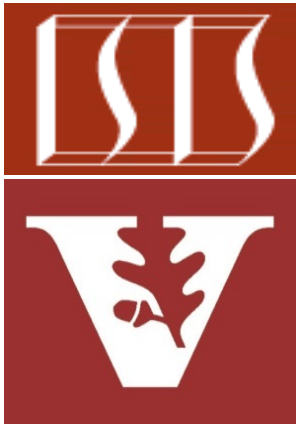
Concurrent Programming with Android Intents

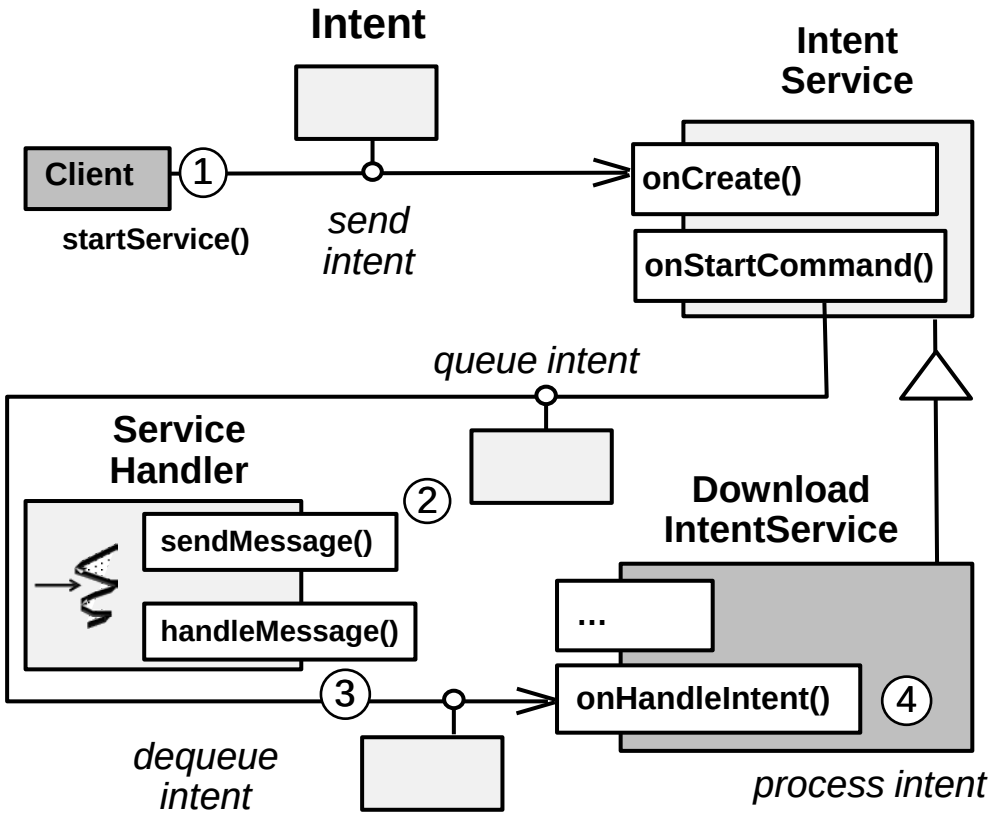


Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Institute for Software Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA

Learning Objectives in this Lesson

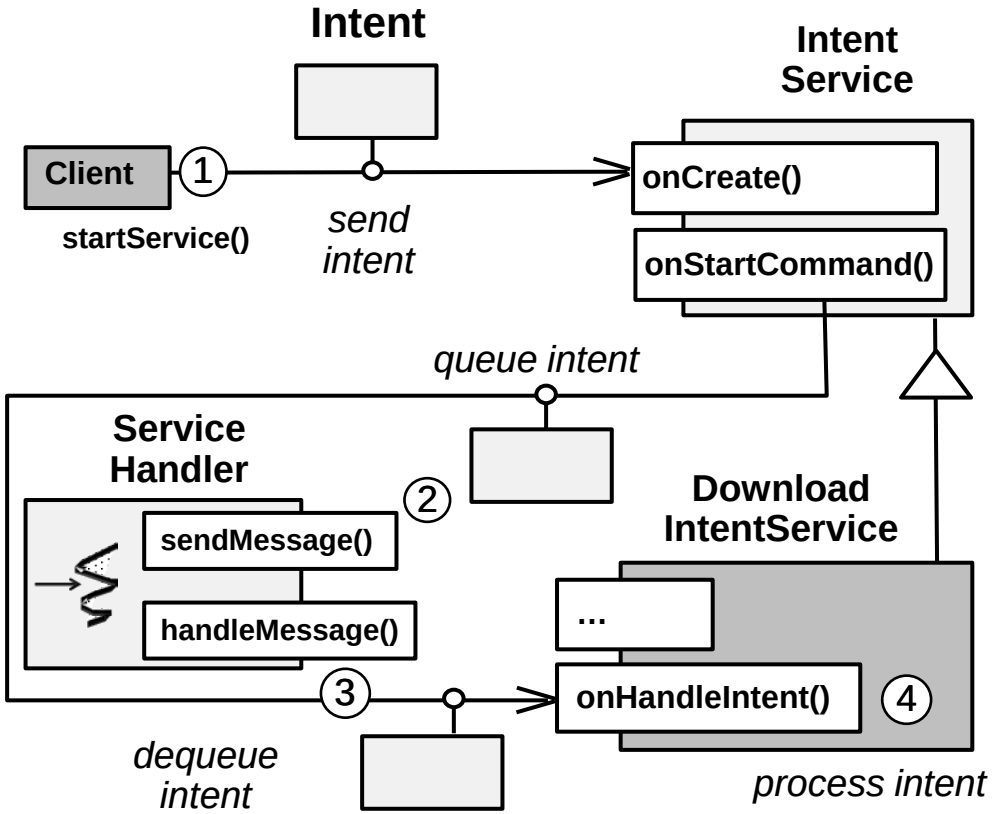
1. Recognize how intents can be used in concurrent apps



See blog.vaporstream.com/blog/2013/07/09/concurrency-patterns-intent-service

Learning Objectives in this Lesson

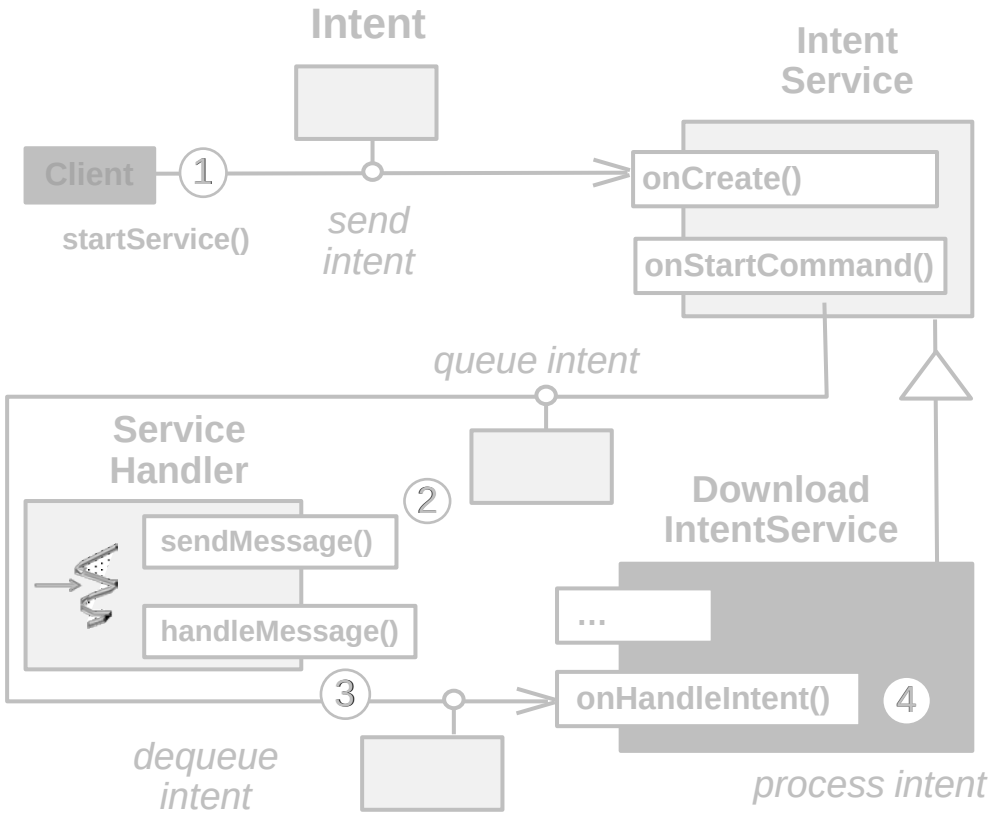
1. Recognize how intents can be used in concurrent apps
2. Know how intents are used to program parts of a concurrent ImageDownloader app



See gitlab.com/vandy-aad-3/ImageDownloader

Learning Objectives in this Lesson

1. Recognize how intents can be used in concurrent apps
2. Know how intents are used to program parts of a concurrent ImageDownloader app

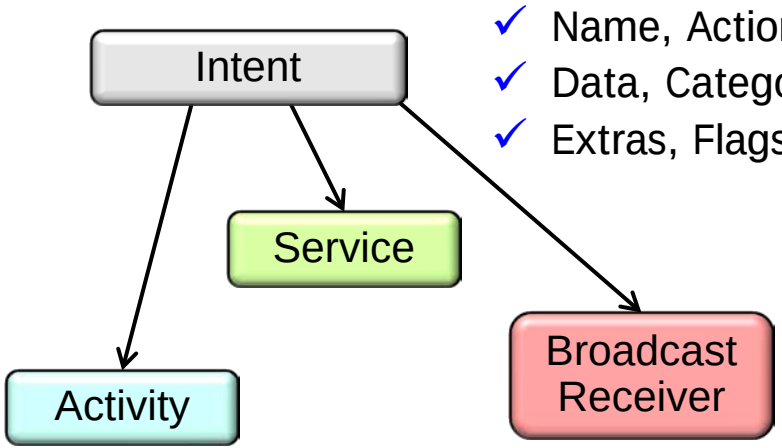


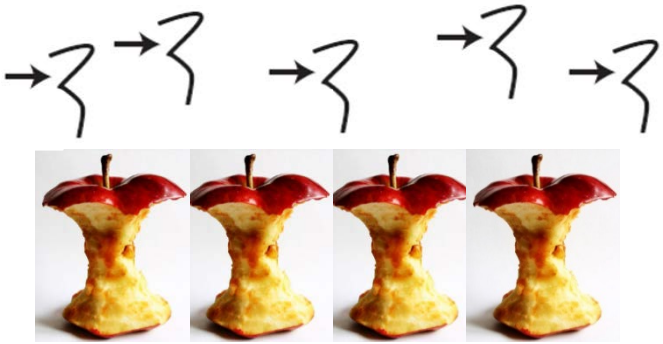
There are forward references in this lesson, so don't get hung up on details now!

Concurrency Considerations for Android Intents

Concurrency Considerations for Android Intents

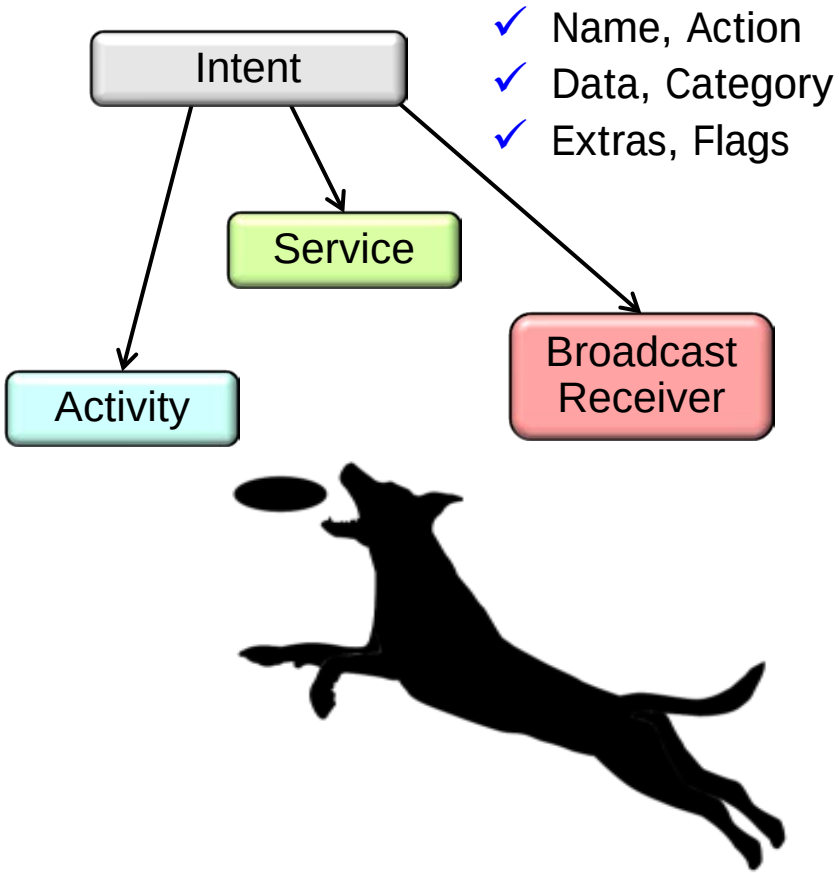
- Intents can be applied in a range of concurrency use cases





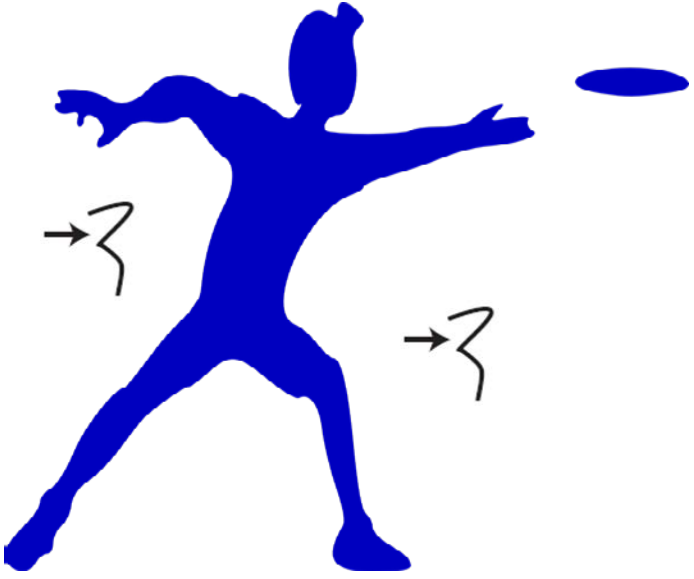
Concurrency Considerations for Android Intents

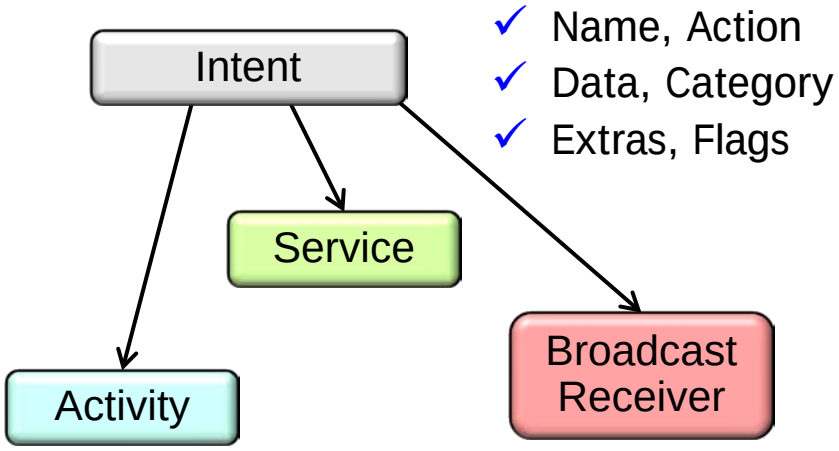
- Intents can be applied in a range of concurrency use cases, e.g.
  - They can be passed as “messages” from one thread/process to another



Concurrency Considerations for Android Intents

- Intents can be applied in a range of concurrency use cases, e.g.
  - They can be passed as “messages” from one thread/process to another



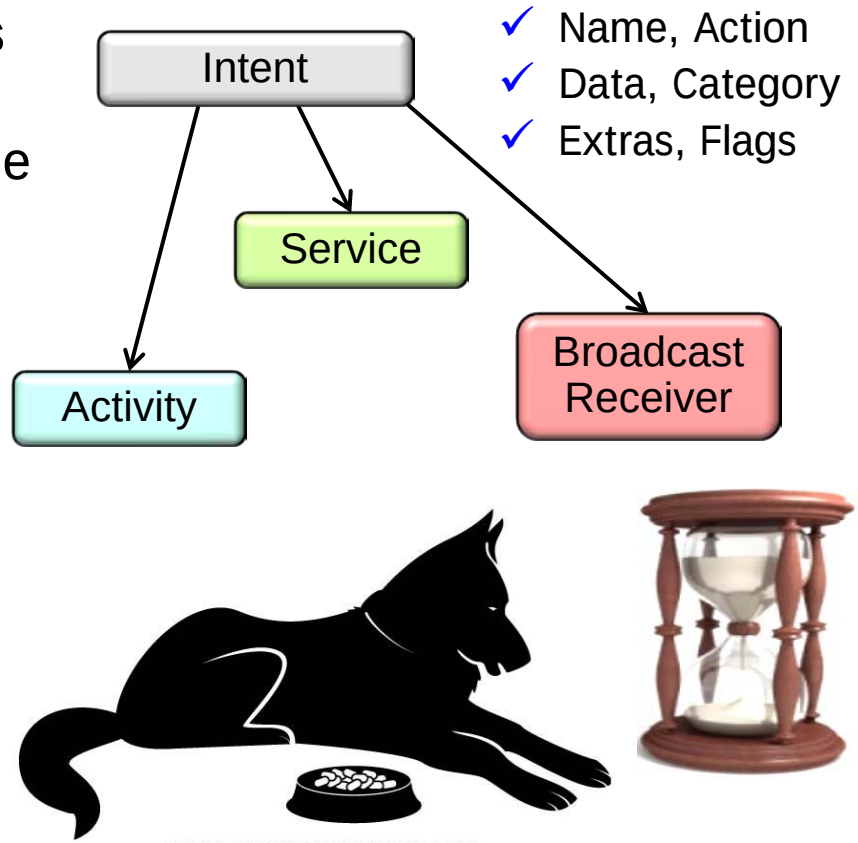


- ✓ Name, Action
- ✓ Data, Category
- ✓ Extras, Flags

An intent’s target component(s) typically runs concurrently wrt the intent sender

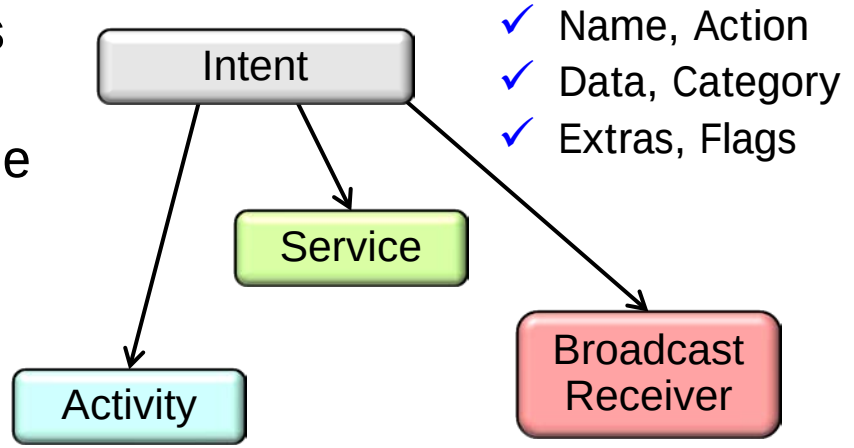
Concurrency Considerations for Android Intents

- Android & Java concurrency mechanisms & frameworks should be used if target component(s) block or run for a long time



Concurrency Considerations for Android Intents

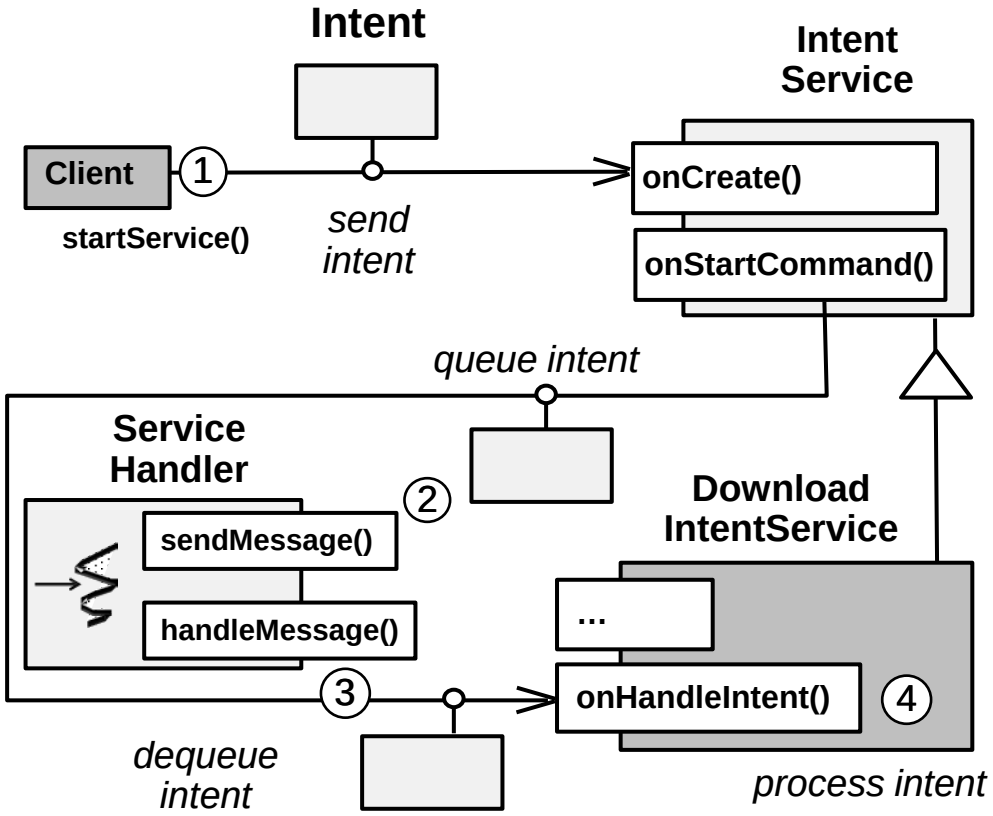
- Android & Java concurrency mechanisms & frameworks should be used if target component(s) block or run for a long time
  - Needed to avoid the “application not responding” dialog



See developer.android.com/training/articles/perf-anr.html

Concurrency Considerations for Android Intents

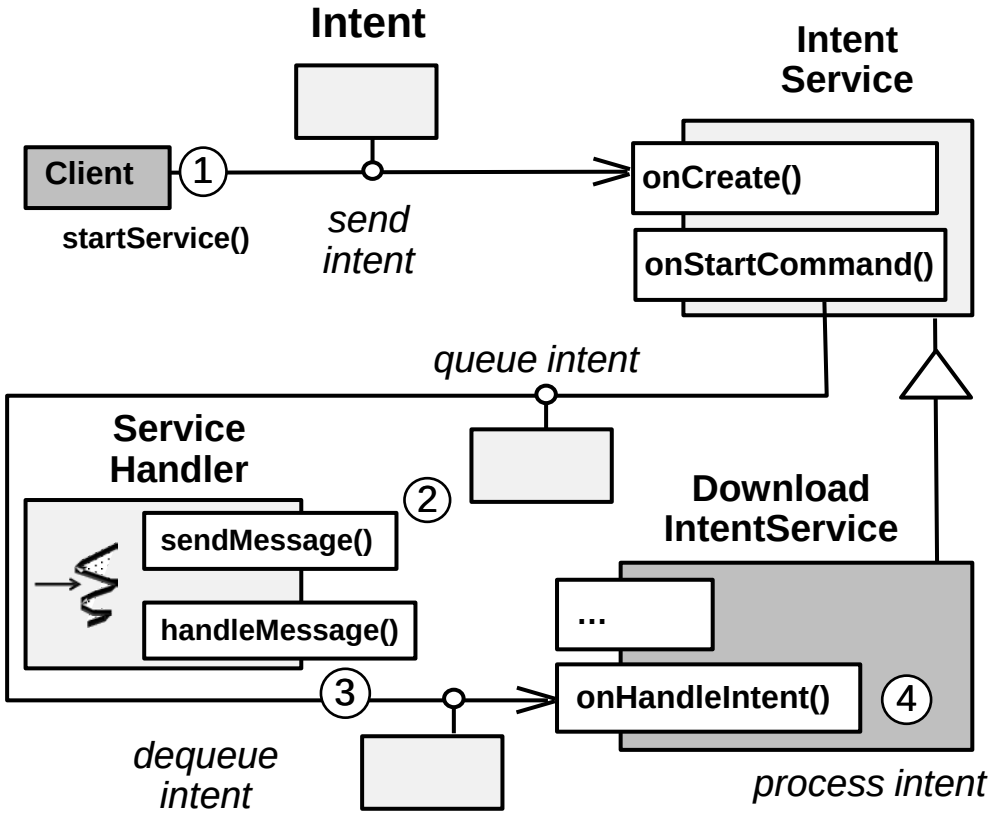
- Android's IntentService provides a reusable component for running intents concurrently



See developer.android.com/reference/android/app/IntentService.html

Concurrency Considerations for Android Intents

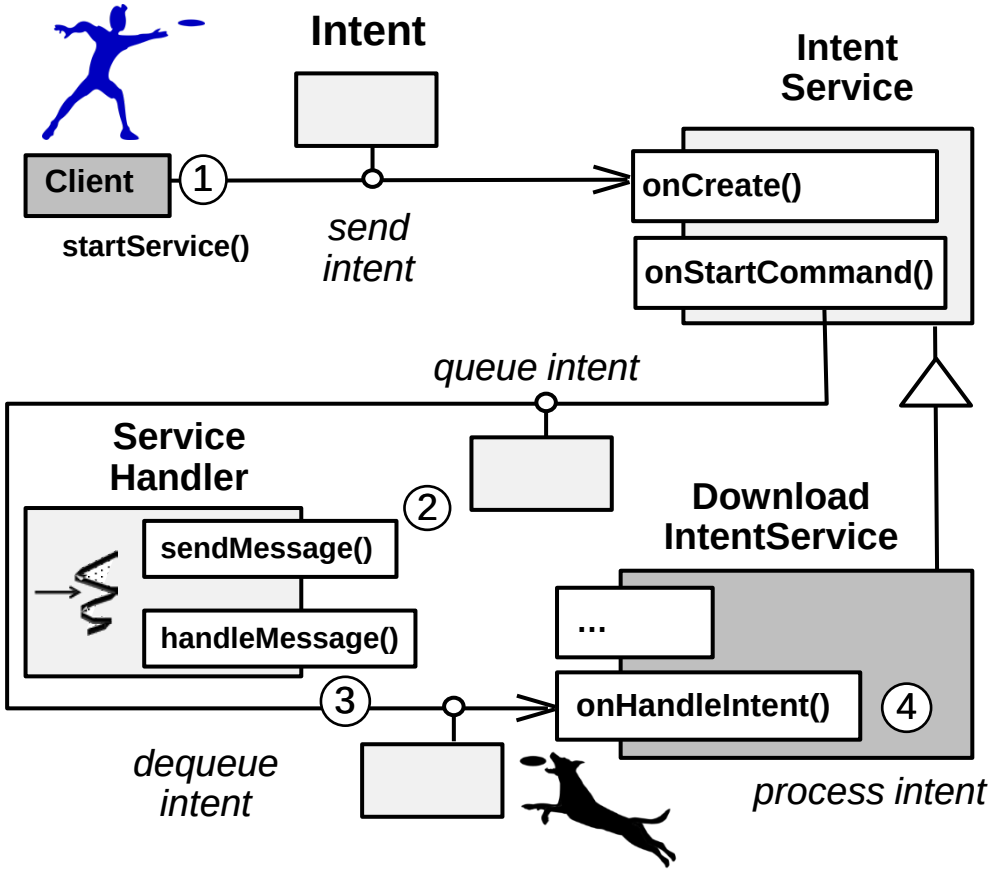
- Android's IntentService provides a reusable component for running intents concurrently
  - IntentService is a framework that handles asynchronous requests on demand



We'll discuss the Android IntentService framework further a later course

Concurrency Considerations for Android Intents

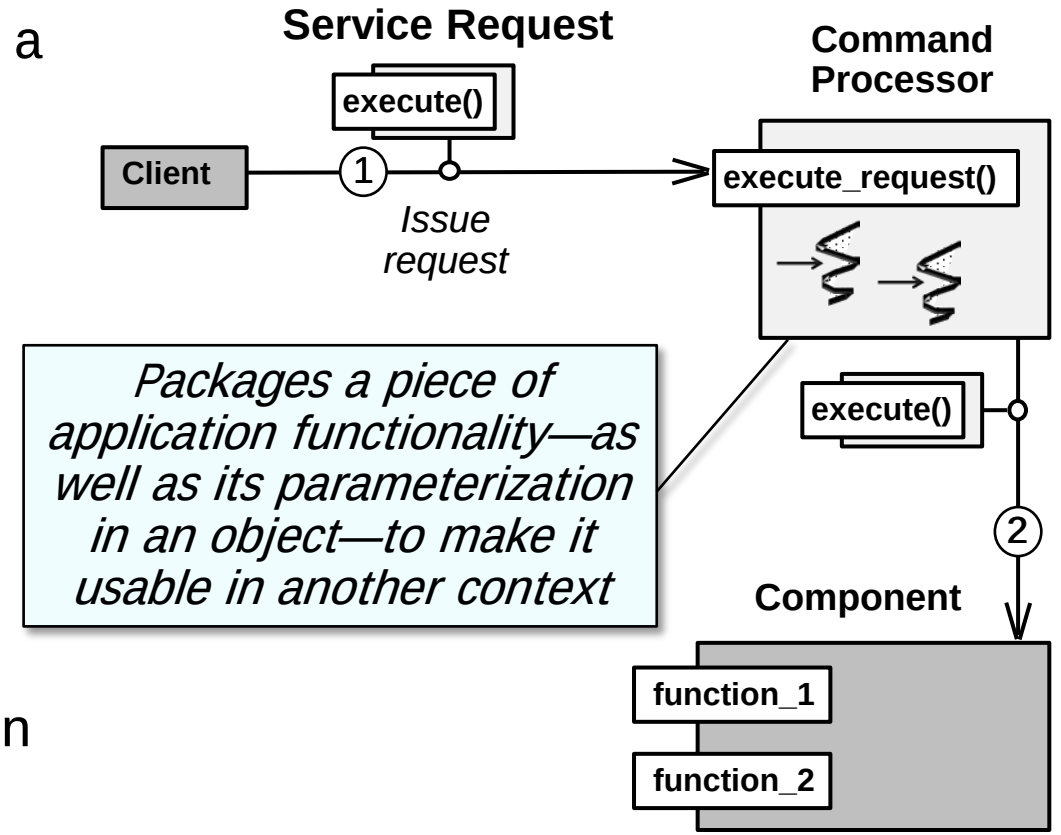
- Android's IntentService provides a reusable component for running intents concurrently
  - IntentService is a framework that handles asynchronous requests on demand
  - These requests are expressed via intents & passed between threads and/or processes



See blog.vaporstream.com/blog/2013/07/09/concurrency-patterns-intent-service

Concurrency Considerations for Android Intents

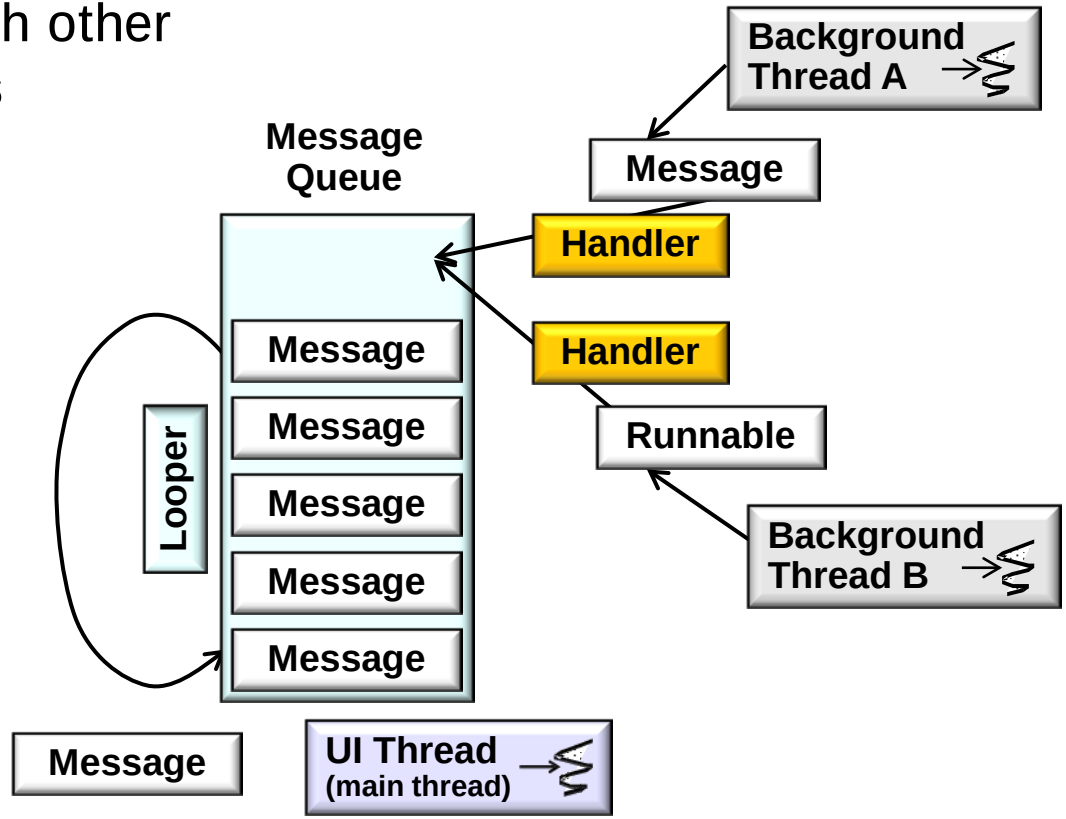
- Android's IntentService provides a reusable component for running intents concurrently
  - IntentService is a framework that handles asynchronous requests on demand
  - These requests are expressed via intents & passed between threads and/or processes
  - The IntentService implements the *CommandProcessor* pattern



See www.dre.vanderbilt.edu/~schmidt/PDF/CommandProcessor.pdf

Concurrency Considerations for Android Intents

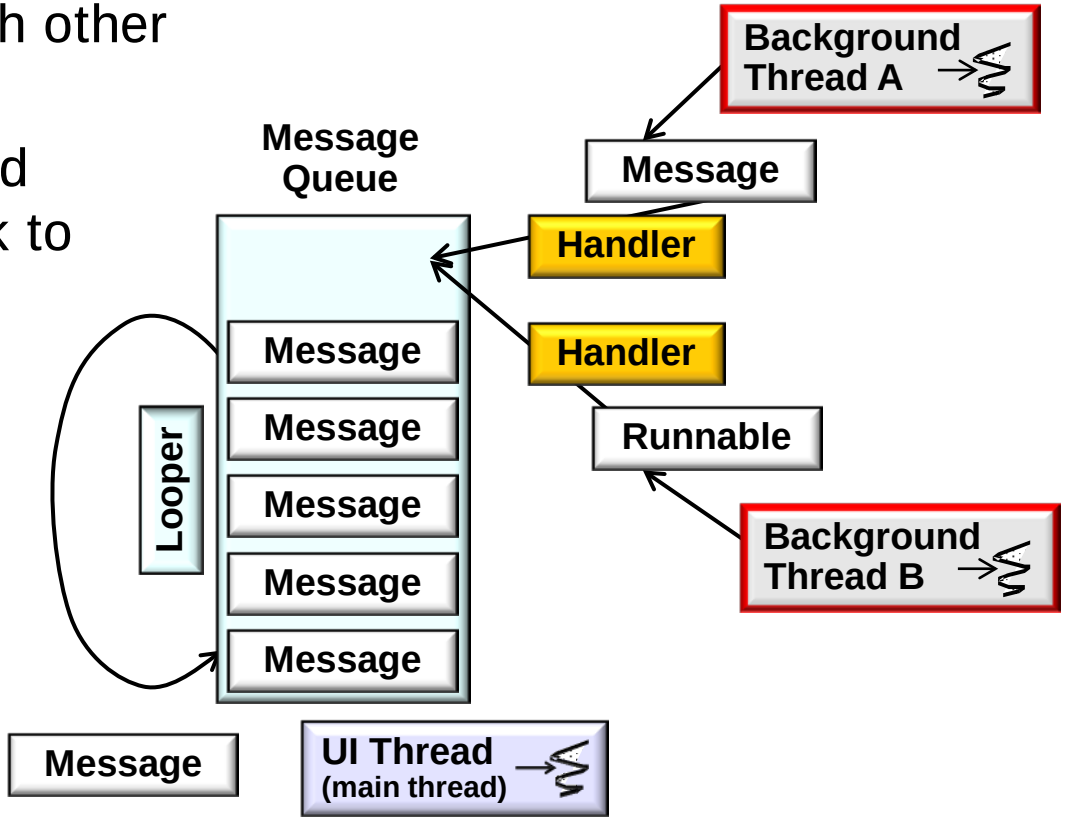
- You can also combine intents with other Android concurrency frameworks



See developer.android.com/guide/components/processes-and-threads.html#Threads

Concurrency Considerations for Android Intents

- You can also combine intents with other Android concurrency frameworks
  - e.g., apps often use the Android HaMeR concurrency framework to pass intents between threads

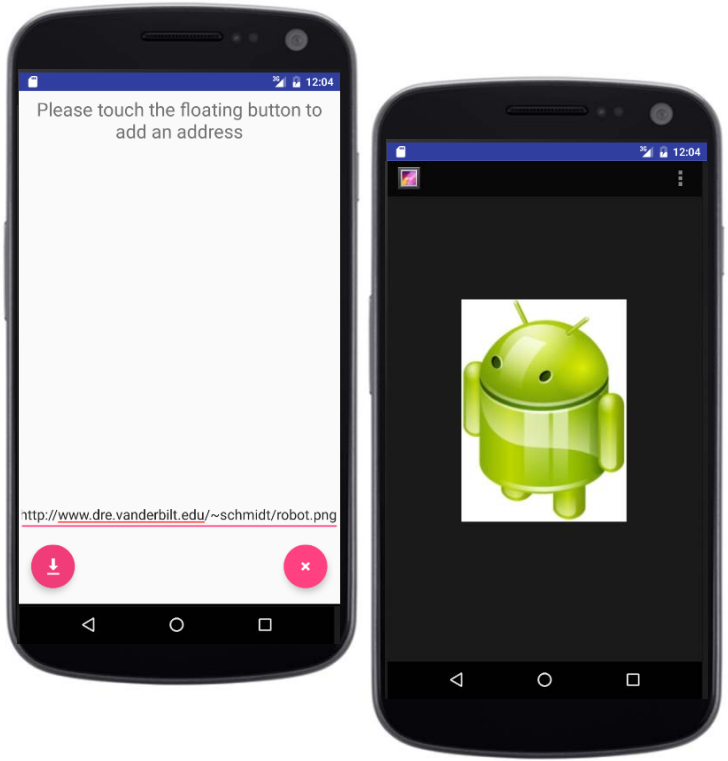


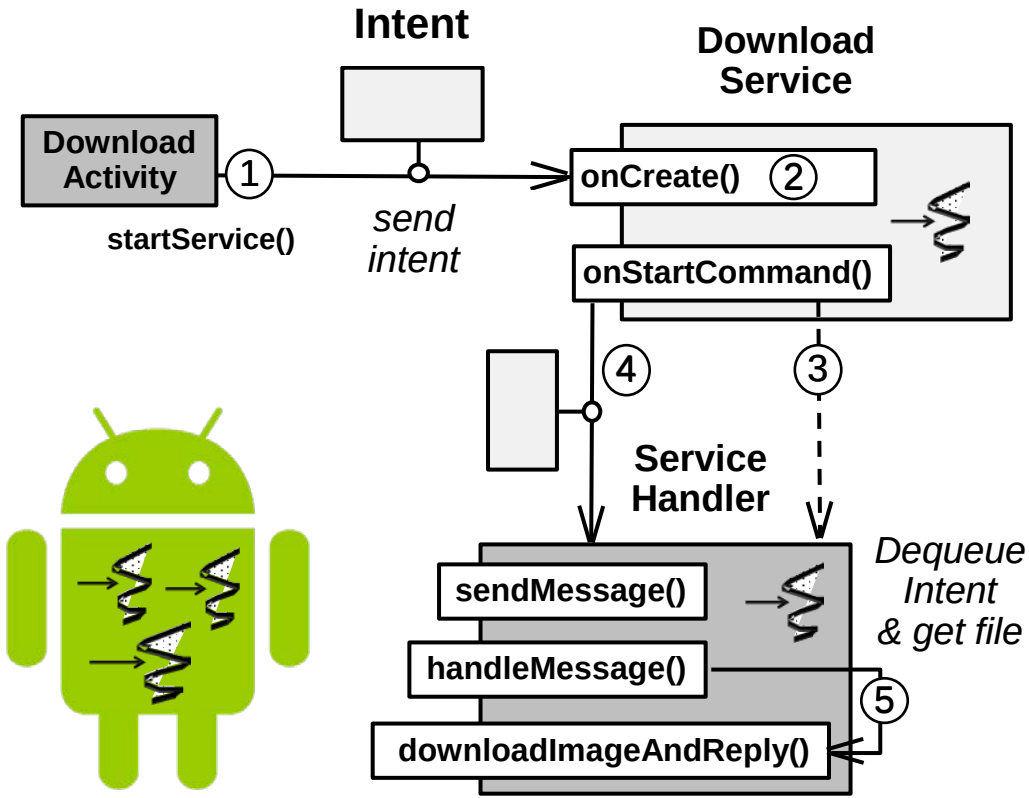
We'll discuss more about the HaMeR concurrency framework shortly..

Example of Concurrent Programming with Intents

Example of Concurrent Programming with Intents

- The ImageDownloader app codifies a common Android concurrency idiom

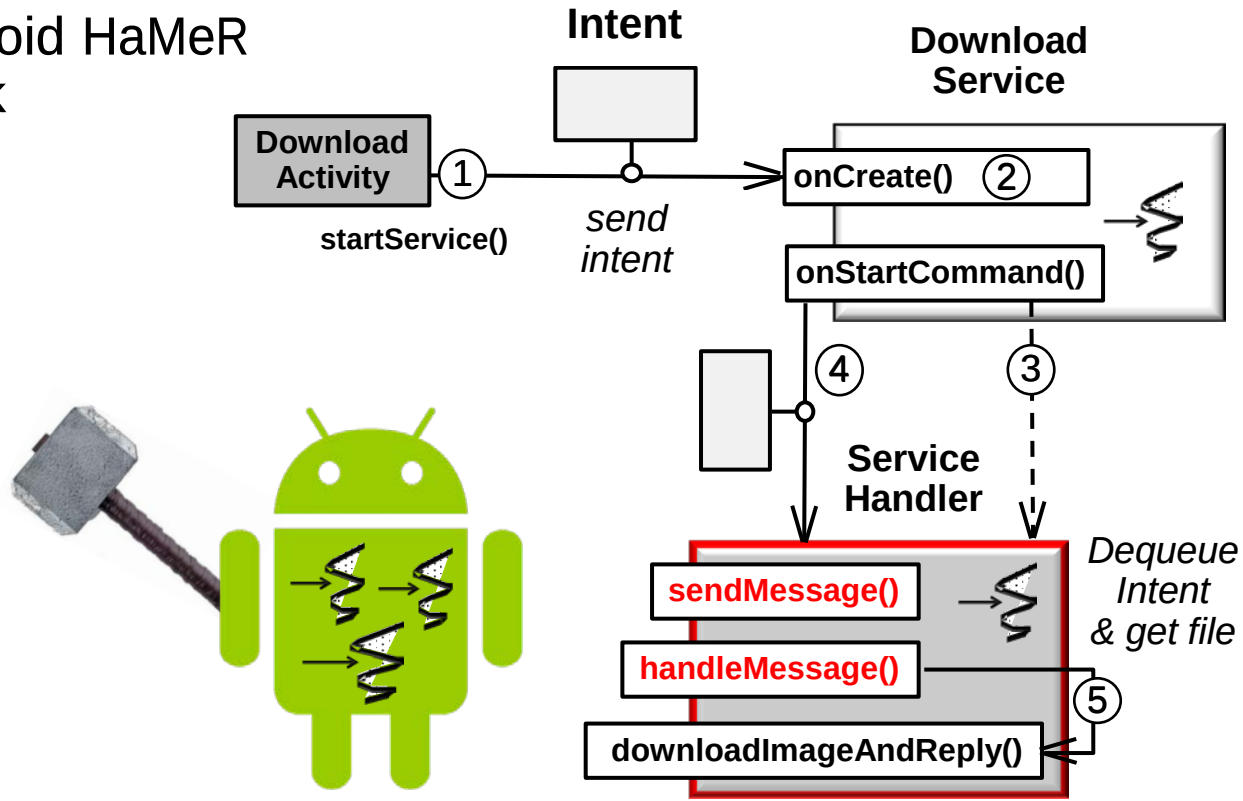




See gitlab.com/vandy-aad-3/ImageDownloader

Example of Concurrent Programming with Intents

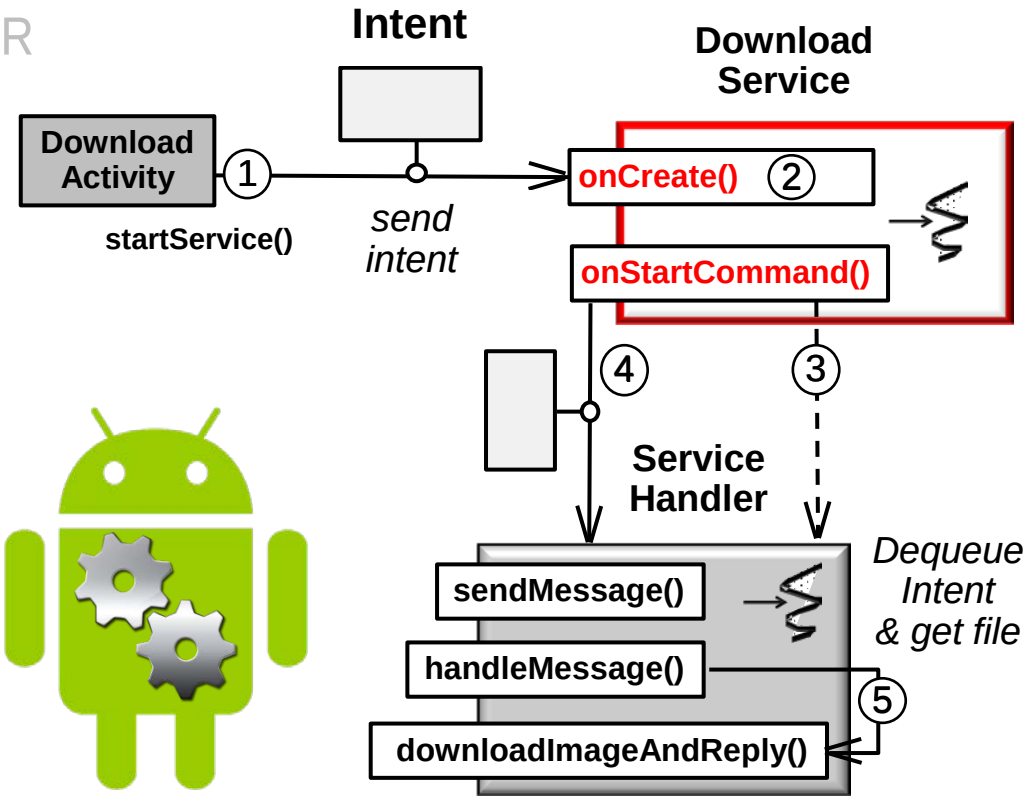
- The ImageDownloader app codifies a common Android concurrency idiom
  - This app uses the Android HaMeR concurrency framework



See developer.android.com/guide/components/processes-and-threads.html#Threads

Example of Concurrent Programming with Intents

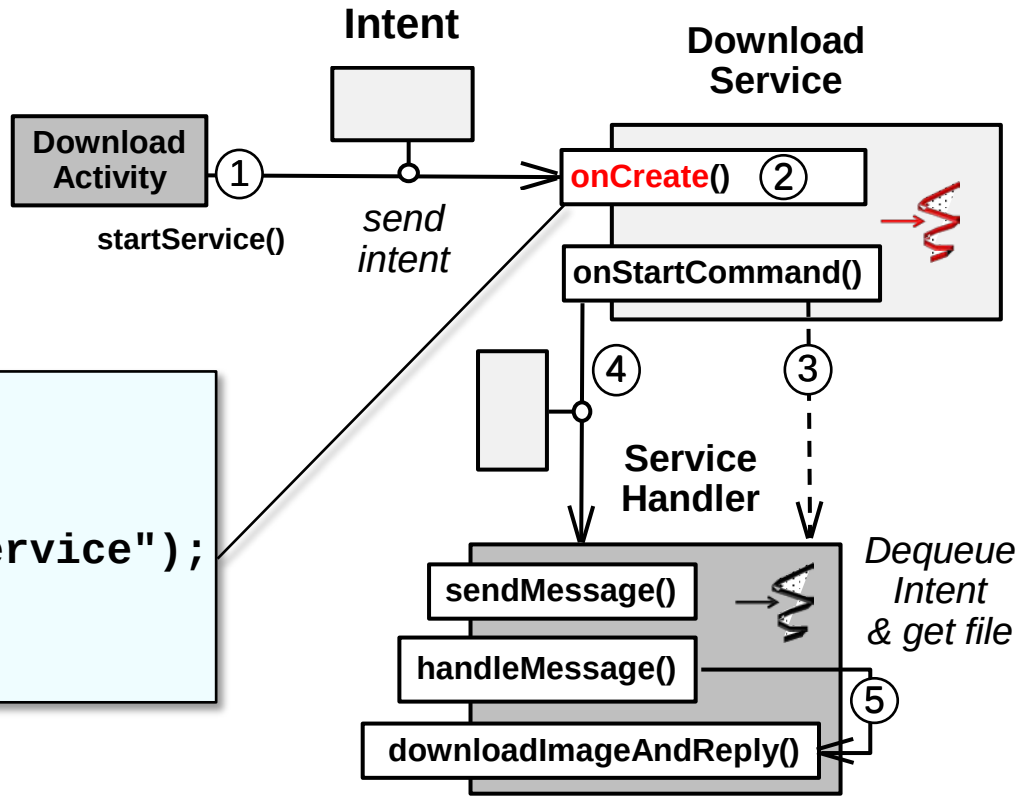
- The ImageDownloader app codifies a common Android concurrency idiom
  - This app uses the Android HaMeR concurrency framework
  - It also uses an Android Service component



See developer.android.com/reference/android/app/Service.html

Example of Concurrent Programming with Intents

- This method in ImageDownloader creates a HandlerThread & ServiceHandler
  - DownloadService.onCreate()
    - Runs in the main thread

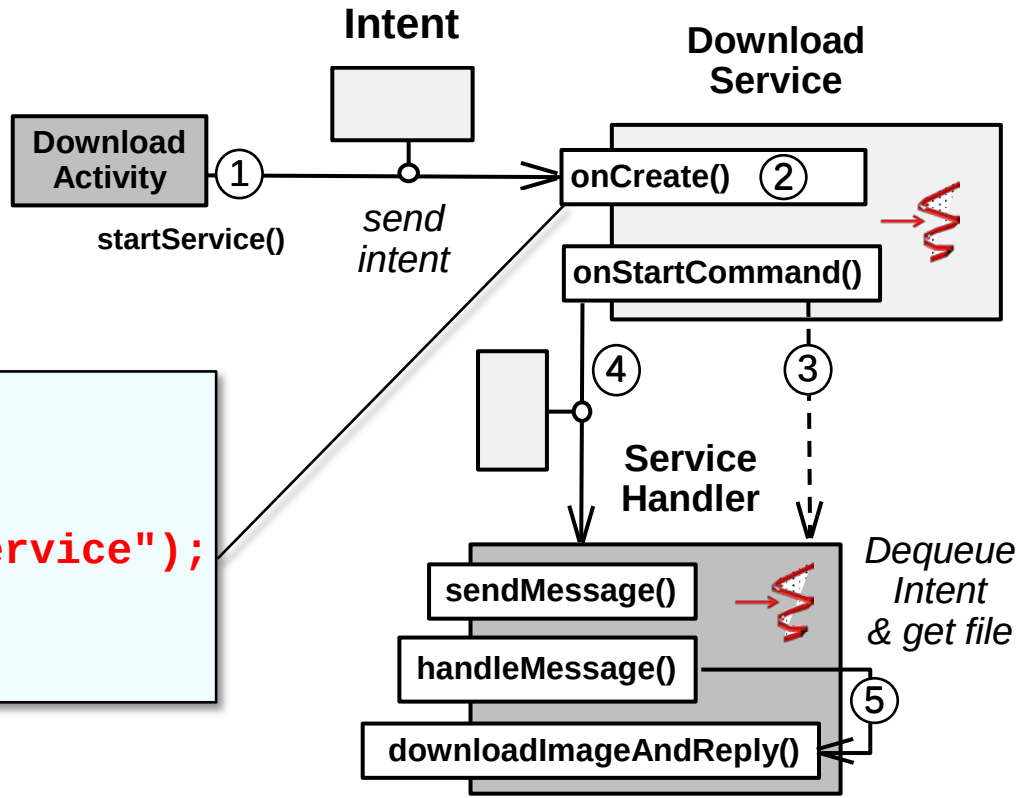



```
void onCreate() {
  ...
  HandlerThread thread = new
    HandlerThread("DownloadService");
  thread.start();
  ...
```

See developer.android.com/reference/android/app/Service.html#onCreate()

Example of Concurrent Programming with Intents

- This method in ImageDownloader creates a HandlerThread & ServiceHandler
  - DownloadService.onCreate()
    1. Runs in the main thread
    2. Creates/starts HandlerThread



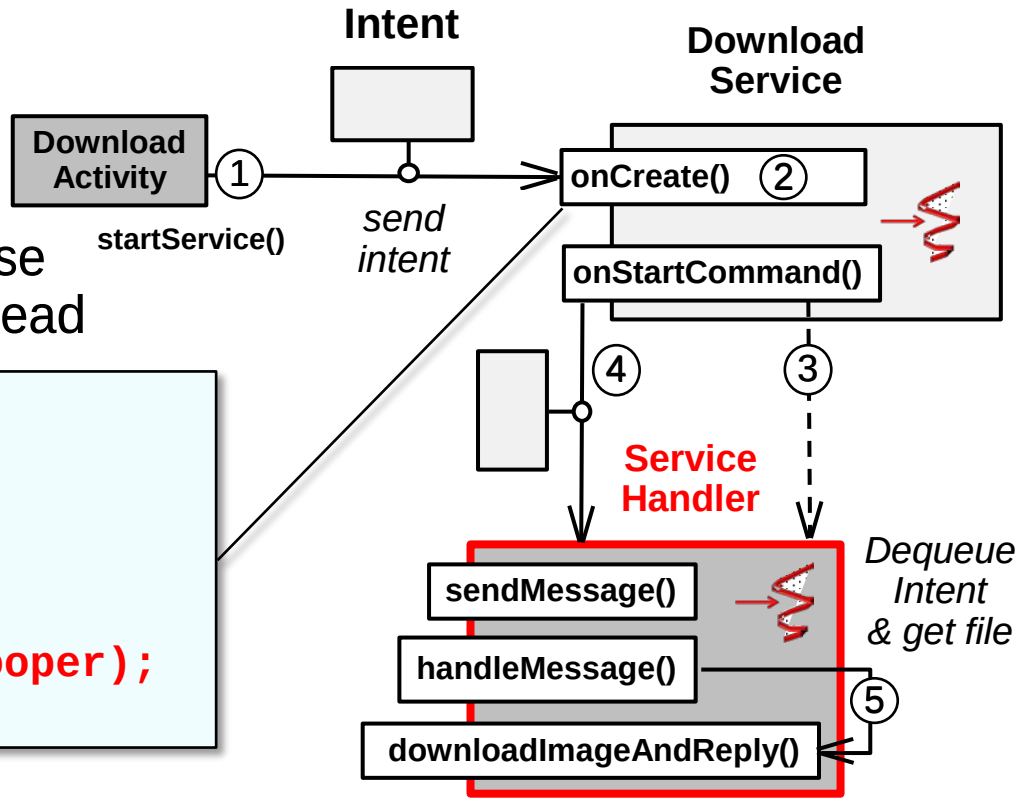

```
void onCreate() {
  ...
  HandlerThread thread = new
    HandlerThread("DownloadService");
  thread.start();
  ...
```

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

Example of Concurrent Programming with Intents

- This method in ImageDownloader creates a HandlerThread & ServiceHandler
  - DownloadService.onCreate()
    1. Runs in the main thread
    2. Creates/starts HandlerThread
    3. Creates a ServiceHandler whose Looper runs in the HandlerThread

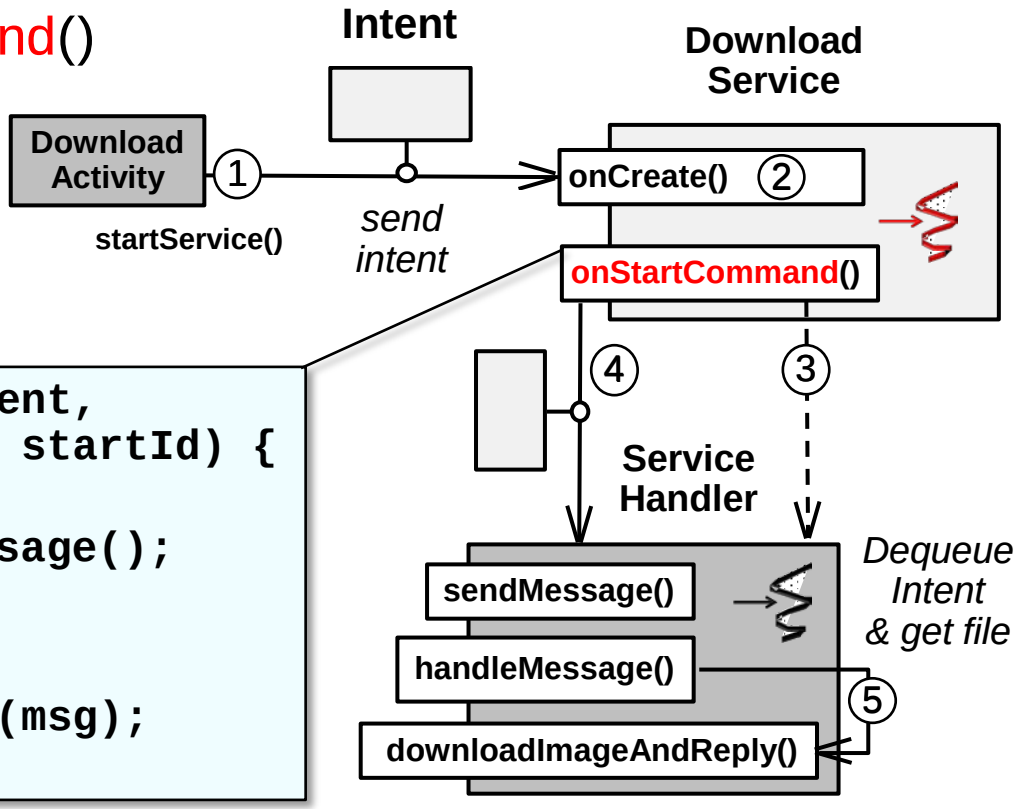
```
void onCreate() {
  ...
  mServiceLooper =
    thread.getLooper();
  mServiceHandler = new
    ServiceHandler(mServiceLooper);
  ...
```



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

Example of Concurrent Programming with Intents

- These methods in ImageDownloader receive & process intents concurrently
  - DownloadService.onStartCommand()

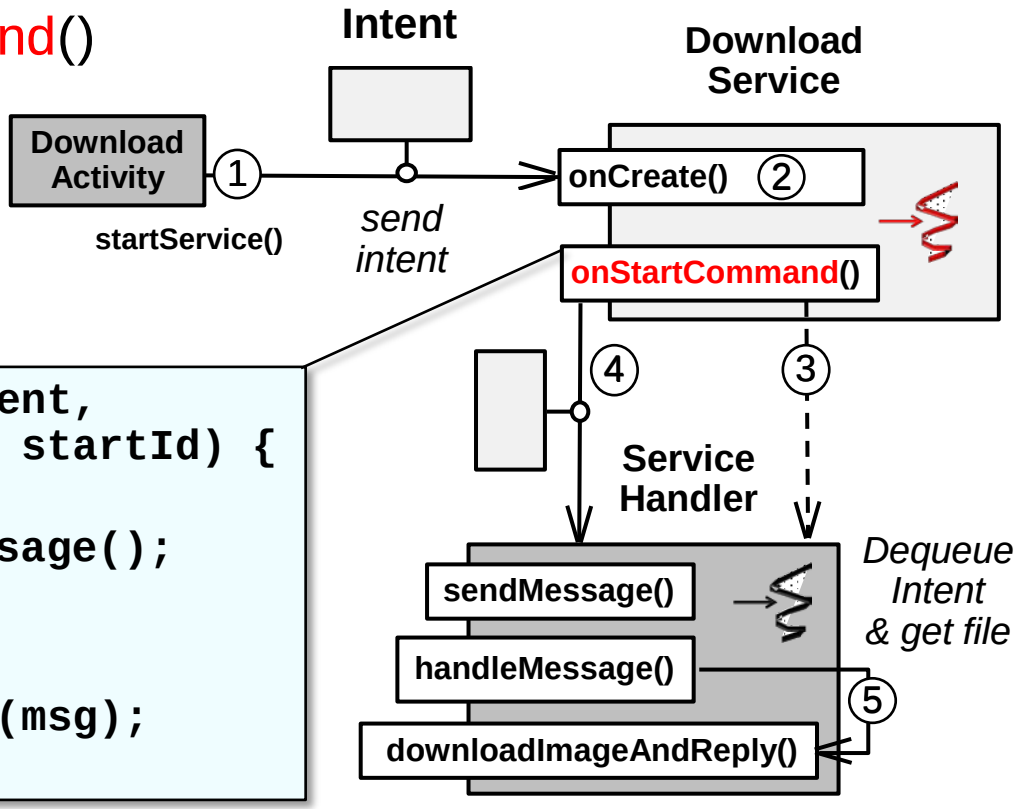



```
int onStartCommand(Intent intent,
                   int f, int startId) {
  Message msg =
    mServiceHandler.obtainMessage();
  msg.arg1 = startId;
  msg.obj = intent;
  mServiceHandler.sendMessage(msg);
  ...
```

See developer.android.com/reference/android/app/Service.html#onStartCommand(android.content.Intent, int, int)

Example of Concurrent Programming with Intents

- These methods in ImageDownloader receive & process intents concurrently
  - DownloadService.onStartCommand()
    - Runs in the main thread

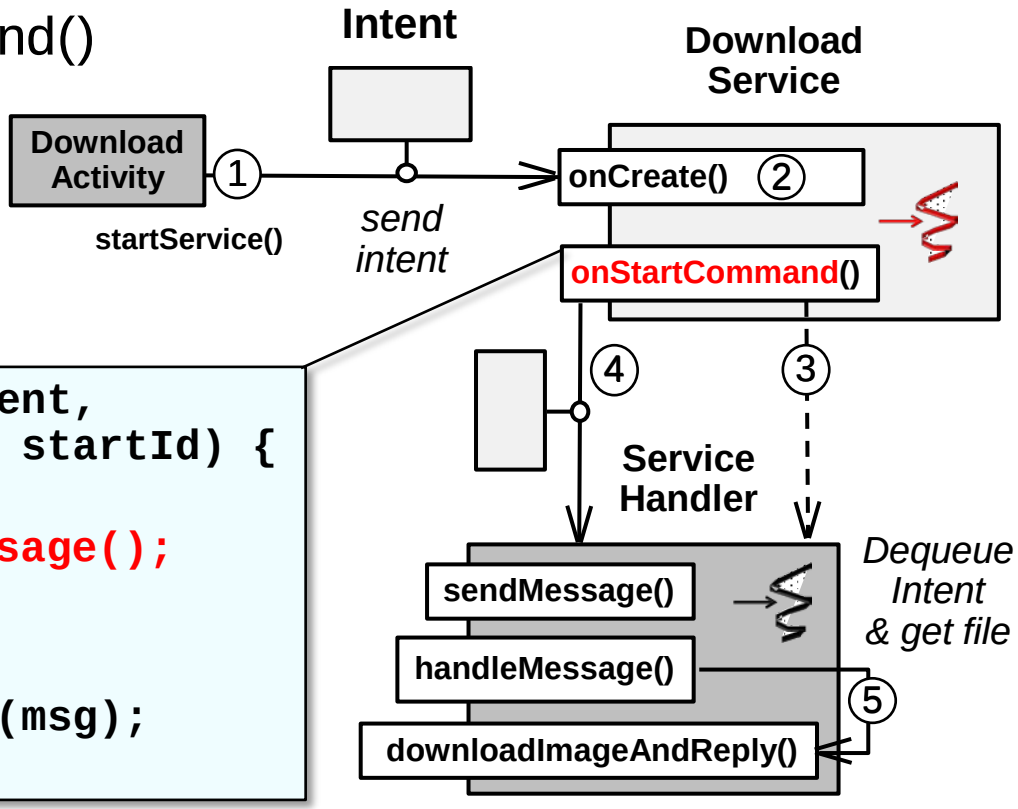



```
int onStartCommand(Intent intent,
                   int f, int startId) {
  Message msg =
    mServiceHandler.obtainMessage();
  msg.arg1 = startId;
  msg.obj = intent;
  mServiceHandler.sendMessage(msg);
  ...
```

See developer.android.com/reference/android/app/Service.html#onStartCommand(android.content.Intent, int, int)

Example of Concurrent Programming with Intents

- These methods in ImageDownloader receive & process intents concurrently
  - DownloadService.onStartCommand()
    1. Runs in the main thread
    2. Create a Message that encapsulates the Intent

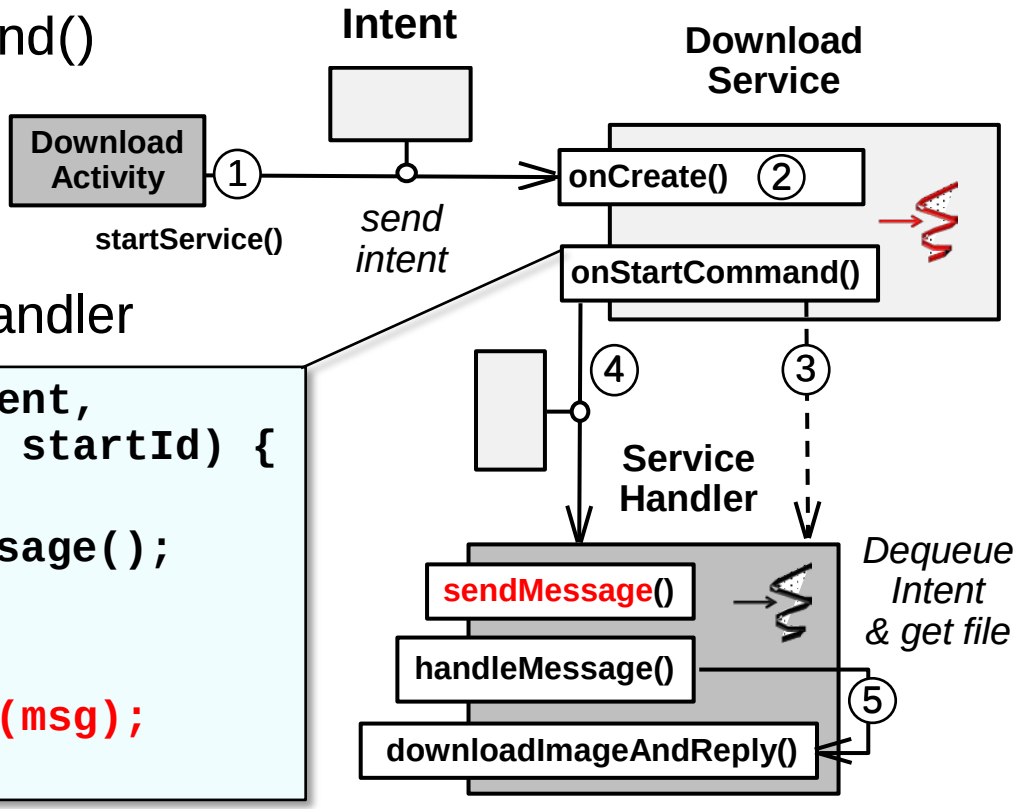



```
int onStartCommand(Intent intent,
                   int f, int startId) {
  Message msg =
    mServiceHandler.obtainMessage();
  msg.arg1 = startId;
  msg.obj = intent;
  mServiceHandler.sendMessage(msg);
  ...
```

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

Example of Concurrent Programming with Intents

- These methods in ImageDownloader receive & process intents concurrently
  - DownloadService.onStartCommand()
    1. Runs in the main thread
    2. Create a Message that encapsulates the Intent
    3. Send Message to the ServiceHandler

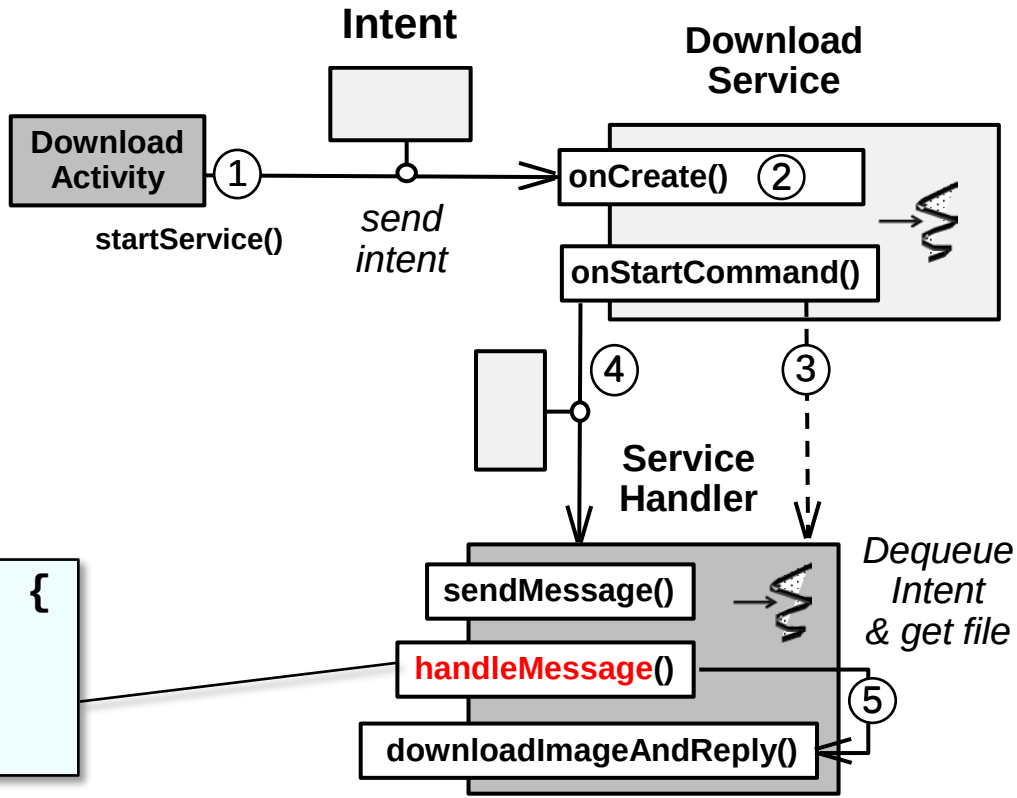



```
int onStartCommand(Intent intent,
                   int f, int startId) {
  Message msg =
    mServiceHandler.obtainMessage();
  msg.arg1 = startId;
  msg.obj = intent;
  mServiceHandler.sendMessage(msg);
  ...
```

See developer.android.com/reference/android/os/Handler.html#sendMessage(android.os.Message)

Example of Concurrent Programming with Intents

- These methods in ImageDownloader receive & process intents concurrently
  - ServiceHandler.handleMessage()

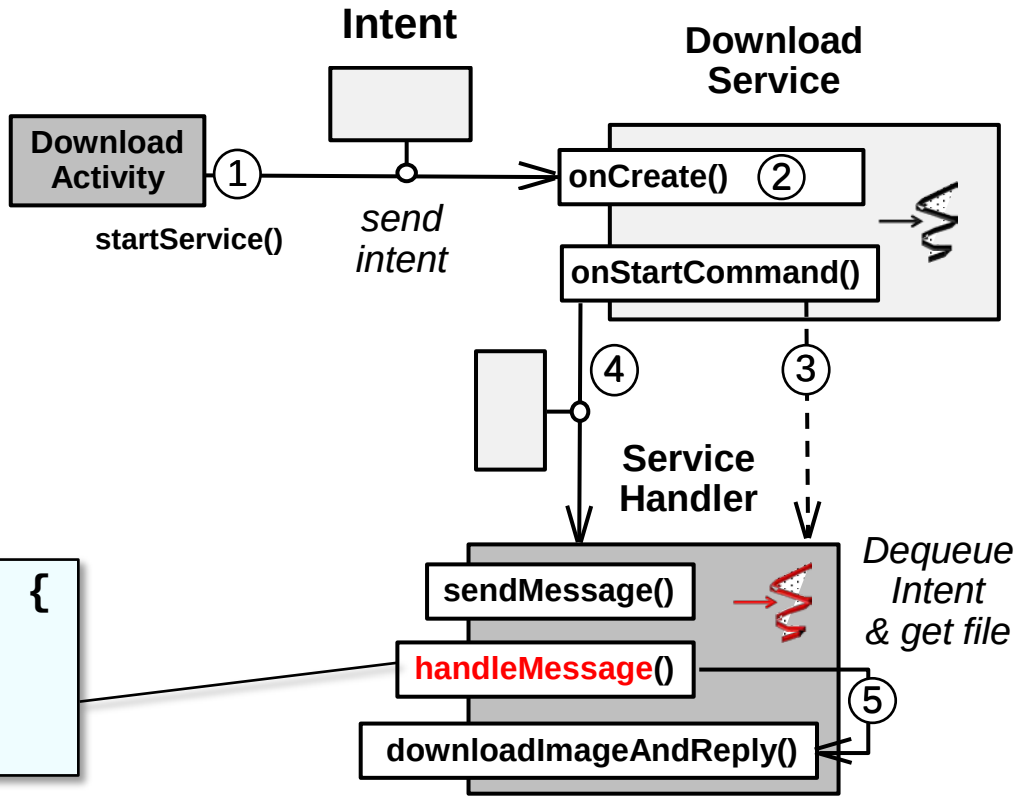



```
void handleMessage(Message msg) {
  downloadImageAndReply
    ((Intent) msg.obj);
  stopSelf(msg.arg1);
```

See developer.android.com/reference/android/os/Handler.html#handleMessage(android.os.Message)

Example of Concurrent Programming with Intents

- These methods in ImageDownloader receive & process intents concurrently
  - ServiceHandler.handleMessage()
    1. Runs in HandlerThread

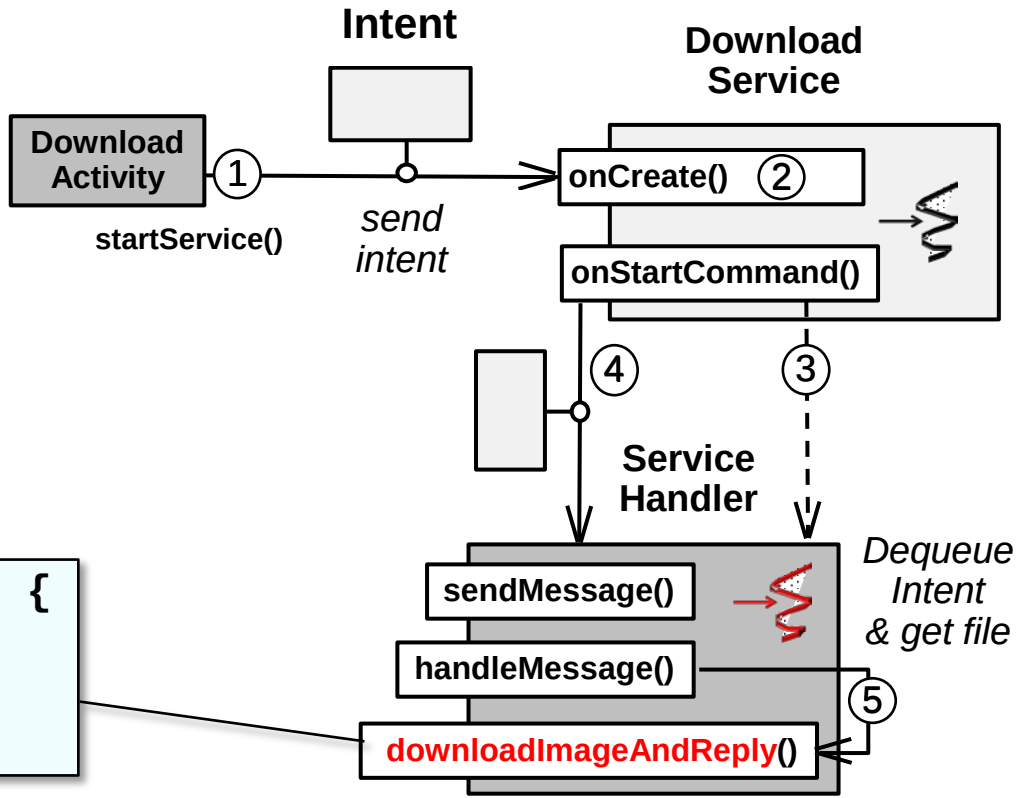



```
void handleMessage(Message msg) {
  downloadImageAndReply
    ((Intent) msg.obj);
  stopSelf(msg.arg1);
```

See developer.android.com/reference/android/os/Handler.html#handleMessage(android.os.Message)

Example of Concurrent Programming with Intents

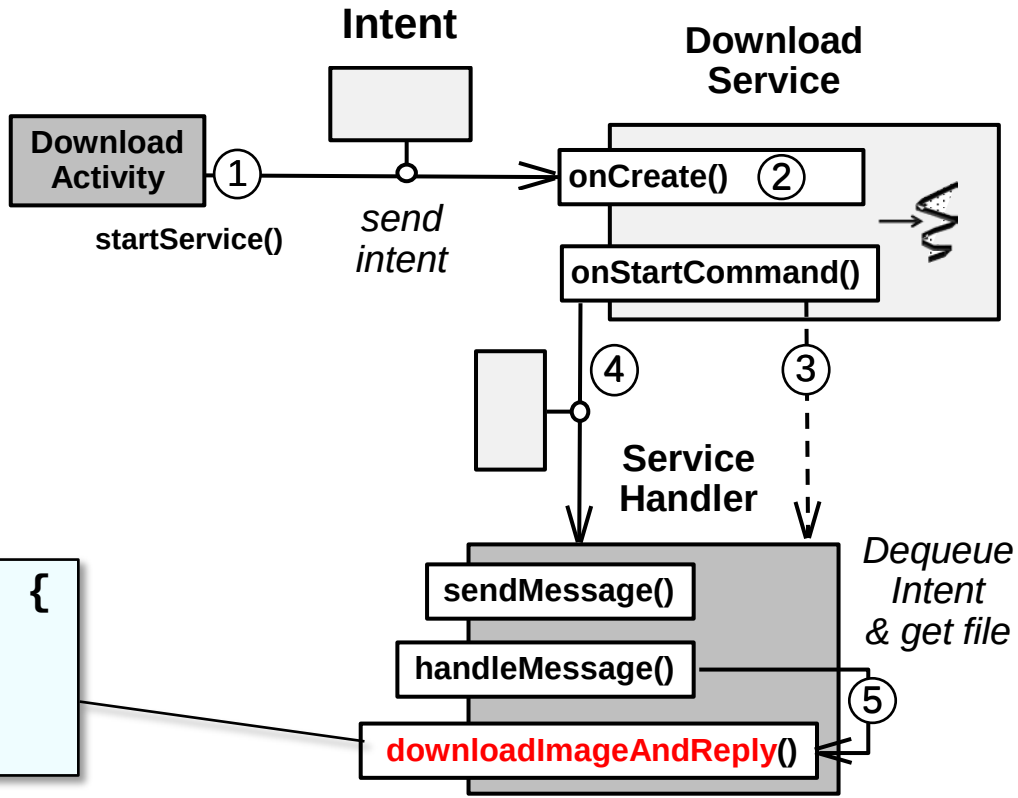
- These methods in ImageDownloader receive & process intents concurrently
  - ServiceHandler.handleMessage()
    1. Runs in HandlerThread
    2. Processes the Message containing the Intent




```
void handleMessage(Message msg) {
  downloadImageAndReply
    ((Intent) msg.obj);
  stopSelf(msg.arg1);
```

Example of Concurrent Programming with Intents

- These methods in ImageDownloader receive & process intents concurrently
  - ServiceHandler.handleMessage()
    1. Runs in HandlerThread
    2. Processes the Message containing the Intent
    3. Has the Service stop itself & terminate the HandlerThread

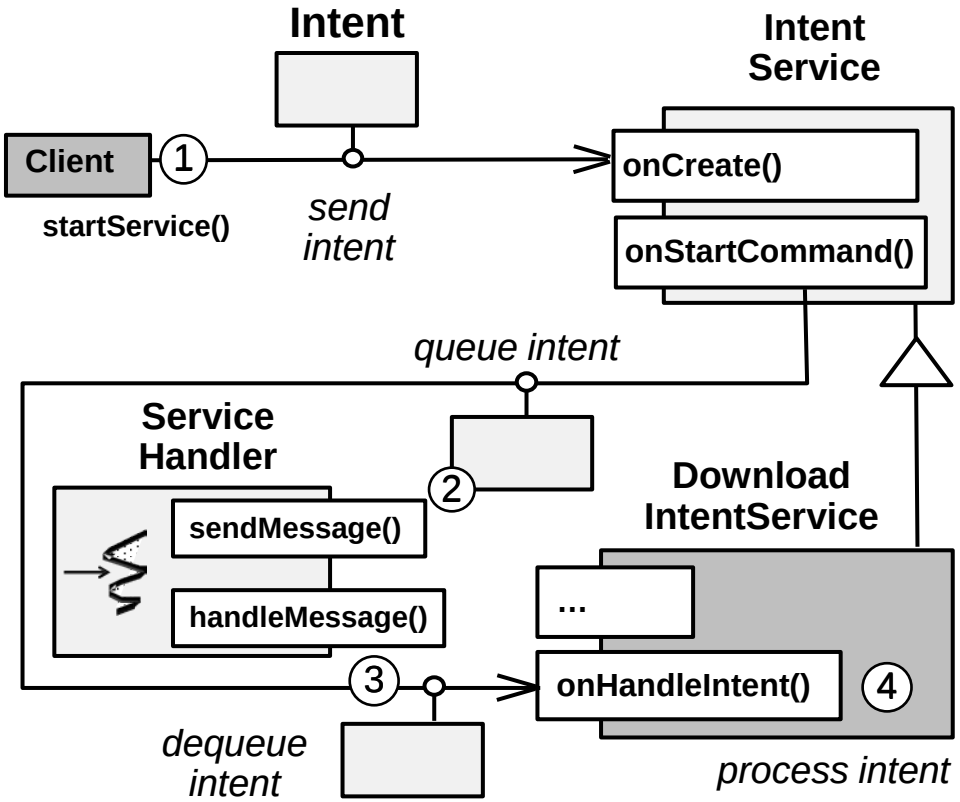



```
void handleMessage(Message msg) {
  downloadImageAndReply
    ((Intent) msg.obj);
  stopSelf(msg.arg1);
```

See developer.android.com/reference/android/app/Service.html#stopSelf(int)

Example of Concurrent Programming with Intents

- This concurrency idiom appears in Android frameworks & packaged apps
  - **frameworks/base/core/java/android/app/IntentService.java**
  - **packages/apps/Calendar/src/com/android/calendar/alerts/AlertService.java**
  - **packages/apps/Mms/src/com/android/mms/transaction/SmsReceiverService.java**
  - **packages/apps/Mms/src/com/android/mms/transaction/TransactionService.java**



We'll cover all of this material in more detail during upcoming courses

End of Concurrent Programming with Android Intents