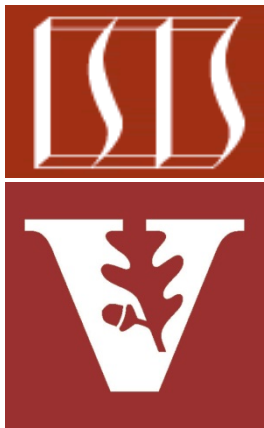


Android Services & Security: Programming Started Services (Part 2)

Douglas C. Schmidt

d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt



Professor of Computer Science

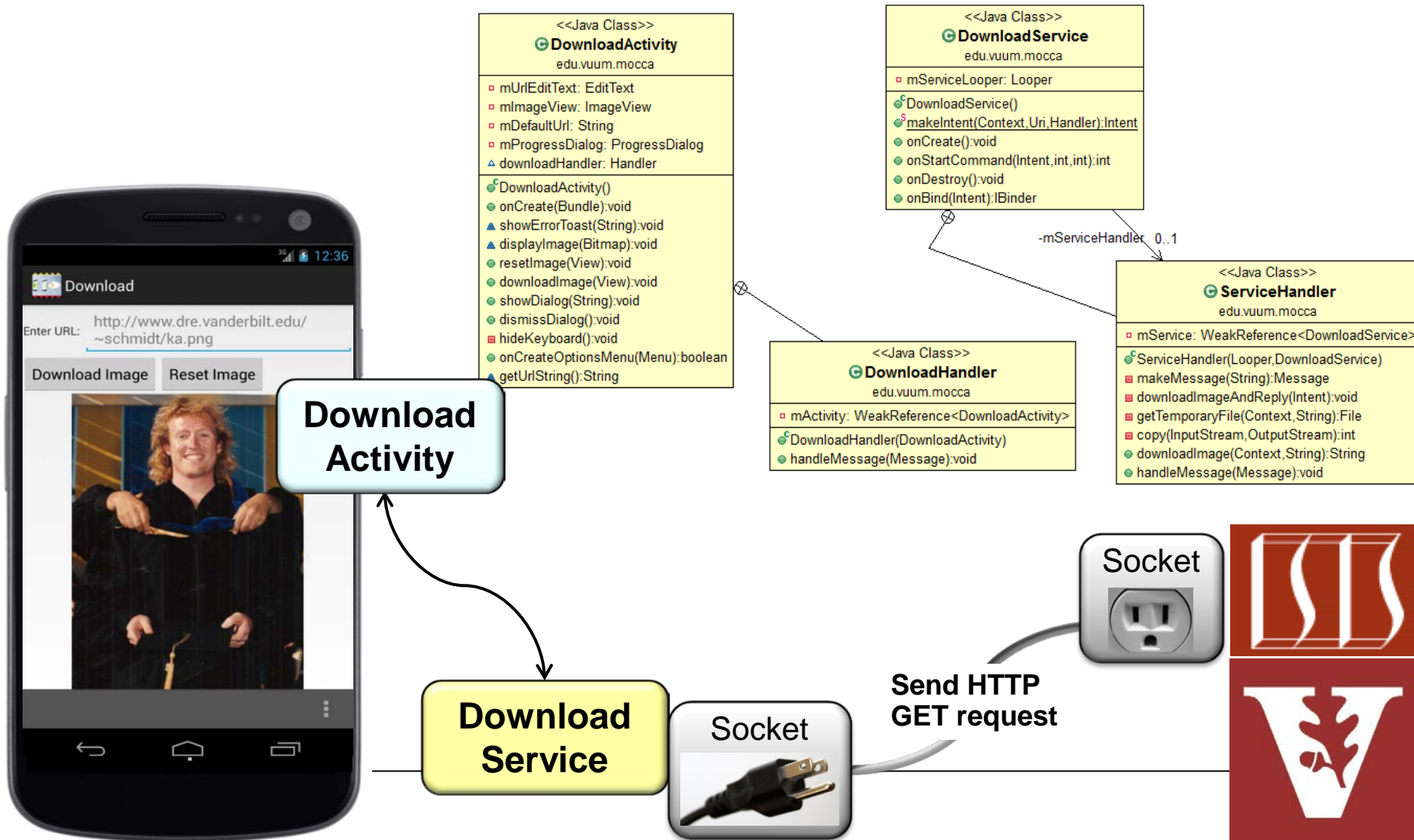
Institute for Software
Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA

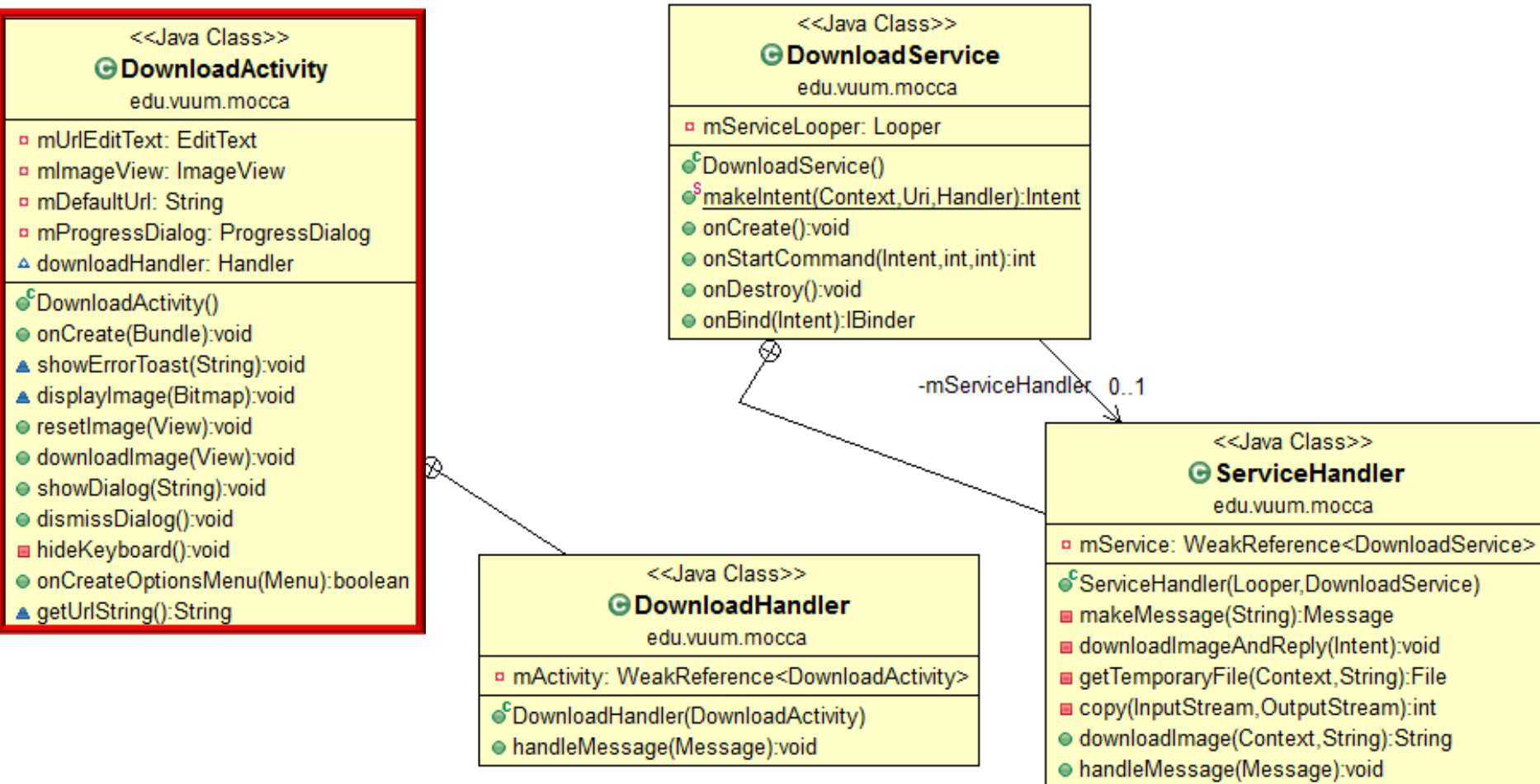


Learning Objectives in this Part of the Module

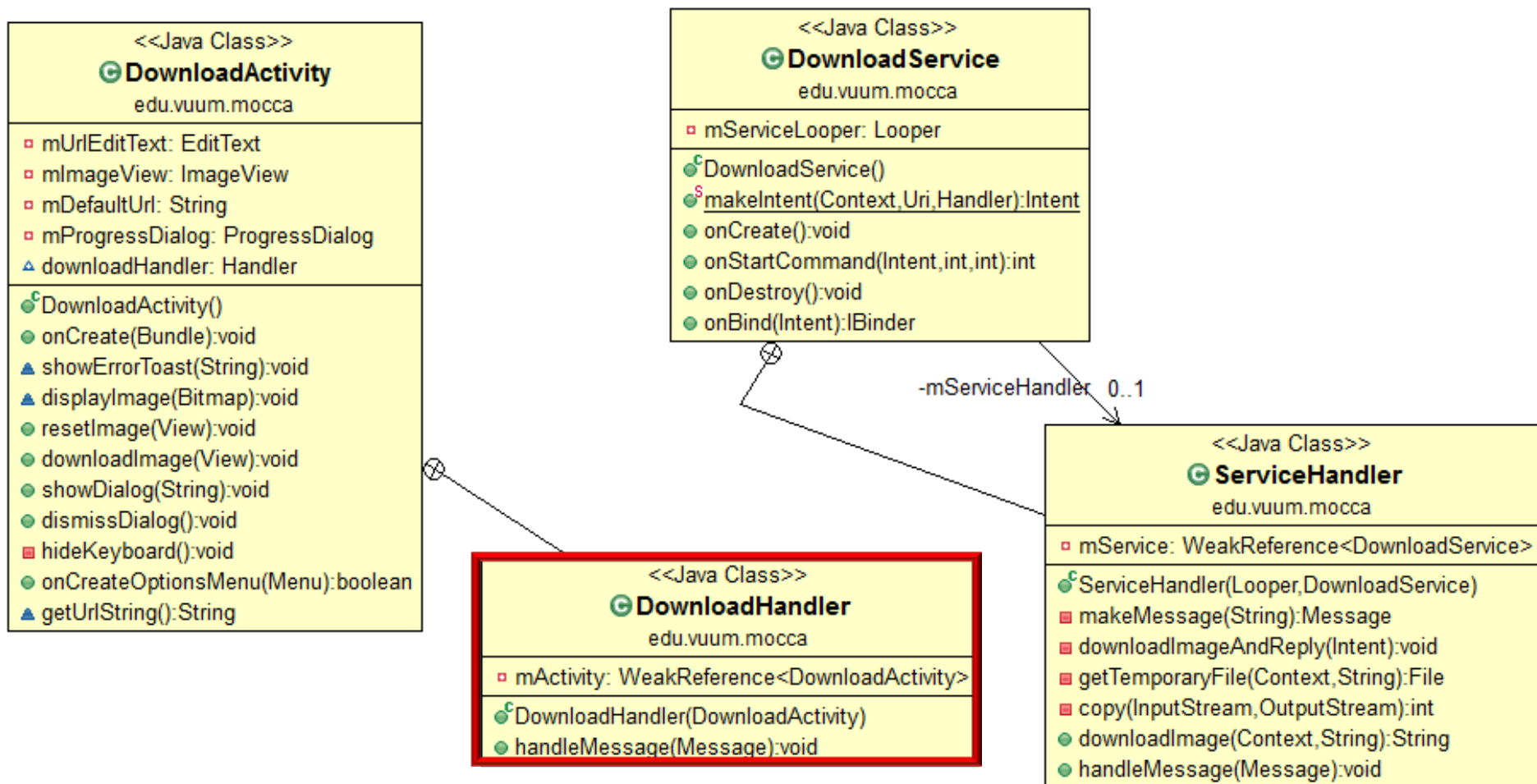
- Understand the implementation of the Download Application



Download Activity Implementation




Download Activity Implementation




Download Activity Implementation

```
public class DownloadActivity extends Activity {  
    ...  
    public void downloadImage(View v) {  
        String url = getUrlString();  
        ...  
        Intent intent = DownloadService.makeIntent  
            (this,  
             Uri.parse(url),  
             downloadHandler);  
  
        startService(intent);  
    }  
  
    ...  
  
    Handler downloadHandler = new DownloadHandler(this);  
    ...  
}
```


 Activity that prompts user
for image URL to download

Download Activity Implementation

```
public class DownloadActivity extends Activity {  
    ...  
    public void downloadImage(View v) {  
        String url = getUrlString();  
        ...  
        Intent intent = DownloadService.makeIntent  
            (this,  
             Uri.parse(url),  
             downloadHandler);  
  
        startService(intent);  
    }  
  
    ...  
  
    Handler downloadHandler = new DownloadHandler(this);  
    ...  
}
```

 Activity that prompts user
for image URL to download


Download Activity Implementation

```
public class DownloadActivity extends Activity {  
    ...  
     Dispatched when user clicks "Download Image" button  
    public void downloadImage(View v) {  
        String url = getUrlString();  
        ...  
        Intent intent = DownloadService.makeIntent  
            (this,  
             Uri.parse(url),  
             downloadHandler);  
  
        startService(intent);  
    }  
  
    ...  
  
    Handler downloadHandler = new DownloadHandler(this);  
    ...  
}
```

Download Activity Implementation

```
public class DownloadActivity extends Activity {  
    ...  
  
    public void downloadImage(View v) {  
        String url = getUrlString();  
        ...  
        Intent intent = DownloadService.makeIntent  
            (this,  
             Uri.parse(url),  
             downloadHandler);  
  
        startService(intent);  
    }  
  
    ...  
  
    Handler downloadHandler = new DownloadHandler(this);  
    ...  
}
```

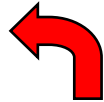
Get URL entered by user



Download Activity Implementation

```
public class DownloadActivity extends Activity {  
    ...  
  
    public void downloadImage(View v) {  
        String url = getUrlString();  
        ...  
        Intent intent = DownloadService.makeIntent  
                                (this,  
                                Uri.parse(url),  
                                downloadHandler);  
        startService(intent);  
    }  
  
    ...  
  
    Handler downloadHandler = new DownloadHandler(this);  
    ...  
}
```

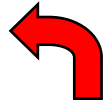
Create Intent associated with DownloadService



Download Activity Implementation

```
public class DownloadActivity extends Activity {  
    ...  
  
    public void downloadImage(View v) {  
        String url = getUrlString();  
        ...  
        Intent intent = DownloadService.makeIntent  
            (this,  
             Uri.parse(url),  
             downloadHandler);  
        startService(intent);  
    }  
  
    ...  
  
    Handler downloadHandler = new DownloadHandler(this);  
    ...  
}
```

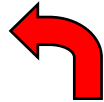
Create Intent associated with DownloadService



Download Activity Implementation

```
public class DownloadActivity extends Activity {  
    ...  
  
    public void downloadImage(View v) {  
        String url = getUrlString();  
        ...  
        Intent intent = DownloadService.makeIntent  
            (this,  
             Uri.parse(url),  
             downloadHandler);  
        startService(intent);  
    }  
  
    ...  
  
    Handler downloadHandler = new DownloadHandler(this);  
    ...  
}
```

Create Intent associated with DownloadService



Download Activity Implementation

```
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

```
    ...
```

```
    Handler downloadHandler = new DownloadHandler(this);  
    ...
```

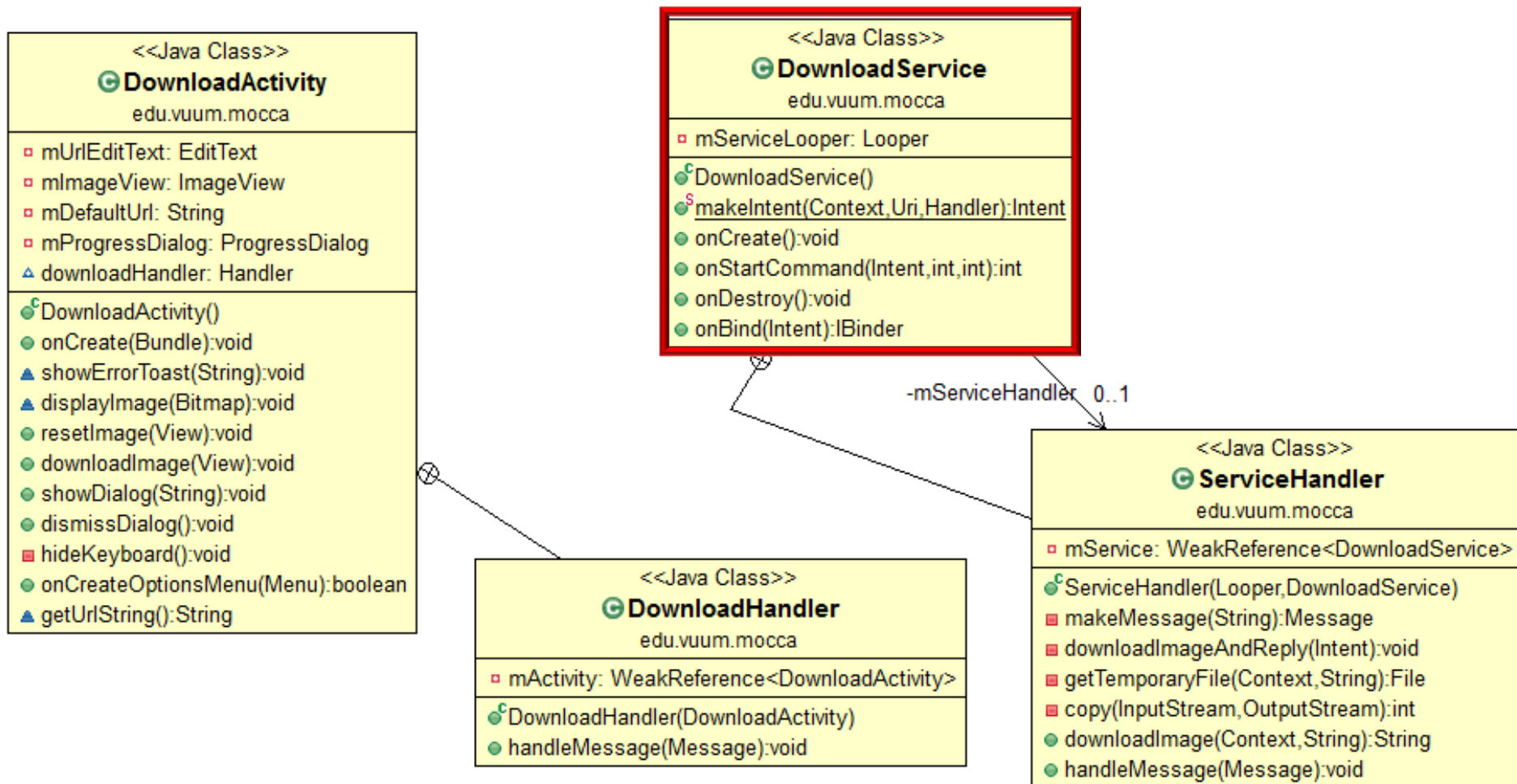
Download Activity Implementation

```
public class DownloadActivity extends Activity {  
    ...  
  
    public void downloadImage(View v) {  
        String url = getUrlString();  
        ...  
        Intent intent = DownloadService.makeIntent  
            (this,  
             Uri.parse(url),  
             downloadHandler);  
  
        startService(intent);  
    }  
  
    ...  
    Code for processing the  
    downloaded image shown later  
    Handler downloadHandler = new DownloadHandler(this);  
    ...
```

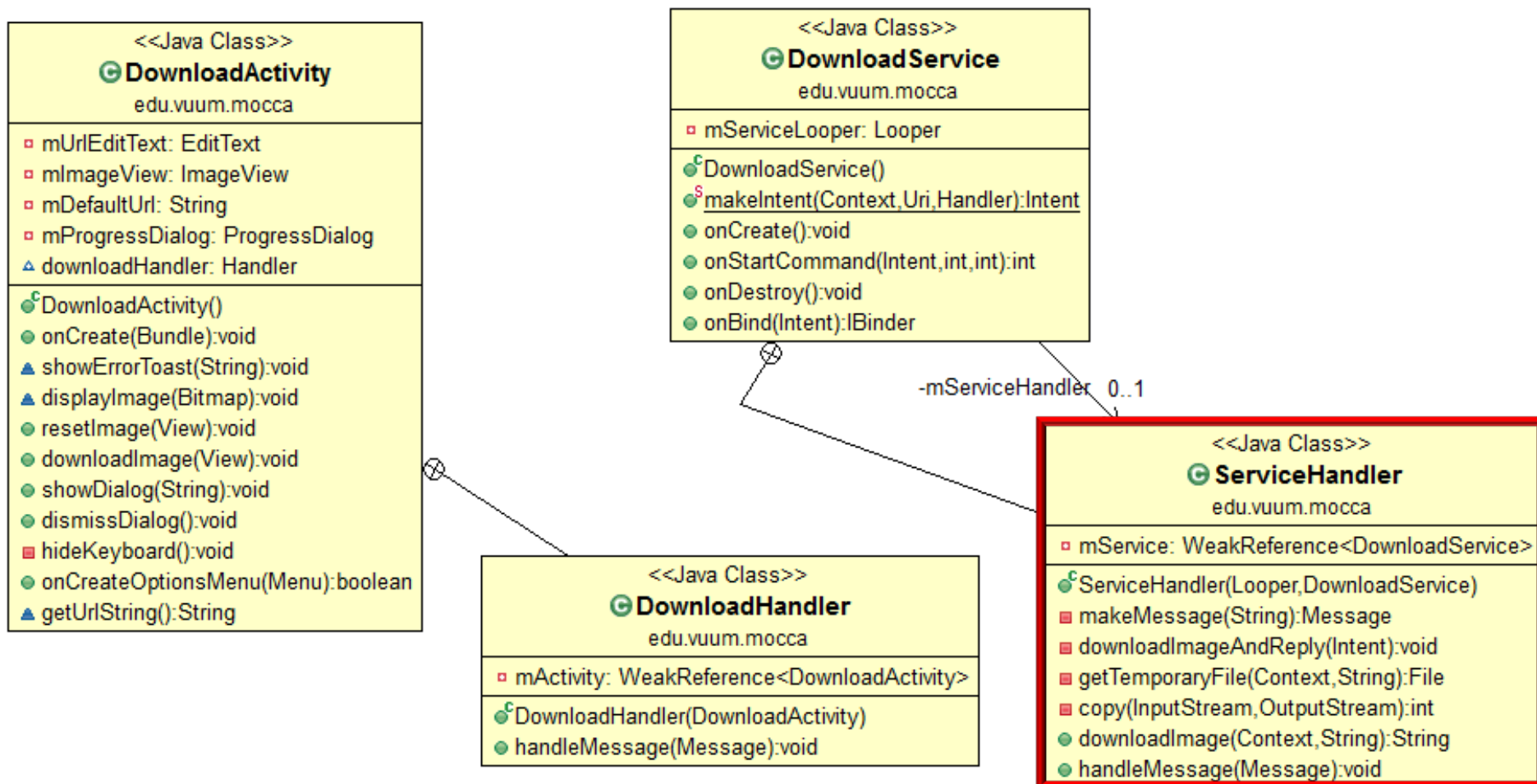


Initializing the Download Service Implementation

Download Service Implementation



Download Service Implementation



Download Service Implementation

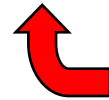
```
public class DownloadService extends Service {  
    ...
```



DownloadService overrides
Service hook methods

Download Service Implementation

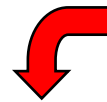
```
public class DownloadService extends Service {  
    ...
```



DownloadService overrides
Service hook methods

Download Service Implementation

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




Factory method 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 is called by the DownloadActivity

Download Service Implementation

```
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

Download Service Implementation


```
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 the URI to the download as data

Download Service Implementation

```
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 Messenger that encapsulates  
DownloadHandler as an extra  
  
        intent.putExtra("MESSENGER",  
                       new Messenger(downloadHandler));  
  
        return intent;  
    }  
    ...  
}
```

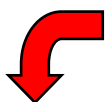
Download Service Implementation

```
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 Messenger that encapsulates  
DownloadHandler as an extra  
  
        intent.putExtra("MESSENGER",  
                       new Messenger(downloadHandler));  
  
        return intent;  
    }  
    ...  
}
```

Download Service Implementation

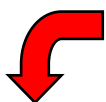
```
public class DownloadService extends Service {  
    ...  
    private volatile Looper mServiceLooper;  
    private volatile ServiceHandler mServiceHandler;  
  
    public void onCreate() {  
        super.onCreate();  
  
        HandlerThread thread = new HandlerThread("DownloadService");  
        thread.start();  
  
        mServiceLooper = thread.getLooper();  
        mServiceHandler = new ServiceHandler(mServiceLooper);  
    }  
}
```

Useful data members



Download Service Implementation

```
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

Download Service Implementation

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

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



Called when DownloadService is first
launched by the Android Activity Manager

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

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

Download Service Implementation

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

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



Create/start a background HandlerThread since by default a Service runs in the UI Thread, which we don't want to block

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

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

Download Service Implementation

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

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

 Create/start a background HandlerThread since by default a Service runs in the UI Thread, which we don't want to block

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

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

Download Service Implementation

```
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);  
}
```

Download Service Implementation

```
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);  
    }
```

Download Service Implementation

```
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);  
    }
```

Processing Intents Concurrently in the Download Service Implementation


Download Service Implementation

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

 Called each time a Started Service
is sent an Intent via startService()

Download Service Implementation


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



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


Download Service Implementation

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


Download Service Implementation

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

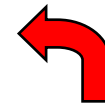
Download Service Implementation

```
public class DownloadService extends Service {  
    ...  
    public int onStartCommand(Intent intent, int f, int startId) {  
  
        Message message =  
            mServiceHandler.makeDownloadMessage(intent, startId);  
  
        mServiceHandler.sendMessage(msg);  
        return START_NOT_STICKY;  
    }  
  
    private Message makeDownloadMessage(Intent intent, int startId){  
        Message message = Message.obtain();  
        message.obj = intent;  
        message.arg1 = startId;  
        return message;  
    }
```

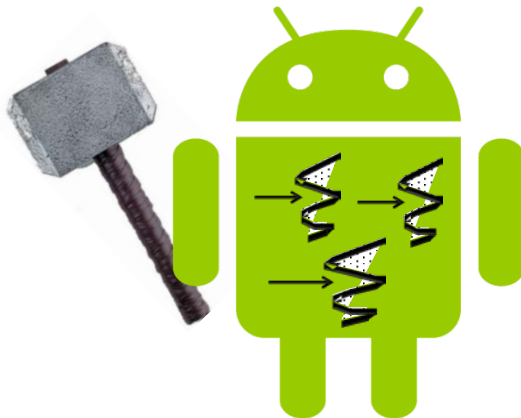
 Include Intent & start ID to indicate
which URI to retrieve & which request is
being stopped when download completes

Download Service Implementation

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

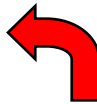


Send Message to ServiceHandler
to retrieve an image based on
contents of the Intent



Download Service Implementation

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



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


Download Service Implementation

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

 Concurrently processes Messages
sent by onStartCommand()

Download Service Implementation


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

 Concurrently processes Messages sent by onStartCommand()




Download Service Implementation

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

 Constructor makes its Looper
the HandlerThread's Looper

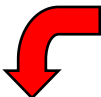
Download Service Implementation

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

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

Download Service Implementation

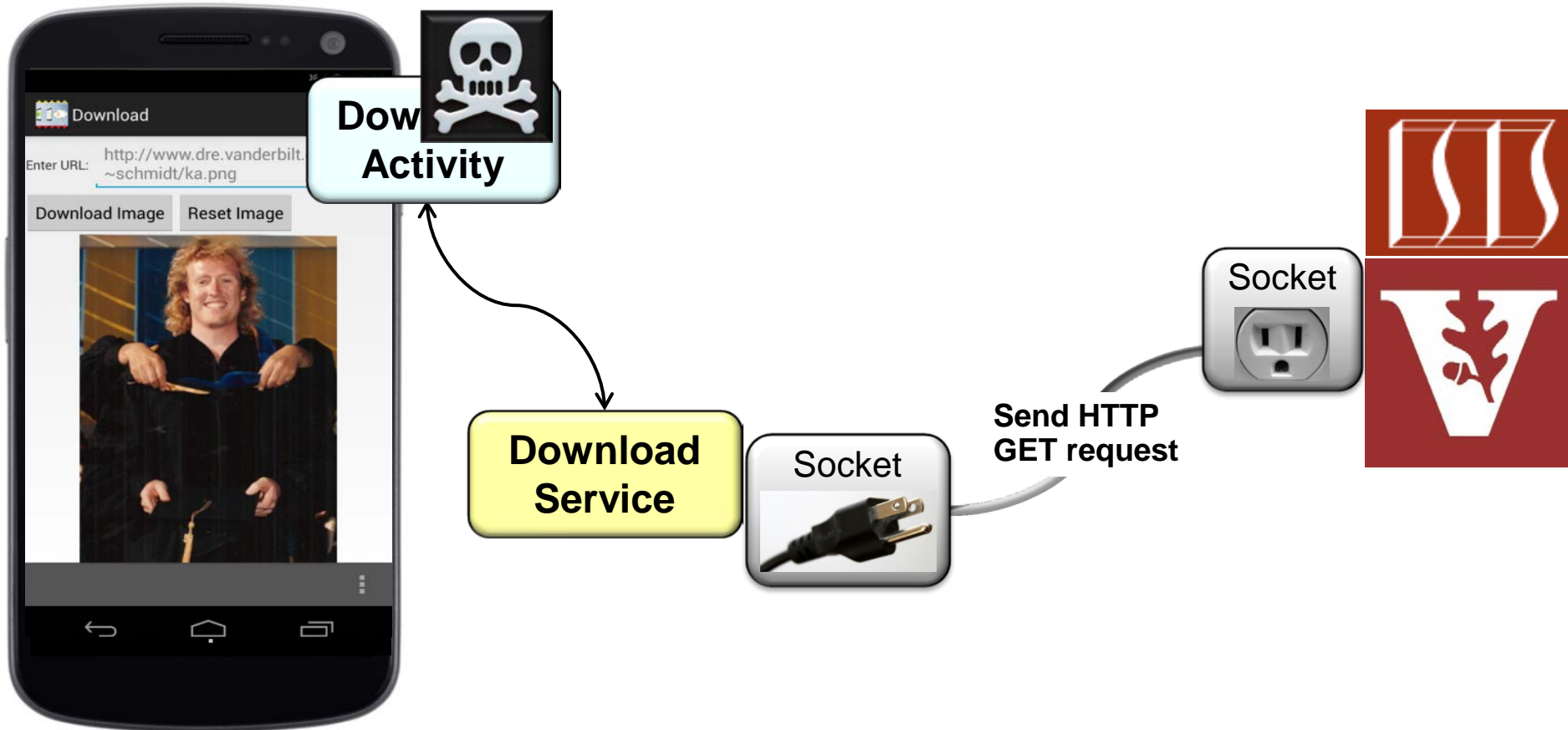
```
public class DownloadService extends Service {  
    ...  
    private final class ServiceHandler extends Handler {  
        ...  
  
        public ServiceHandler(Looper looper,  
                               DownloadService service) {  
            super(looper);  
            ...  
        }  
  
        public void handleMessage(Message message) {  
            downloadImageAndReply((Intent) message.obj);  
  
            stopSelf(msg.arg1);  
        }  
        ...  
    }  
}
```

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

Stopping the Download Service Implementation

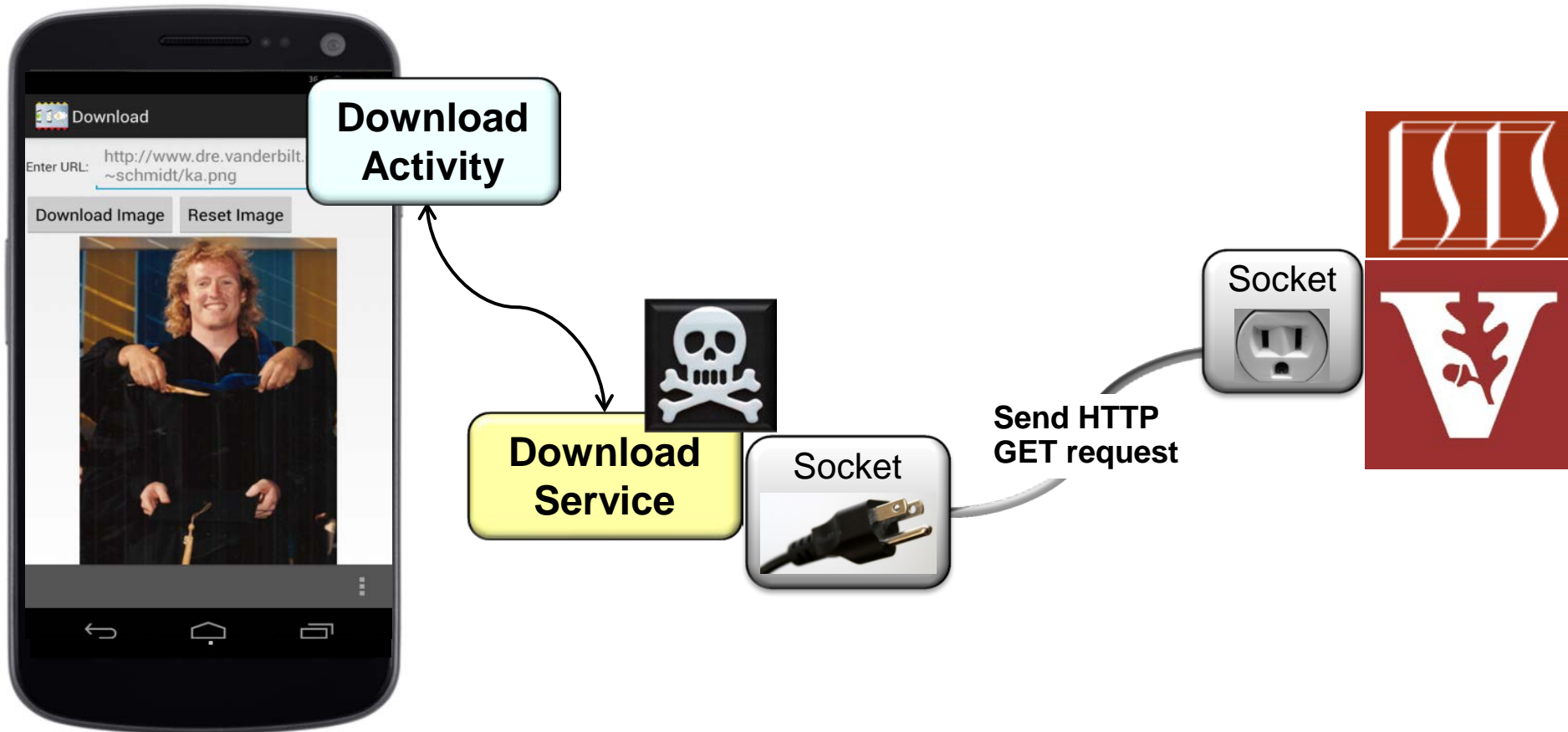
Download Service Implementation

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



Download Service Implementation

- 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!




Download Service Implementation

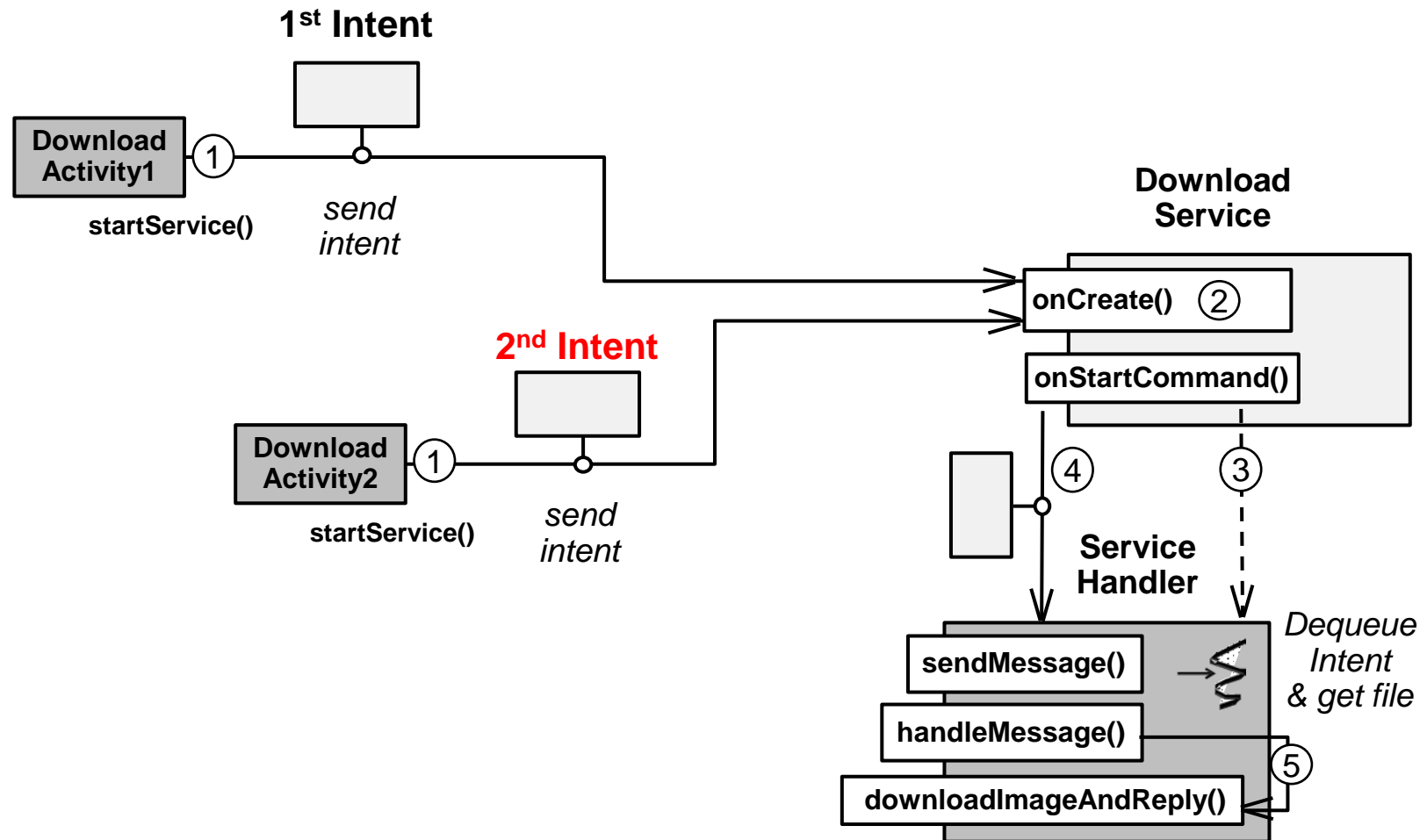
```
public class DownloadService extends Service {  
    ...  
    private final class ServiceHandler extends Handler {  
        ...  
  
        public ServiceHandler(Looper looper,  
                               DownloadService service) {  
            super(looper);  
            ...  
  
            public void handleMessage(Message message) {  
                downloadImageAndReply((Intent) message.obj);  
  
                stopSelf(message.arg1);  
            }  
            ...  
        }  
    }  
}
```


Download Service Implementation

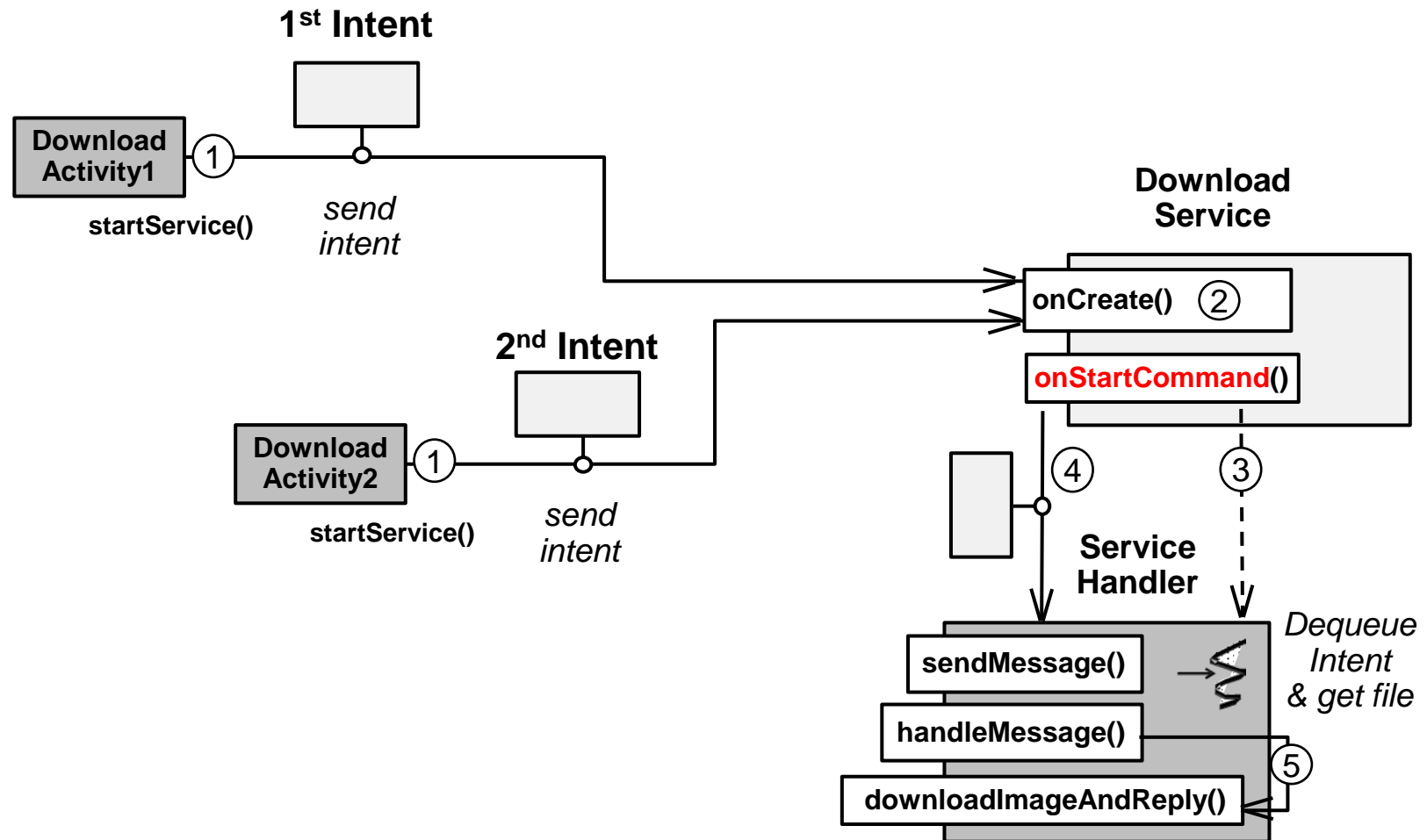
```
public class DownloadService extends Service {  
    ...  
    private final class ServiceHandler extends Handler {  
        ...  
  
        public ServiceHandler(Looper looper,  
                               DownloadService service) {  
            super(looper);  
            ...  
  
            public void handleMessage(Message message) {  
                downloadImageAndReply((Intent) message.obj);  
  
                stopSelf(message.arg1);  
            }  
        }  
    }  
    ...
```

 Stop the Service using a common Android idiom

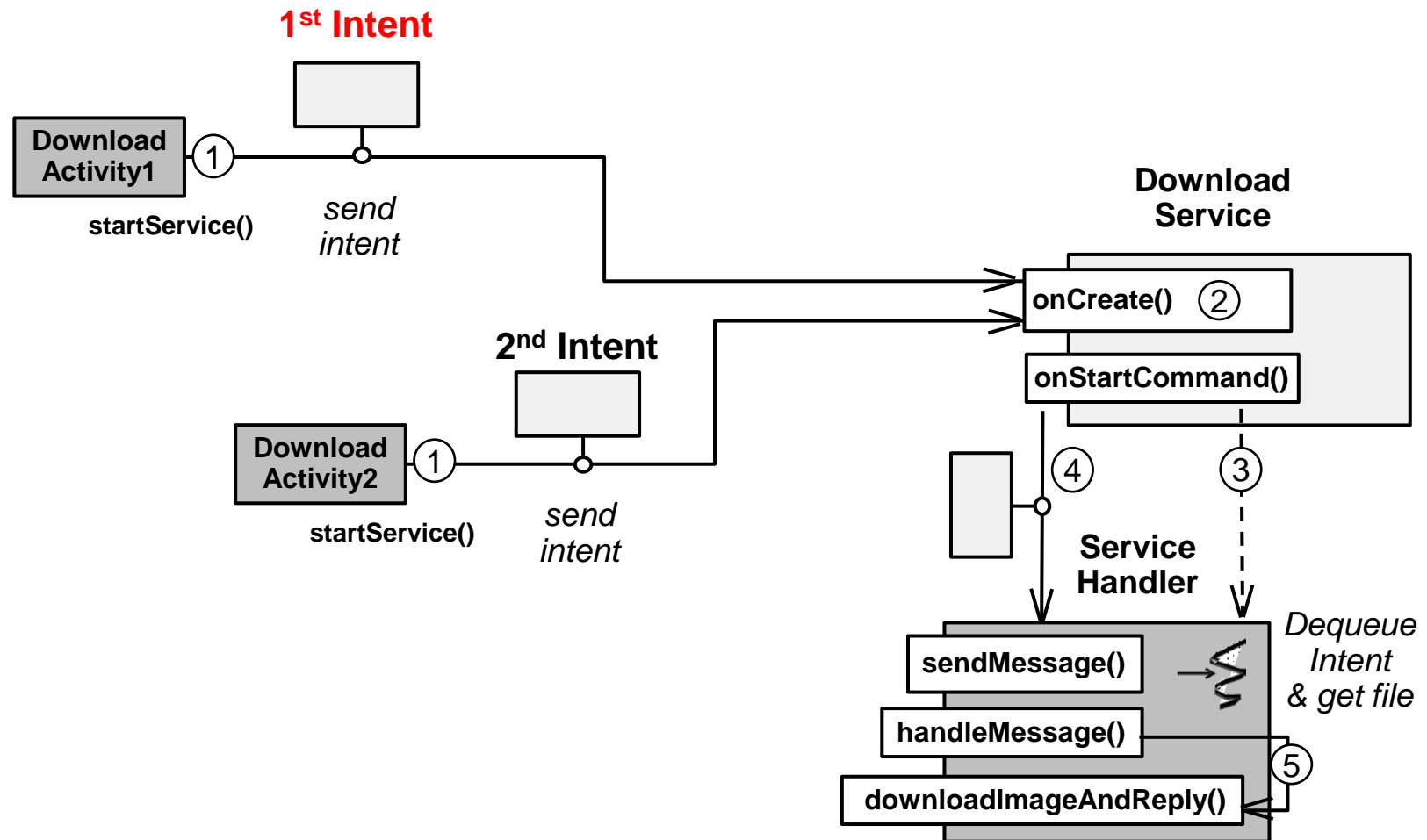
Download Service Implementation



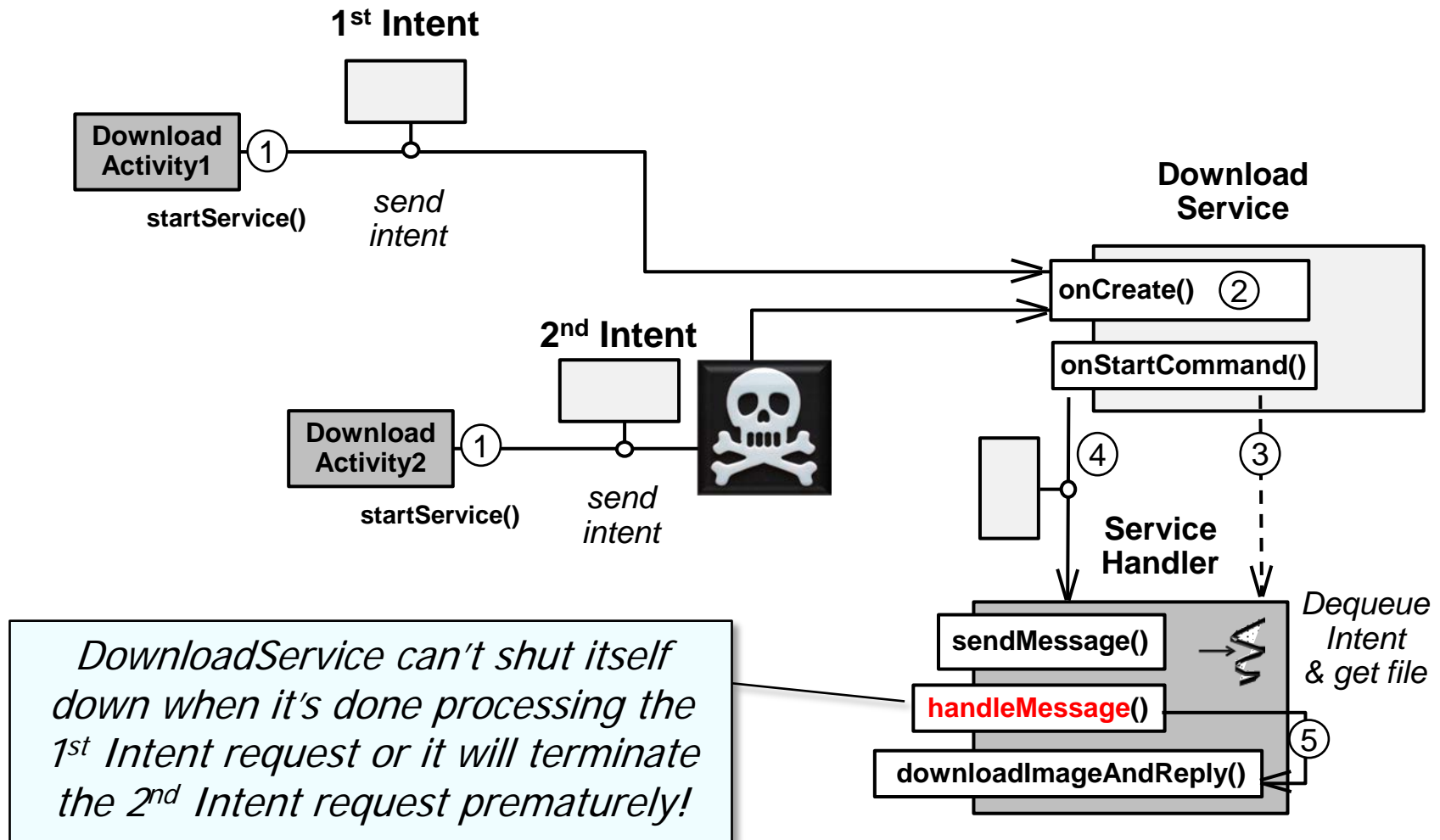
Download Service Implementation



Download Service Implementation



Download Service Implementation



Download Service Implementation

```
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 the original Intent request, so it doesn't stop in the middle of handling another request


Download Service Implementation

```
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 the original Intent request, so it doesn't stop in the middle of handling another request

Download Service Implementation

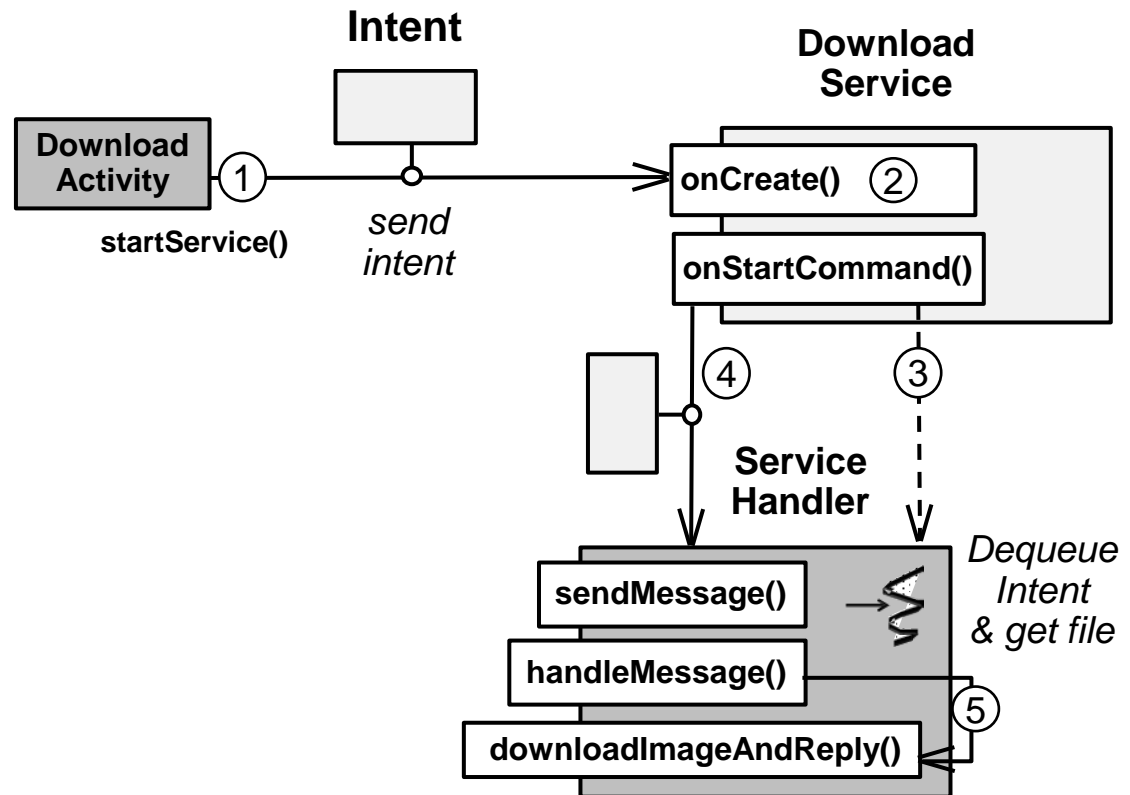
```
public class DownloadService extends Service {  
    ...  
    public void onDestroy() {  
        mServiceLooper.quit();  Shutdown the looper  
    }  
    ...  
}
```


Summary



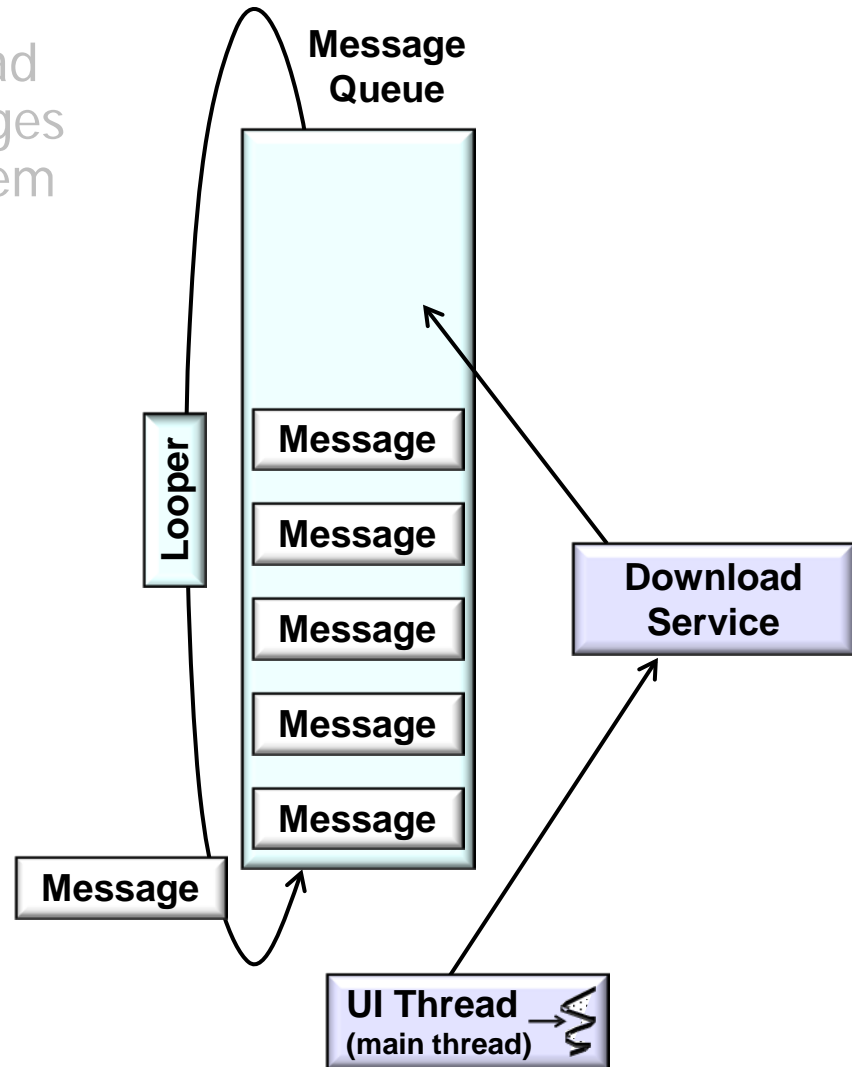
Summary

- The Download Activity & Download Service concurrently retrieve images from remote servers & display them



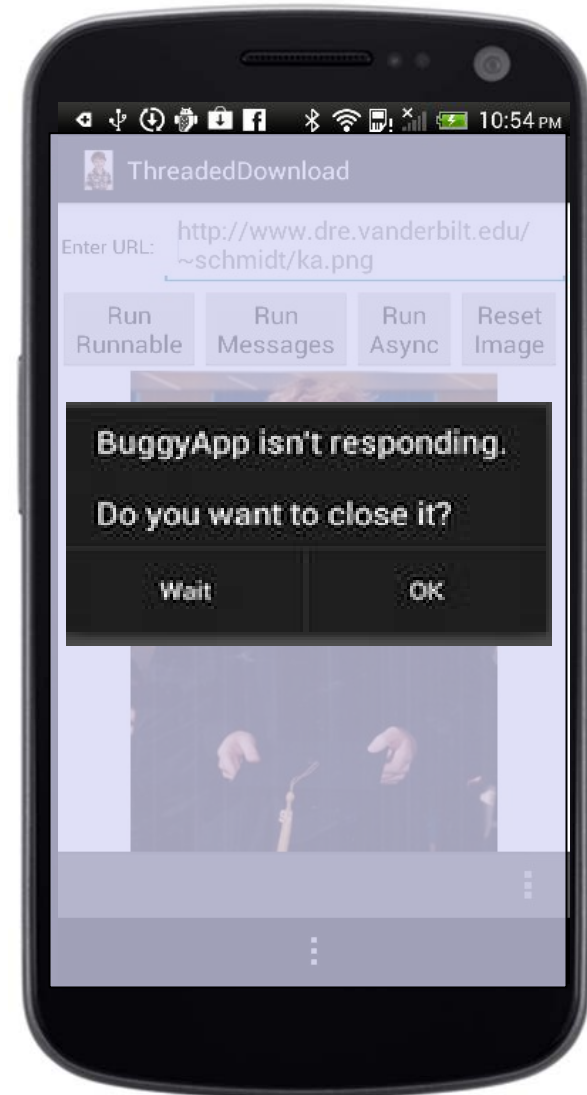
Summary

- The Download Activity & Download Service concurrently retrieve images from remote servers & display them
- A Service uses the UI Thread by default



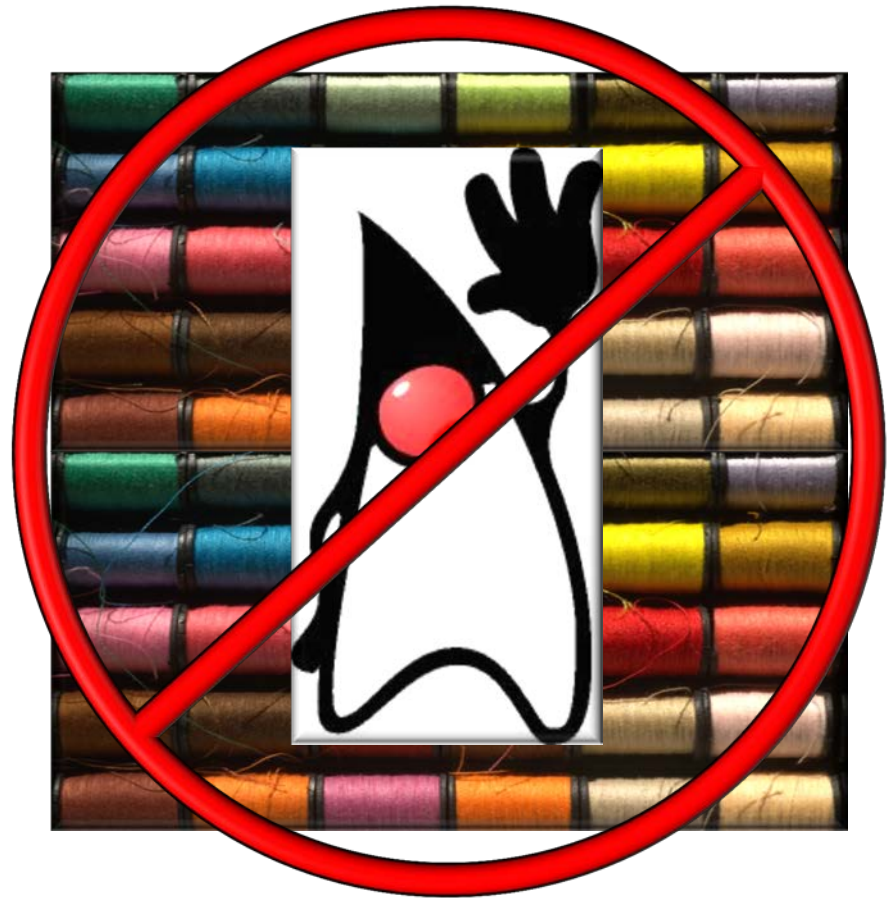
Summary

- The Download Activity & Download Service concurrently retrieve images from remote servers & display them
- A Service uses the UI Thread by default
 - It thus can incur “Application Not Responding” errors if it blocks too long



Summary

- The Download Activity & Download Service concurrently retrieve images from remote servers & display them
- A Service uses the UI Thread by default
- A Service is *not* a Thread



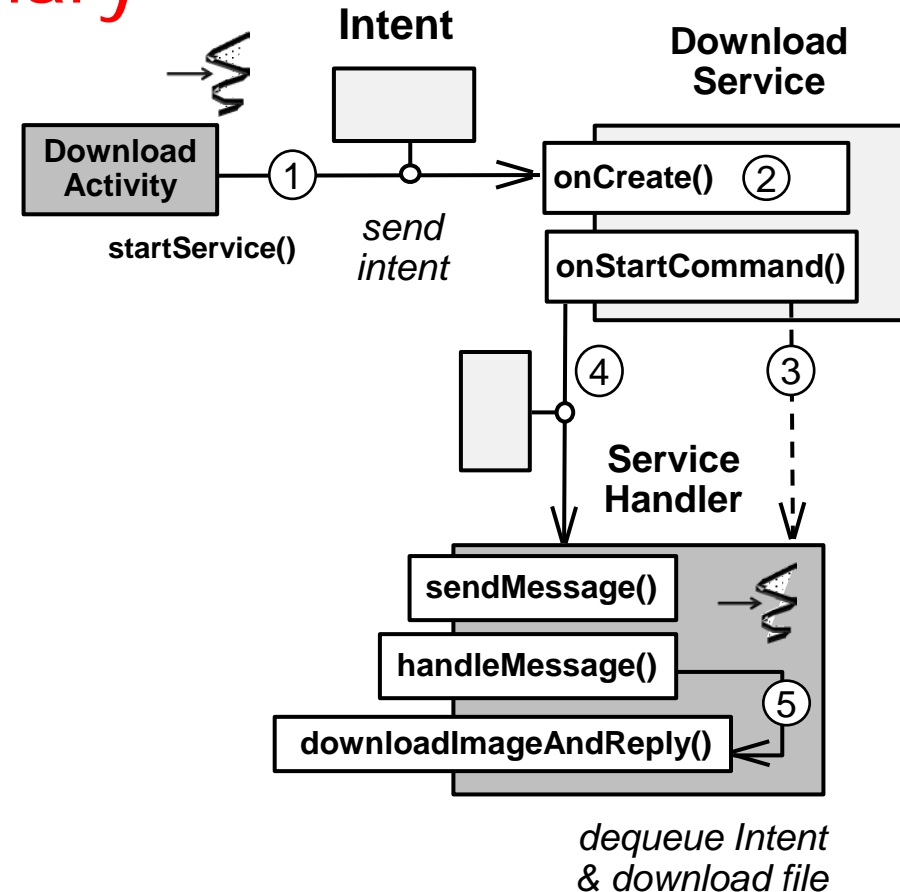
Summary

- The Download Activity & Download Service concurrently retrieve images from remote servers & display them
- A Service uses the UI Thread by default
- A Service is *not* a Thread
 - It must be programmed to run in a separate Thread or Process

```
public void onCreate() {  
    super.onCreate();  
  
    HandlerThread thread =  
        new HandlerThread  
            ("DownloadService");  
    thread.start();  
  
    mServiceLooper =  
        thread.getLooper();  
    mServiceHandler =  
        new ServiceHandler  
            (mServiceLooper);  
}
```

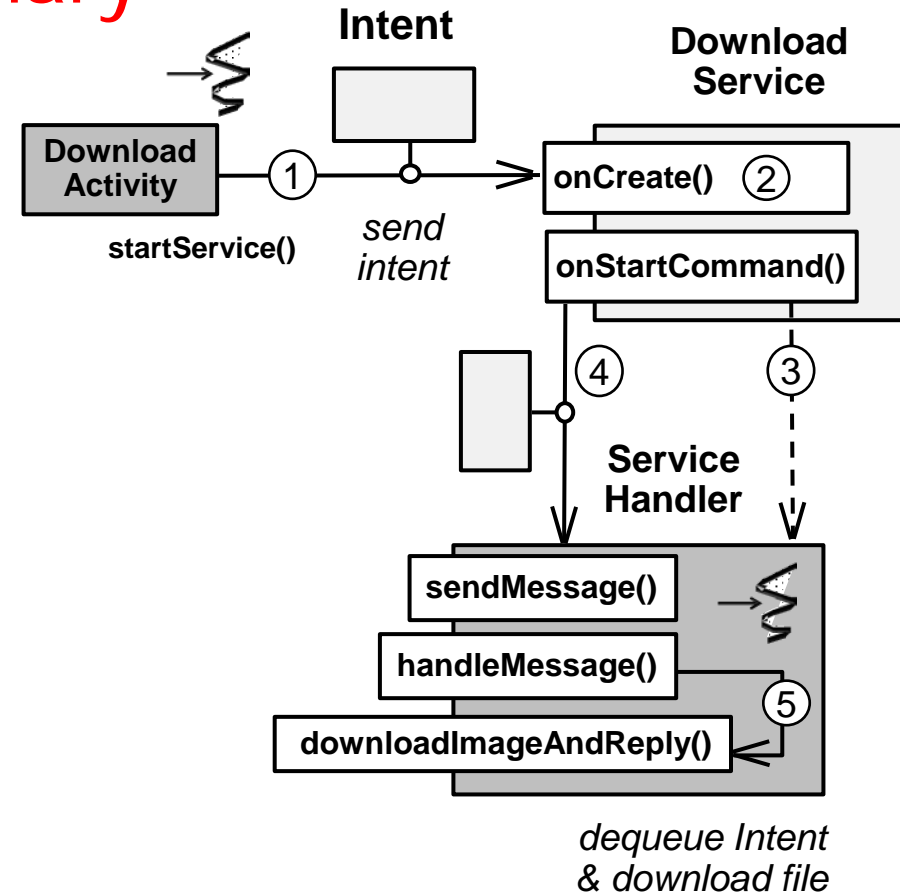
Summary

- The Download Activity & Download Service concurrently retrieve images from remote servers & display them
- A Service uses the UI Thread by default
- A Service is *not* a Thread
- Compute- or I/O-intensive Services should run in background Thread(s)



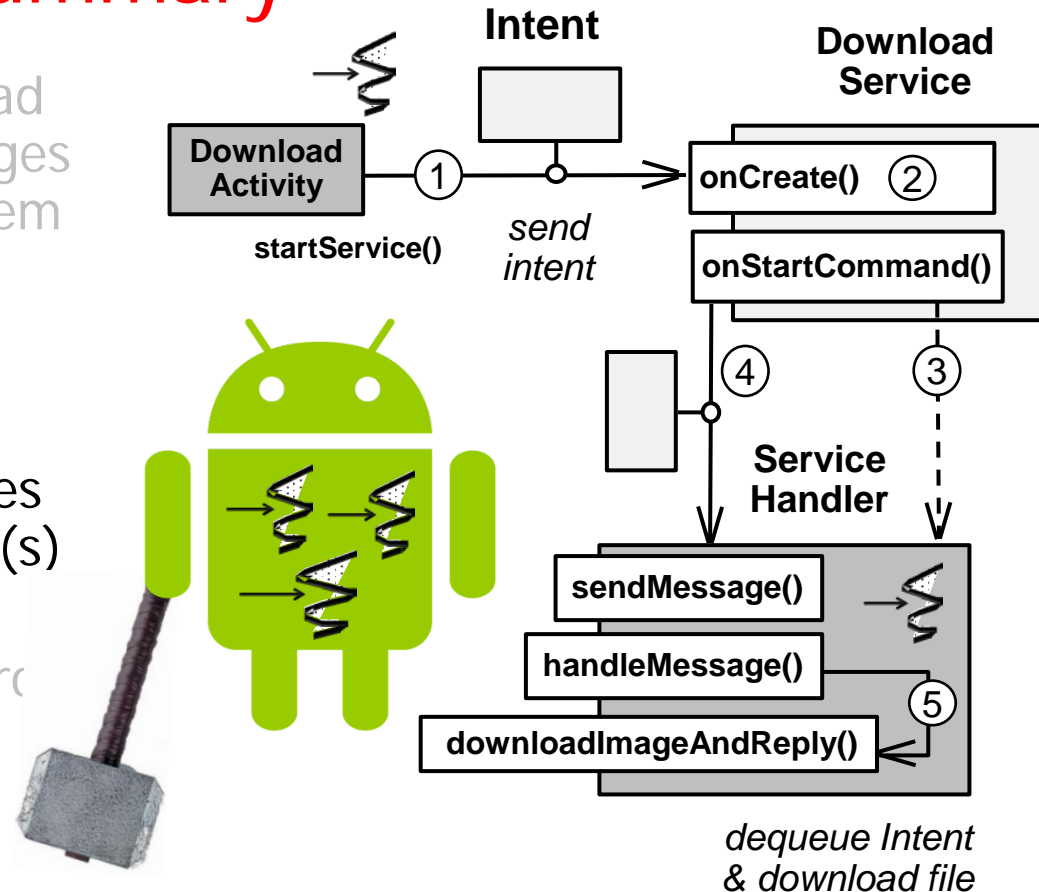
Summary

- The Download Activity & Download Service concurrently retrieve images from remote servers & display them
- A Service uses the UI Thread by default
- A Service is *not* a Thread
- Compute- or I/O-intensive Services should run in background Thread(s)
 - Service implementations often manually program various Android concurrency & communication mechanisms



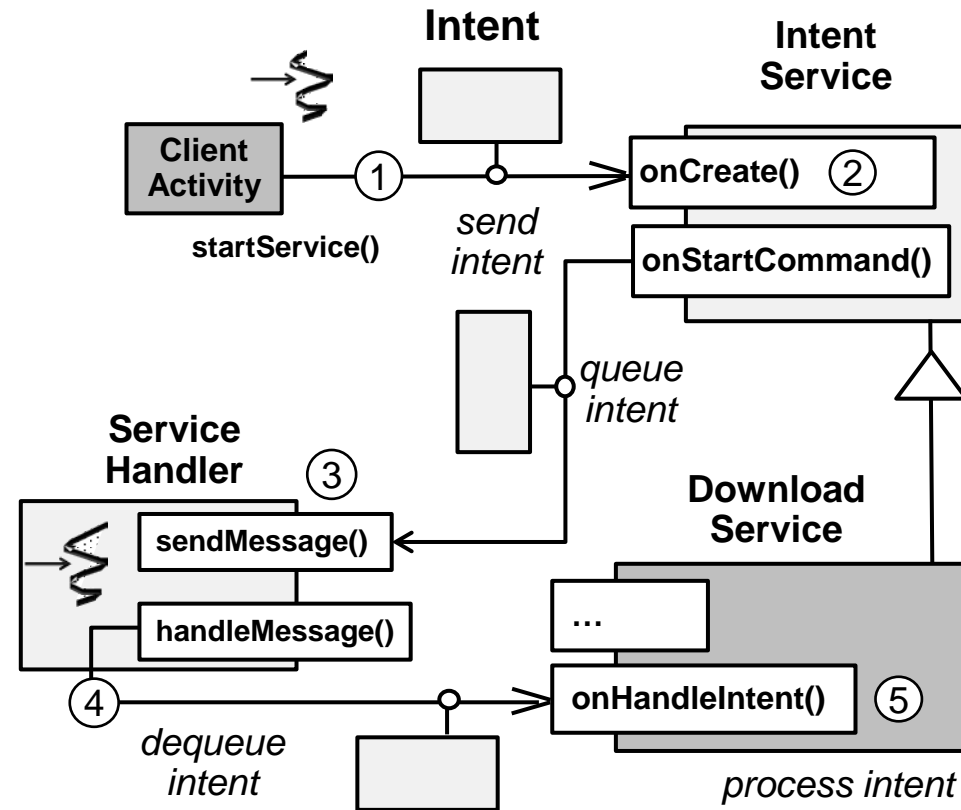
Summary

- The Download Activity & Download Service concurrently retrieve images from remote servers & display them
- A Service uses the UI Thread by default
- A Service is *not* a Thread
- Compute- or I/O-intensive Services should run in background Thread(s)
 - Service implementations often manually program various Android concurrency & communication mechanisms
- e.g., Download Service applies various classes from the Android HaMeR framework



Summary

- The Download Activity & Download Service concurrently retrieve images from remote servers & display them
- A Service uses the UI Thread by default
- A Service is *not* a Thread
- Compute- or I/O-intensive Services should run in background Thread(s)
- Android's IntentService codifies this concurrent Intent processing model in a reusable framework



Summary

- The Download Activity & Download Service concurrently retrieve images from remote servers & display them
- A Service uses the UI Thread by default
- A Service is *not* a Thread
- Compute- or I/O-intensive Services should run in background Thread(s)
- Android's IntentService codifies this concurrent Intent processing model in a reusable framework
 - It applies the *Command Processor* pattern to handle asynchronous commands expressed as Intents in a single background thread

