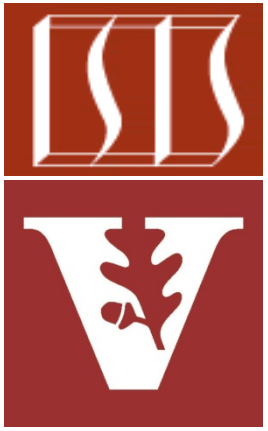
Android Services & Security: Programming Started Services with Intents & Messengers (Part 2)

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



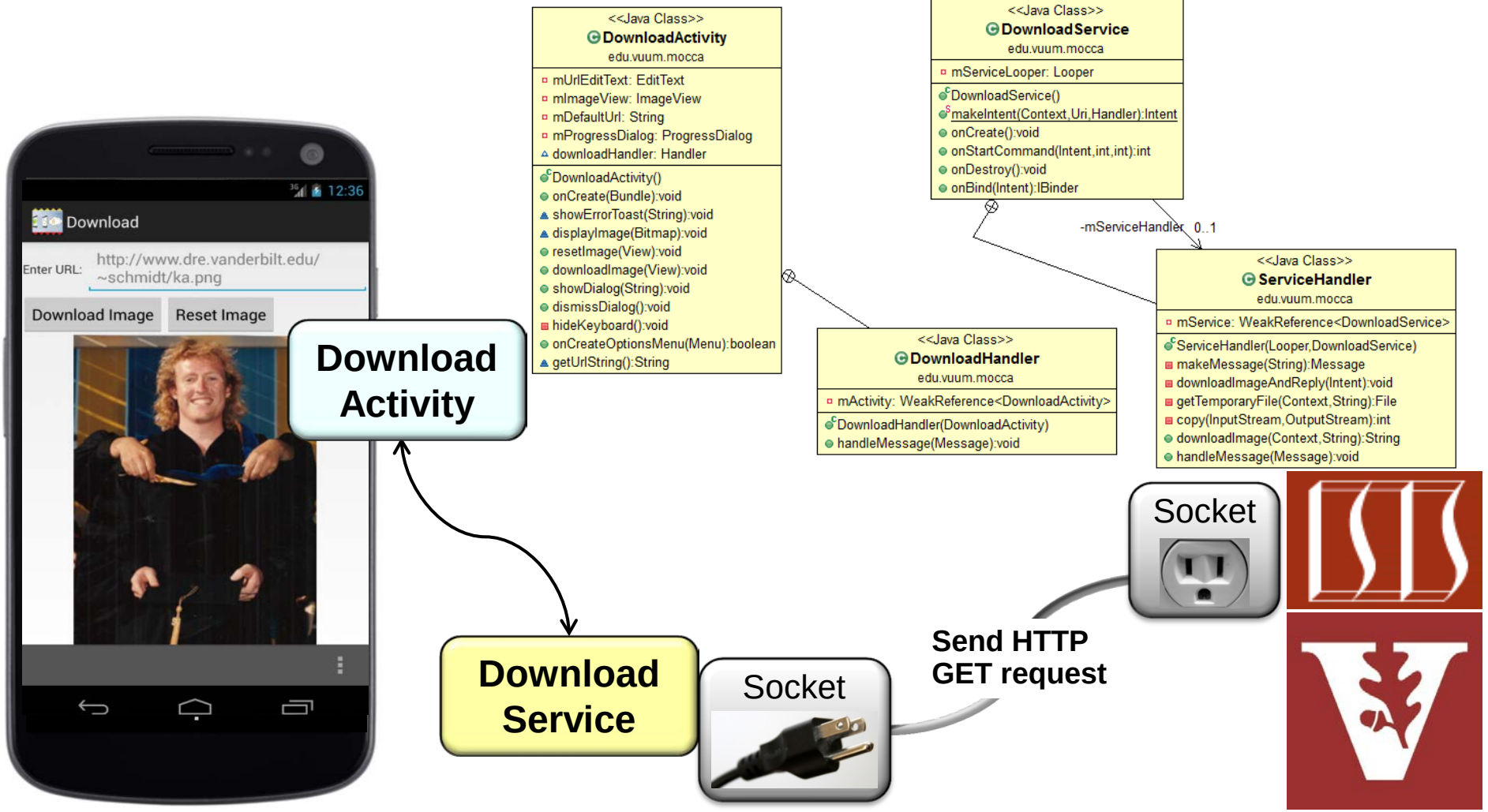
Professor of Computer Science

Institute for Software Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA

Learning Objectives in this Part of the Module

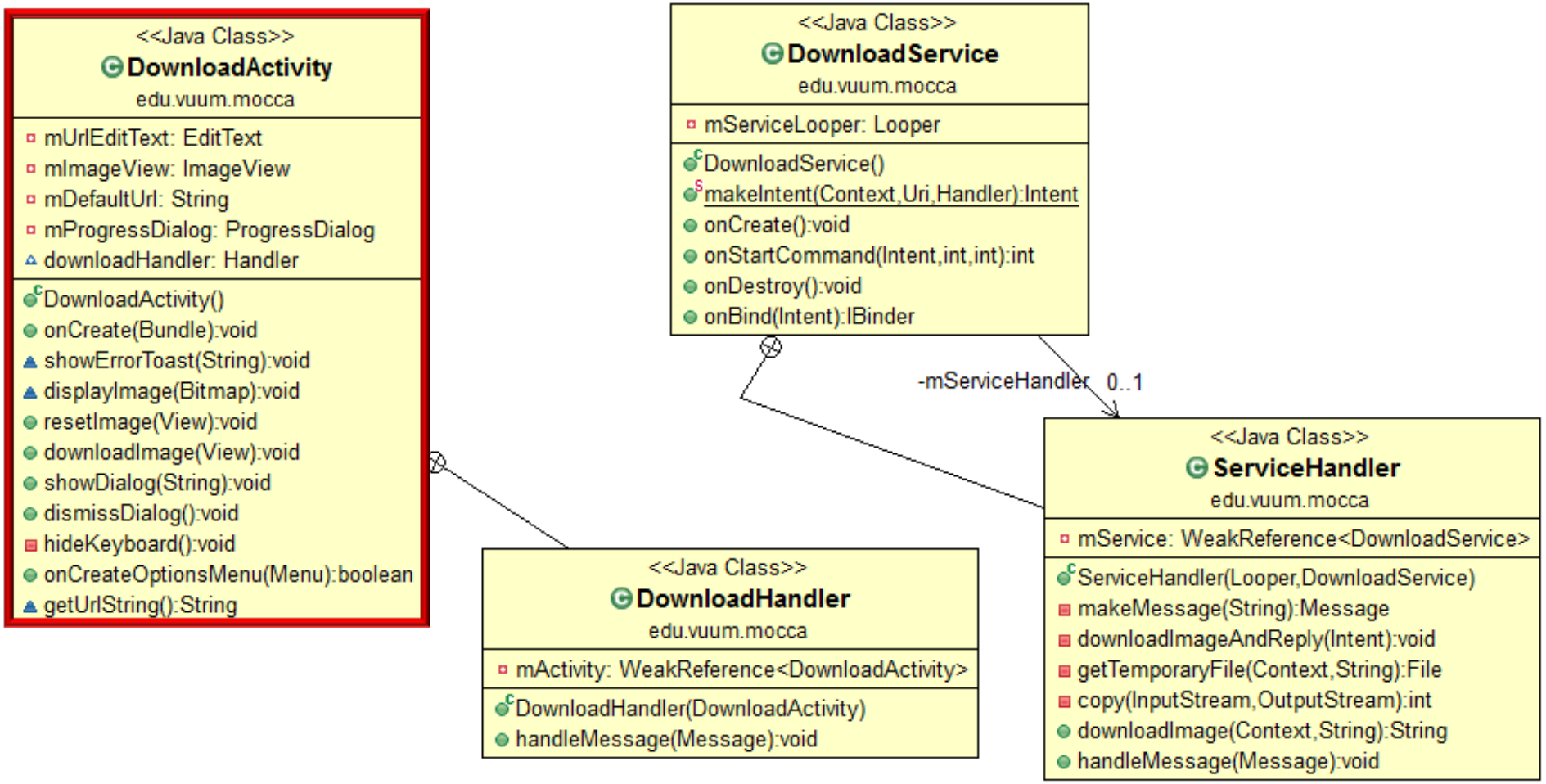
- Understand how to use a Started Service, Intent, & Messengers to implement a Download Application that retrieves/displays images from a remote server



Implementation of the DownloadActivity

Implementation of the DownloadActivity

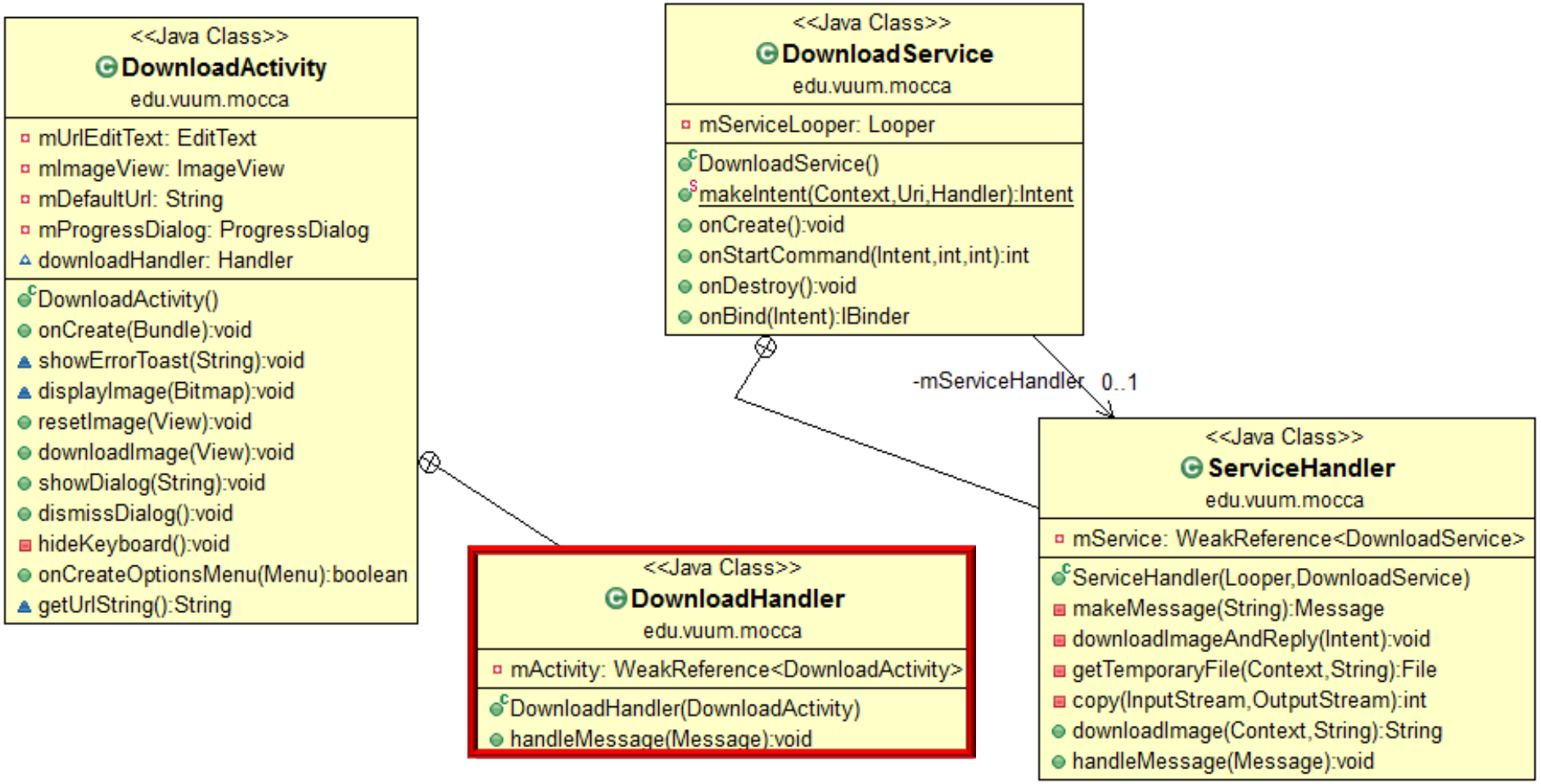
- DownloadActivity provides user-facing capabilities of DownloadApplication



Allows a user to retrieve & display an image

Implementation of the DownloadActivity

- DownloadActivity provides user-facing capabilities of DownloadApplication



Processes Messages sent to it from DownloadService

Implementation of the DownloadActivity

- DownloadActivity provides user-facing capabilities of DownloadApplication

```
public class DownloadActivity extends Activity {
   ...

   Handler mDownloadHandler = null;



   public void onCreate(Bundle savedInstanceState) {
      ...
      mDownloadHandler = new DownloadHandler(this);
   }
```

Activity that prompts user for image URL to download

Implementation of the DownloadActivity

- DownloadActivity provides user-facing capabilities of DownloadApplication

```
public class DownloadActivity extends Activity {
   ...

   Handler mDownloadHandler = null;



   public void onCreate(Bundle savedInstanceState) {
      ...
      mDownloadHandler = new DownloadHandler(this);
   }
```

Activity that prompts user for image URL to download

Implementation of the DownloadActivity

- DownloadActivity provides user-facing capabilities of DownloadApplication

```
public class DownloadActivity extends Activity {
   ...

   Handler mDownloadHandler = null;
```

Inherits from Handler & uses its handleMessage() hook method to process Messages sent from downloadService

```
   public void onCreate(Bundle savedInstanceState) {
      ...
      mDownloadHandler = new DownloadHandler(this);
   }
```

Implementation of the DownloadActivity

- DownloadActivity provides user-facing capabilities of DownloadApplication

```
public class DownloadActivity extends Activity {
   ...

   Handler mDownloadHandler = null;

                                  Hook method called to initialize DownloadActivity

   public void onCreate(Bundle savedInstanceState) {
      ...
      mDownloadHandler = new DownloadHandler(this);
   }
```

Implementation of the DownloadActivity

- DownloadActivity provides user-facing capabilities of DownloadApplication

```
public class DownloadActivity extends Activity {
   ...

   Handler mDownloadHandler = null;



   public void onCreate(Bundle savedInstanceState) {
      ...
      mDownloadHandler = new DownloadHandler(this);
   }
```

Initialize the DownloadHandler

Source code for displaying the downloaded image shown later

Implementation of the DownloadActivity

- DownloadActivity provides user-facing capabilities of DownloadApplication

```
public class DownloadActivity extends Activity {
  ...

  public void downloadImage(View v) {
    String url = getUrlString();
    ...
    Intent intent = DownloadService.makeIntent
                      (this,
                       Uri.parse(url),
                       downloadHandler);

    startService(intent);
  }
```

Dispatched when user clicks “Download Image” button

Implementation of the DownloadActivity

- DownloadActivity provides user-facing capabilities of DownloadApplication

```
public class DownloadActivity extends Activity {
  ...

  public void downloadImage(View v) {
    String url = getUrlString();
    ...
    Intent intent = DownloadService.makeIntent
                      (this,
                       Uri.parse(url),
                       downloadHandler);

    startService(intent);
  }
```

Get URL entered by user

Implementation of the DownloadActivity

- DownloadActivity provides user-facing capabilities of DownloadApplication

```
public class DownloadActivity extends Activity {
  ...

  public void downloadImage(View v) {
    String url = getUrlString();
    ...
    Intent intent = DownloadService.makeIntent
                      (this,
                       Uri.parse(url),
                       downloadHandler);

    startService(intent);
  }
```

Create Intent associated with DownloadService

Implementation of the DownloadActivity

- DownloadActivity provides user-facing capabilities of DownloadApplication

```
public class DownloadActivity extends Activity {
  ...

  public void downloadImage(View v) {
    String url = getUrlString();
    ...
    Intent intent = DownloadService.makeIntent
                      (this,
                       Uri.parse(url),
                       downloadHandler);

    startService(intent);
  }
```

Create Intent associated with DownloadService

Implementation of the DownloadActivity

- DownloadActivity provides user-facing capabilities of DownloadApplication

```
public class DownloadActivity extends Activity {
  ...

  public void downloadImage(View v) {
    String url = getUrlString();
    ...
    Intent intent = DownloadService.makeIntent
                      (this,
                       Uri.parse(url),
                       downloadHandler);

    startService(intent);
  }
```

Create Intent associated with DownloadService

Implementation of the DownloadActivity

- DownloadActivity provides user-facing capabilities of DownloadApplication

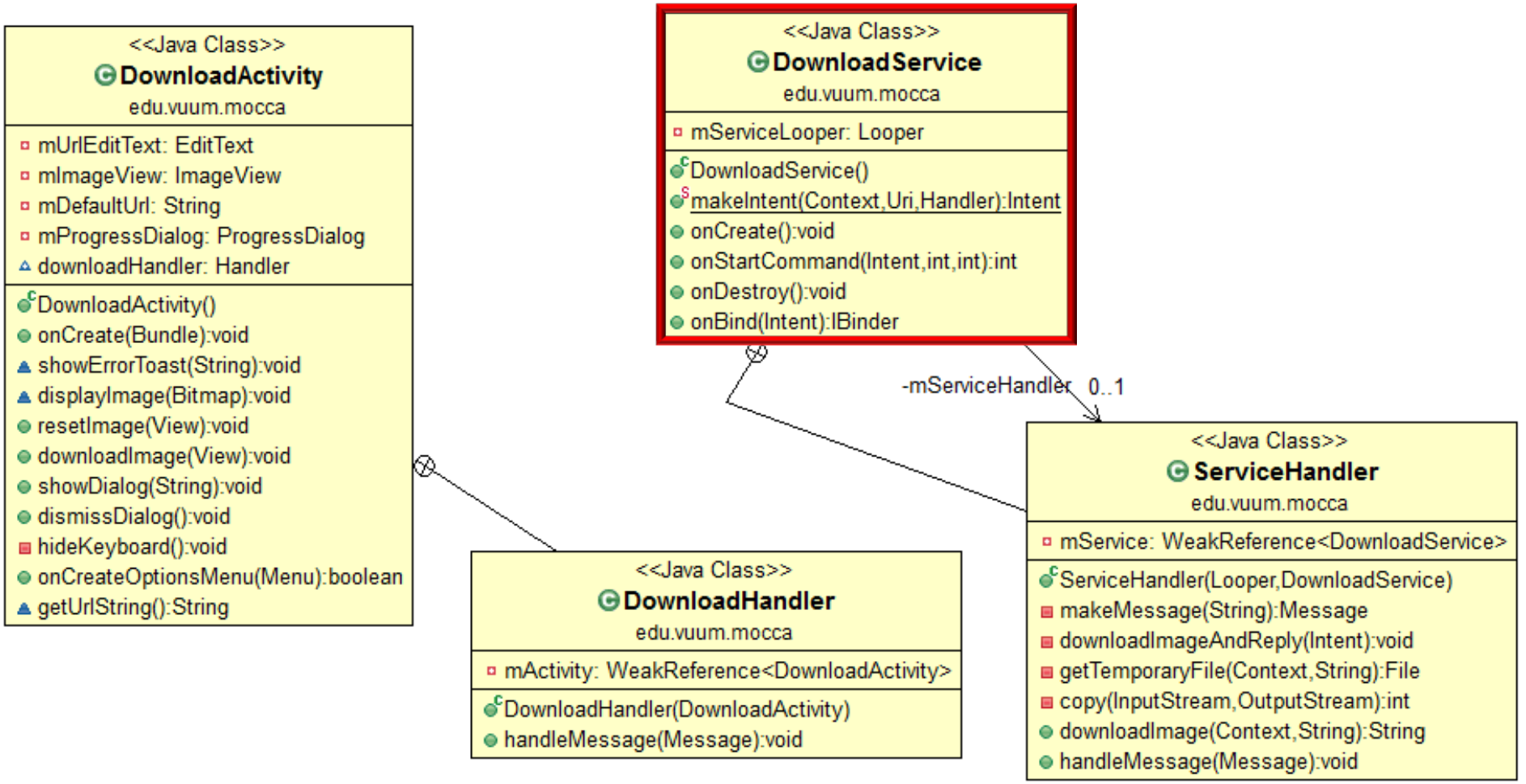
```
public class DownloadActivity extends Activity {
  ...

  public void downloadImage(View v) {
    String url = getUrlString();
    ...
    Intent intent = DownloadService.makeIntent
                      (this,
                       Uri.parse(url),
                       downloadHandler);

    startService(intent);
  }
```

Launch the DownloadService to handle this Intent

Implementation of the DownloadService

Implementation of DownloadService

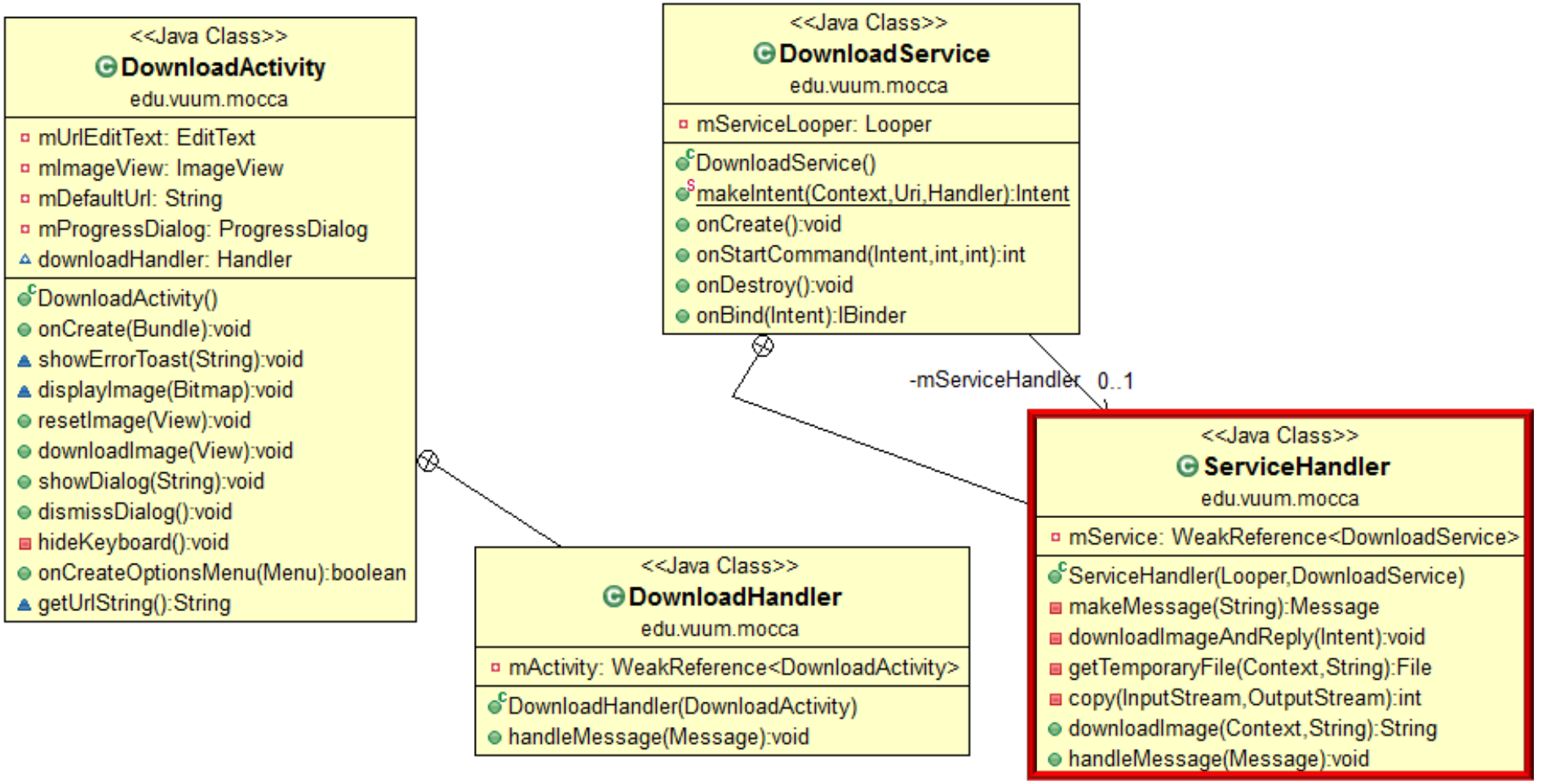
- DownloadActivity provides user-facing capabilities of DownloadApplication



Downloads & stores a bitmap image on behalf of the DownloadActivity

Implementation of DownloadService

- DownloadActivity provides user-facing capabilities of DownloadApplication



Concurrently downloads an image from URL provided via an Intent, stores it on the file system, & returns pathname to DownloadActivity

Initializing the DownloadService

Initializing the DownloadService

- DownloadService runs in background & retrieves images from remote server

```
public class DownloadService extends Service {
   ...
```

DownloadService overrides Service hook methods

Initializing the DownloadService

- DownloadService runs in background & retrieves images from remote server

```
public class DownloadService extends Service {
   ...
```

DownloadService overrides Service hook methods

Initializing the DownloadService

- DownloadService runs in background & retrieves images from remote server

```
public class DownloadService extends Service {
```

Factory method called by DownloadActivity to make the right type of Intent

```
  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;
  }
  ...
```

This factory method shields clients from details of creating an Intent

Initializing the DownloadService

- DownloadService runs in background & retrieves images from remote server

```
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;
  }
  ...
```

Create an Intent that's associated with the DownloadService class

Initializing the DownloadService

- DownloadService runs in background & retrieves images from remote server

```
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);
```

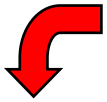
Add the URI to the download as data

```
    intent.putExtra(MESSENGER,
                    new Messenger(downloadHandler));

    return intent;
  }
  ...
```

Initializing the DownloadService

- DownloadService runs in background & retrieves images from remote server

```
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;
  }
  ...
```



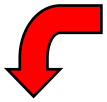
Add Messenger that encapsulates DownloadHandler as an extra

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

Initializing the DownloadService

- DownloadService runs in background & retrieves images from remote server

```
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;
  }
  ...
```



Add Messenger that encapsulates DownloadHandler as an extra

See earlier part on “Activity & Service Communication”

Initializing the DownloadService

- DownloadService runs in background & retrieves images from remote server

```
public class DownloadService extends Service {
  ...
  private volatile Looper mServiceLooper;
  private volatile ServiceHandler mServiceHandler;

  public void onCreate() {
    super.onCreate();



    HandlerThread thread = new HandlerThread("DownloadService");
    thread.start();



    mServiceLooper = thread.getLooper();
    mServiceHandler = new ServiceHandler(mServiceLooper);
  }
```

Useful data members

Initializing the DownloadService

- DownloadService runs in background & retrieves images from remote server

```
public class DownloadService extends Service {
  ...
  private volatile Looper mServiceLooper;
  private volatile ServiceHandler mServiceHandler;

  public void onCreate() {
    super.onCreate();



    HandlerThread thread = new HandlerThread("DownloadService");
    thread.start();



    mServiceLooper = thread.getLooper();
    mServiceHandler = new ServiceHandler(mServiceLooper);
  }
```

Useful data members

See en.wikipedia.org/wiki/Volatile_variable#In_Java

Initializing the DownloadService

- DownloadService runs in background & retrieves images from remote server

```
public class DownloadService extends Service {
  ...
  private volatile Looper mServiceLooper;
  private volatile ServiceHandler mServiceHandler;

  public void onCreate() {
    super.onCreate();

    HandlerThread thread = new HandlerThread("DownloadService");
    thread.start();


    mServiceLooper = thread.getLooper();
    mServiceHandler = new ServiceHandler(mServiceLooper);
  }
```

Called when DownloadService is launched by the Android Activity Manager Service

By default, a Service runs in the UI Thread, which should not block..

Initializing the DownloadService

- DownloadService runs in background & retrieves images from remote server

```
public class DownloadService extends Service {
  ...
  private volatile Looper mServiceLooper;
  private volatile ServiceHandler mServiceHandler;

  public void onCreate() {
    super.onCreate();
```

Create a HandlerThread since we don't know whether this Service has been configured to run in its own process

```
    HandlerThread thread = new HandlerThread("DownloadService");
    thread.start();



    mServiceLooper = thread.getLooper();
    mServiceHandler = new ServiceHandler(mServiceLooper);
  }
```

See earlier part on "Android Looper"

Initializing the DownloadService

- DownloadService runs in background & retrieves images from remote server

```
public class DownloadService extends Service {
  ...
  private volatile Looper mServiceLooper;
  private volatile ServiceHandler mServiceHandler;

  public void onCreate() {
    super.onCreate();



    HandlerThread thread = new HandlerThread("DownloadService");
    thread.start();

    mServiceLooper = thread.getLooper();
    mServiceHandler = new ServiceHandler(mServiceLooper);
  }
```

Start the HandlerThread & use it to retrieve the image concurrently in the background

Initializing the DownloadService

- DownloadService runs in background & retrieves images from remote server

```
public class DownloadService extends Service {
  ...
  private volatile Looper mServiceLooper;
  private volatile ServiceHandler mServiceHandler;

  public void onCreate() {
    super.onCreate();



    HandlerThread thread = new HandlerThread("DownloadService");
    thread.start();

                Get the HandlerThread's Looper & use it for our Handler

    mServiceLooper = thread.getLooper();
    mServiceHandler = new ServiceHandler(mServiceLooper);
  }
```

Initializing the DownloadService

- DownloadService runs in background & retrieves images from remote server

```
public class DownloadService extends Service {
  ...
  private volatile Looper mServiceLooper;
  private volatile ServiceHandler mServiceHandler;

  public void onCreate() {
    super.onCreate();



    HandlerThread thread = new HandlerThread("DownloadService");
    thread.start();

                Get the HandlerThread's Looper & use it for our Handler

    mServiceLooper = thread.getLooper();
    mServiceHandler = new ServiceHandler(mServiceLooper);
  }
```

ServiceHandler.handleMessage() will be dispatched in the HandlerThread

Initializing the DownloadService

- DownloadService runs in background & retrieves images from remote server

```
public class DownloadService extends Service {
  ...
  private volatile Looper mServiceLooper;
  private volatile ServiceHandler mServiceHandler;

  public void onCreate() {
    super.onCreate();



    HandlerThread thread = new HandlerThread("DownloadService");
    thread.start();

                Get the HandlerThread's Looper & use it for our Handler

    mServiceLooper = thread.getLooper();
    mServiceHandler = new ServiceHandler(mServiceLooper);
  }
```

See upcoming part on "Android IntentService"

Processing Intents in DownloadService

Processing Intents in DownloadService

- DownloadService runs in background & retrieves images from remote server

```
public class DownloadService extends Service {
  ...
  public int onStartCommand(Intent intent, int f, int startId) {
    Message message =
      mServiceHandler.makeDownloadMessage(intent, startId);
    mServiceHandler.sendMessage(message);
    return START_NOT_STICKY;
  }
  ...
```

Called for each Intent send via startService()

Processing Intents in DownloadService

- DownloadService runs in background & retrieves images from remote server

```
public class DownloadService extends Service {
  ...
  public int onStartCommand(Intent intent, int f, int startId) {


    Message message =
      mServiceHandler.makeDownloadMessage(intent, startId);
    mServiceHandler.sendMessage(message);
    return START_NOT_STICKY;
  }
  ...
```

Create a Message to sent to ServiceHandler to retrieve an image based on URI in the Intent

Processing Intents in DownloadService

- DownloadService runs in background & retrieves images from remote server

```
public class DownloadService extends Service {
  ...
  public int onStartCommand(Intent intent, int f, int startId) {


    Message message =
      mServiceHandler.makeDownloadMessage(intent, startId);
    mServiceHandler.sendMessage(message);
    return START_NOT_STICKY;
  }

  private Message makeDownloadMessage(Intent intent,
                                      int startId) {
    Message message = Message.obtain();
    message.obj = intent;
    message.arg1 = startId;
    return message;
  }
```

Include Intent & start ID to indicate which URI to retrieve & which request is being stopped when download completes

Processing Intents in DownloadService

- DownloadService runs in background & retrieves images from remote server

```
public class DownloadService extends Service {
  ...
  public int onStartCommand(Intent intent, int f, int startId) {


    Message message =
      mServiceHandler.makeDownloadMessage(intent, startId);
    mServiceHandler.sendMessage(message);
    return START_NOT_STICKY;
  }

  private Message makeDownloadMessage(Intent intent,
                                      int startId){
    Message message = Message.obtain();
    message.obj = intent;
    message.arg1 = startId;
    return message;
  }
```

Include Intent & start ID to indicate which URI to retrieve & which request is being stopped when download completes

Processing Intents in DownloadService

- DownloadService runs in background & retrieves images from remote server

```
public class DownloadService extends Service {
  ...
  public int onStartCommand(Intent intent, int f, int startId) {


    Message message =
      mServiceHandler.makeDownloadMessage(intent, startId);
    mServiceHandler.sendMessage(message);
    return START_NOT_STICKY;
  }

  private Message makeDownloadMessage(Intent intent,
                                      int startId){
    Message message = Message.obtain();
    message.obj = intent;
    message.arg1 = startId;
    return message;
  }
```

Include Intent & start ID to indicate which URI to retrieve & which request is being stopped when download completes

The Service is only stopped when startId matches the last start request

Processing Intents in DownloadService

- DownloadService runs in background & retrieves images from remote server

```
public class DownloadService extends Service {
  ...
  public int onStartCommand(Intent intent, int f, int startId) {


    Message message =
      mServiceHandler.makeDownloadMessage(intent, startId);
    mServiceHandler.sendMessage(message);
    return START_NOT_STICKY;
  }
  ...
```

Send Message to ServiceHandler to retrieve an image based on contents of the Intent

Processing Intents in DownloadService

- DownloadService runs in background & retrieves images from remote server

```
public class DownloadService extends Service {
  ...
  public int onStartCommand(Intent intent, int f, int startId) {


    Message message =
      mServiceHandler.makeDownloadMessage(intent, startId);
    mServiceHandler.sendMessage(message);
    return START_NOT_STICKY;
  }
  ...
```

Don’t restart the DownloadService automatically if its process is killed while it’s running

Processing Intents in DownloadService

- DownloadService runs in background & retrieves images from remote server

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

    public ServiceHandler(Looper looper) {
      super(looper);
      ...


    public void handleMessage(Message message) {
      downloadImageAndReply((Intent) message.obj);

      stopSelf(msg.arg1);
    }
    ...
```

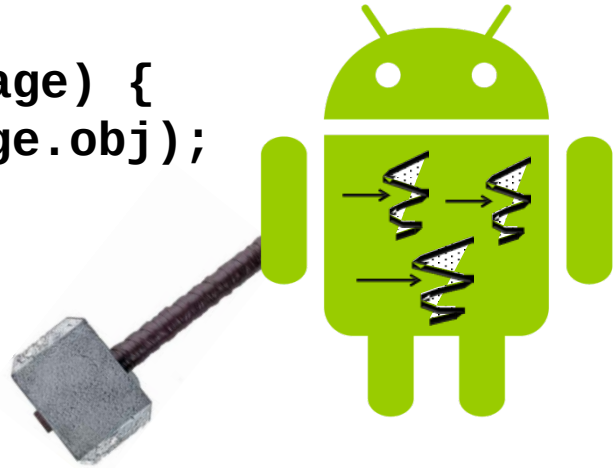
Concurrently processes Messages sent by onStartCommand()

Processing Intents in DownloadService

- DownloadService runs in background & retrieves images from remote server

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

    public ServiceHandler(Looper looper) {
      super(looper);
      ...


    public void handleMessage(Message message) {
      downloadImageAndReply((Intent) message.obj);

      stopSelf(msg.arg1);
    }
    ...
```

Concurrently processes Messages sent by onStartCommand()



Processing Intents in DownloadService

- DownloadService runs in background & retrieves images from remote server

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

    public ServiceHandler(Looper looper) {
      super(looper);
      ...


    public void handleMessage(Message message) {
      downloadImageAndReply((Intent) message.obj);

      stopSelf(msg.arg1);
    }
    ...
```

Constructor makes its Looper the HandlerThread's Looper

Processing Intents in DownloadService

- DownloadService runs in background & retrieves images from remote server

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


    public ServiceHandler(Looper looper) {
      super(looper);
      ...

    public void handleMessage(Message message) {
      downloadImageAndReply((Intent) message.obj);

      stopSelf(msg.arg1);
    }
    ...
```

Hook method that retrieves an image from a remote server & replies to client

Processing Intents in DownloadService

- DownloadService runs in background & retrieves images from remote server

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


    public ServiceHandler(Looper looper) {
      super(looper);
      ...

    public void handleMessage(Message message) {
      downloadImageAndReply((Intent) message.obj);

      stopSelf(msg.arg1);
    }
    ...
```

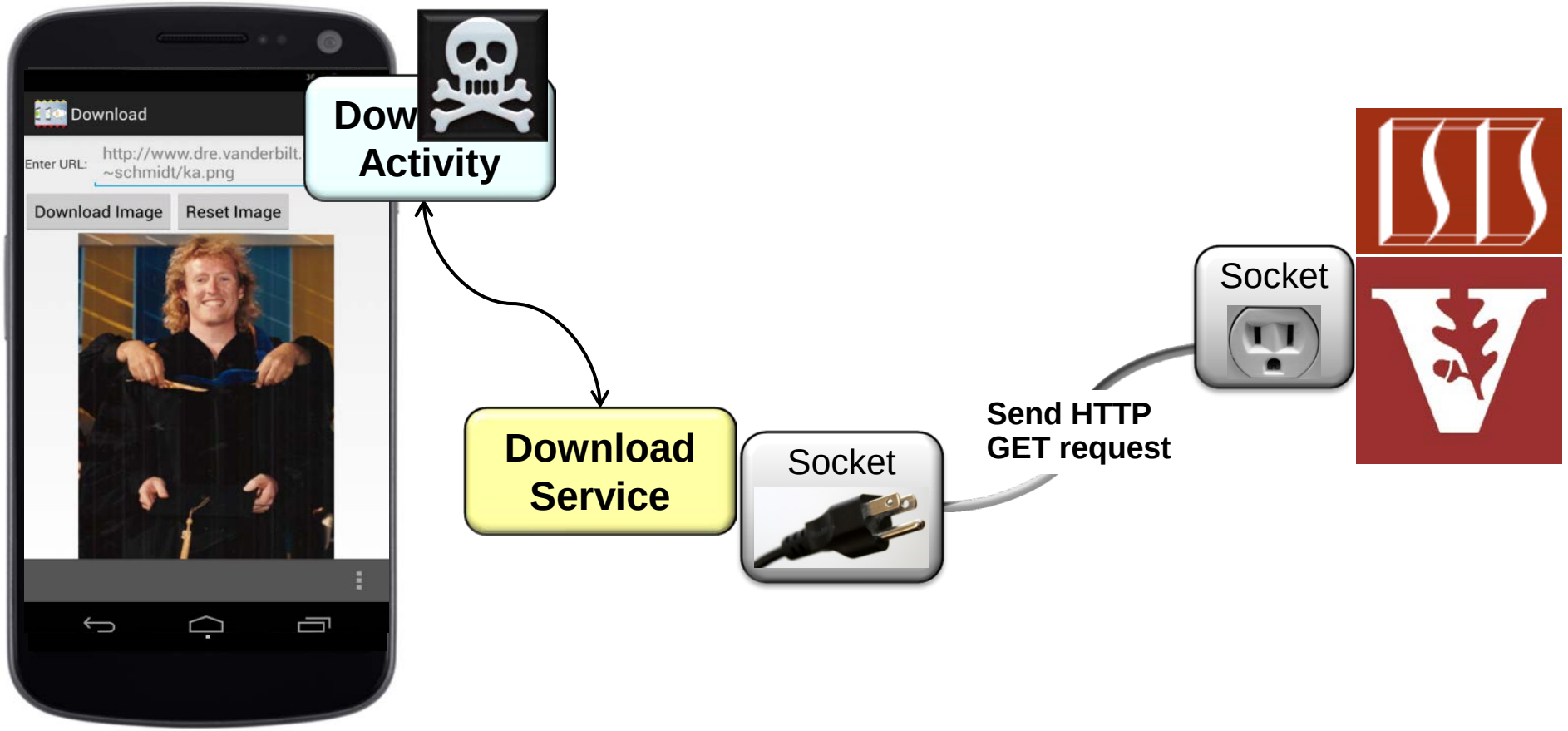
Hook method that retrieves an image from a remote server & replies to client

See earlier part on “Service & Activity Communication via Android Messenger”

Stopping the DownloadService

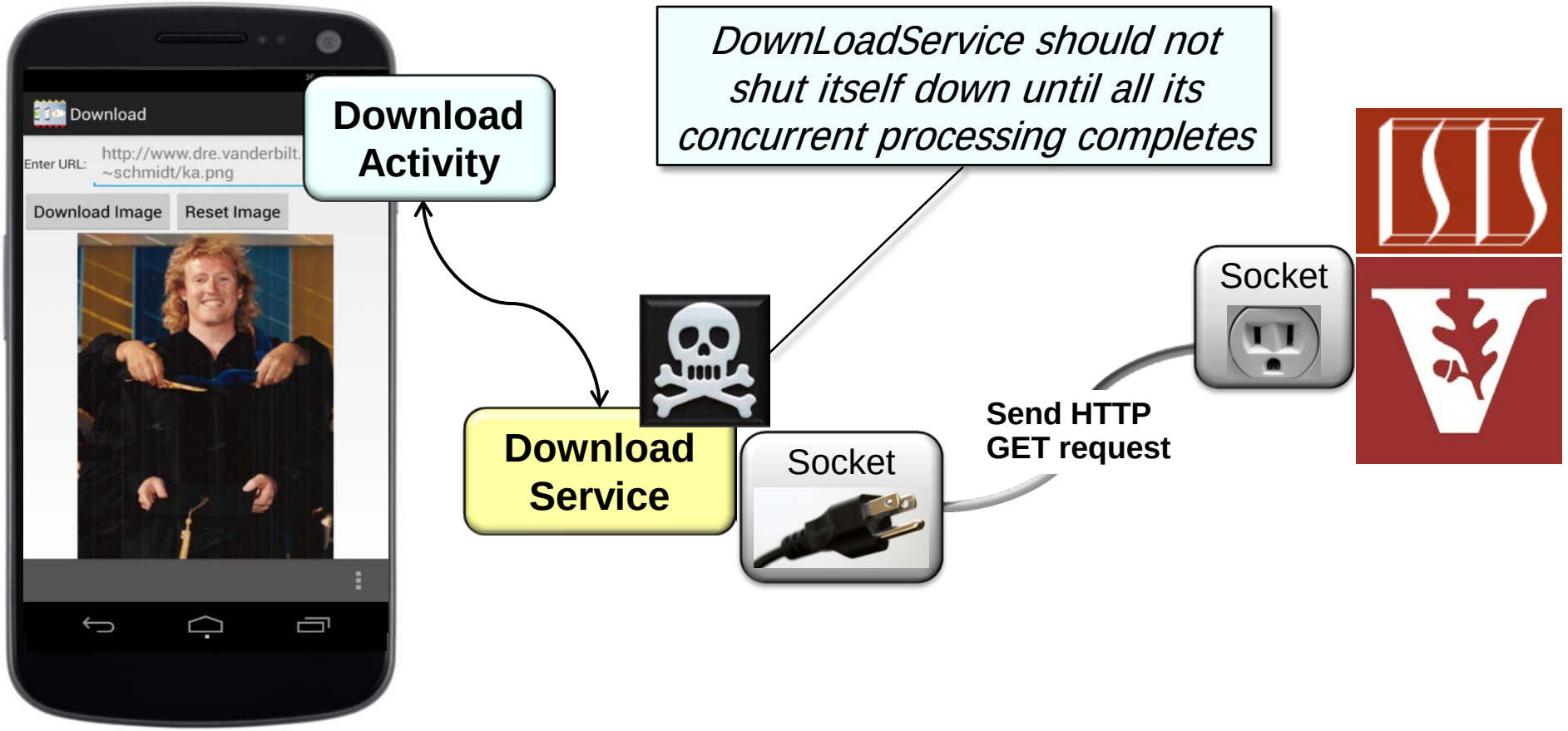
Stopping the DownloadService

- The lifecycle of a Service is independent of the component that launched it



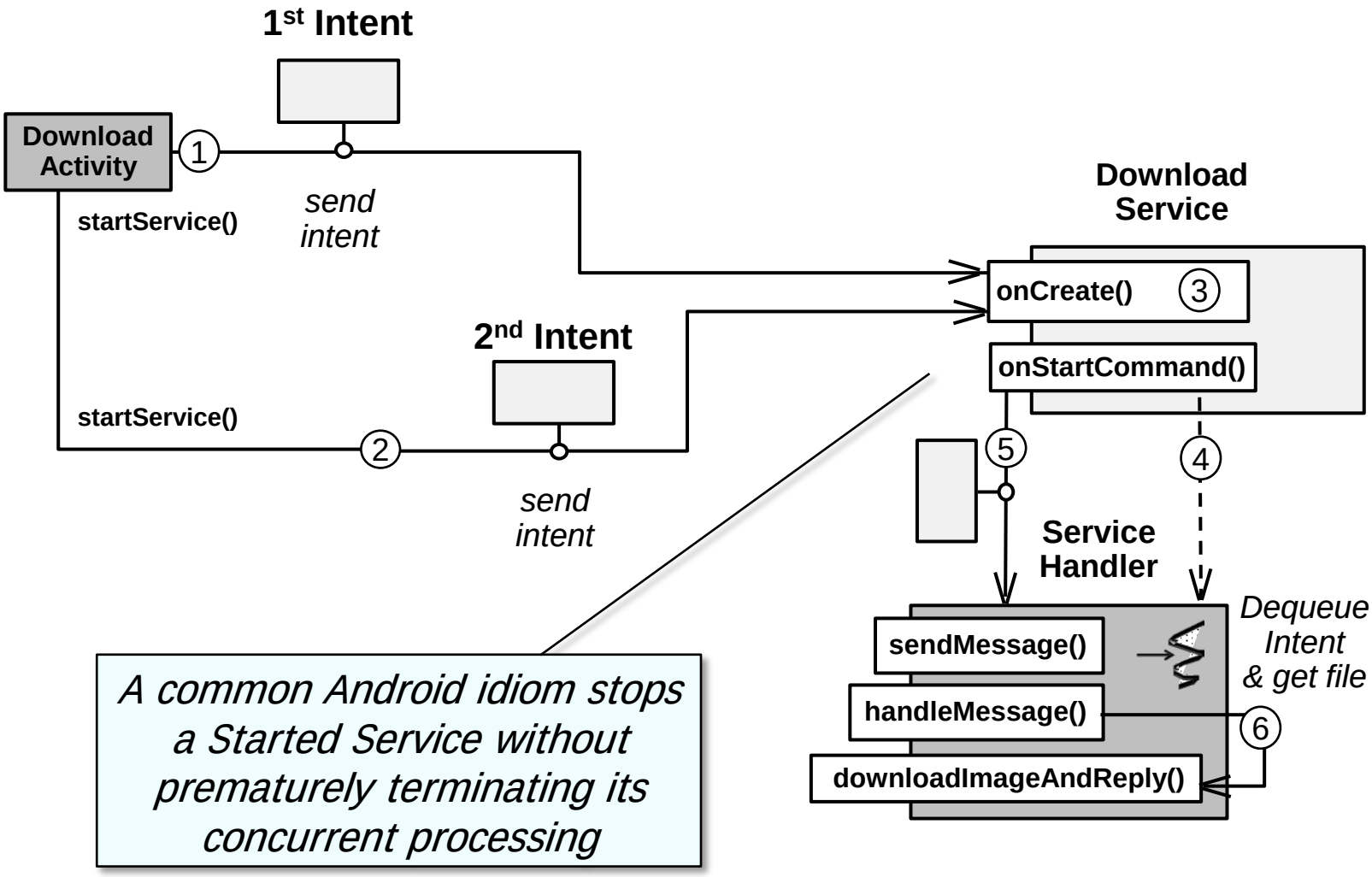
Stopping the DownloadService

- The lifecycle of a Service is independent of the component that launched it
  - The Download Service must therefore shut itself down or it will run forever!



Stopping the DownloadService

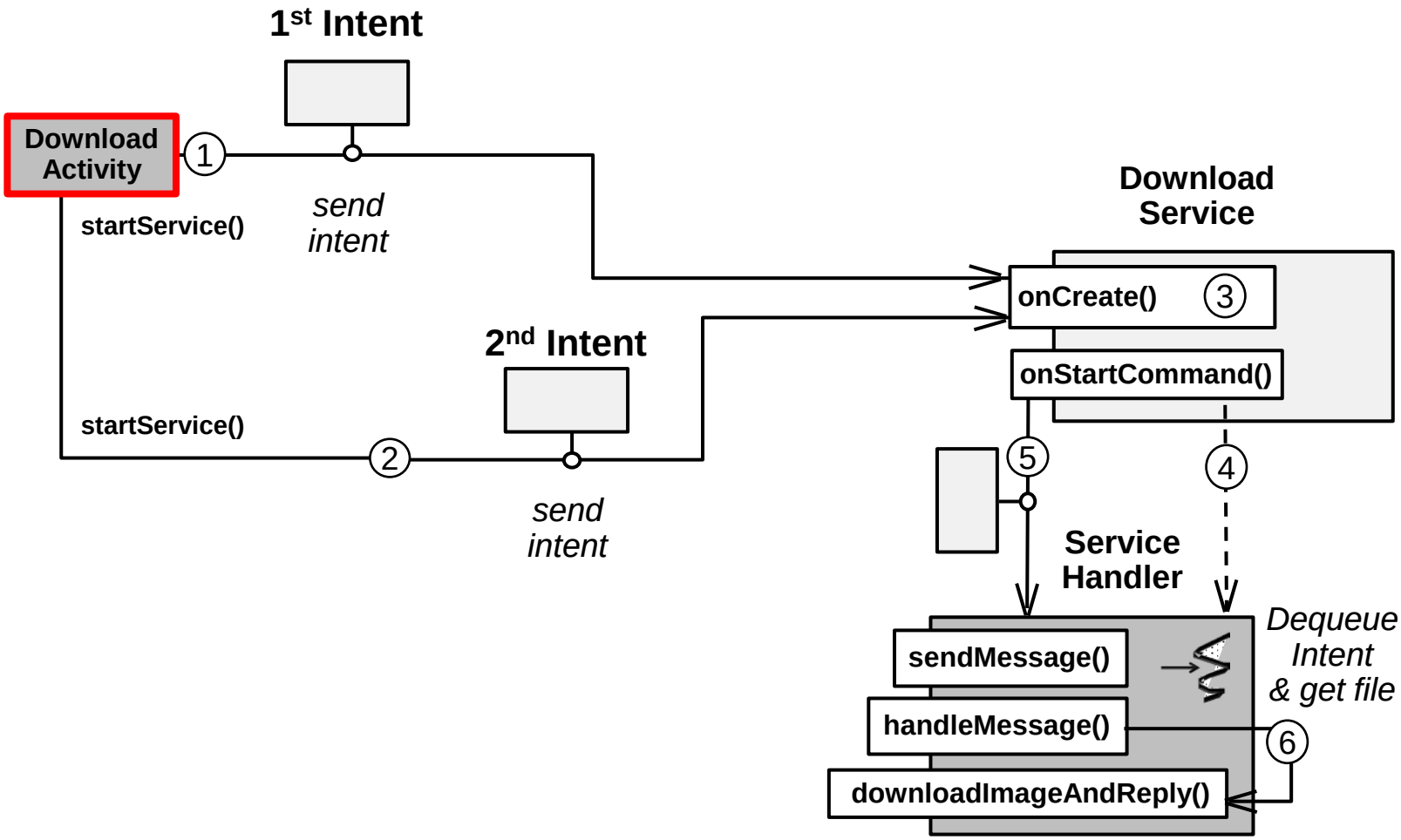
- The lifecycle of a Service is independent of the component that launched it



See developer.android.com/guide/components/services.html#Stopping

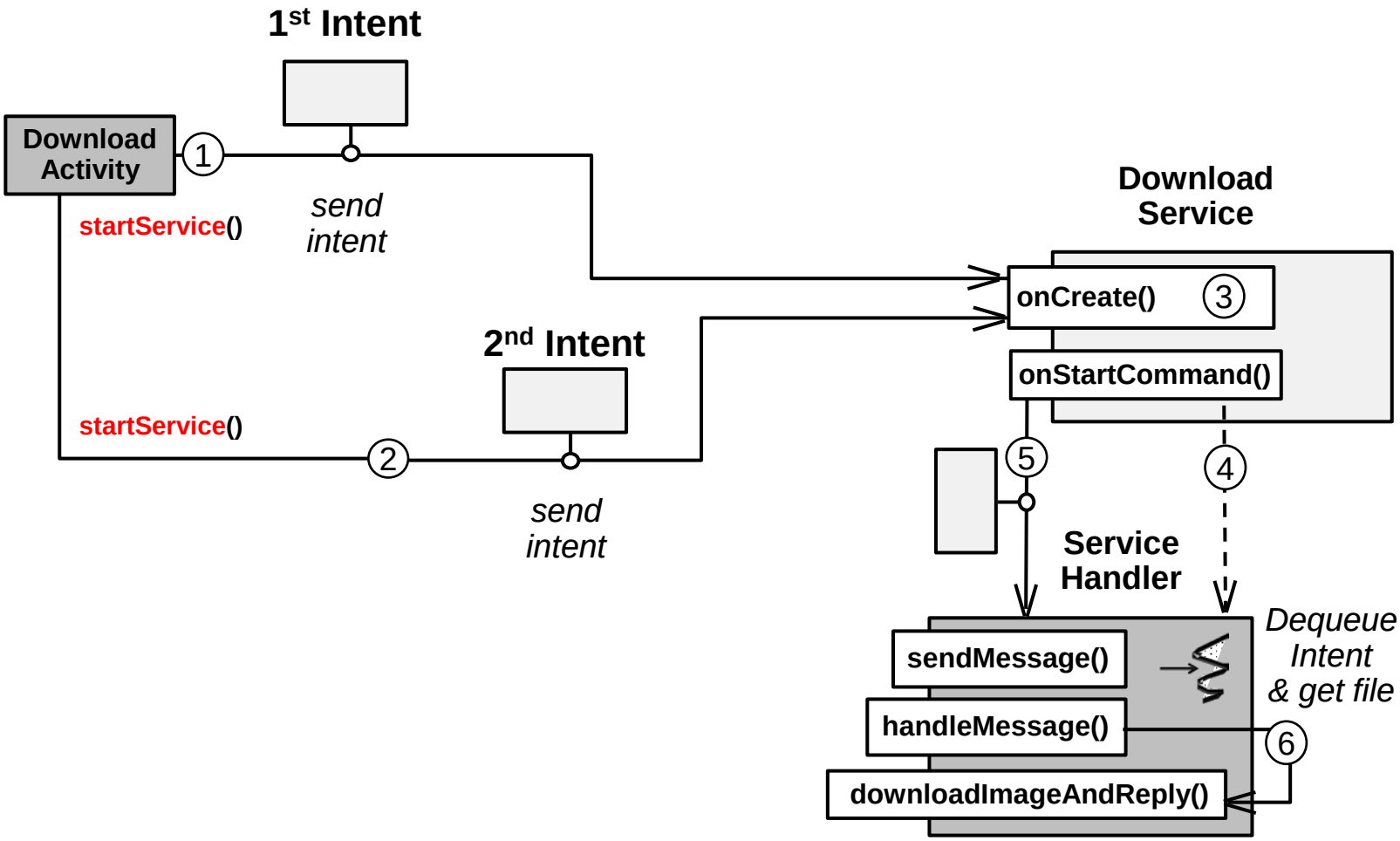
Stopping the DownloadService

- The lifecycle of a Service is independent of the component that launched it



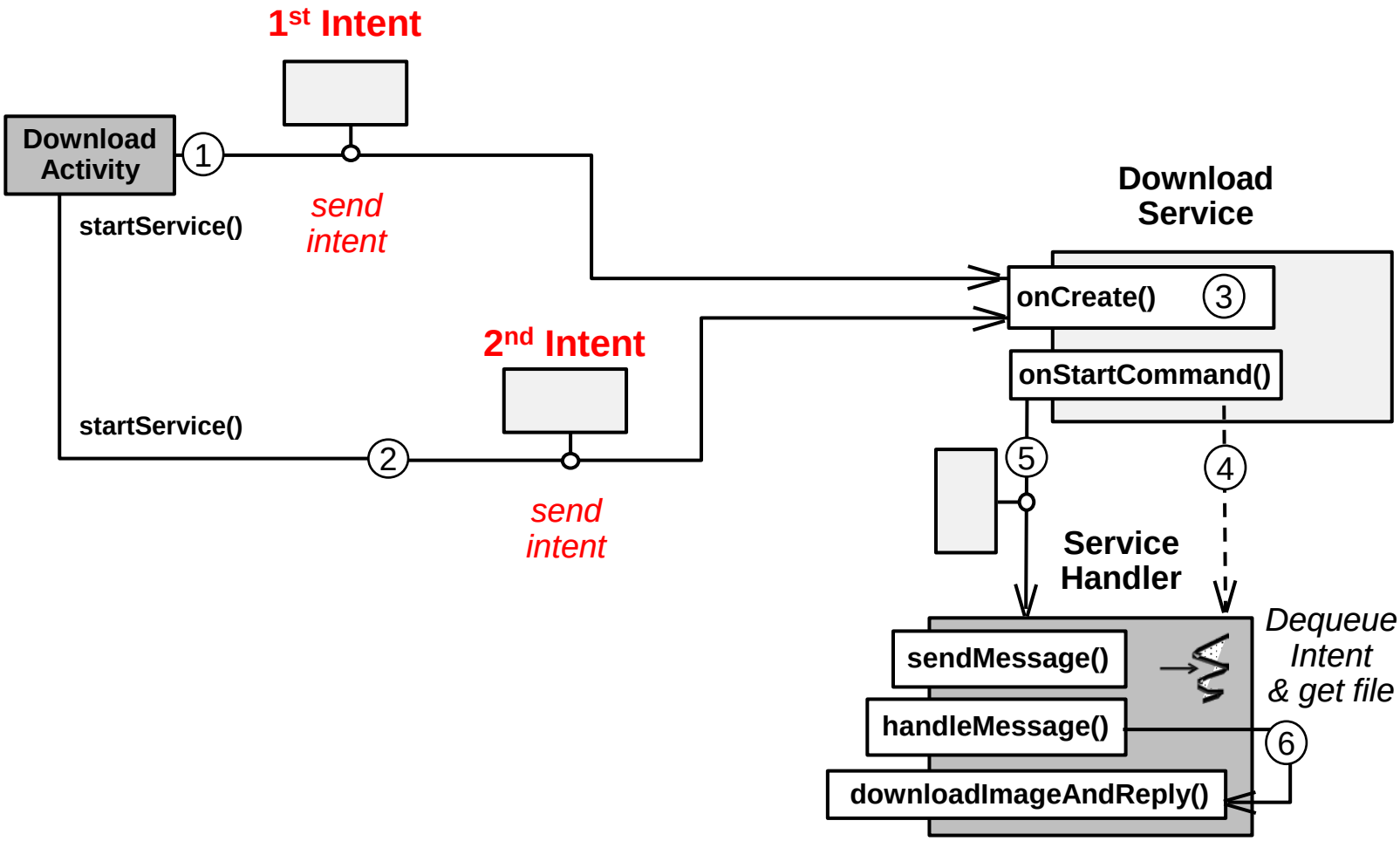
Stopping the DownloadService

- The lifecycle of a Service is independent of the component that launched it



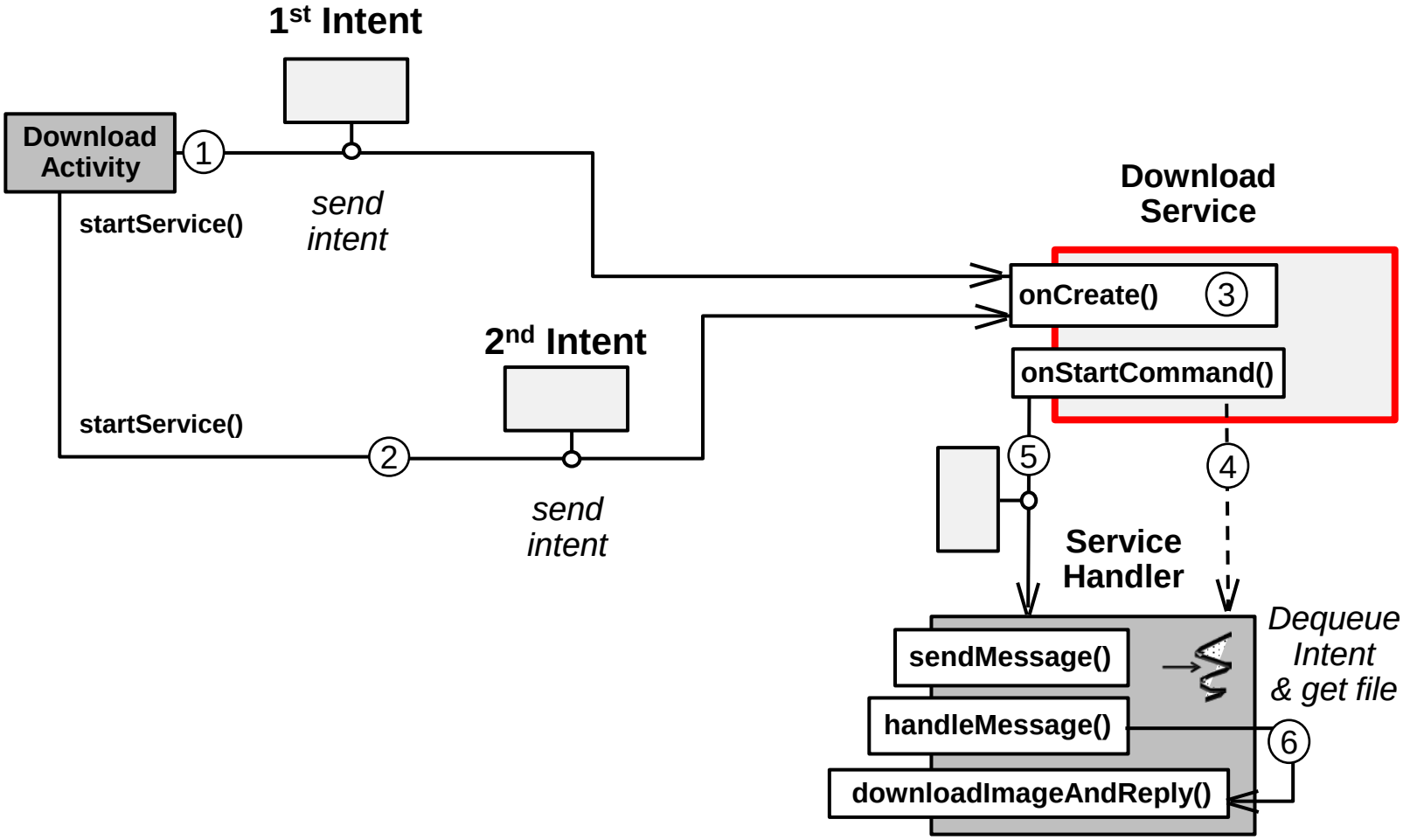
Stopping the DownloadService

- The lifecycle of a Service is independent of the component that launched it



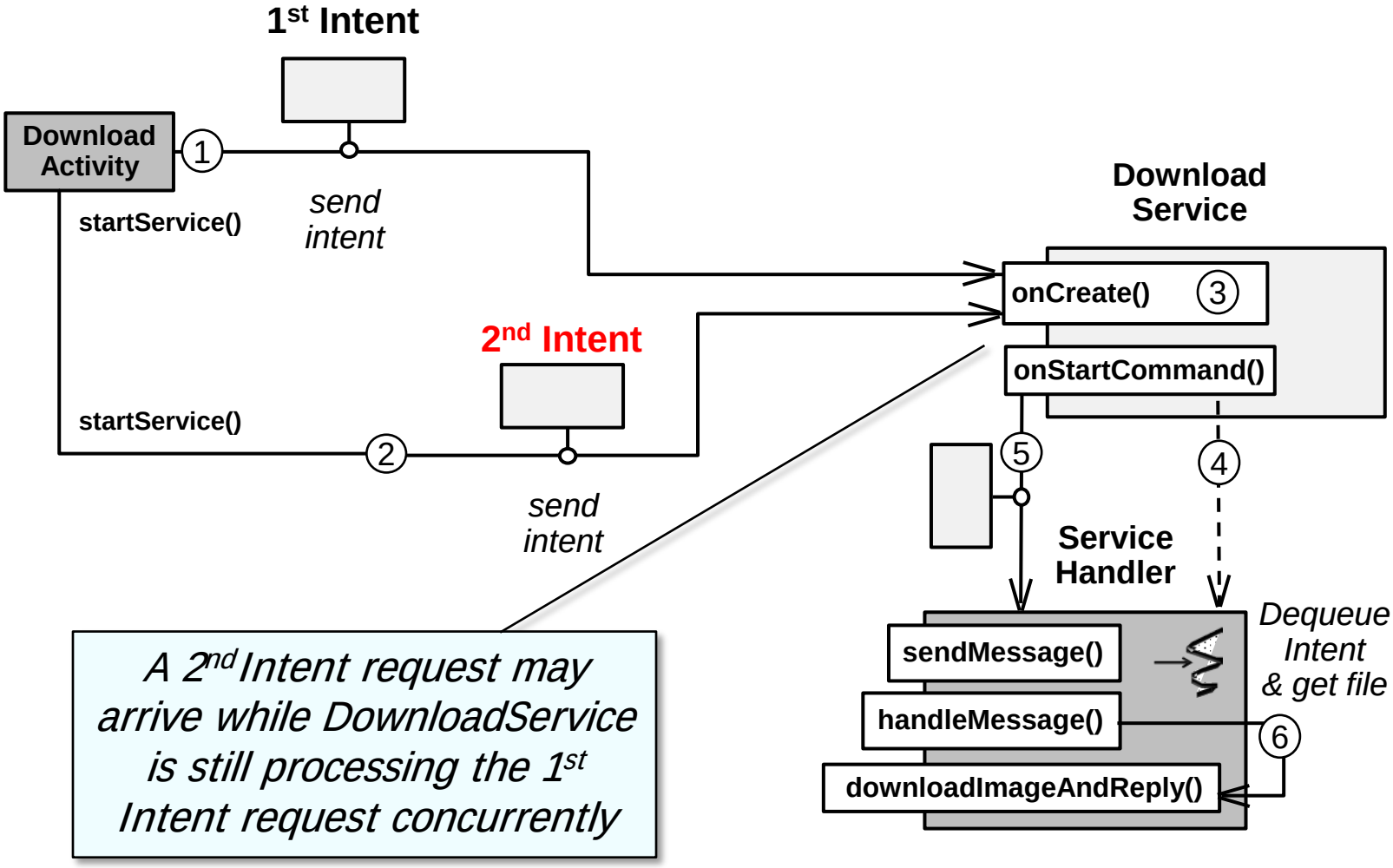
Stopping the DownloadService

- The lifecycle of a Service is independent of the component that launched it



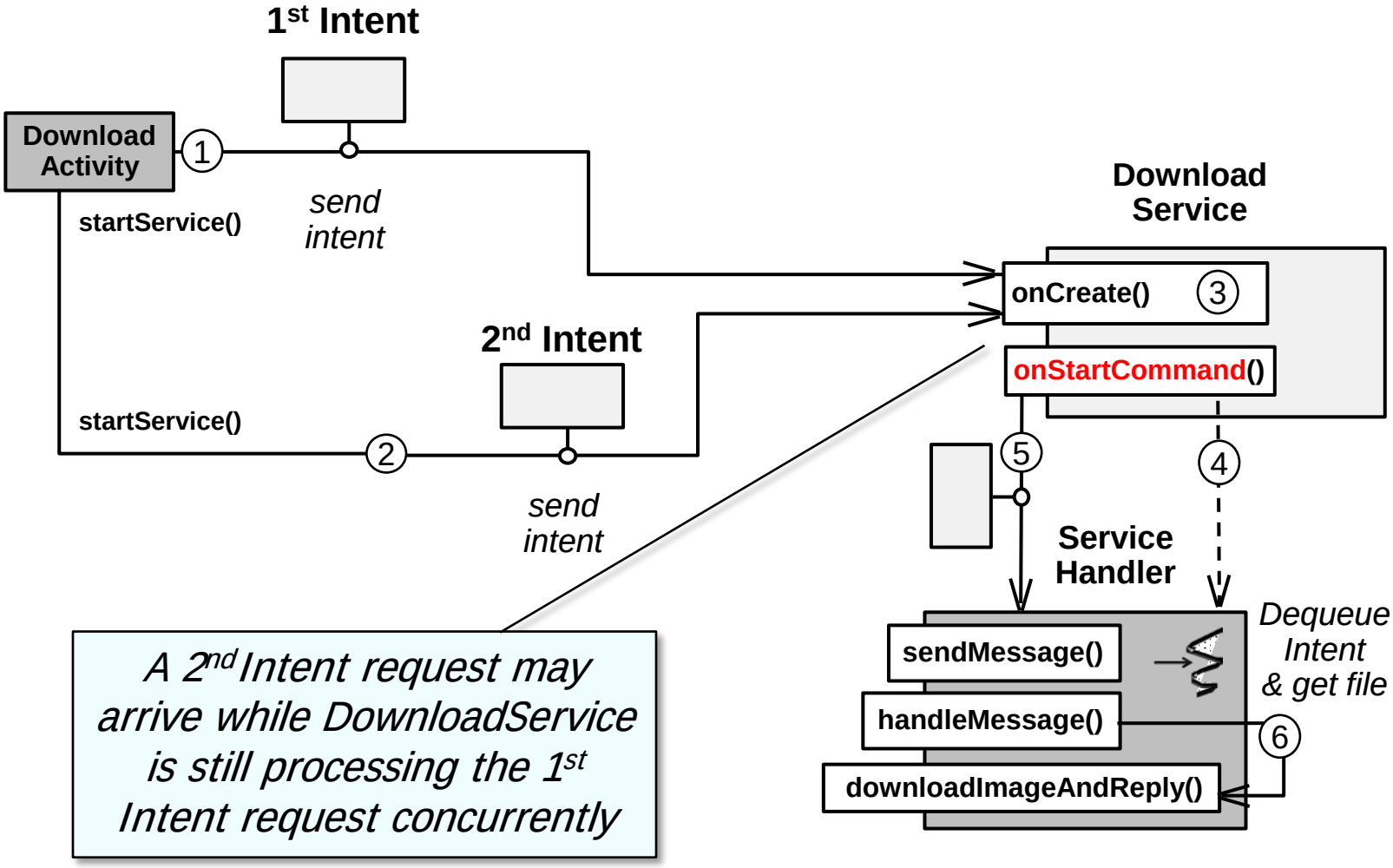
Stopping the DownloadService

- The lifecycle of a Service is independent of the component that launched it



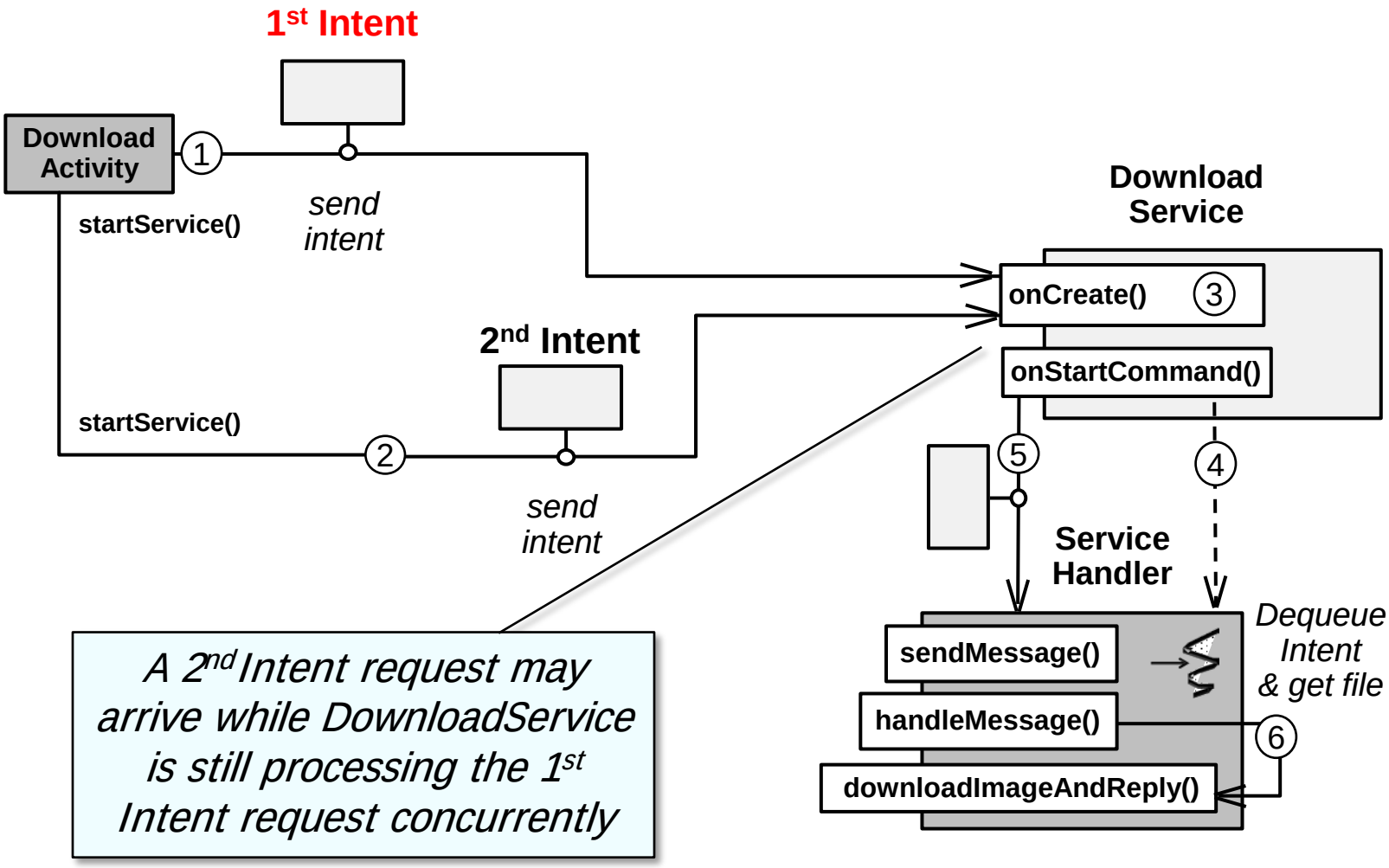
Stopping the DownloadService

- The lifecycle of a Service is independent of the component that launched it



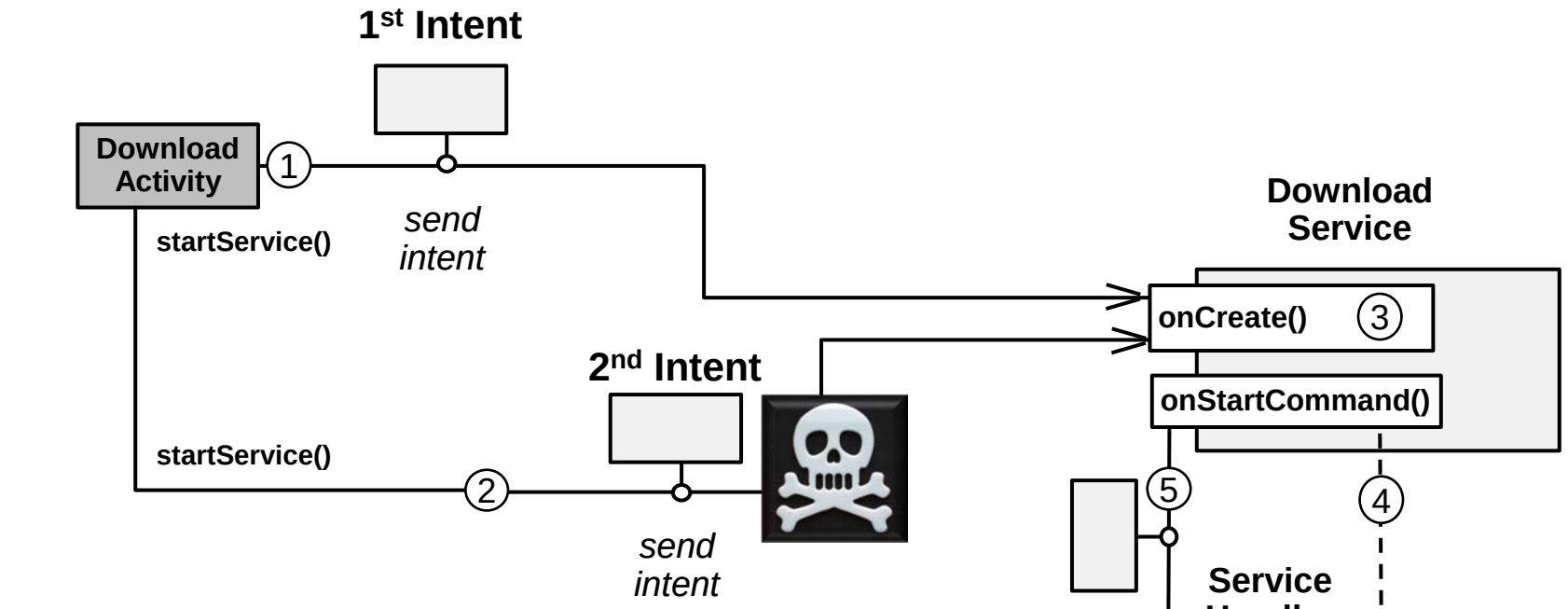
Stopping the DownloadService

- The lifecycle of a Service is independent of the component that launched it



Stopping the DownloadService

- The lifecycle of a Service is independent of the component that launched it



DownloadService can't shut itself down when it's done processing the 1st Intent request or it will terminate the 2nd Intent request prematurely!

Stopping the DownloadService

- The lifecycle of a Service is independent of the component that launched it

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


    public ServiceHandler(Looper looper,
                          DownloadService service) {
      super(looper);
      ...

    public void handleMessage(Message message) {
      downloadImageAndReply((Intent) message.obj);

      stopSelf(message.arg1);
    }
    ...
```

Stopping the DownloadService

- The lifecycle of a Service is independent of the component that launched it

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


    public ServiceHandler(Looper looper,
                          DownloadService service) {
      super(looper);
      ...

    public void handleMessage(Message message) {
      downloadImageAndReply((Intent) message.obj);

      stopSelf(message.arg1);
    }
    ...
```

Stopping the DownloadService

- The lifecycle of a Service is independent of the component that launched it

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


    public ServiceHandler(Looper looper,
                          DownloadService service) {
      super(looper);
      ...

    public void handleMessage(Message message) {
      downloadImageAndReply((Intent) message.obj);

      stopSelf(message.arg1);
    }
  ...
```

Stop Service using startId of original Intent request, so it doesn't stop while handling another request

See developer.android.com/guide/components/services.html#Stopping

Stopping the DownloadService

- The lifecycle of a Service is independent of the component that launched it

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


    public ServiceHandler(Looper looper,
                          DownloadService service) {
      super(looper);
      ...

    public void handleMessage(Message message) {
      downloadImageAndReply((Intent) message.obj);

      stopSelf(message.arg1);
    }
  ...
```

Stop Service using startId of original Intent request, so it doesn't stop while handling another request

Android only stops the Service when startId matches the last start request

Stopping the DownloadService

- The lifecycle of a Service is independent of the component that launched it

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


    public ServiceHandler(Looper looper,
                          DownloadService service) {
      super(looper);
      ...

    public void handleMessage(Message message) {
      downloadImageAndReply((Intent) message.obj);

      stopSelf(message.arg1);
    }
  ...
```

Stop Service using startId of original Intent request, so it doesn't stop while handling another request

More complicated mechanisms are needed to stop Services that process multiple Intents in multiple Threads

Stopping the DownloadService

- The lifecycle of a Service is independent of the component that launched it

```
public class DownloadService extends Service {
  ...

  public void onDestroy() {
    mServiceLooper.quit();
  }
  ...
```

This hook method is automatically dispatched after the Service is stopped

Stopping the DownloadService

- The lifecycle of a Service is independent of the component that launched it

```
public class DownloadService extends Service {
  ...



  public void onDestroy() {
    mServiceLooper.quit();
  }
  ...
```

Shutdown the looper