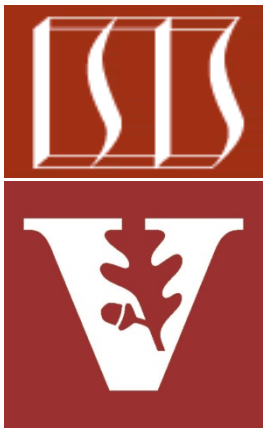
Android Services & Security: Service to Activity Communication via Android Messenger

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt



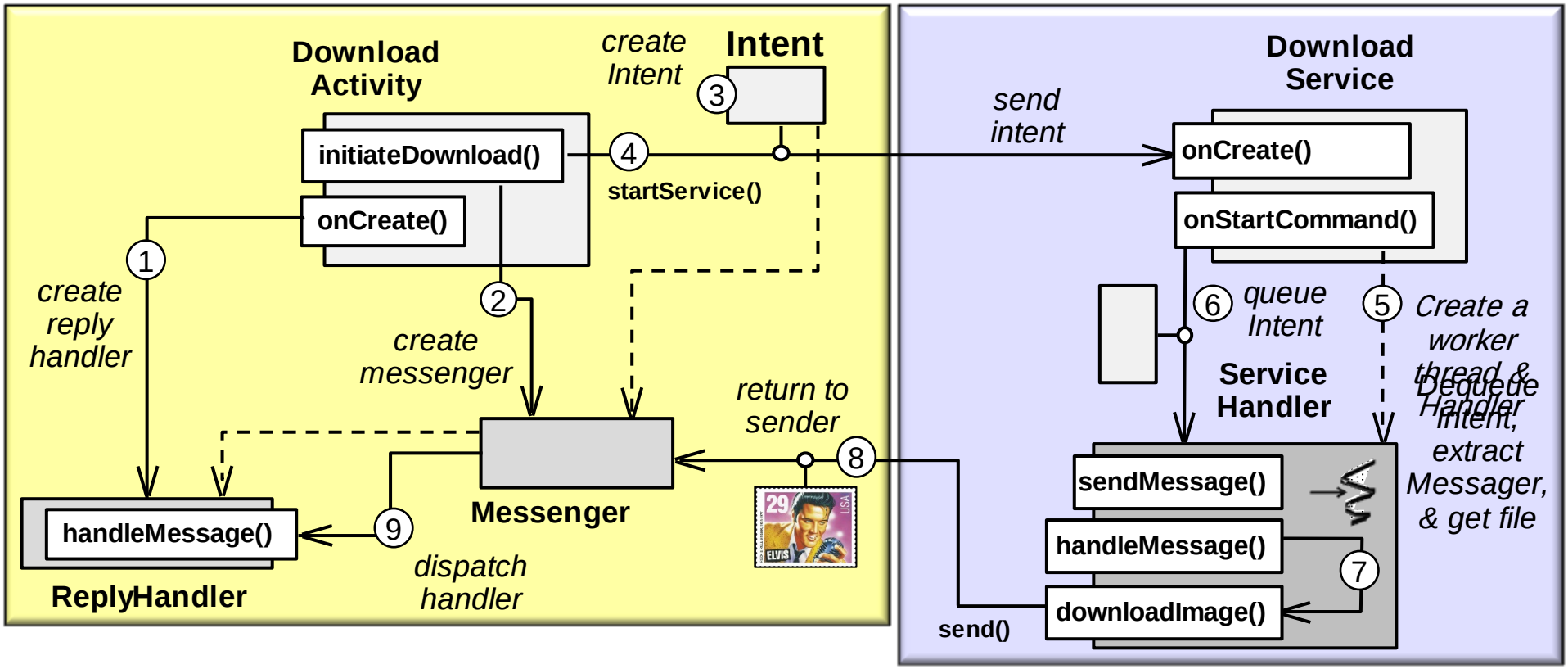
Professor of Computer Science

Institute for Software Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA

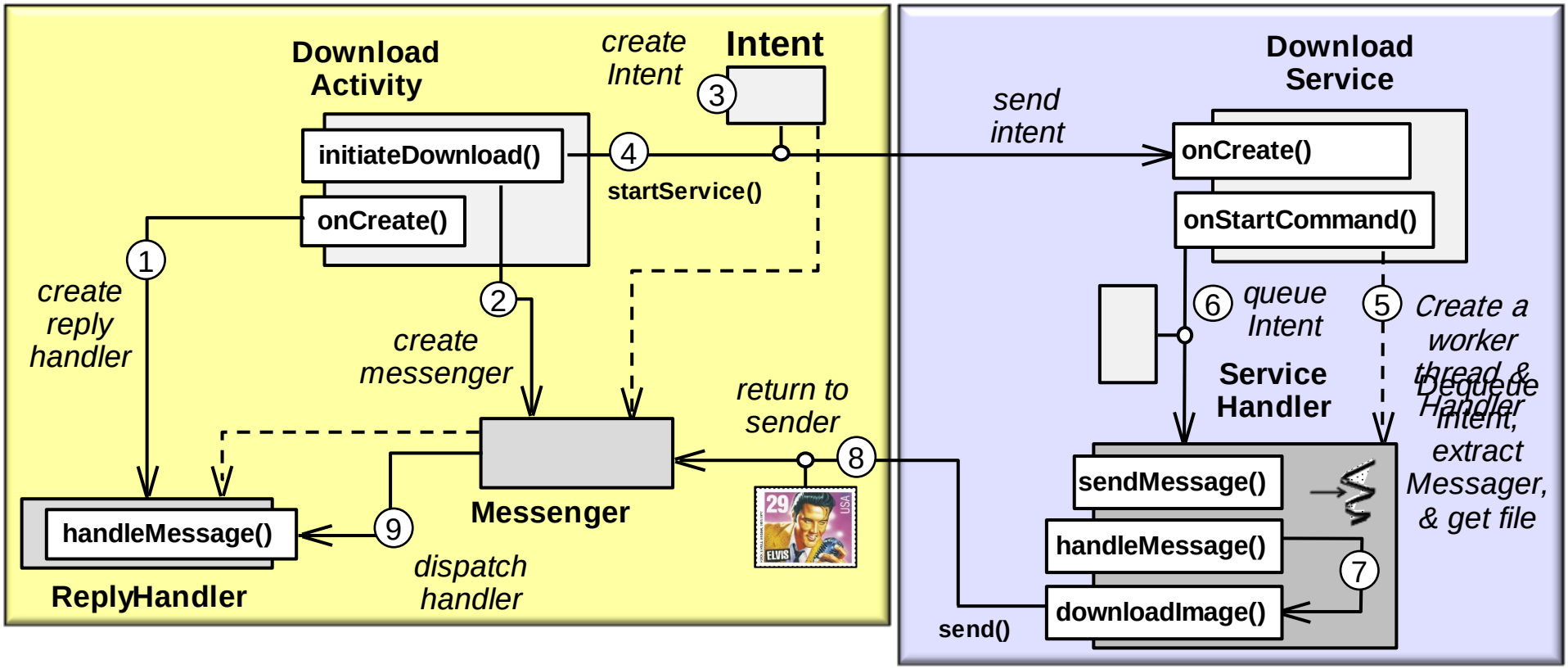
Learning Objectives in this Part of the Module

- Understand how the Messenger generalizes the HaMeR concurrency framework's sendMessage() & handleMessage() mechanisms



Learning Objectives in this Part of the Module

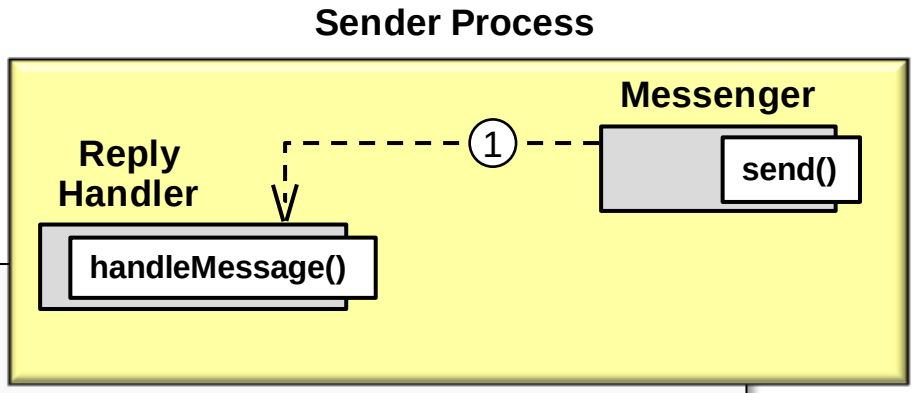
- Understand how the Messenger generalizes the HaMeR concurrency framework's sendMessage() & handleMessage() mechanisms
- Recognize how to use a Messenger to communicate from the DownloadService back to the DownloadActivity



We'll focus on using Messengers to send Messages from Started Services to Activities

Overview of Messengers

- A Messenger provides a proxy that encapsulates access to a Handler in a component



Messenger

extends Object
implements Parcelable

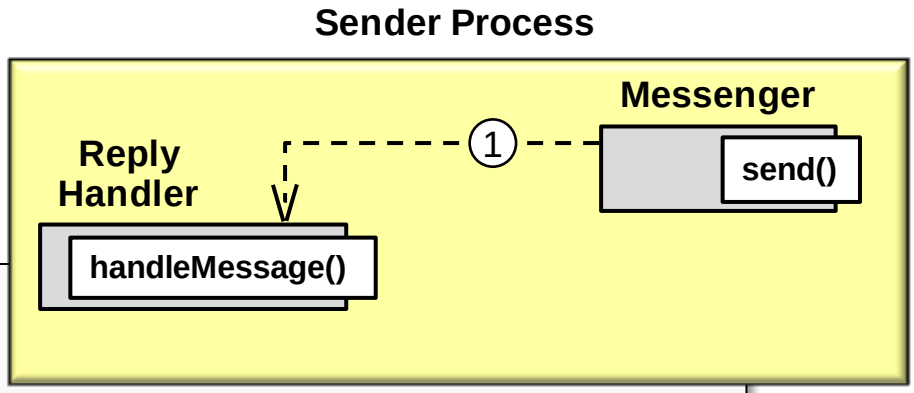
java.lang.Object
↳android.os.Messenger

Class Overview

Reference to a Handler, which others can use to send messages to it. This allows for the implementation of message-based communication across processes, by creating a Messenger pointing to a Handler in one process, and handing that Messenger to another process.

Overview of Messengers

- A Messenger provides a proxy that encapsulates access to a Handler in a component
  - e.g., an Activity or Service



Messenger

extends Object
implements Parcelable

java.lang.Object
↳android.os.Messenger

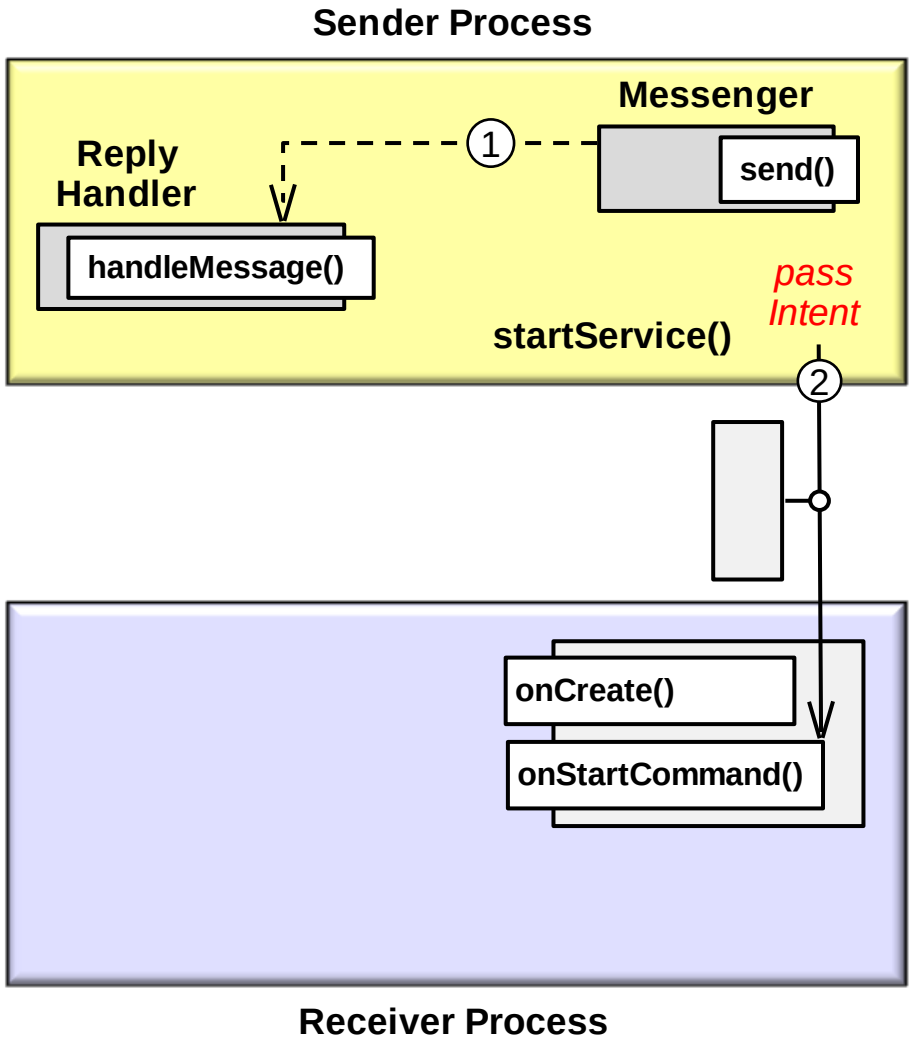
Class Overview

Reference to a Handler, which others can use to send messages to it. This allows for the implementation of message-based communication across processes, by creating a Messenger pointing to a Handler in one process, and handing that Messenger to another process.

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

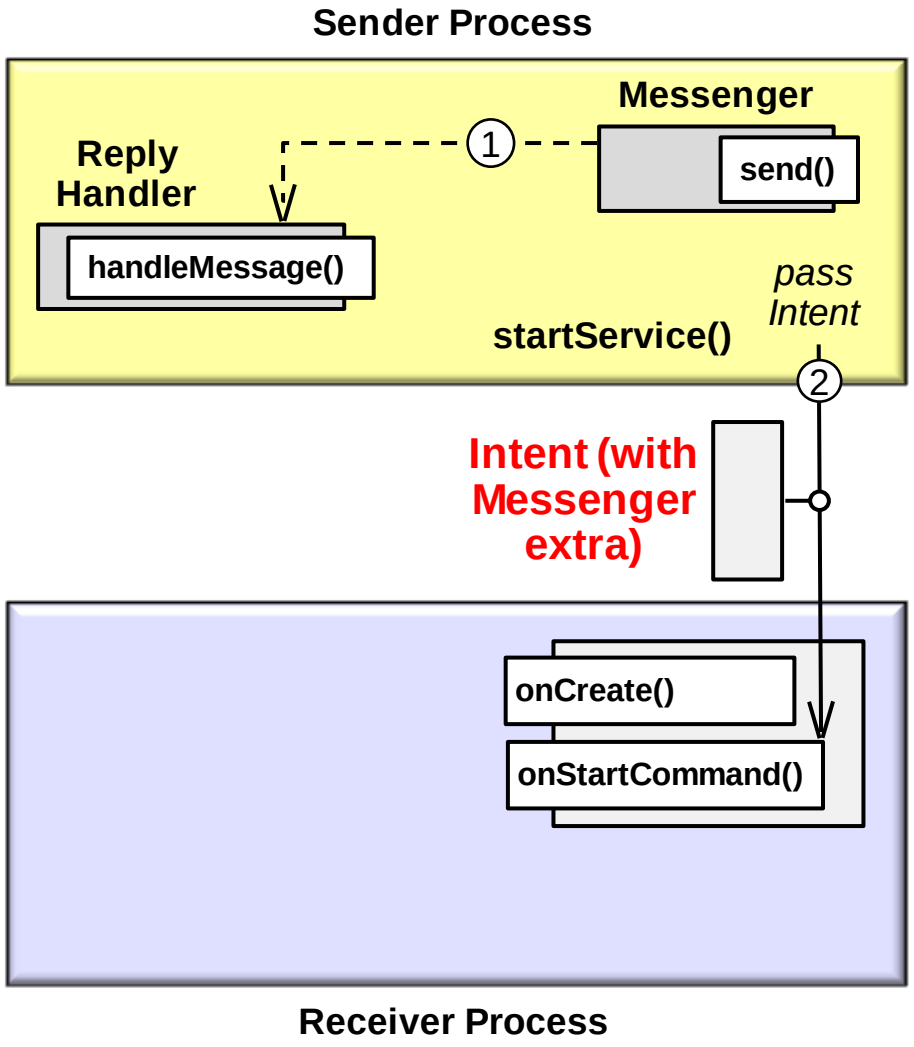
Overview of Messengers

- A Messenger provides a proxy that encapsulates access to a Handler in a component
- A reference to a Messenger can be passed to other components via some communication mechanism



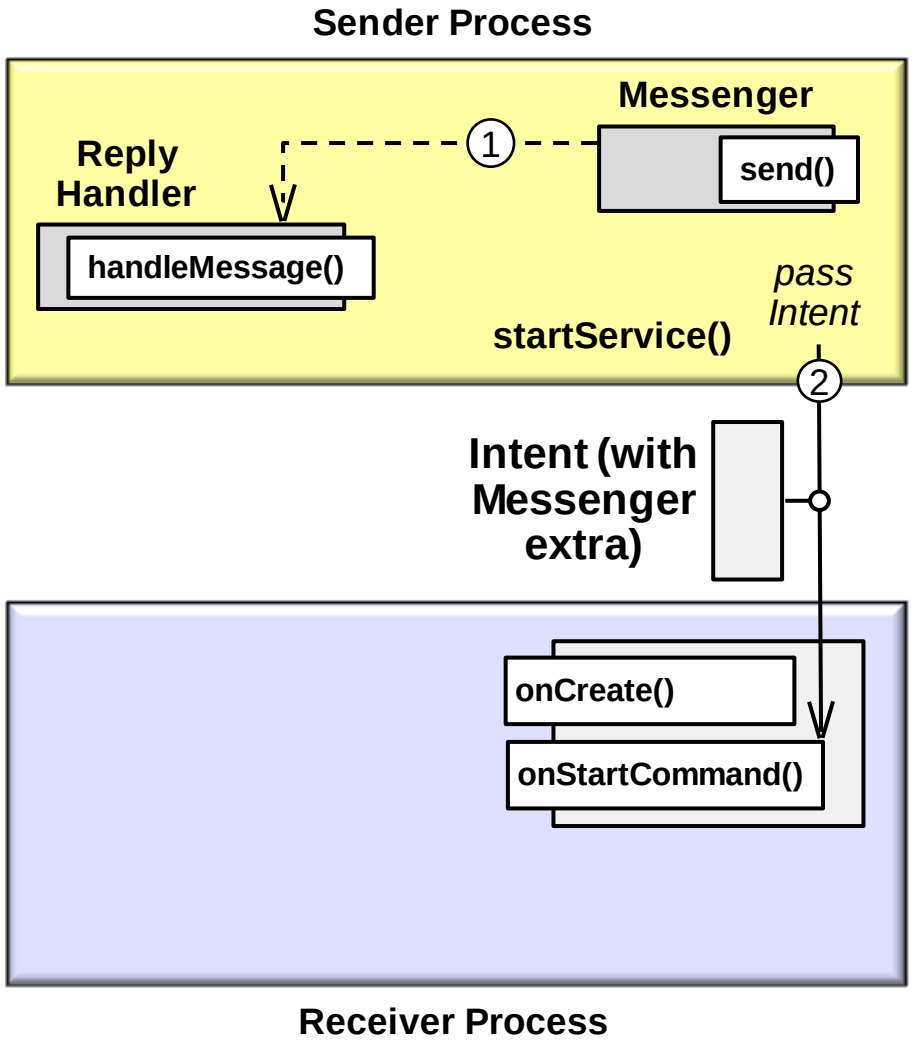
Overview of Messengers

- A Messenger provides a proxy that encapsulates access to a Handler in a component
- A reference to a Messenger can be passed to other components via some communication mechanism
  - e.g., as an "extra" to an Intent



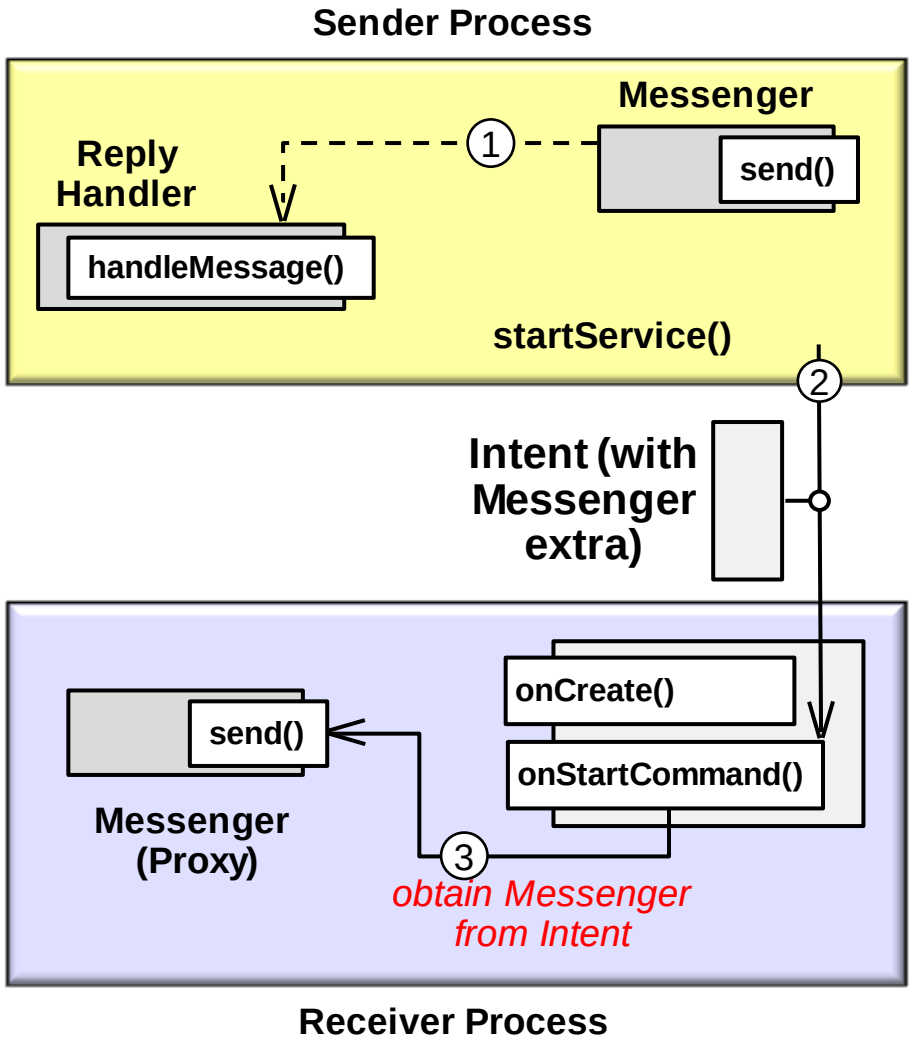
Overview of Messengers

- A Messenger provides a proxy that encapsulates access to a Handler in a component
- A reference to a Messenger can be passed to other components via some communication mechanism
- The receiver then does three things



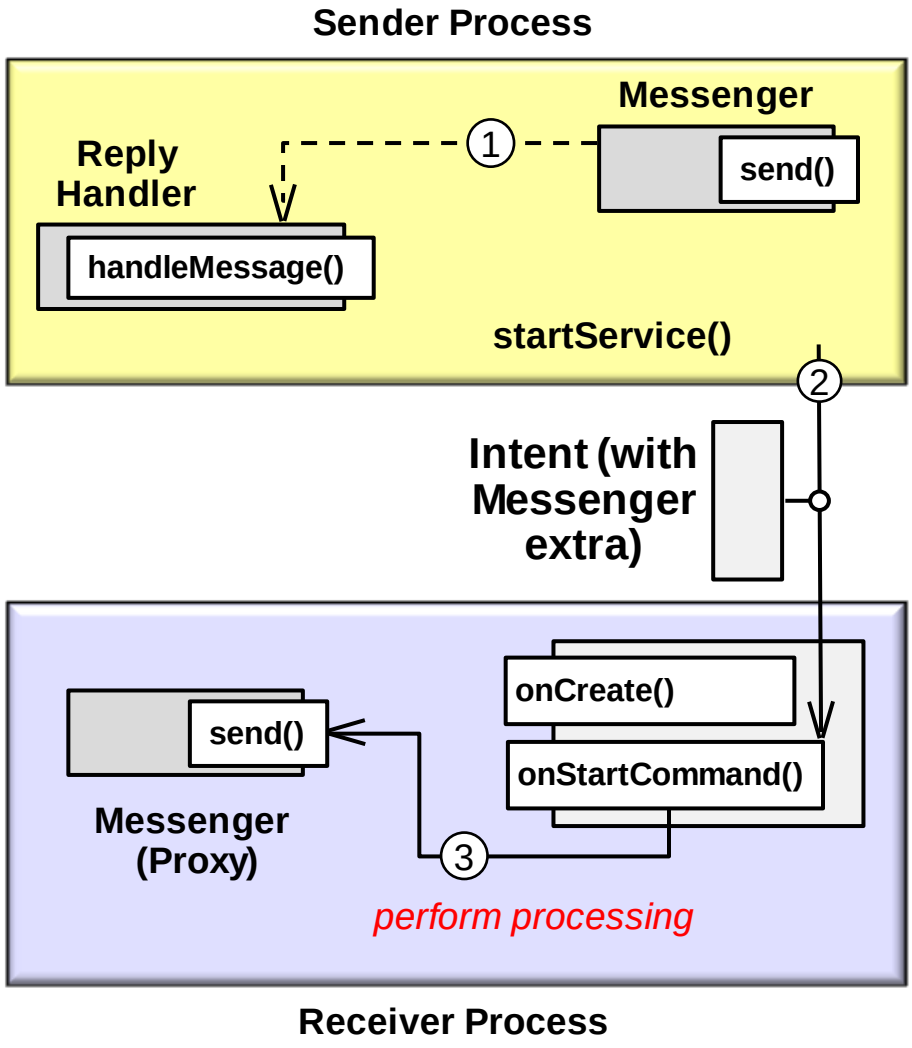
Overview of Messengers

- A Messenger provides a proxy that encapsulates access to a Handler in a component
- A reference to a Messenger can be passed to other components via some communication mechanism
- The receiver then does three things
  - Obtains the Messenger



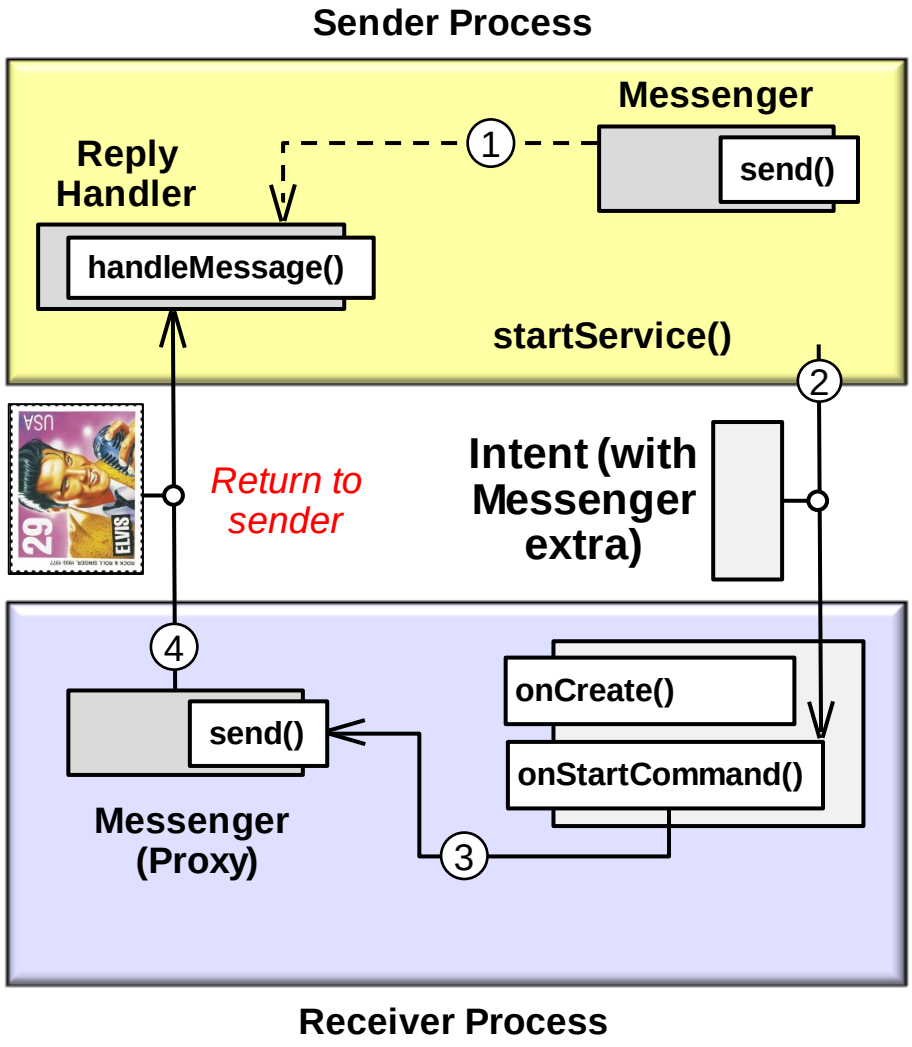
Overview of Messengers

- A Messenger provides a proxy that encapsulates access to a Handler in a component
- A reference to a Messenger can be passed to other components via some communication mechanism
- The receiver then does three things
  - Obtains the Messenger
  - Performs some processing



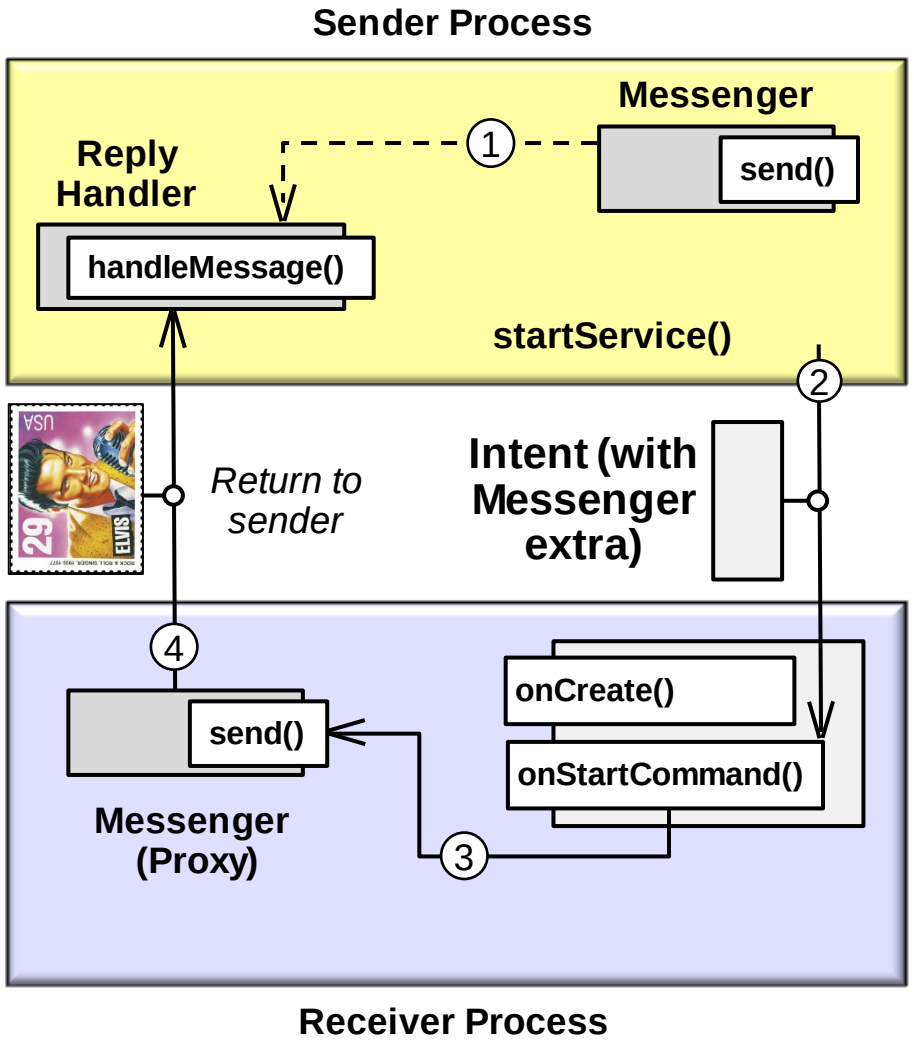
Overview of Messengers

- A Messenger provides a proxy that encapsulates access to a Handler in a component
- A reference to a Messenger can be passed to other components via some communication mechanism
- The receiver then does three things
  - Obtains the Messenger
  - Performs some processing
  - Returns the results back to the sender process



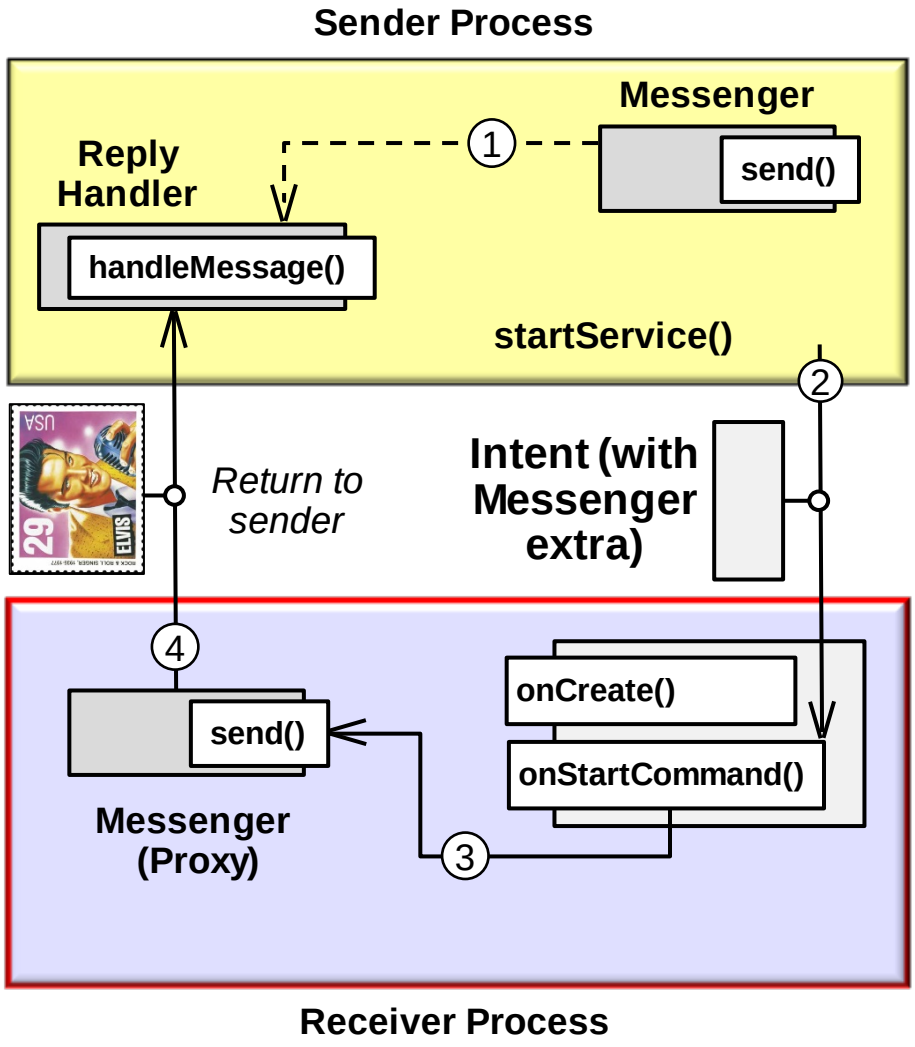
Overview of Messengers

- A Messenger provides a proxy that encapsulates access to a Handler in a component
- A reference to a Messenger can be passed to other components via some communication mechanism
- The receiver then does three things
  - Obtains the Messenger
  - Performs some processing
  - Returns the results back to the sender process
    - Even if the Handler resides in a different process than the Service



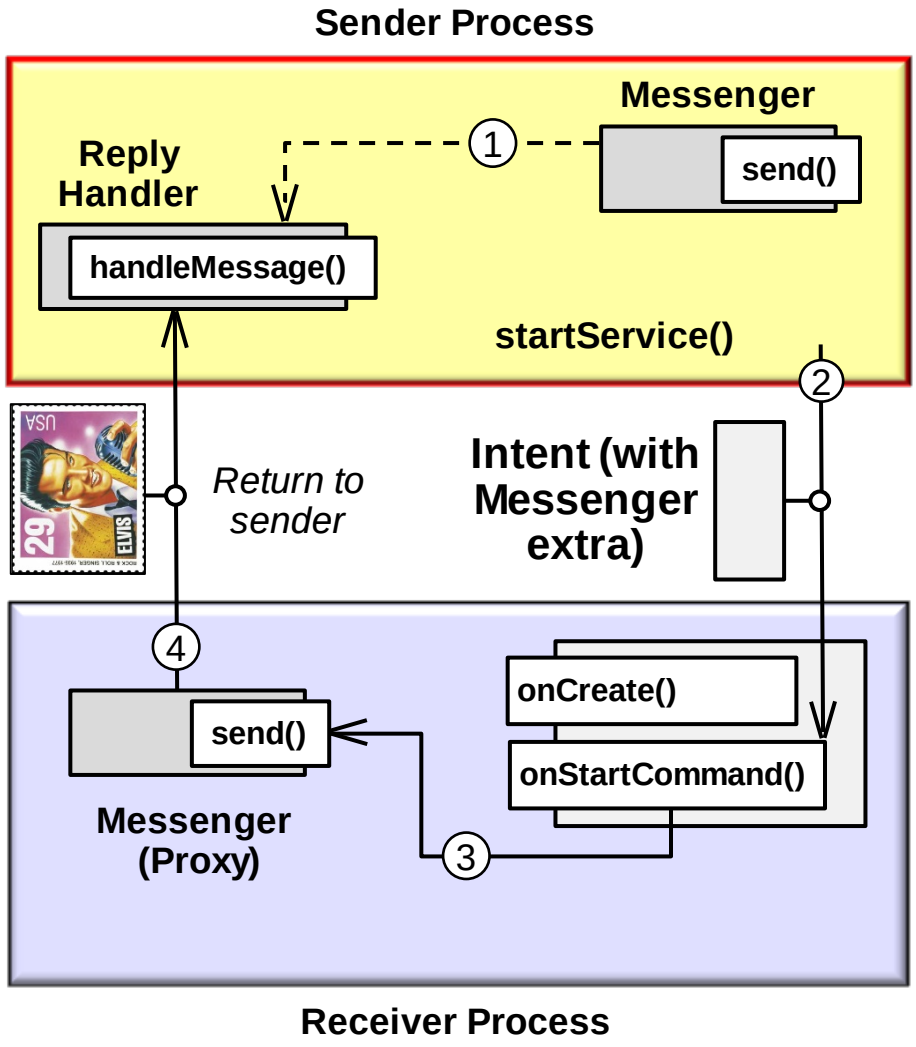
Overview of Messengers

- A Messenger provides a proxy that encapsulates access to a Handler in a component
- A reference to a Messenger can be passed to other components via some communication mechanism
- The receiver then does three things
  - Obtains the Messenger
  - Performs some processing
  - Returns the results back to the sender process
    - Even if the Handler resides in a different process than the Service



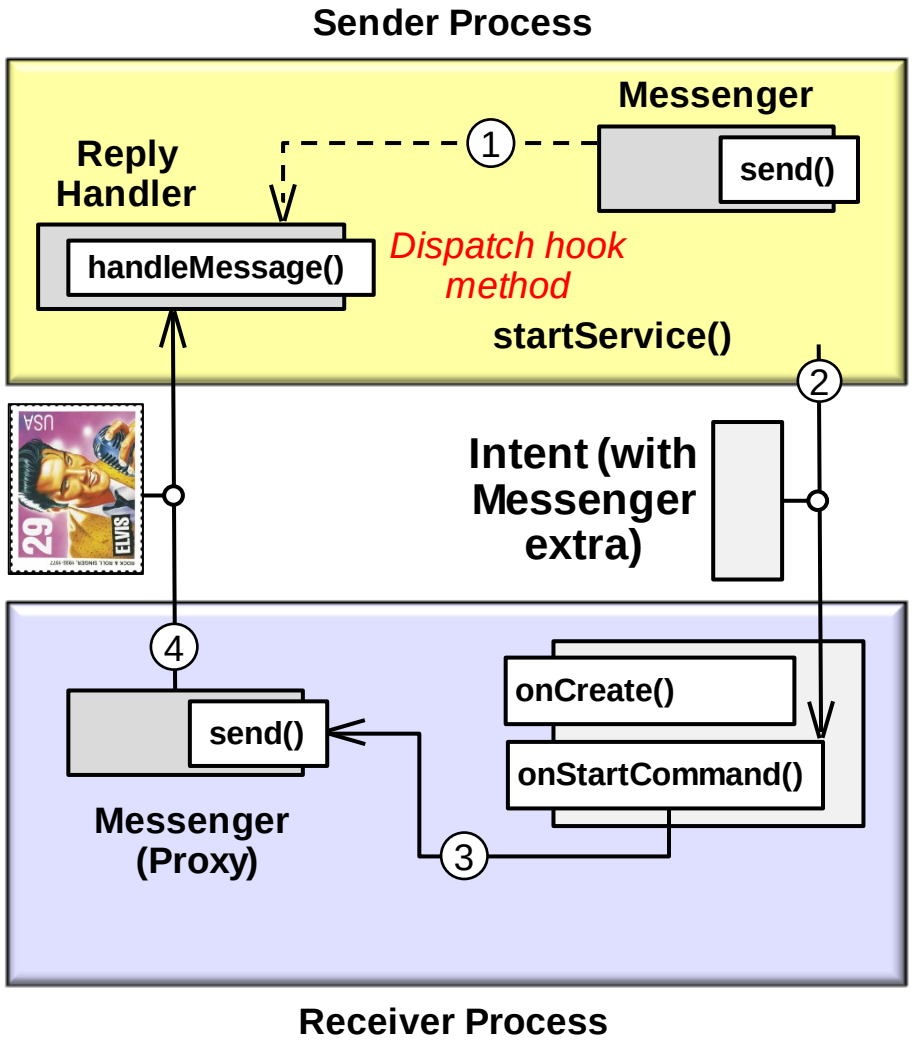
Overview of Messengers

- A Messenger provides a proxy that encapsulates access to a Handler in a component
- A reference to a Messenger can be passed to other components via some communication mechanism
- The receiver then does three things
  - Obtains the Messenger
  - Performs some processing
  - Returns the results back to the sender process
    - Even if the Handler resides in a different process than the Service



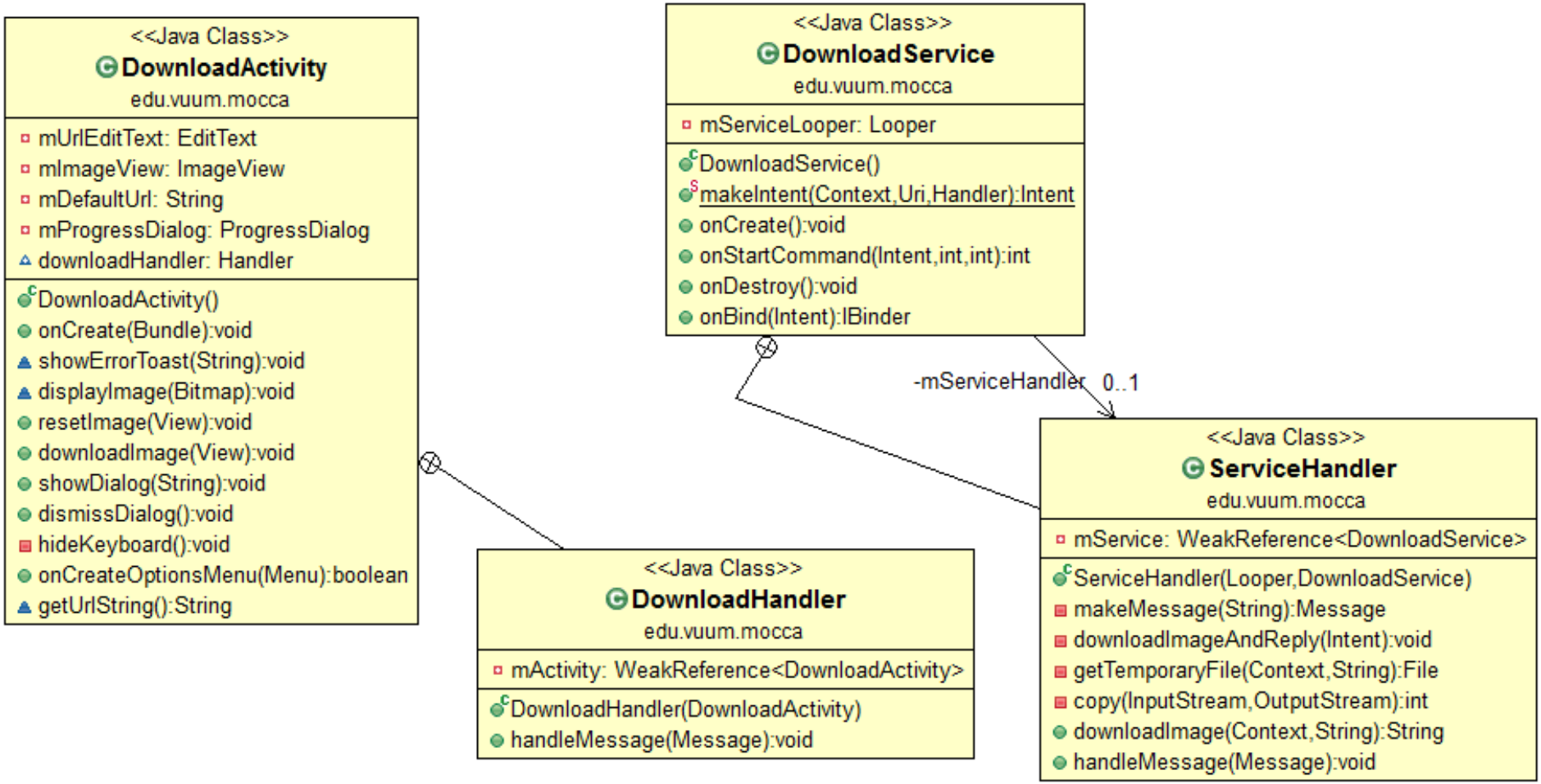
Overview of Messengers

- A Messenger provides a proxy that encapsulates access to a Handler in a component
- A reference to a Messenger can be passed to other components via some communication mechanism
- The receiver then does three things
- The Message sent by the receiver it then dispatched to the Handler's handleMessage() hook method



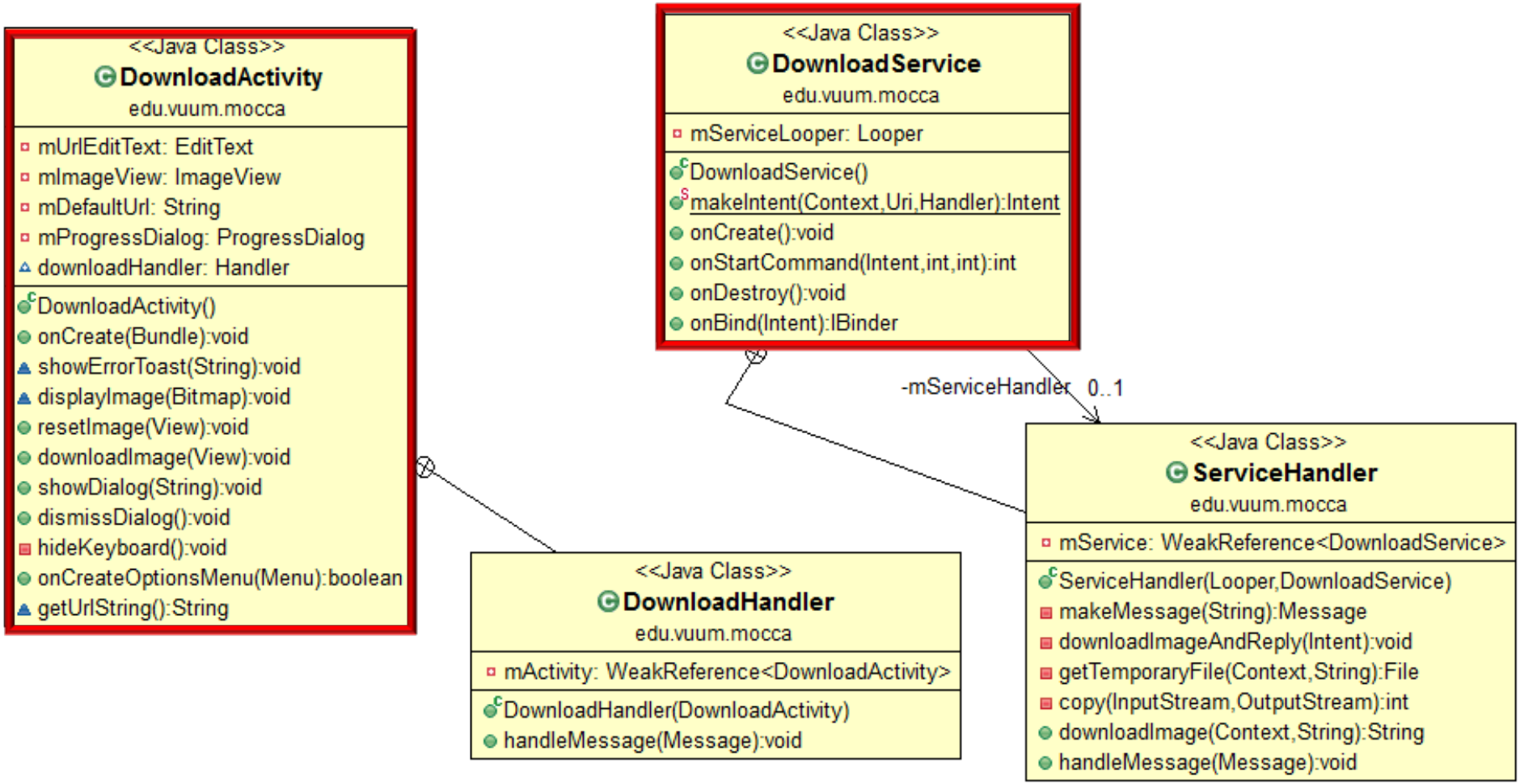
Overview of Using Messenger in the Download Application

Using Messenger in the Download Application

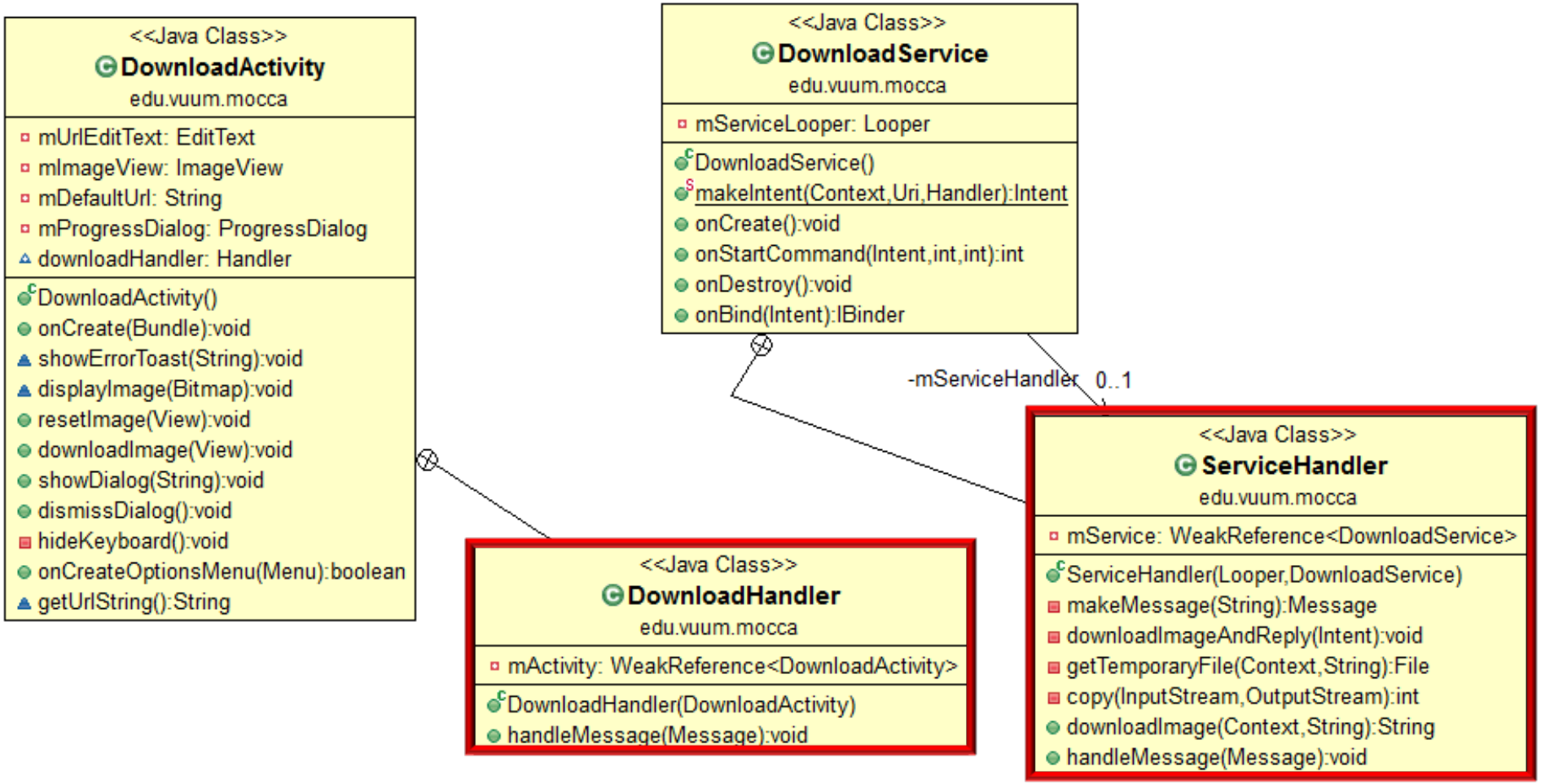


See earlier parts on “Programming Started Services”

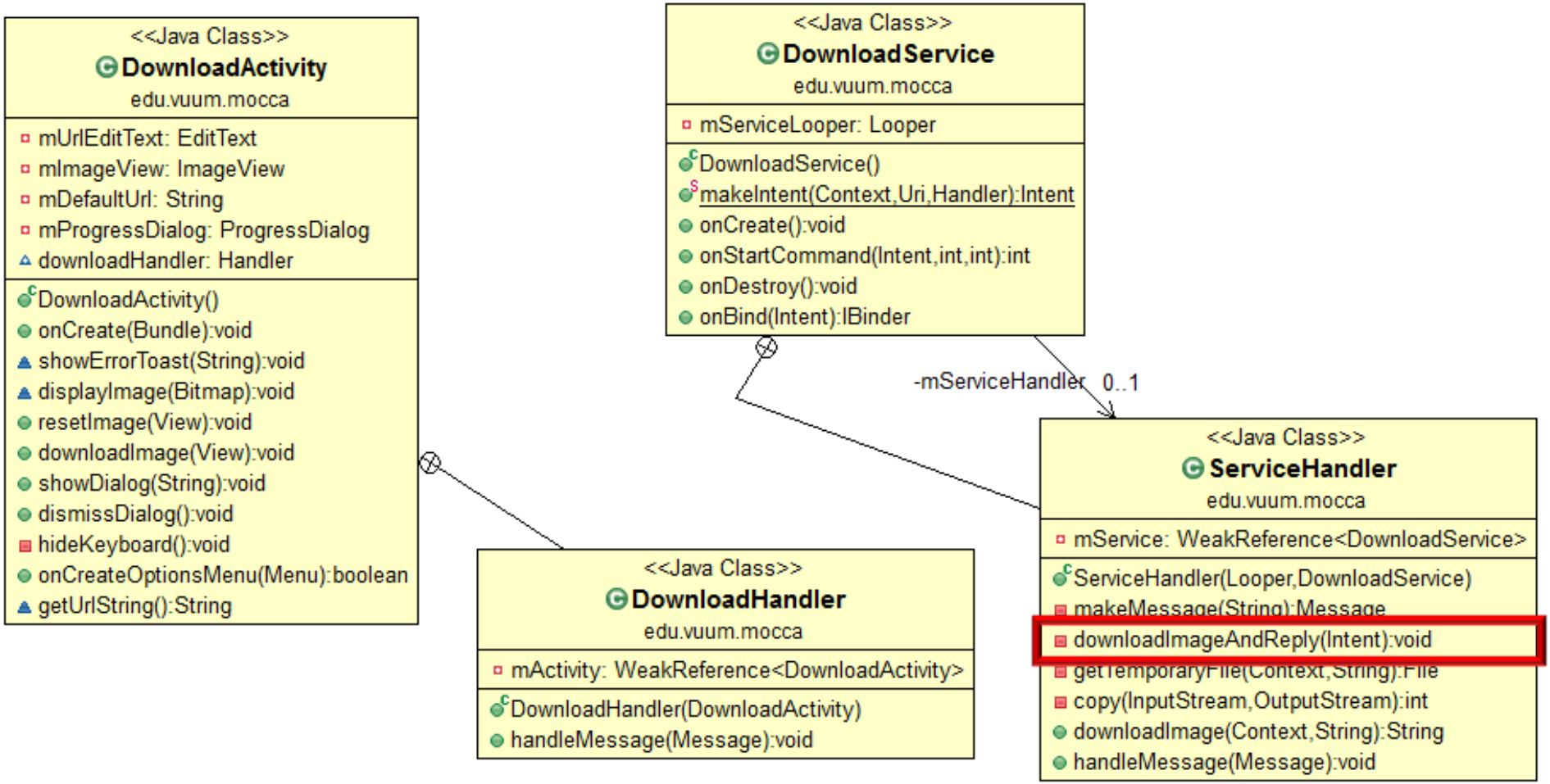
Using Messenger in the Download Application



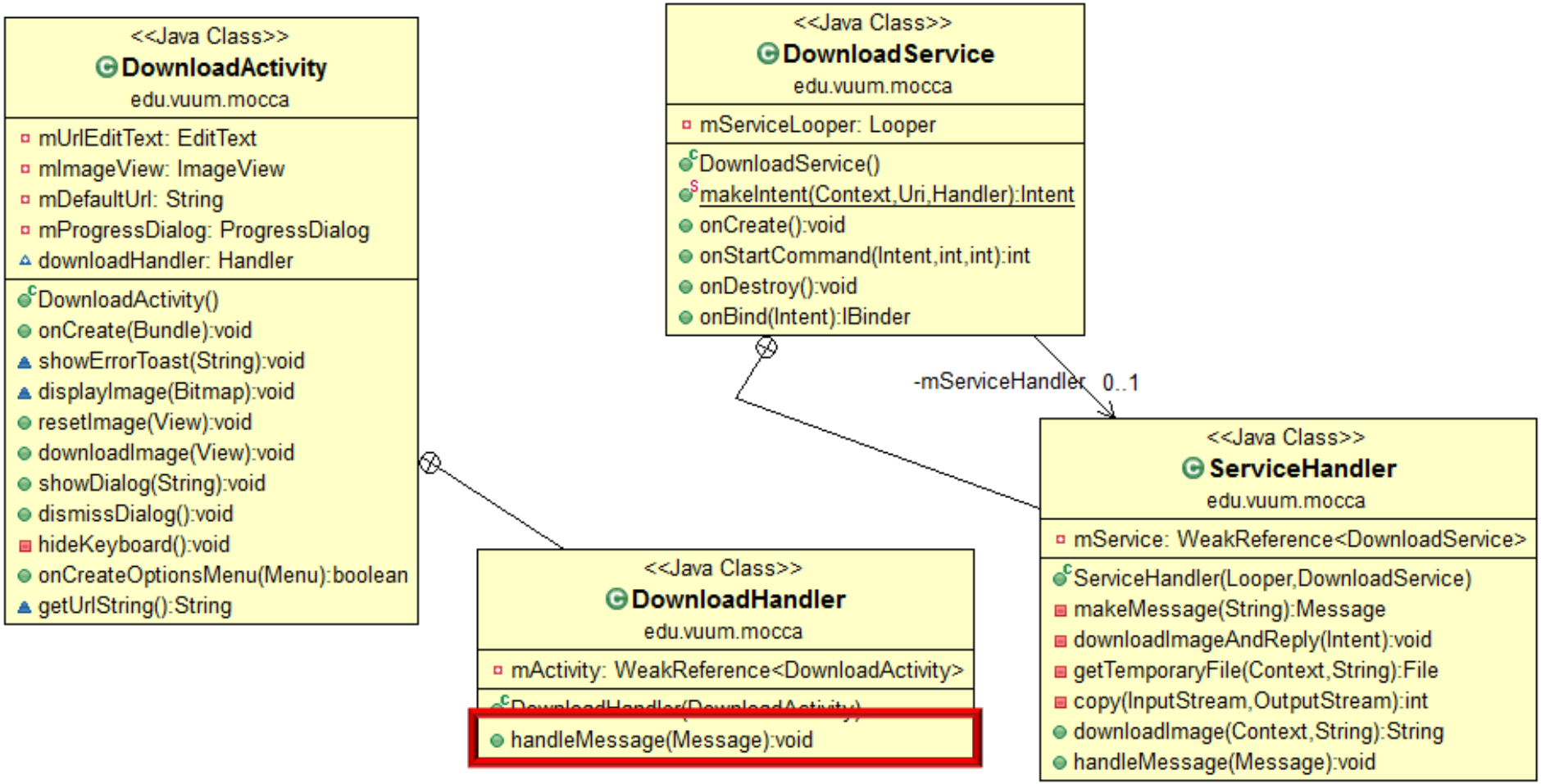
Using Messenger in the Download Application



Using Messenger in the Download Application

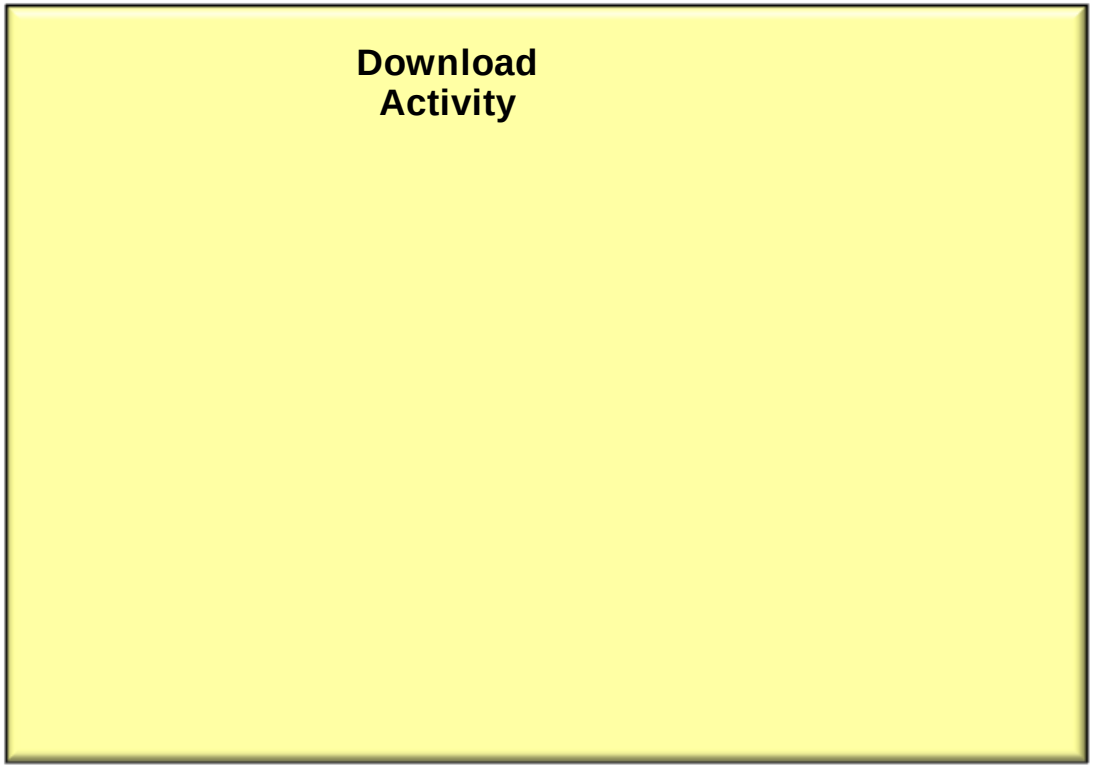


Using Messenger in the Download Application



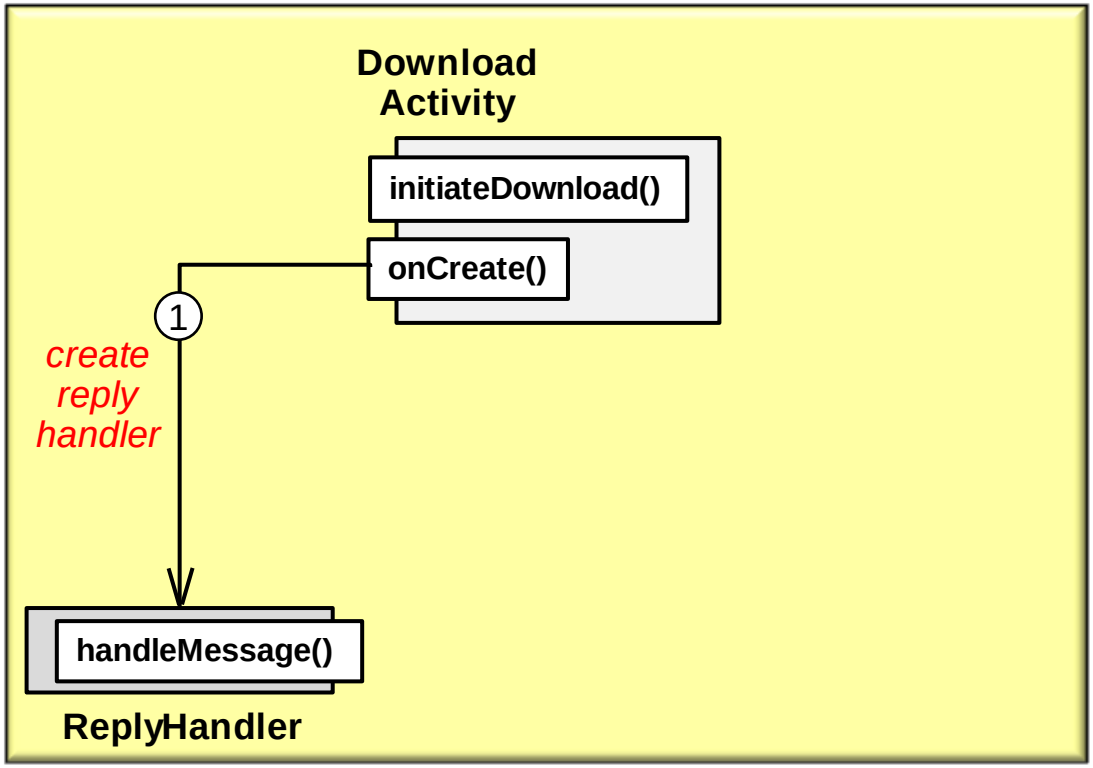
Using Messenger in the DownloadActivity

- DownloadActivity performs several steps



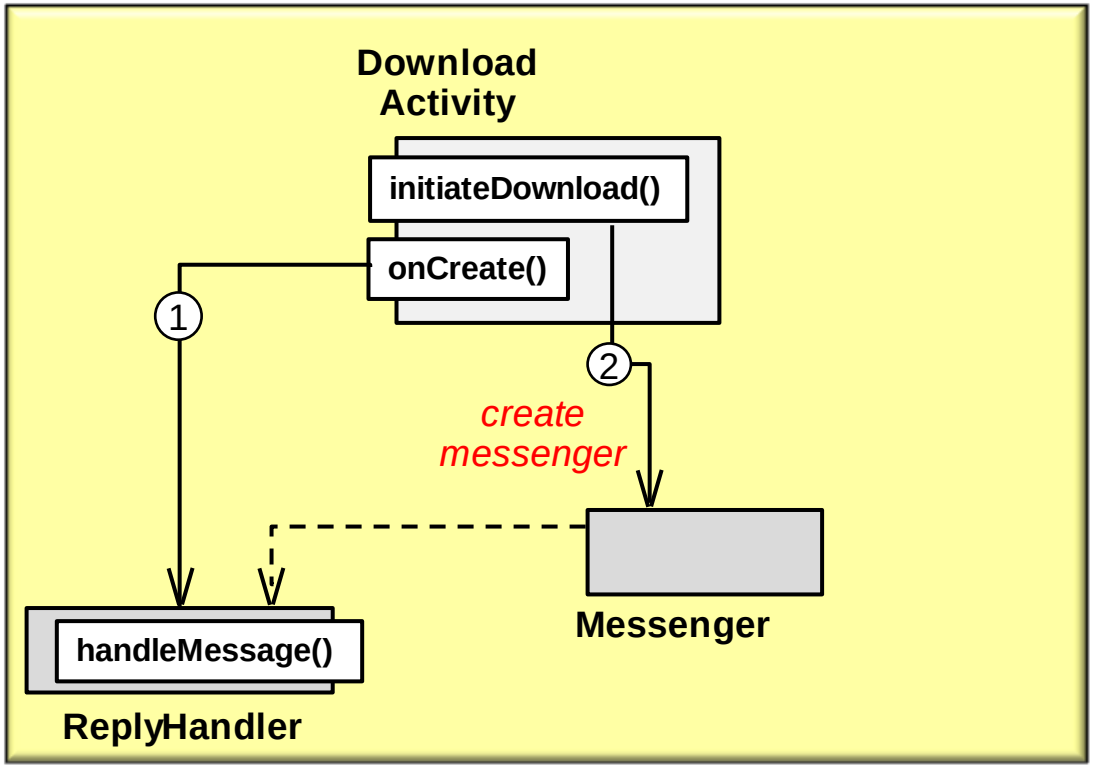
Using Messenger in the DownloadActivity

- DownloadActivity performs several steps
  - Creates a DownloadHandler & a Messenger that encapsulates the Handler



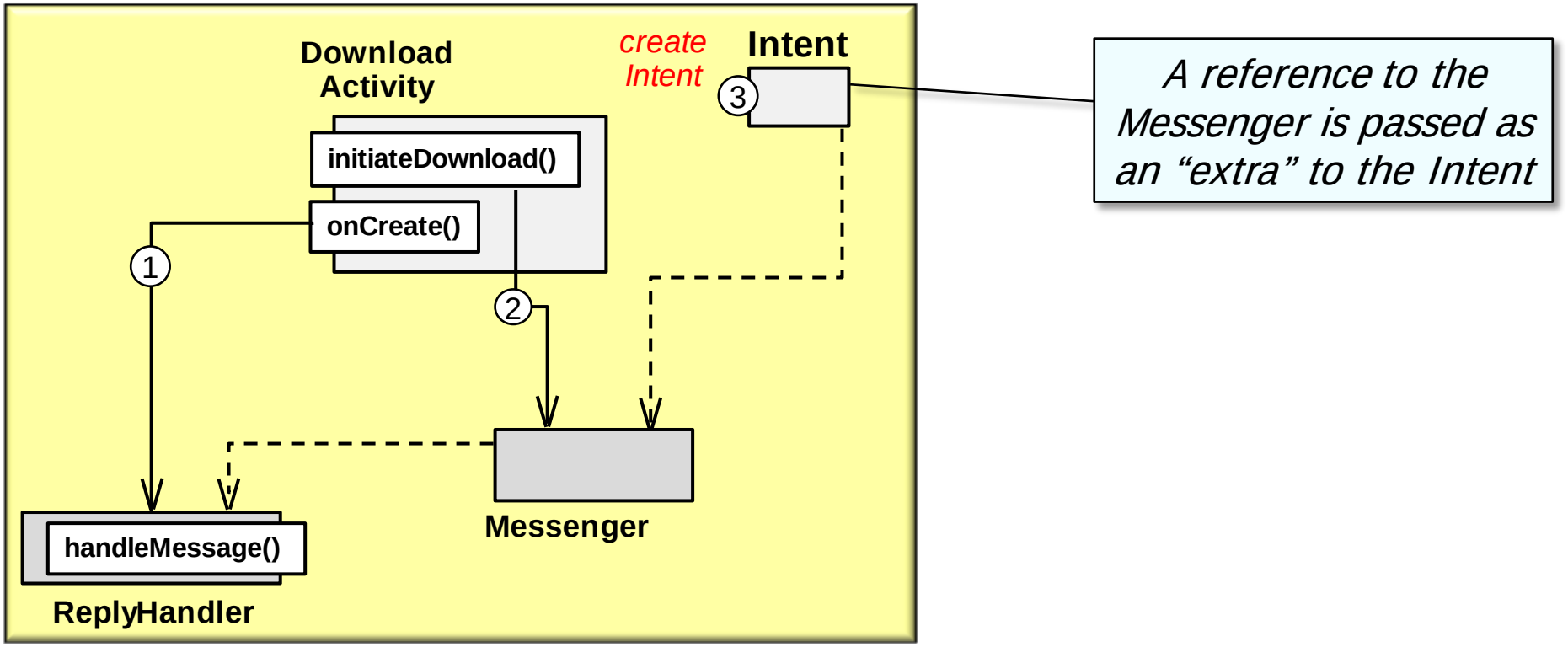
Using Messenger in the DownloadActivity

- DownloadActivity performs several steps
  - Creates a DownloadHandler & a Messenger that encapsulates the Handler



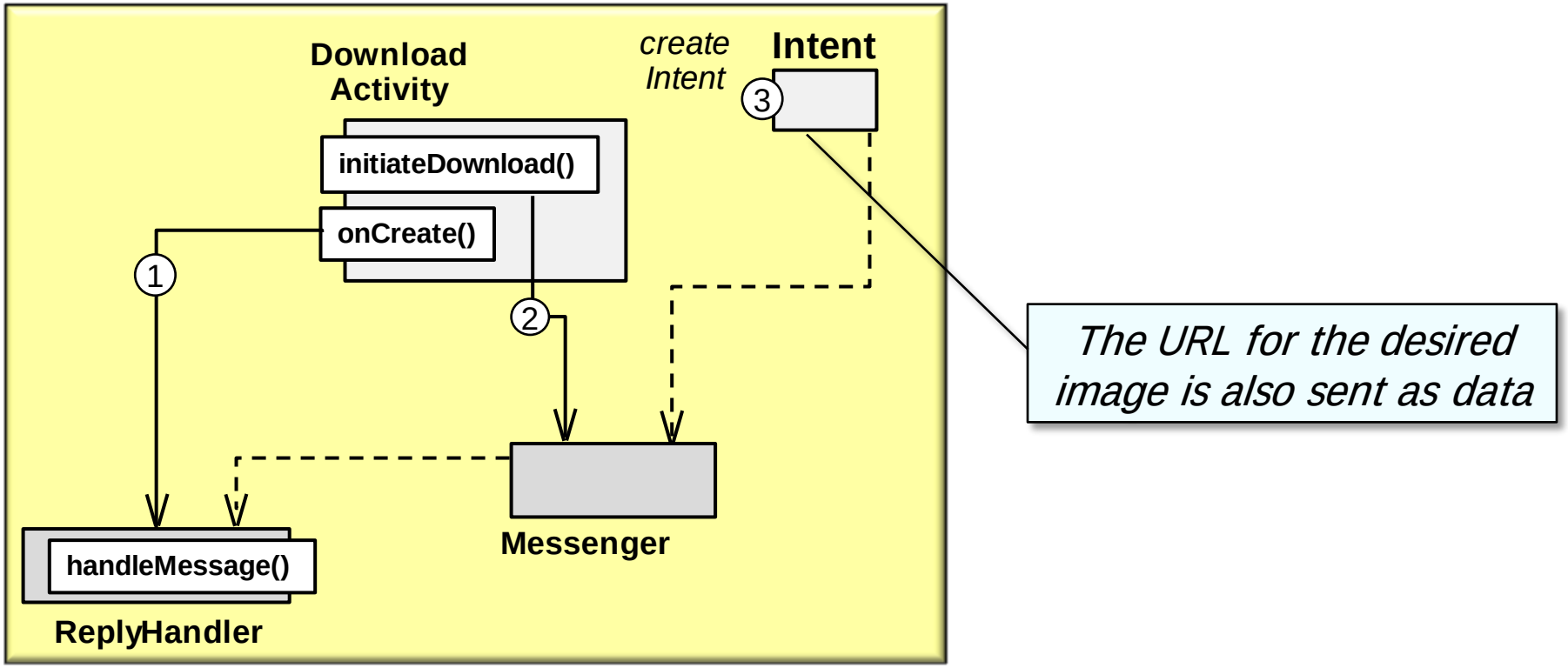
Using Messenger in the DownloadActivity

- DownloadActivity performs several steps
  - Creates a DownloadHandler & a Messenger that encapsulates the Handler
  - Creates an Intent used to start DownloadService



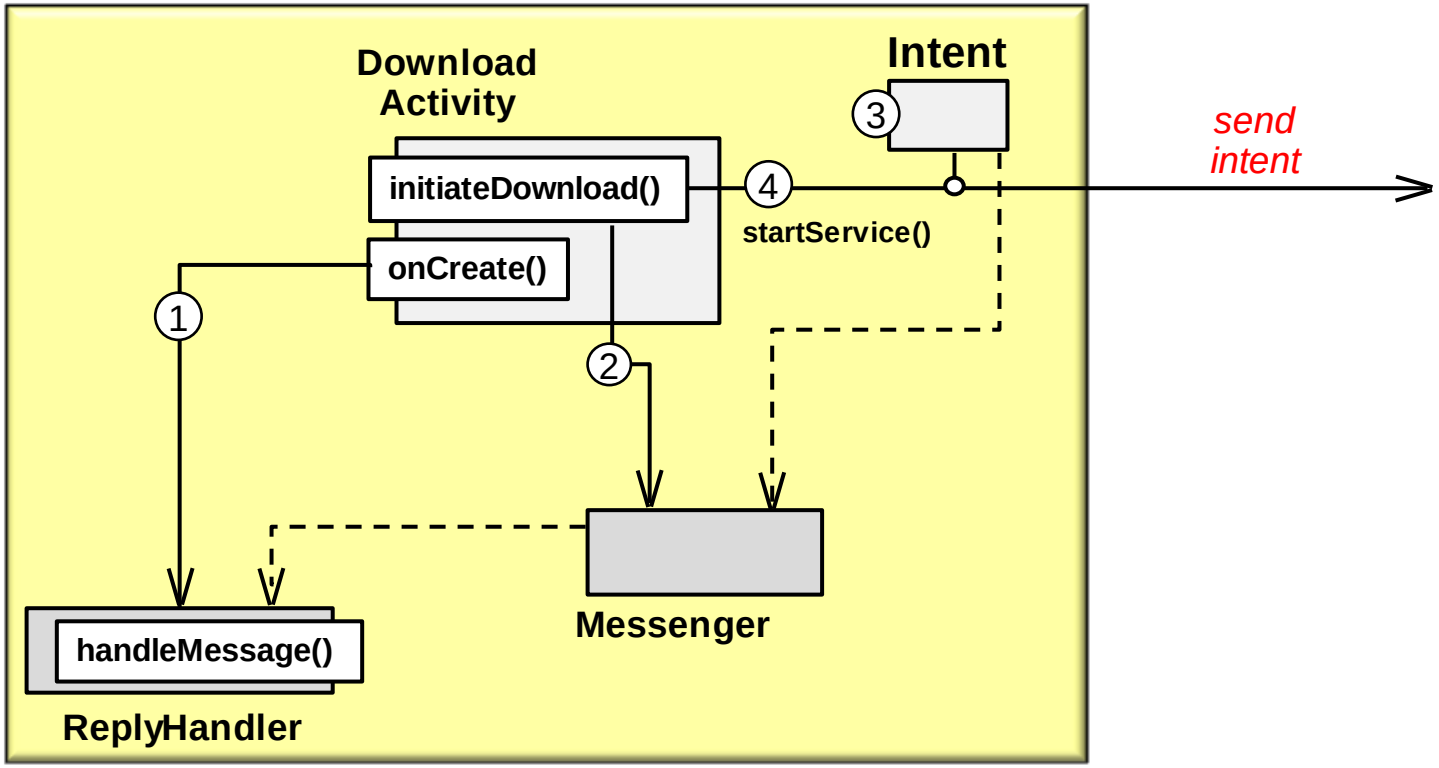
Using Messenger in the DownloadActivity

- DownloadActivity performs several steps
  - Creates a DownloadHandler & a Messenger that encapsulates the Handler
  - Creates an Intent used to start DownloadService



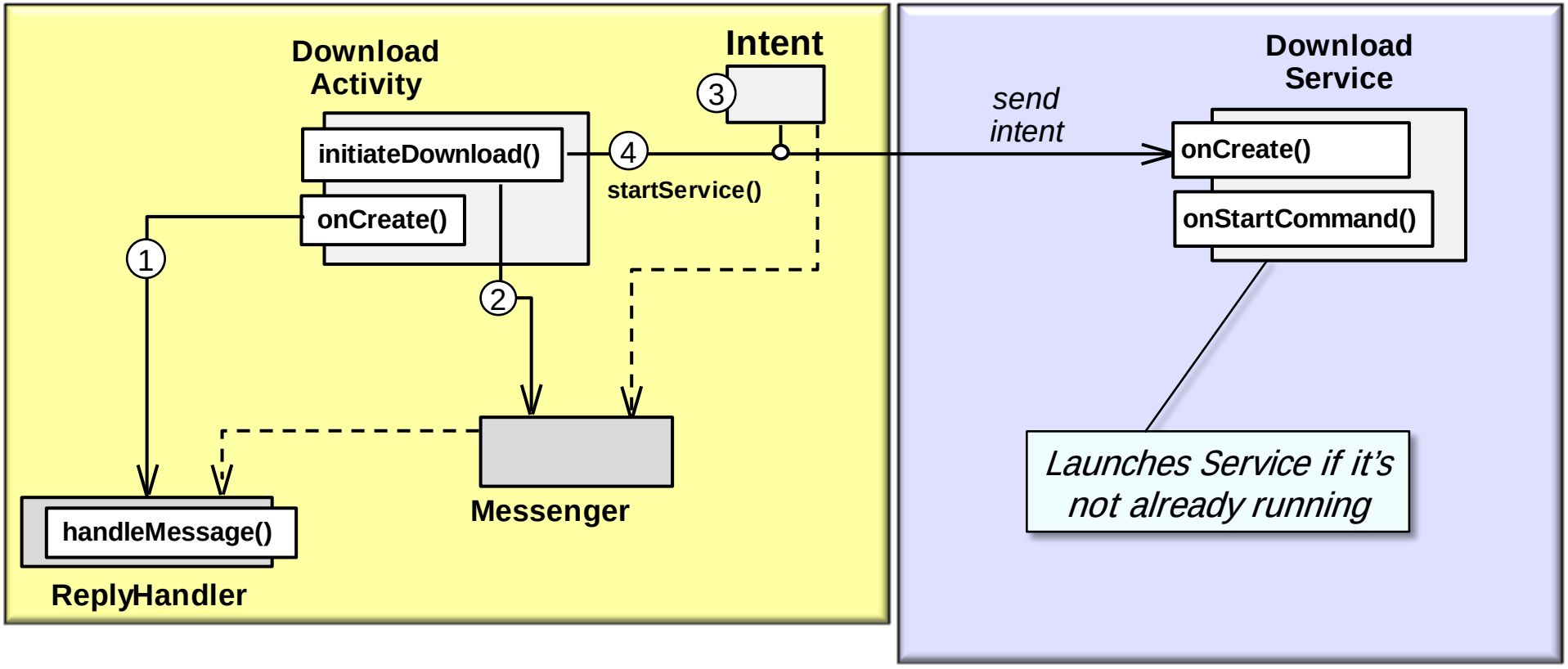
Using Messenger in the DownloadActivity

- DownloadActivity performs several steps
  - Creates a DownloadHandler & a Messenger that encapsulates the Handler
  - Creates an Intent used to start DownloadService
  - Calls startService(), passing the Intent command as a parameter



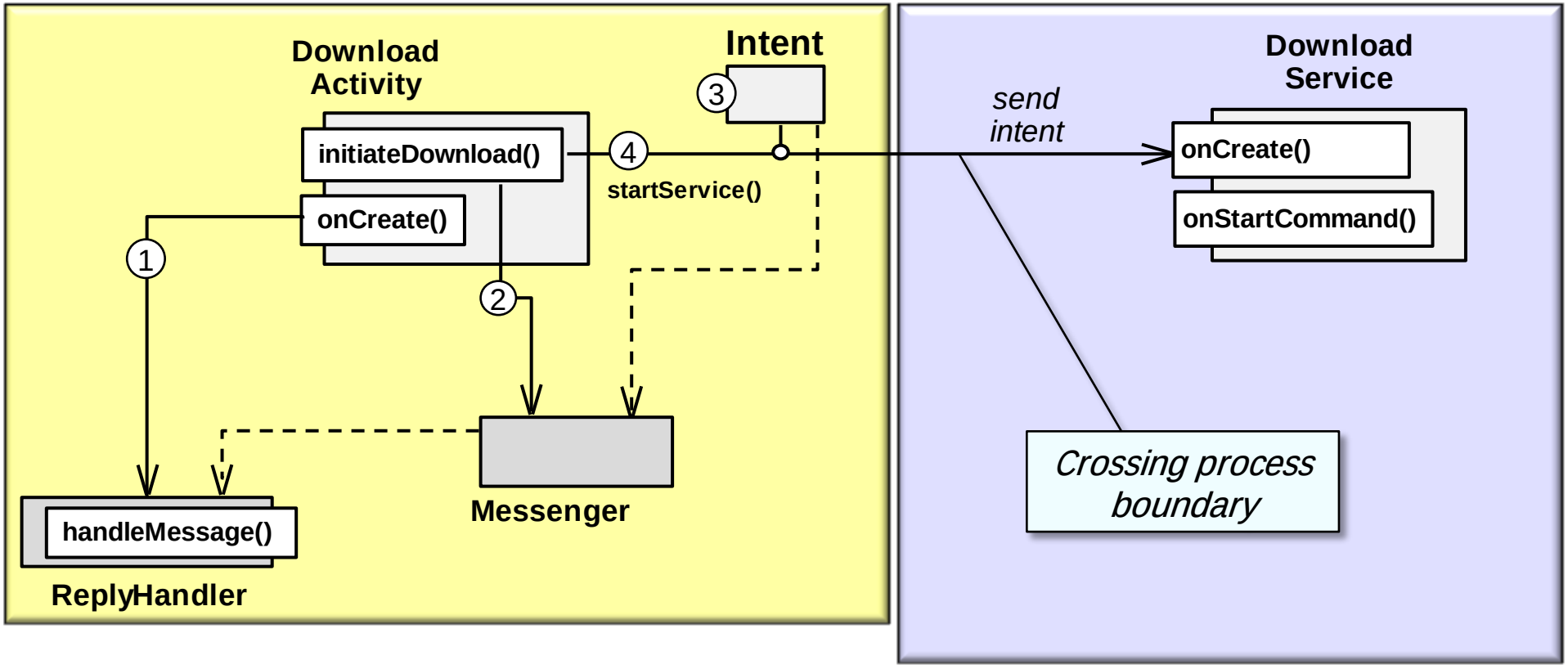
Using Messenger in the DownloadActivity

- DownloadActivity performs several steps
  - Creates a DownloadHandler & a Messenger that encapsulates the Handler
  - Creates an Intent used to start DownloadService
  - Calls startService(), passing the Intent command as a parameter



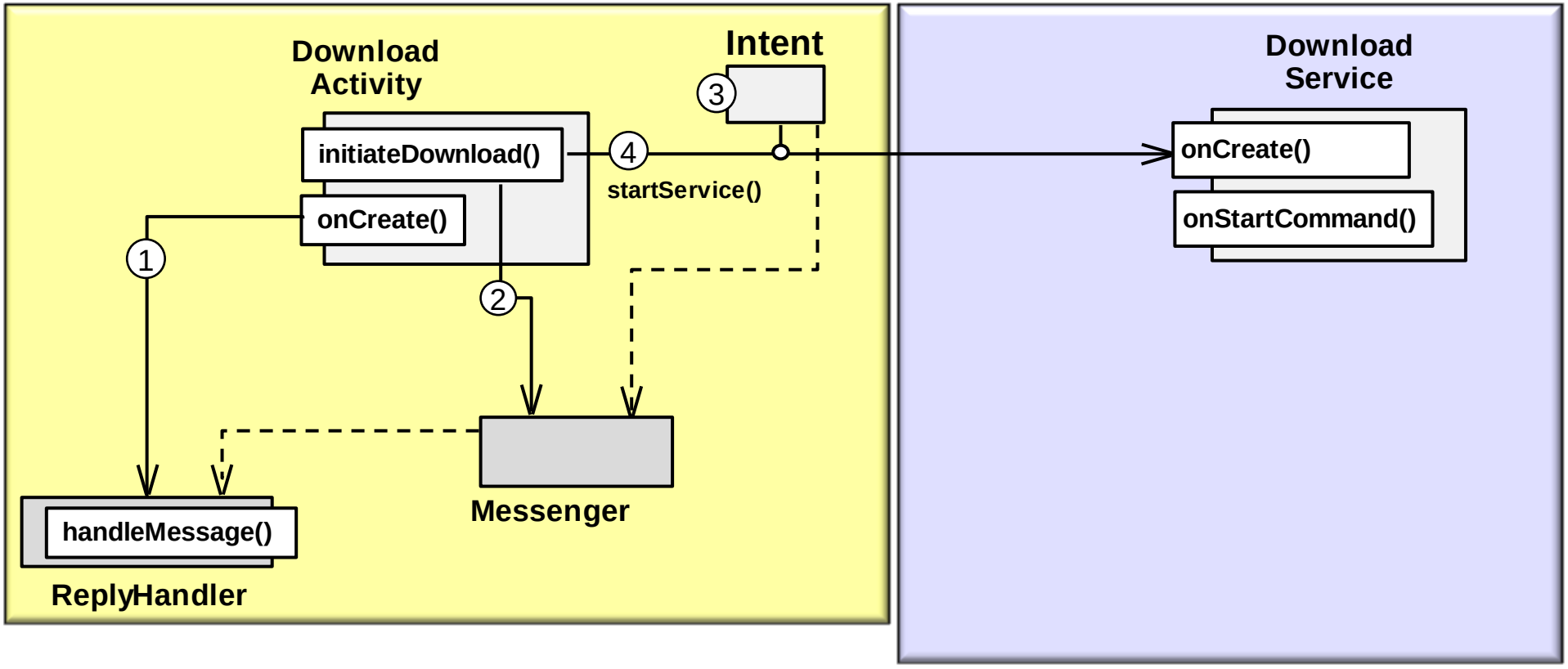
Using Messenger in the DownloadService

- DownloadActivity performs several steps
  - Creates a DownloadHandler & a Messenger that encapsulates the Handler
  - Creates an Intent used to start DownloadService
  - Calls startService(), passing the Intent command as a parameter



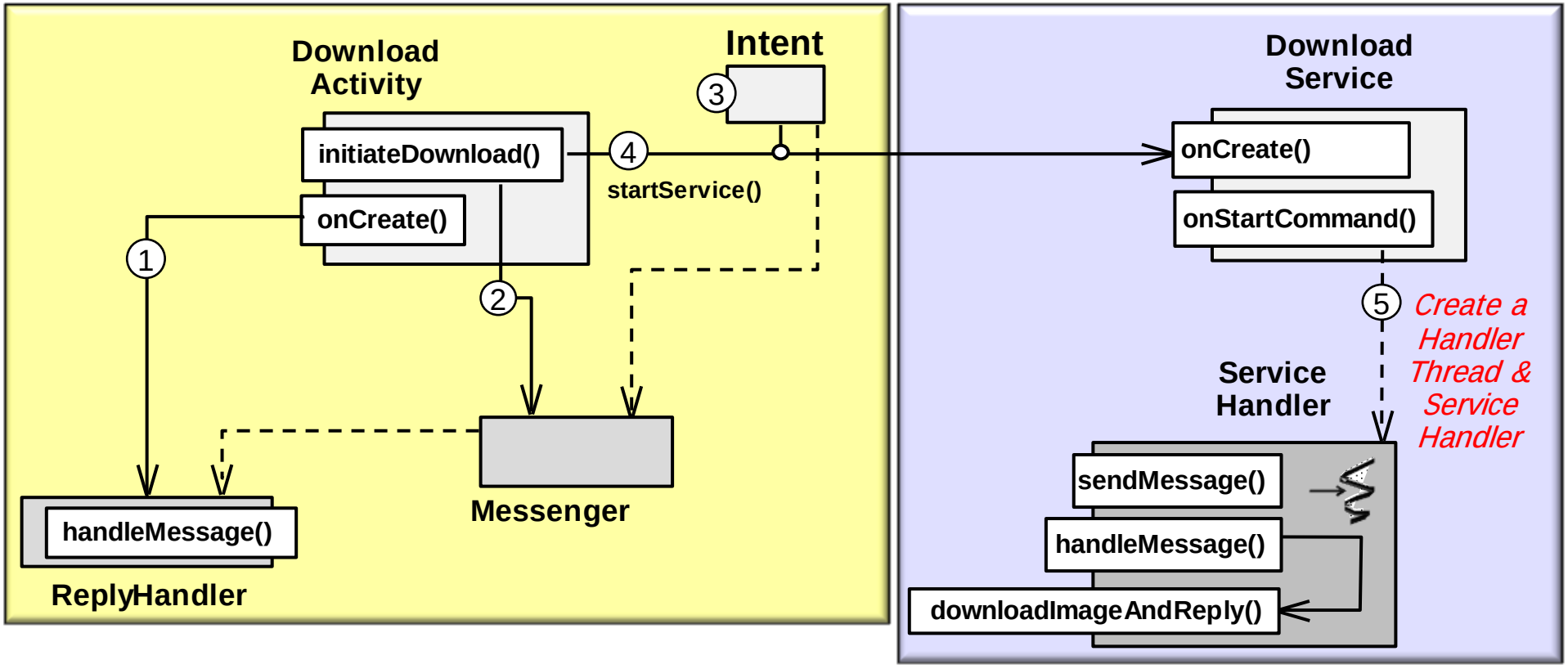
Using Messenger in the DownloadService

- Download Service also performs several steps



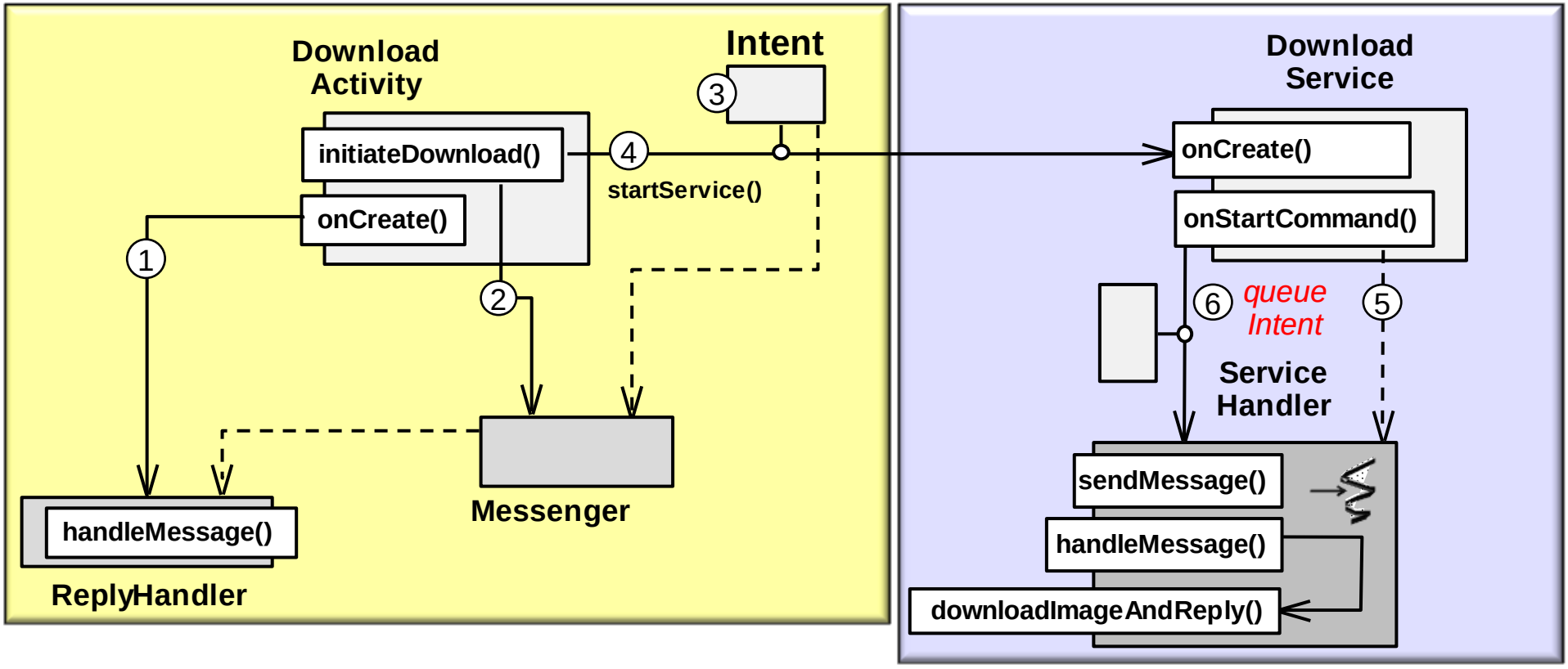
Using Messenger in the DownloadService

- Download Service also performs several steps
  - Creates a Handler Thread & Service Handler



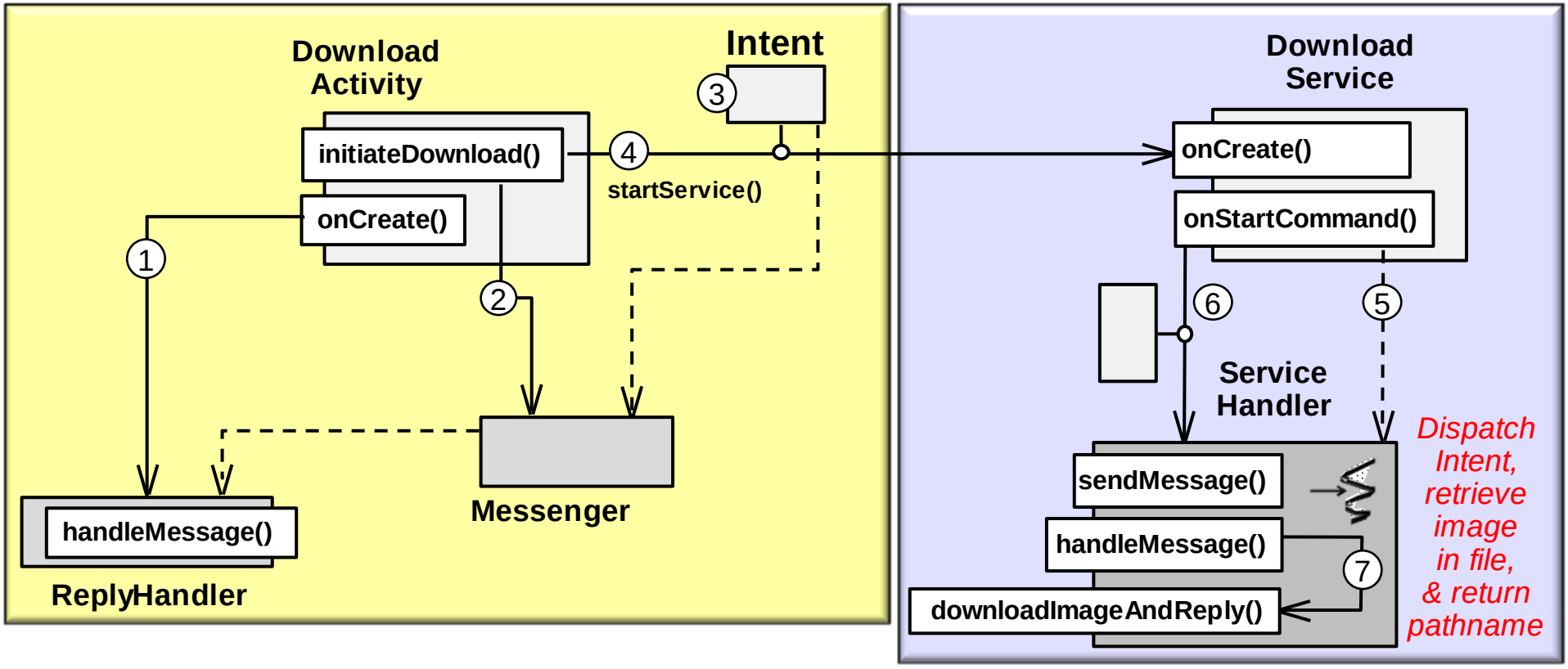
Using Messenger in the DownloadService

- Download Service also performs several steps
  - Creates a Handler Thread & Service Handler
  - Queues Intent in the Service Handler's MessageQueue



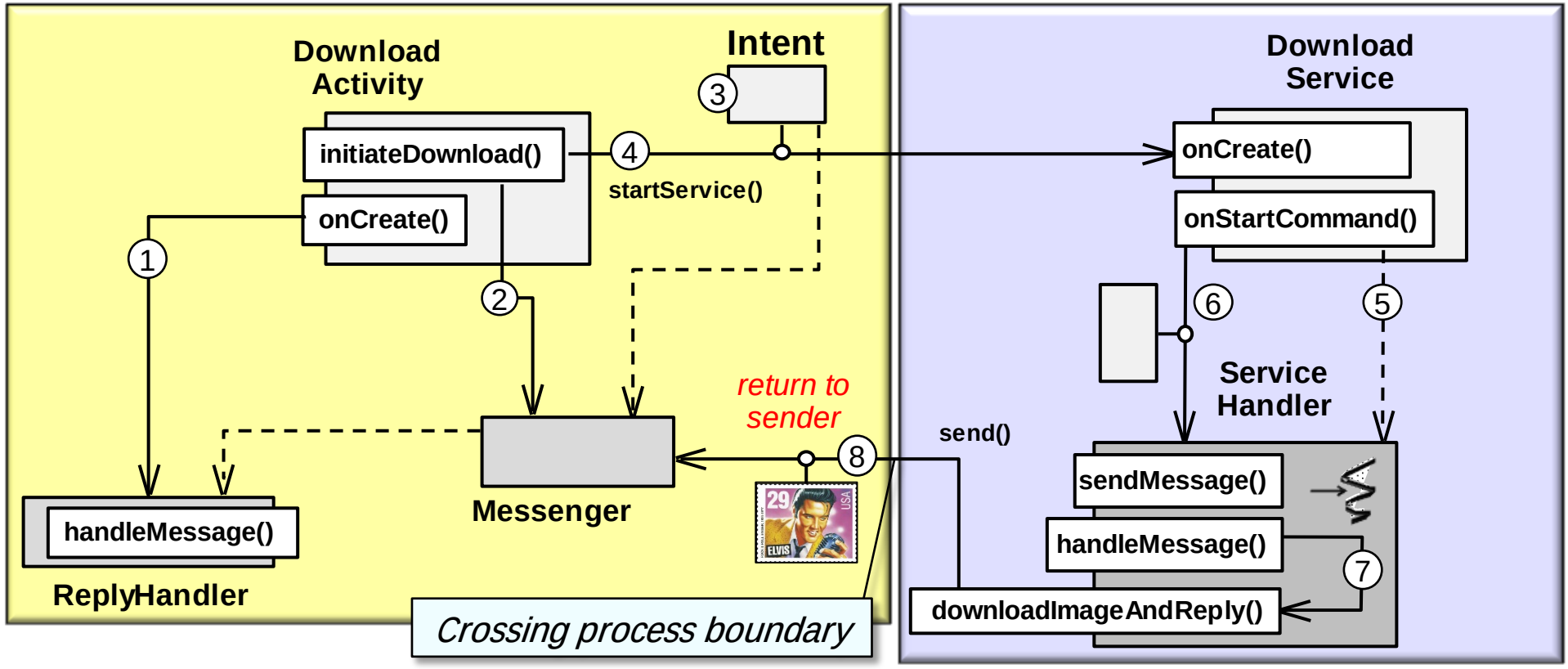
Using Messenger in the DownloadService

- Download Service also performs several steps
  - Creates a Handler Thread & Service Handler
  - Queues Intent in the Service Handler's MessageQueue
  - Dispatch Intent to handleMessage(), which calls downloadImageAndReply()



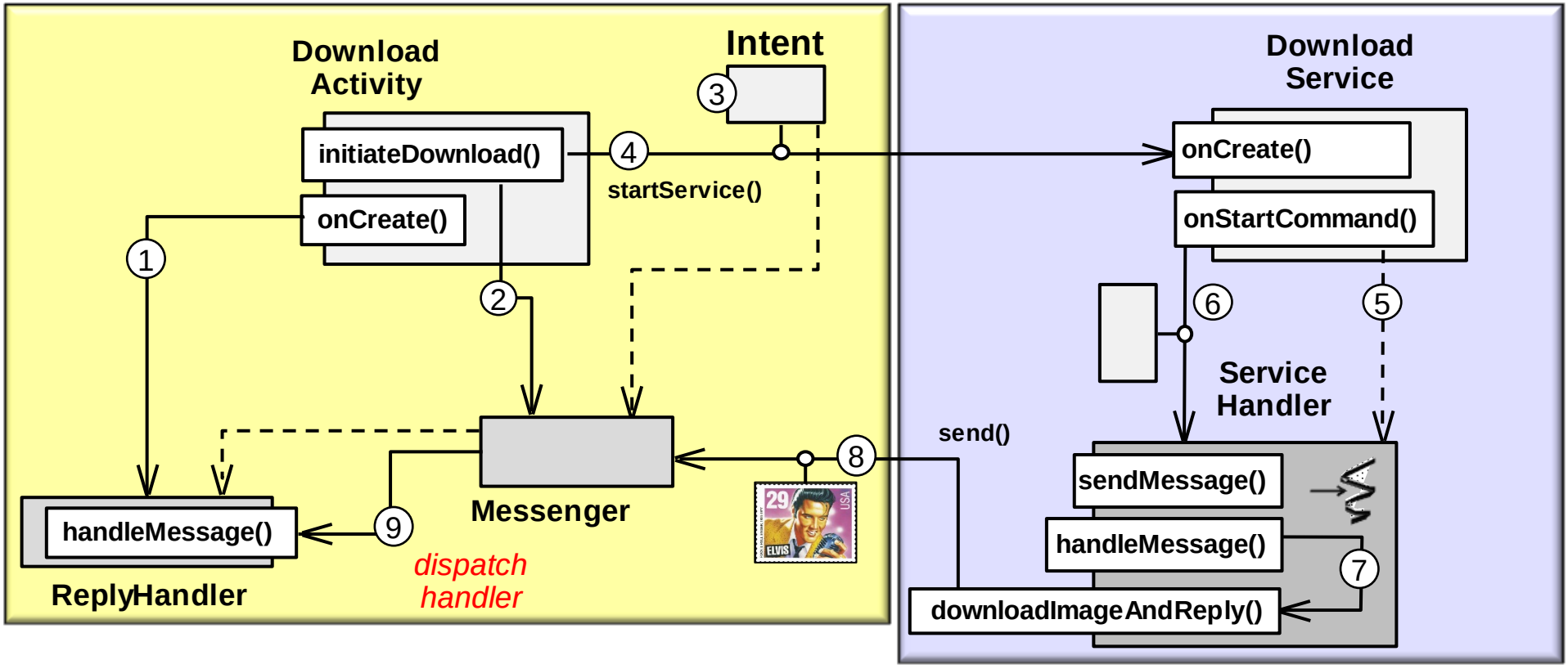
Using Messenger in the DownloadService

- Download Service also performs several steps
  - Creates a Handler Thread & Service Handler
  - Queues Intent in the Service Handler's MessageQueue
  - Dispatch Intent to handleMessage(), which calls downloadImageAndReply()



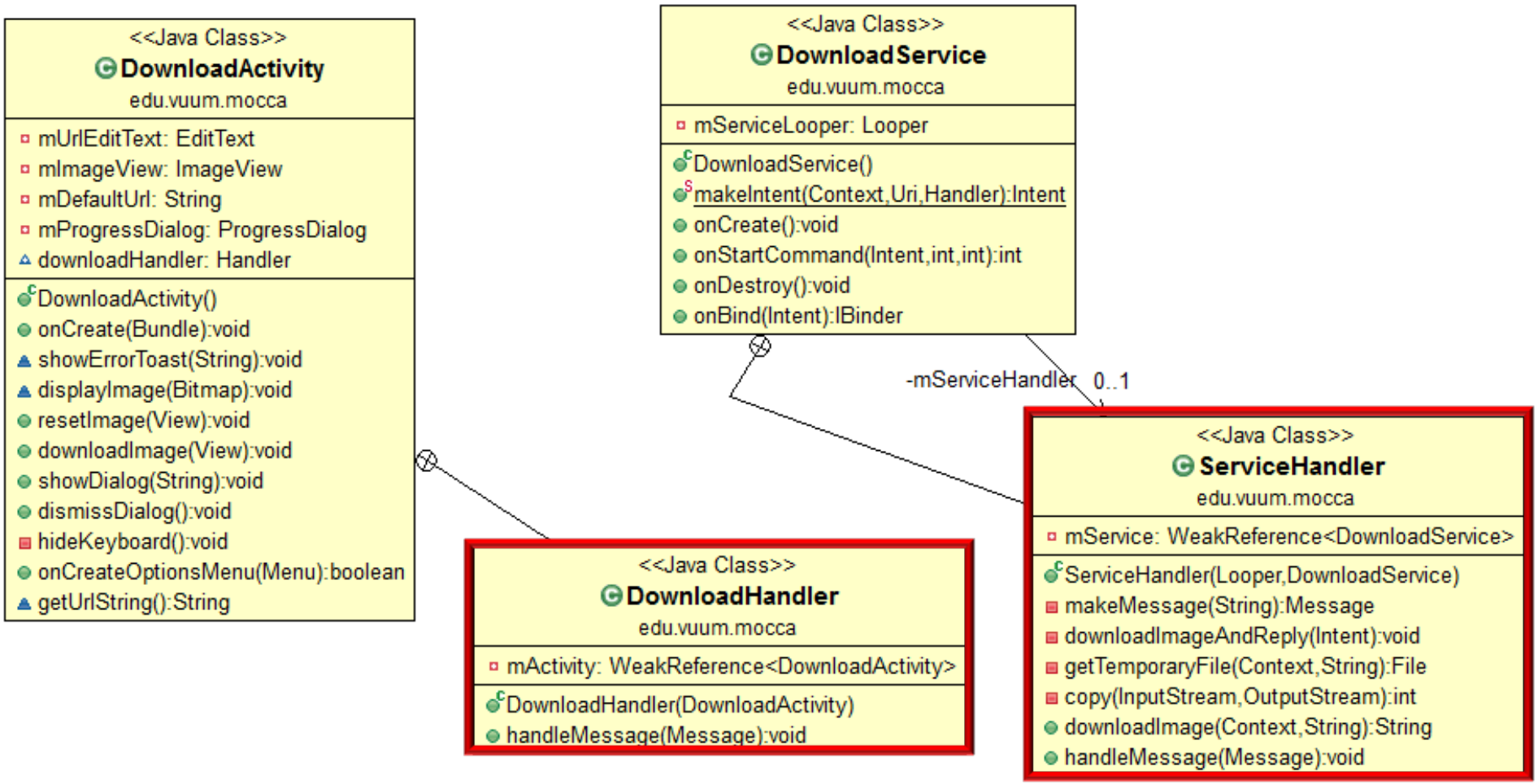
Using Messenger in the DownloadService

- Download Service also performs several steps
  - Creates a Handler Thread & Service Handler
  - Queues Intent in the Service Handler's MessageQueue
  - Dispatch Intent to handleMessage(), which calls downloadImageAndReply()



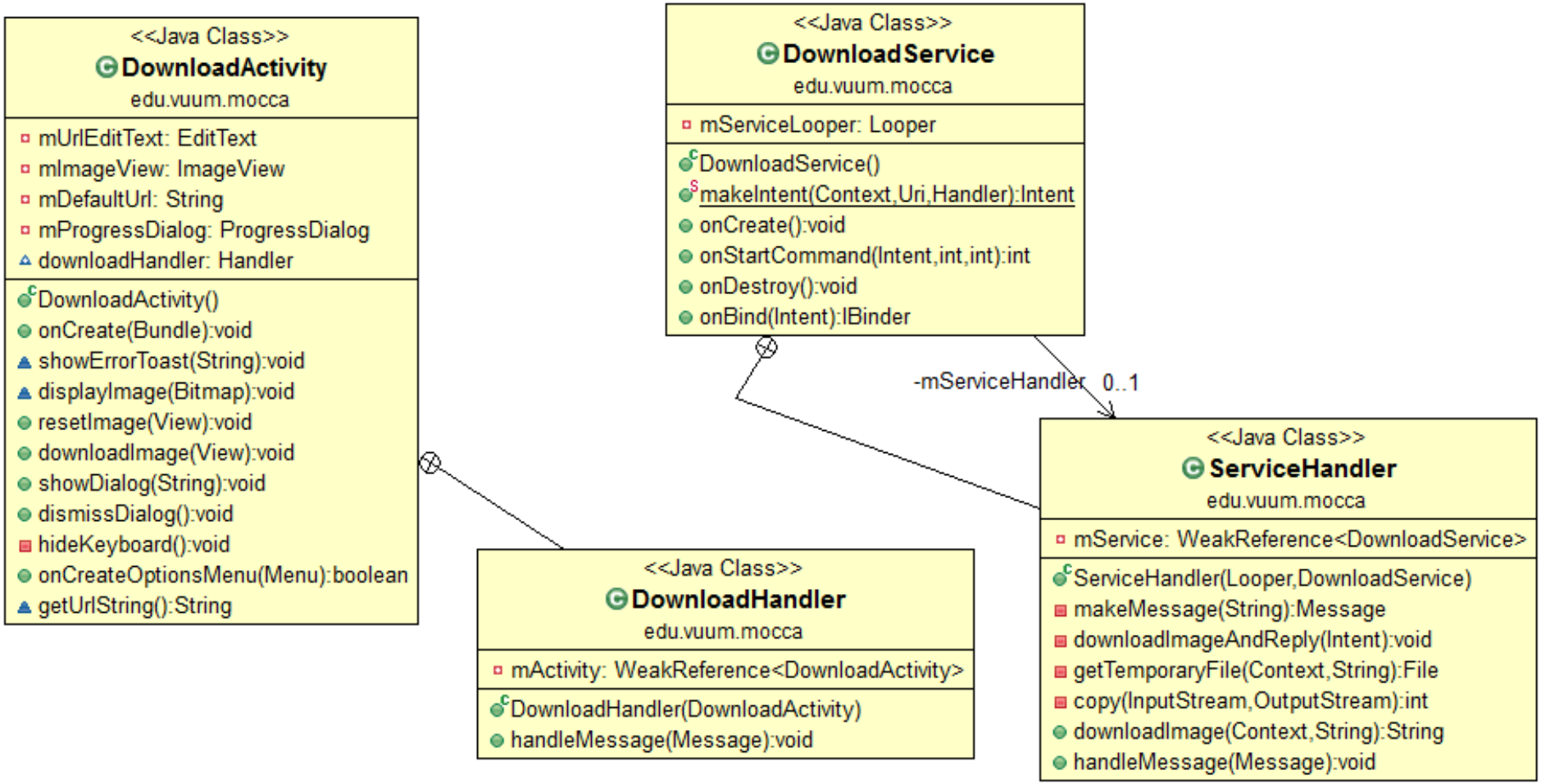
Programming the Messenger in the Download Application (Part 1)

Programming Messenger in Download Application



See earlier parts on “Programming Started Services”

Programming Messenger in Download Application



github.com/douglascraigschmidt/POSA-14/tree/master/ex/DownloadApplication

Programming Messenger in Download Application

- DownloadActivity passes a Messenger to the DownloadService

```
public class DownloadActivity extends Activity {
   ...
   Handler downloadHandler = new DownloadHandler(this);



   public void downloadImage(View v) {
      Uri uri = getUrlString();



      Intent intent = DownloadService.makeIntent
                          (this, Uri.parse(uri), downloadHandler);

      startService(intent);
   }

   ...
```

Programming Messenger in Download Application

- DownloadActivity passes a Messenger to the DownloadService

```
public class DownloadActivity extends Activity {
   ...
   Handler downloadHandler = new DownloadHandler(this);
```

Create a Handler to process replies from DownloadService

```
   public void downloadImage(View v) {
      Uri uri = getUrlString();



      Intent intent = DownloadService.makeIntent
                        (this, Uri.parse(uri), downloadHandler);

      startService(intent);
   }

   ...
```

Programming Messenger in Download Application

- DownloadActivity passes a Messenger to the DownloadService

```
public class DownloadActivity extends Activity {
  ...
  Handler downloadHandler = new DownloadHandler(this);



  public void downloadImage(View v) {
     Uri uri = getUrlString();

     Factory method to make the Intent

     Intent intent = DownloadService.makeIntent
                        (this, Uri.parse(uri), downloadHandler);

     startService(intent);
  }

  ...
```

Programming Messenger in Download Application

- DownloadService replies to Activity via Messenger's send() method

```
public class DownloadService extends Service {

  public static Intent makeIntent(Context context,
                                  Uri uri,
                                  Handler downloadHandler) {
    Intent intent = new Intent(context,
                               DownloadService.class);

    intent.setData(uri);

    intent.putExtra("MESSENGER",
                    new Messenger(downloadHandler));
    return intent;
  }

  ...
```

Factory method to make the right intent

Programming Messenger in Download Application

- DownloadService replies to Activity via Messenger's send() method

```
public class DownloadService extends Service {



  public static Intent makeIntent(Context context,
                                  Uri uri,
                                  Handler downloadHandler) {
    Intent intent = new Intent(context,
                               DownloadService.class);

    intent.setData(uri);

    intent.putExtra("MESSENGER",
                    new Messenger(downloadHandler));
    return intent;
  }

  ...
```

Pass a Messenger as "extra" in Intent used to start DownloadService

Programming Messenger in Download Application

- DownloadService replies to Activity via Messenger's send() method

```
public class DownloadService extends Service {



  public static Intent makeIntent(Context context,
                                  Uri uri,
                                  Handler downloadHandler) {
    Intent intent = new Intent(context,
                               DownloadService.class);

    intent.setData(uri);

    intent.putExtra("MESSENGER",
                    new Messenger(downloadHandler));
    return intent;
  }

  ...
```

Pass a Messenger as "extra" in Intent used to start DownloadService

Programming Messenger in Download Application

- DownloadService replies to Activity via Messenger's send() method

```
public class DownloadService extends Service {



  public static Intent makeIntent(Context context,
                                  Uri uri,
                                  Handler downloadHandler) {
    Intent intent = new Intent(context,
                               DownloadService.class);

    intent.setData(uri);

    intent.putExtra("MESSENGER",
                    new Messenger(downloadHandler));
    return intent;
  }

  ...
```

Pass a Messenger as "extra" in Intent used to start DownloadService

Programming Messenger in Download Application

- DownloadActivity passes a Messenger to the DownloadService

```
public class DownloadActivity extends Activity {
  ...
  Handler downloadHandler = new DownloadHandler(this);



  public void downloadImage(View v) {
     Uri uri = getUrlString();

     Factory method to make the Intent

     Intent intent = DownloadService.makeIntent
                        (this, Uri.parse(uri), downloadHandler);

     startService(intent);
  }

  ...
```

Programming Messenger in Download Application

- DownloadActivity passes a Messenger to the DownloadService

```
public class DownloadActivity extends Activity {
  ...
  Handler downloadHandler = new DownloadHandler(this);



  public void downloadImage(View v) {
     Uri uri = getUrlString();



     Intent intent = DownloadService.makeIntent
                        (this, Uri.parse(uri), downloadHandler);

     startService(intent);
  }
  ...
```

Start the DownloadService

Programming Messenger in Download Application

- DownloadActivity passes a Messenger to the DownloadService

```
public class DownloadActivity extends Activity {
  ...
  Handler downloadHandler = new DownloadHandler(this);



  public void downloadImage(View v) {
     Uri uri = getUrlString();



     Intent intent = DownloadService.makeIntent
                        (this, Uri.parse(uri), downloadHandler);

     startService(intent);
  }
  ...
```

Start the DownloadService

Programming the Messenger in the Download Application (Part 2)

Programming Messenger in Download Application

- DownloadService replies to Activity via Messenger's send() method

```
public class DownloadService extends Service {
  ...
  private final class ServiceHandler extends Handler {

    private void downloadImageAndReply(Intent intent) {


      String pathname =
        downloadImage(DownloadService.this,
                      intent.getData().toString());


      Messenger messenger = (Messenger) intent.getExtras().
                                          get("MESSENGER");

      sendPath(messenger, pathname);
    }
    ...
```

Programming Messenger in Download Application

- DownloadService replies to Activity via Messenger's send() method

```
public class DownloadService extends Service {
  ...
  private final class ServiceHandler extends Handler {

    private void downloadImageAndReply(Intent intent) {

      String pathname =
        downloadImage(DownloadService.this,
                      intent.getData().toString());


      Messenger messenger = (Messenger) intent.getExtras().
                                              get("MESSENGER");

      sendPath(messenger, pathname);
    }
    ...
```

Retrieve the designated image & reply to the DownloadActivity

Programming Messenger in Download Application

- DownloadService replies to Activity via Messenger's send() method

```
public class DownloadService extends Service {
  ...
  private final class ServiceHandler extends Handler {

    private void downloadImageAndReply(Intent intent) {


      String pathname =
        downloadImage(DownloadService.this,
                      intent.getData().toString());

      Messenger messenger = (Messenger) intent.getExtras().
                                          get("MESSENGER");

      sendPath(messenger, pathname);
    }
    ...
```

Retrieve the designated image

Programming Messenger in Download Application

- DownloadService replies to Activity via Messenger's send() method

```
public class DownloadService extends Service {
  ...
  private final class ServiceHandler extends Handler {

    private void downloadImageAndReply(Intent intent) {


      String pathname =
        downloadImage(DownloadService.this,
                      intent.getData().toString());


      Messenger messenger = (Messenger) intent.getExtras().
                                          get("MESSENGER");


      sendPath(messenger, pathname);
    }
    ...
```

Extract Messenger from the Intent

Programming Messenger in Download Application

- DownloadService replies to Activity via Messenger's send() method

```
public class DownloadService extends Service {
  ...
  private final class ServiceHandler extends Handler {

    private void downloadImageAndReply(Intent intent) {


      String pathname =
        downloadImage(DownloadService.this,
                      intent.getData().toString());


      Messenger messenger = (Messenger) intent.getExtras().
                                              get("MESSENGER");

      sendPath(messenger, pathname);
  }
  ...
```

Send the pathname via the Messenger

Programming Messenger in Download Application

- DownloadService replies to Activity via Messenger's send() method

```
public class DownloadService extends Service {
  ...
  private final class ServiceHandler extends Handler {

    private void sendPath(Messenger messenger,
                          String pathname) {
      Message message = makeReplyMessage(pathname);



      try {
        messenger.send(message);
      }
      ...
```

Programming Messenger in Download Application

- DownloadService replies to Activity via Messenger's send() method

```
public class DownloadService extends Service {
  ...
  private final class ServiceHandler extends Handler {

    private void sendPath(Messenger messenger,
                          String pathname) {
      Message message = makeReplyMessage(pathname);

      try {
        messenger.send(message);
      }
      ...
```

Factory method that creates a Message to return to the Download Activity

Programming Messenger in Download Application

- DownloadService replies to Activity via Messenger's send() method

```
public class DownloadService extends Service {
  ...
  private final class ServiceHandler extends Handler {

    private Message makeReplyMessage(String pathname){
```

Factory method creates a Message that encapsulates pathname for downloaded image

```
      Message message = Message.obtain();
      message.arg1 = pathname == null
        ? Activity.RESULT_CANCELED : Activity.RESULT_OK;

      Bundle bundle = new Bundle();
      bundle.putString("PATHNAME", pathname);
      message.setData(bundle);
      return message;
    }
    ...
```

Programming Messenger in Download Application

- DownloadService replies to Activity via Messenger's send() method

```
public class DownloadService extends Service {
  ...
  private final class ServiceHandler extends Handler {

    private Message makeReplyMessage(String pathname){

      Message message = Message.obtain();
      message.arg1 = pathname == null
        ? Activity.RESULT_CANCELED : Activity.RESULT_OK;

      Bundle bundle = new Bundle();
      bundle.putString("PATHNAME", pathname);
      message.setData(bundle);
      return message;
   }
   ...
```

Store the result indicating whether the download succeeded or failed

Programming Messenger in Download Application

- DownloadService replies to Activity via Messenger's send() method

```
public class DownloadService extends Service {
  ...
  private final class ServiceHandler extends Handler {

    private Message makeReplyMessage(String pathname){



      Message message = Message.obtain();
      message.arg1 = pathname == null
        ? Activity.RESULT_CANCELED : Activity.RESULT_OK;

      Bundle bundle = new Bundle();
      bundle.putString("PATHNAME", pathname);
      message.setData(bundle);
      return message;
  }
  ...
```

Create a Bundle to store the pathname String for the downloaded image

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

Programming Messenger in Download Application

- DownloadService replies to Activity via Messenger's send() method

```
public class DownloadService extends Service {
  ...
  private final class ServiceHandler extends Handler {

    private void sendPath(Messenger messenger,
                          String pathname) {
      Message message = makeReplyMessage(pathname);

      try {
        messenger.send(message);
      }
      ...
```

Factory method that creates a Message to return to the Download Activity

Programming Messenger in Download Application

- DownloadService replies to Activity via Messenger's send() method

```
public class DownloadService extends Service {
  ...
  private final class ServiceHandler extends Handler {

    private void sendPath(Messenger messenger,
                          String pathname) {
      Message message = makeReplyMessage(pathname);



      try {
        messenger.send(message);
      }
      ...
```

Pass the Message back to the Download Activity

Programming Messenger in Download Application

- DownloadService replies to Activity via Messenger's send() method

```
public class DownloadService extends Service {
  ...
  private final class ServiceHandler extends Handler {

    private void sendPath(Messenger messenger,
                          String pathname) {
      Message message = makeReplyMessage(pathname);



      try {
        messenger.send(message);
      }
      ...
```

Pass the Message back to the Download Activity

Programming Messenger in Download Application

- DownloadActivity receives Message via its Handler in the UI Thread

```
public class DownloadActivity extends Activity {
  ...
  private static class DownloadHandler extends Handler {
    public void handleMessage(Message msg) {
      ...

      Bundle data = msg.getData();

      String pathname = data.getString("PATHNAME");


      if (msg.arg1 != RESULT_OK || pathname == null)
         activity.showDialog("failed download");

      ...
      activity.displayImage
        (BitmapFactory.decodeFile(pathname));
      ...
```

Programming Messenger in Download Application

- DownloadActivity receives Message via its Handler in the UI Thread

```
public class DownloadActivity extends Activity {
  ...
  private static class DownloadHandler extends Handler {
    public void handleMessage(Message msg) {
      ...
      Bundle data = msg.getData();

      String pathname = data.getString("PATHNAME");


      if (msg.arg1 != RESULT_OK || pathname == null)
         activity.showDialog("failed download");

      ...
      activity.displayImage
        (BitmapFactory.decodeFile(pathname));
      ...
```

Dispatched to process reply from DownloadService

This hook method runs in the context of the UI Thread

Programming Messenger in Download Application

- DownloadActivity receives Message via its Handler in the UI Thread

```
public class DownloadActivity extends Activity {
  ...
  private static class DownloadHandler extends Handler {
    public void handleMessage(Message msg) {
      ...

      Bundle data = msg.getData();

      String pathname = data.getString("PATHNAME");


      if (msg.arg1 != RESULT_OK || pathname == null)
         activity.showDialog("failed download");


      ...
      activity.displayImage
        (BitmapFactory.decodeFile(pathname));
      ...
```

Extract data Message, which is in a Bundle passed across processes

Programming Messenger in Download Application

- DownloadActivity receives Message via its Handler in the UI Thread

```
public class DownloadActivity extends Activity {
  ...
  private static class DownloadHandler extends Handler {
    public void handleMessage(Message msg) {
      ...

      Bundle data = msg.getData();

      String pathname = data.getString("PATHNAME");


      if (msg.arg1 != RESULT_OK || pathname == null)
         activity.showDialog("failed download");

      ...
      activity.displayImage
        (BitmapFactory.decodeFile(pathname));
      ...
```

Extract the pathname from the Bundle

Programming Messenger in Download Application

- DownloadActivity receives Message via its Handler in the UI Thread

```
public class DownloadActivity extends Activity {
  ...
  private static class DownloadHandler extends Handler {
    public void handleMessage(Message msg) {
      ...

      Bundle data = msg.getData();

      String pathname = data.getString("PATHNAME");

                                  See if the download succeeded or not
      if (msg.arg1 != RESULT_OK || pathname == null)
         activity.showDialog("failed download");

      ...
      activity.displayImage
        (BitmapFactory.decodeFile(pathname));
      ...
```

Programming Messenger in Download Application

- DownloadActivity receives Message via its Handler in the UI Thread

```
public class DownloadActivity extends Activity {
  ...
  private static class DownloadHandler extends Handler {
    public void handleMessage(Message msg) {
      ...

      Bundle data = msg.getData();

      String pathname = data.getString("PATHNAME");


      if (msg.arg1 != RESULT_OK || pathname == null)
         activity.showDialog("failed download");

      ...
      activity.displayImage
        (BitmapFactory.decodeFile(pathname));
      ...
```

Decode & display the image in UI Thread

Programming Messenger in Download Application

- DownloadActivity receives Message via its Handler in the UI Thread

```
public class DownloadActivity extends Activity {
  ...
  private static class DownloadHandler extends Handler {
    public void handleMessage(Message msg) {
      ...

      Bundle data = msg.getData();

      String pathname = data.getString("PATHNAME");


      if (msg.arg1 != RESULT_OK || pathname == null)
         activity.showDialog("failed download");

      ...
      activity.displayImage
        (BitmapFactory.decodeFile(pathname));
      ...
```

Decode & display the image in UI Thread

Programming Messenger in Download Application

- DownloadActivity receives Message via its Handler in the UI Thread

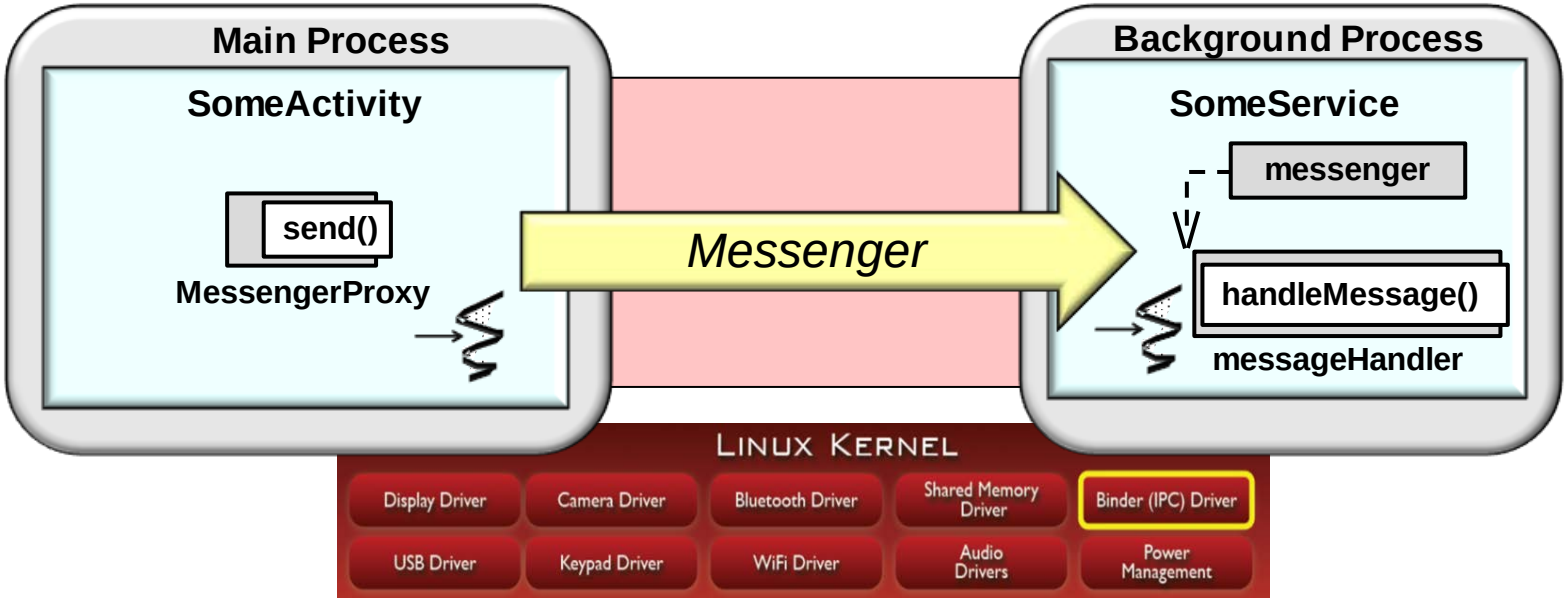
```
public class DownloadActivity extends Activity {
  ...
  private static class DownloadHandler extends Handler {
    public void handleMessage(Message msg) {
      ...

      Bundle data = msg.getData();

      String pathname = data.getString("PATHNAME");


      if (msg.arg1 != RESULT_OK || pathname == null)
         activity.showDialog("failed download");


      ...
      activity.displayImage
        (BitmapFactory.decodeFile(pathname));
      ...
```

Note how the DownloadActivity never blocks synchronously on any long-duration operations

Summary

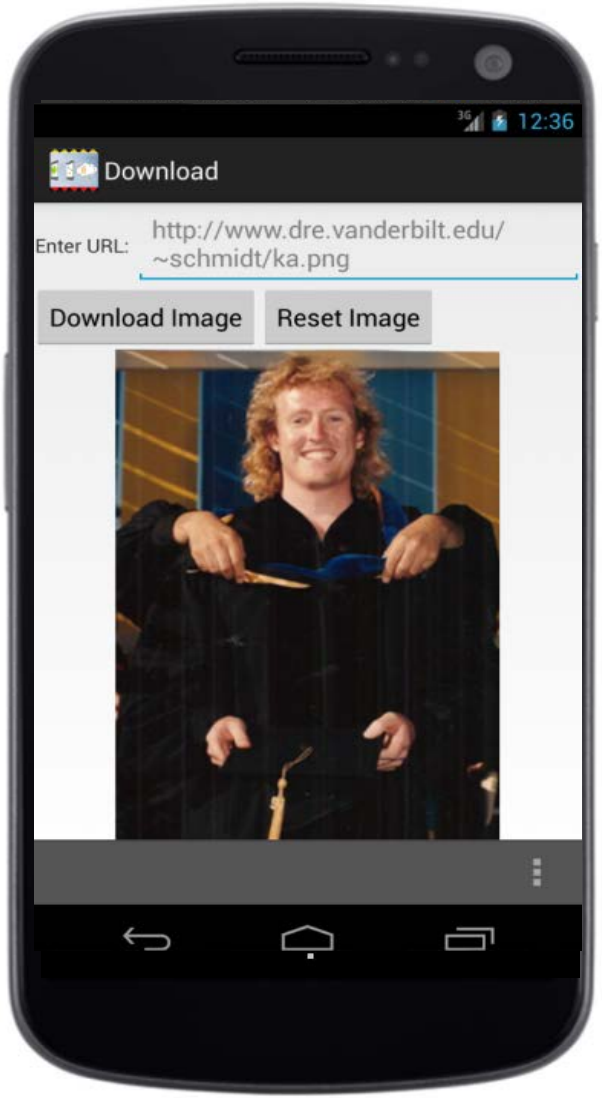
Summary

- Messengers provide a flexible framework for intra- & inter-process communication between Activities & Services in Android



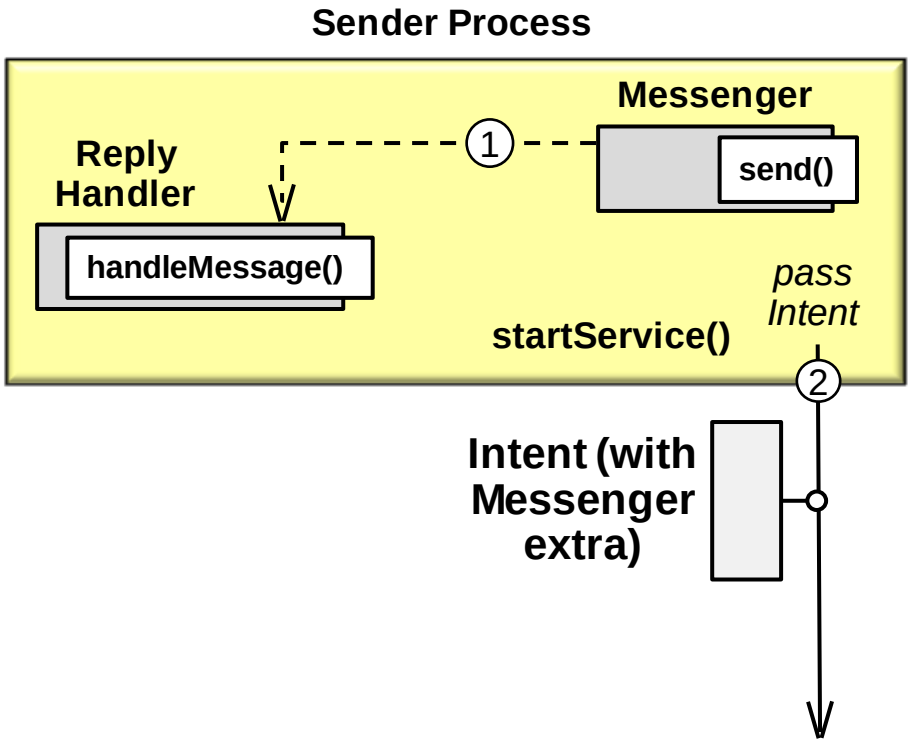
Summary

- Messengers provide a flexible framework for intra- & inter-process communication between Activities & Services in Android
  - We applied Messengers to the DownloadApplication



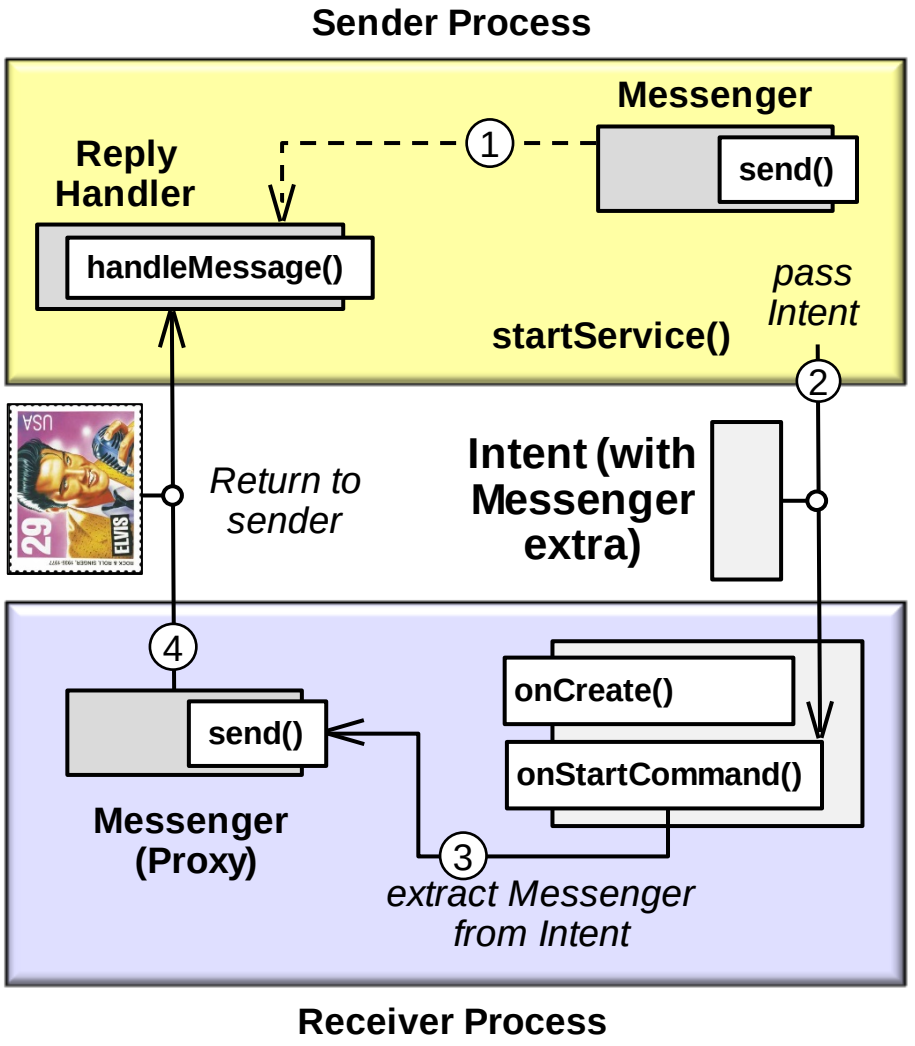
Summary

- Messengers provide a flexible framework for intra- & inter-process communication between Activities & Services in Android
  - We applied Messengers to the DownloadApplication
  - This application uses a common idiom in Android



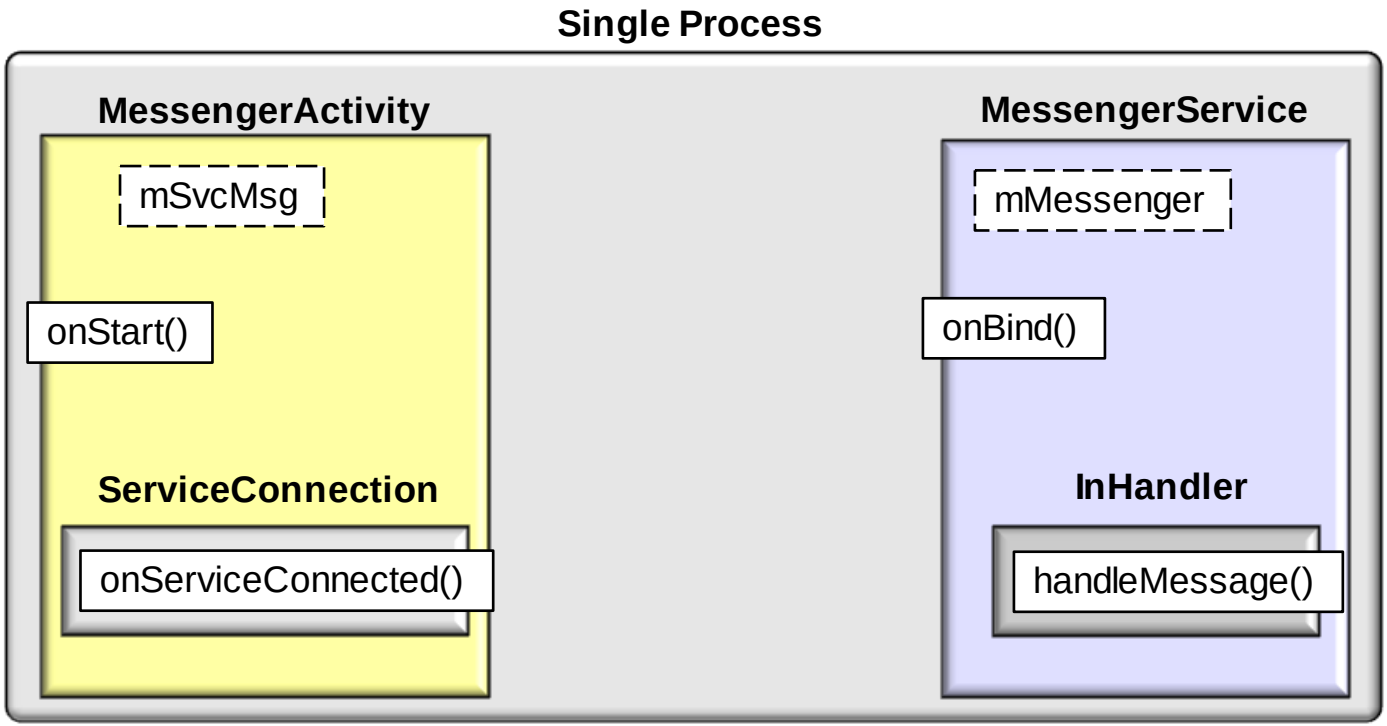
Summary

- Messengers provide a flexible framework for intra- & inter-process communication between Activities & Services in Android
  - We applied Messengers to the DownloadApplication
  - This application uses a common idiom in Android



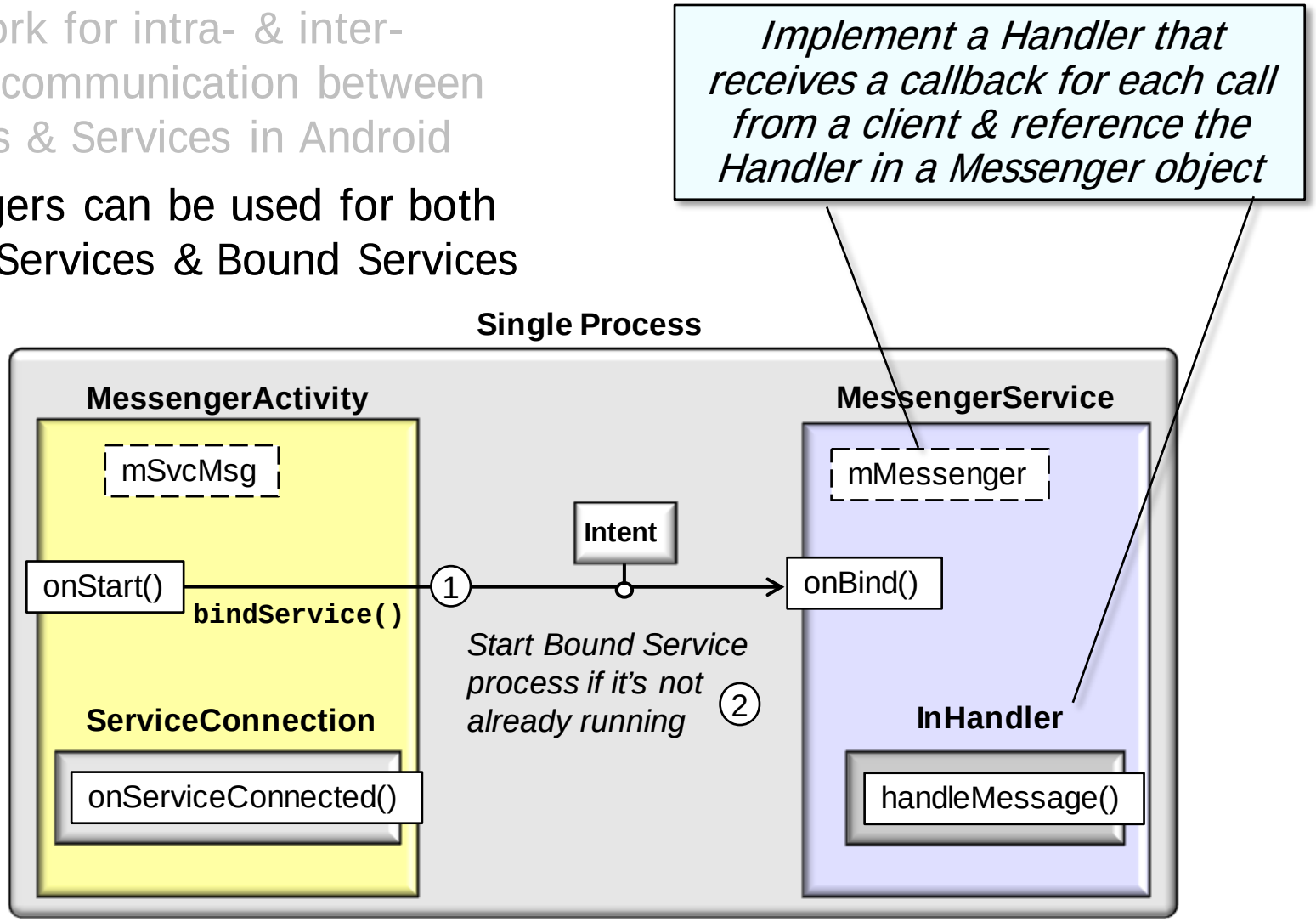
Summary

- Messengers provide a flexible framework for intra- & inter-process communication between Activities & Services in Android
- Messengers can be used for both Started Services & Bound Services



Summary

- Messengers provide a flexible framework for intra- & inter-process communication between Activities & Services in Android
- Messengers can be used for both Started Services & Bound Services

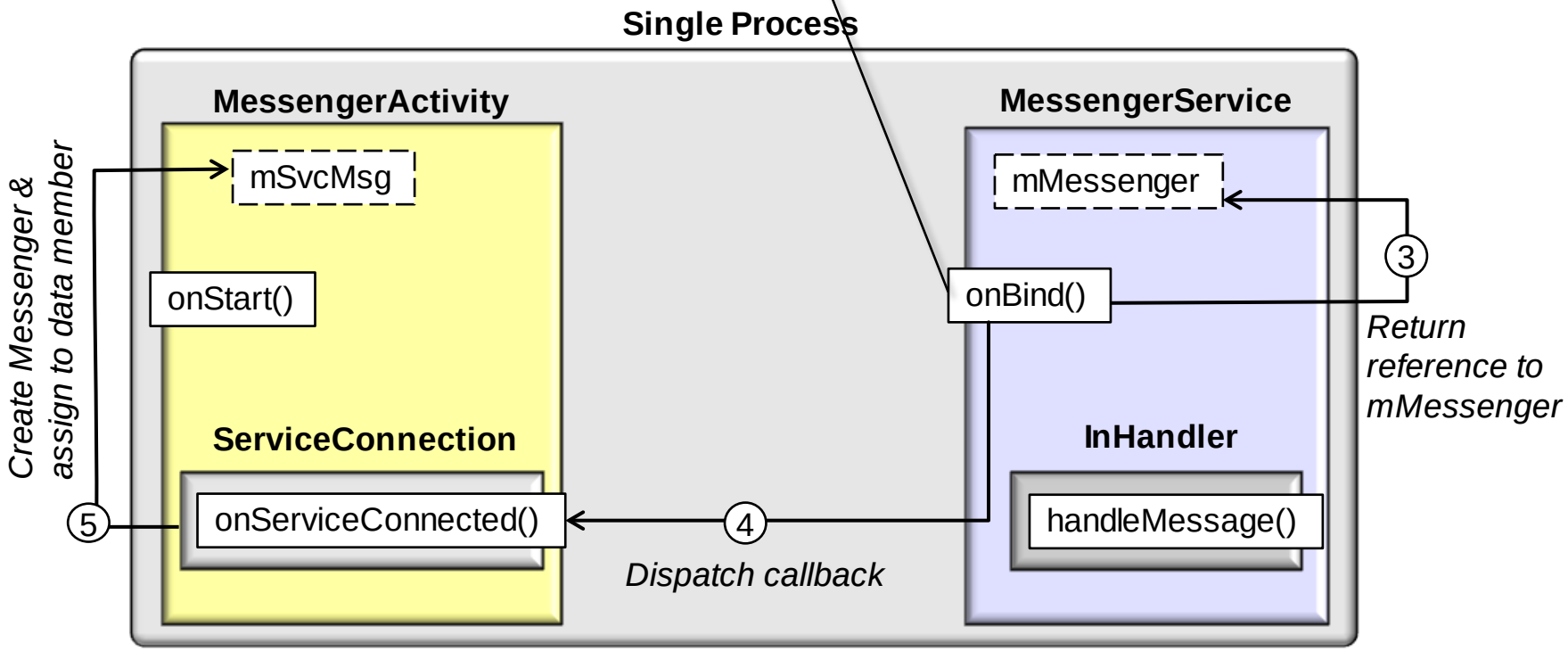


Summary

- Messengers provide a flexible framework for intra- & inter-process communication between Activities & Services in Android
- Messengers can be used for both Started Services & Bound Services

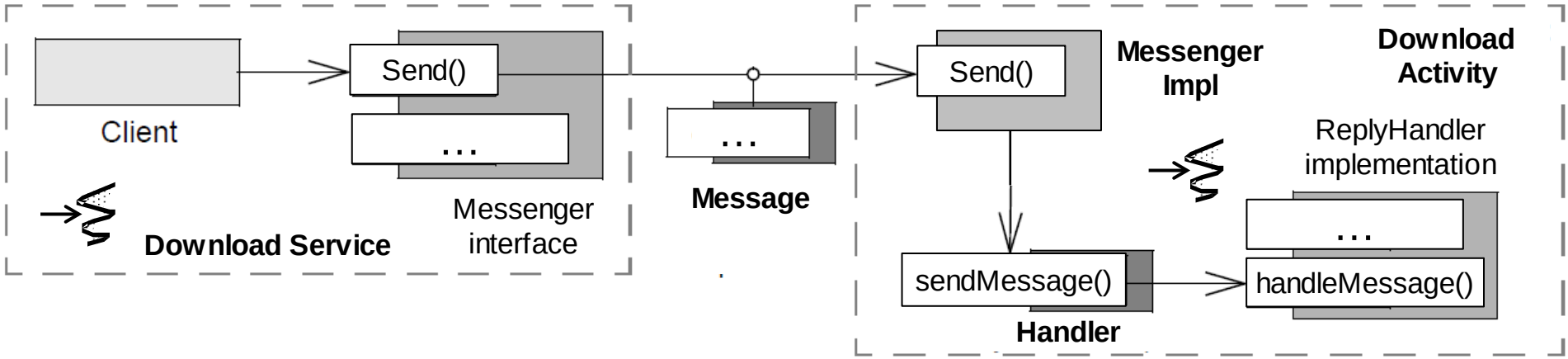
Messenger creates IBinder that Service returns to clients from onBind()

```
public IBinder onBind(Intent intent)
{ return mMessenger.getBinder(); }
```



Summary

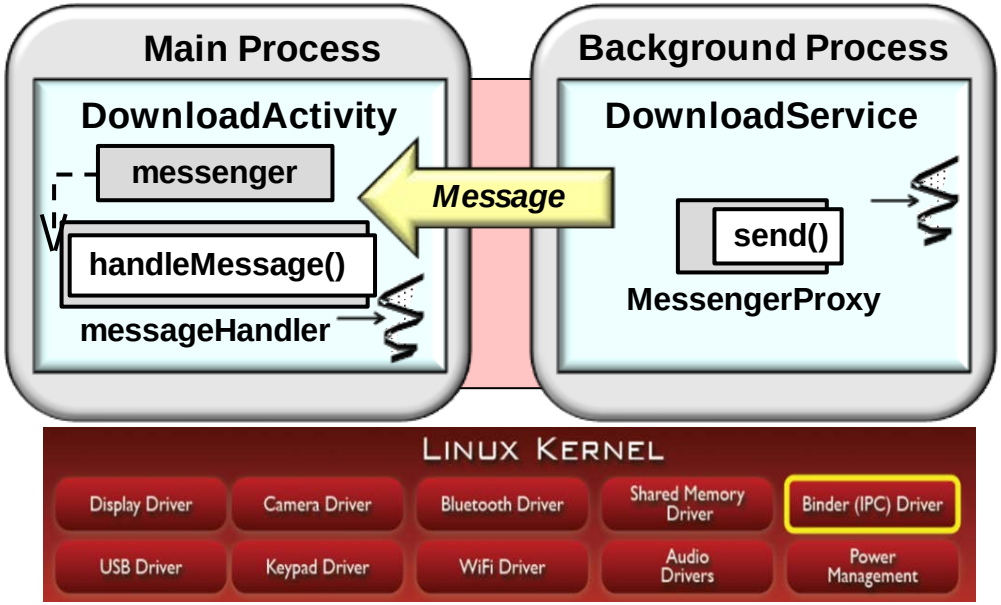
- Messengers provide a flexible framework for intra- & inter-process communication between Activities & Services in Android
- Messengers can be used for both Started Services & Bound Services
- Messenger-based programs apply the *Active Object* pattern



See upcoming parts on "The Active Object Pattern"

Summary

- Messengers provide a flexible framework for intra- & inter-process communication between Activities & Services in Android
- Messengers can be used for both Started Services & Bound Services
- Messenger-based programs apply the *Active Object* pattern
- Messengers are best suited for relative simple interactions & data types



Summary

- Messengers provide a flexible framework for intra- & inter-process communication between Activities & Services in Android
- Messengers can be used for both Started Services & Bound Services
- Messenger-based programs apply the *Active Object* pattern
- Messengers are best suited for relative simple interactions & data types
- More sophisticated interactions & complex data types may benefit from Android Interface Definition Language (AIDL)-based approaches