

Overview of the Java Parallel ImageStreamGang Case Study

Douglas C. Schmidt

d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt



Professor of Computer Science

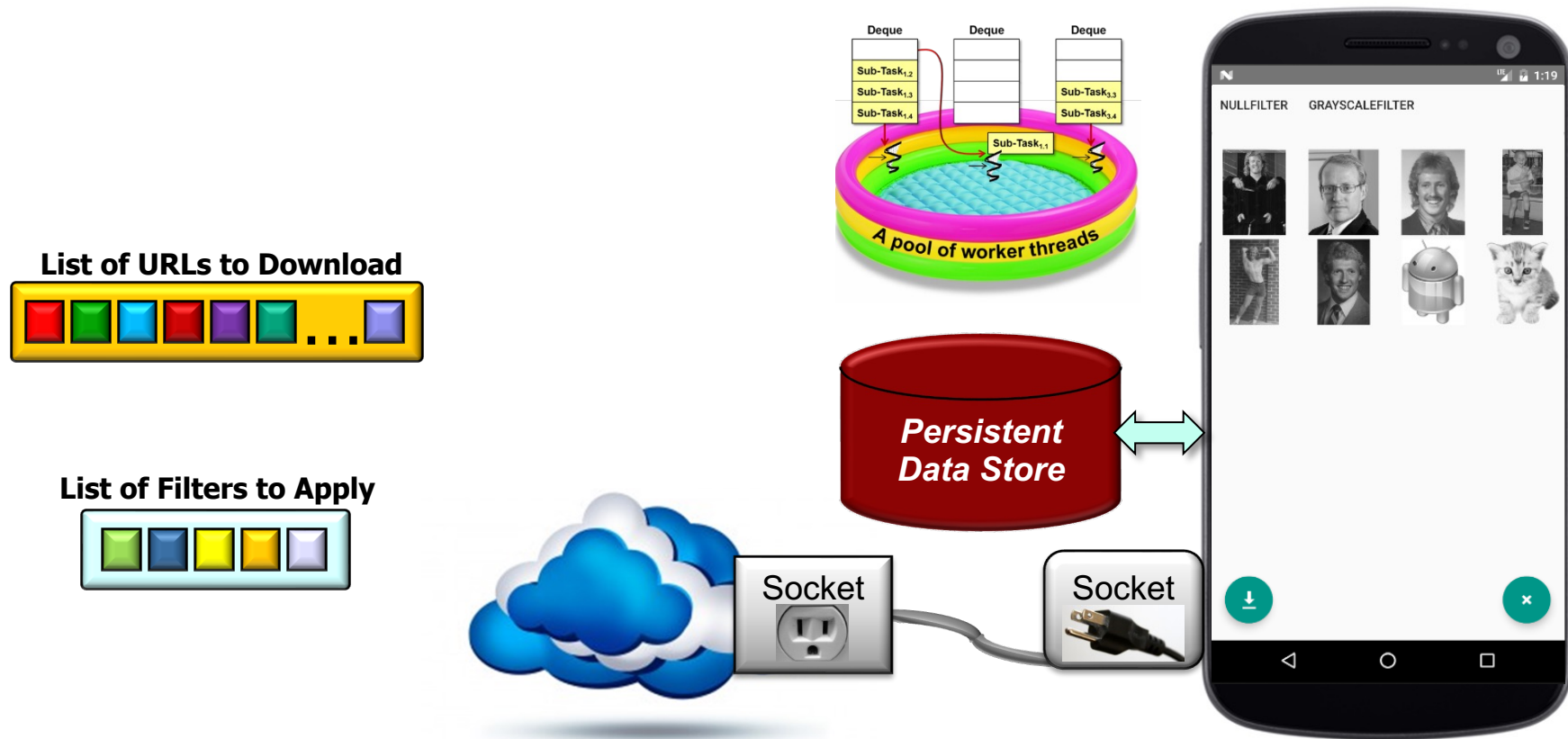
**Institute for Software
Integrated Systems**

**Vanderbilt University
Nashville, Tennessee, USA**



Learning Objectives in this Part of the Lesson

- Understand purpose of the ImageStreamGang app

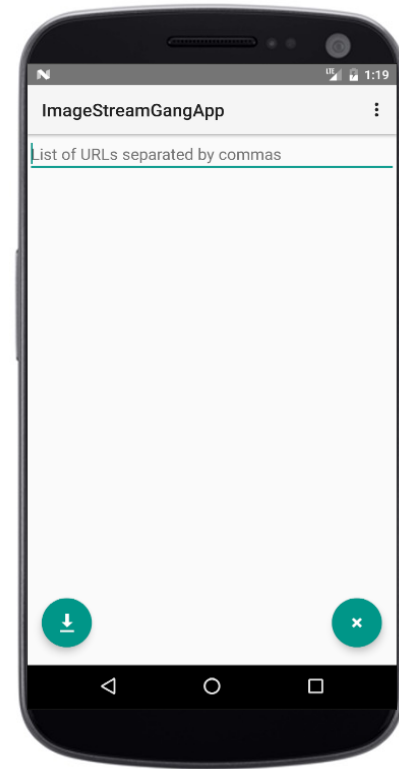
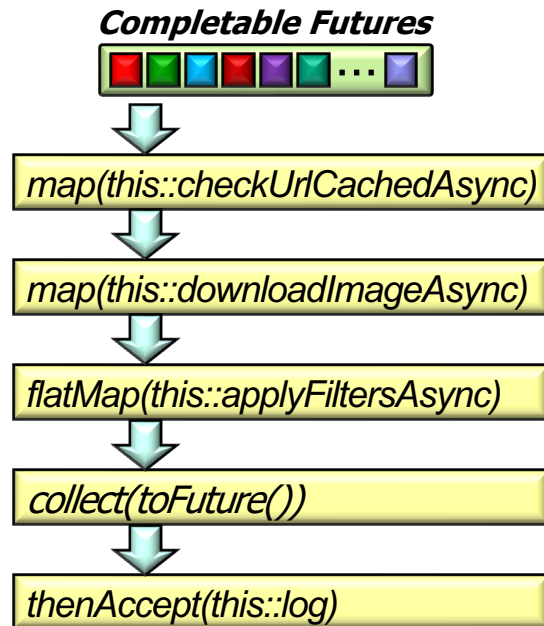
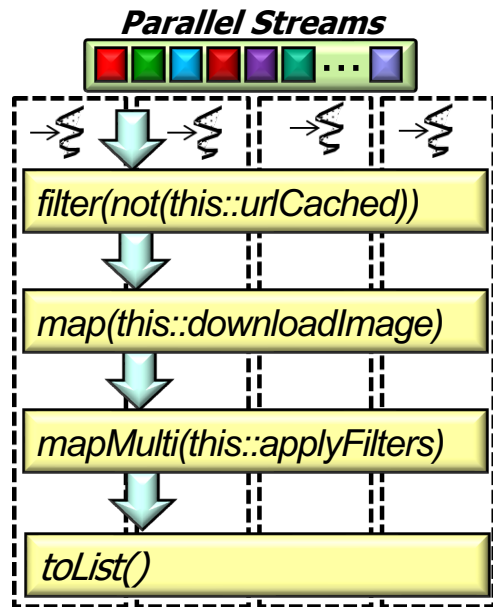


See github.com/douglasraigschmidt/LiveLessons/tree/master/ImageStreamGang

Overview of the Pattern– Oriented ImageStream Gang App

Overview of the Pattern-Oriented ImageStreamGang App

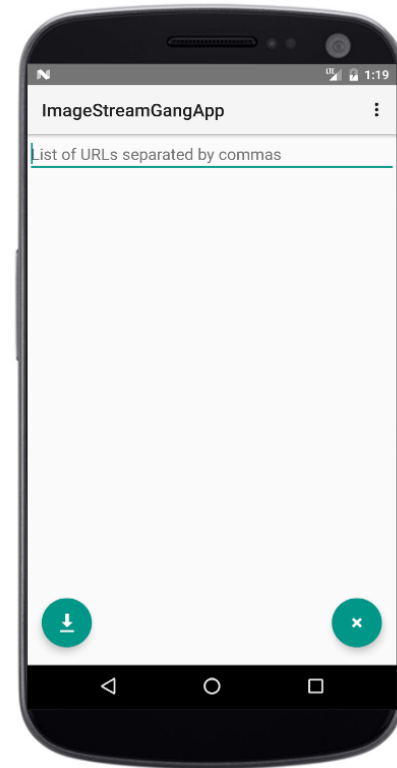
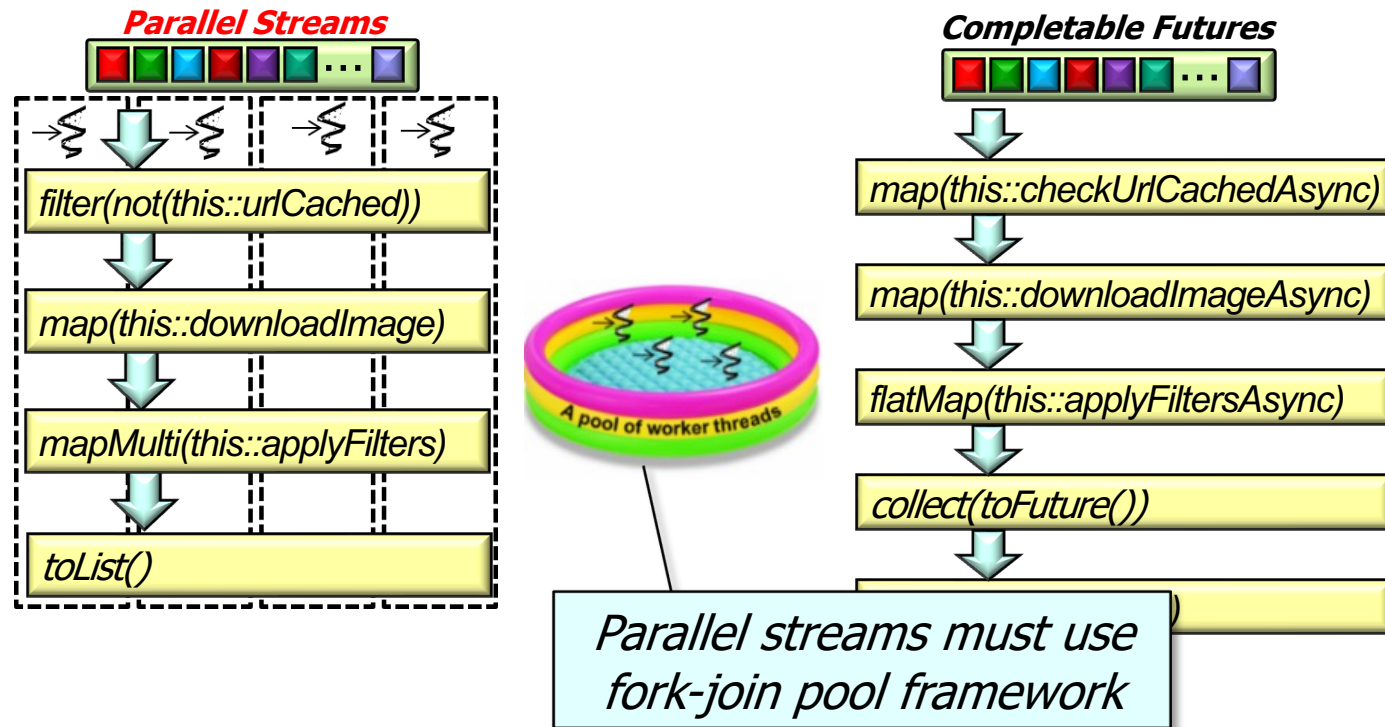
- This app combines streams, completable futures, & reactive streams with the StreamGang framework to download, transform, store, & display images



See github.com/douglasraigschmidt/LiveLessons/tree/master/ImageStreamGang

Overview of the Pattern-Oriented ImageStreamGang App

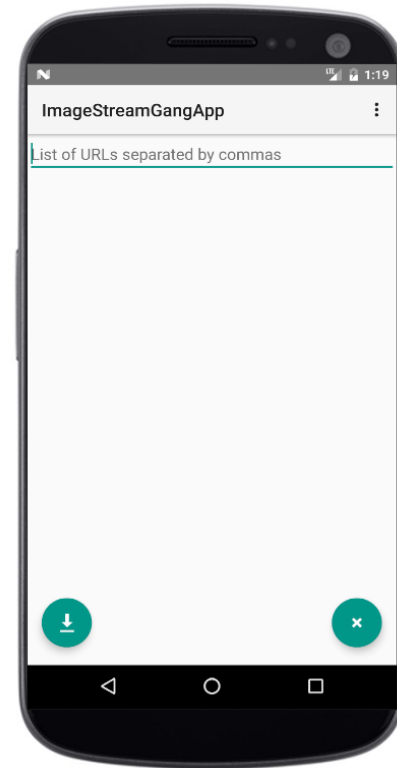
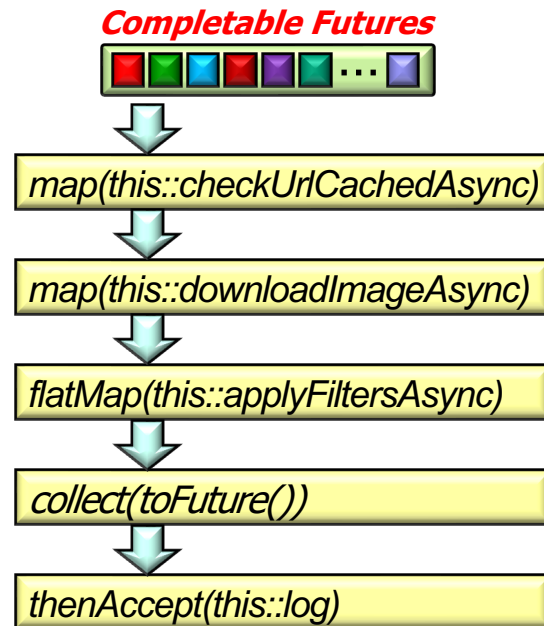
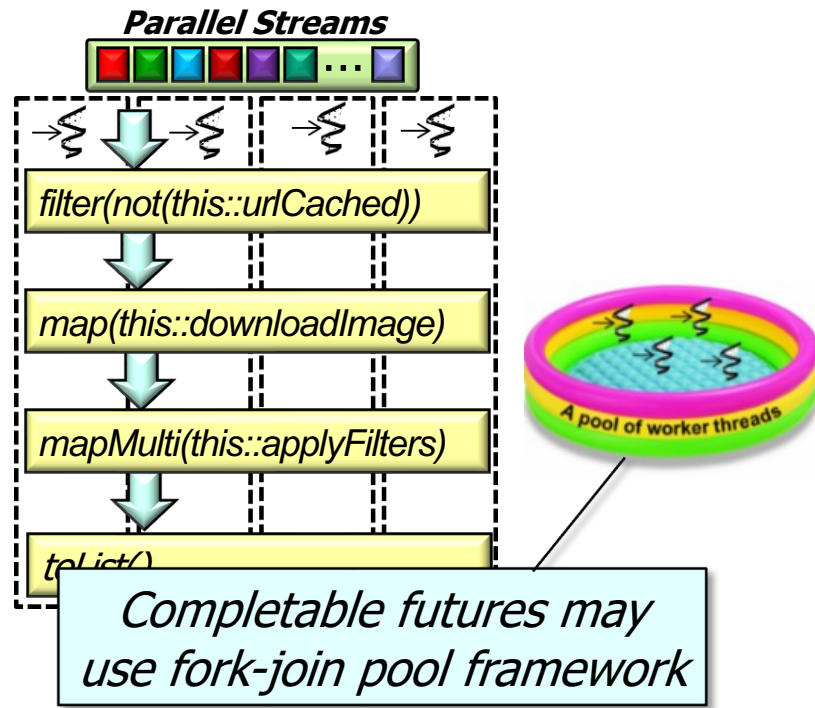
- This app combines streams, completable futures, & reactive streams with the StreamGang framework to download, transform, store, & display images



See docs.oracle.com/javase/tutorial/collections/streams/parallelism.html

Overview of the Pattern-Oriented ImageStreamGang App

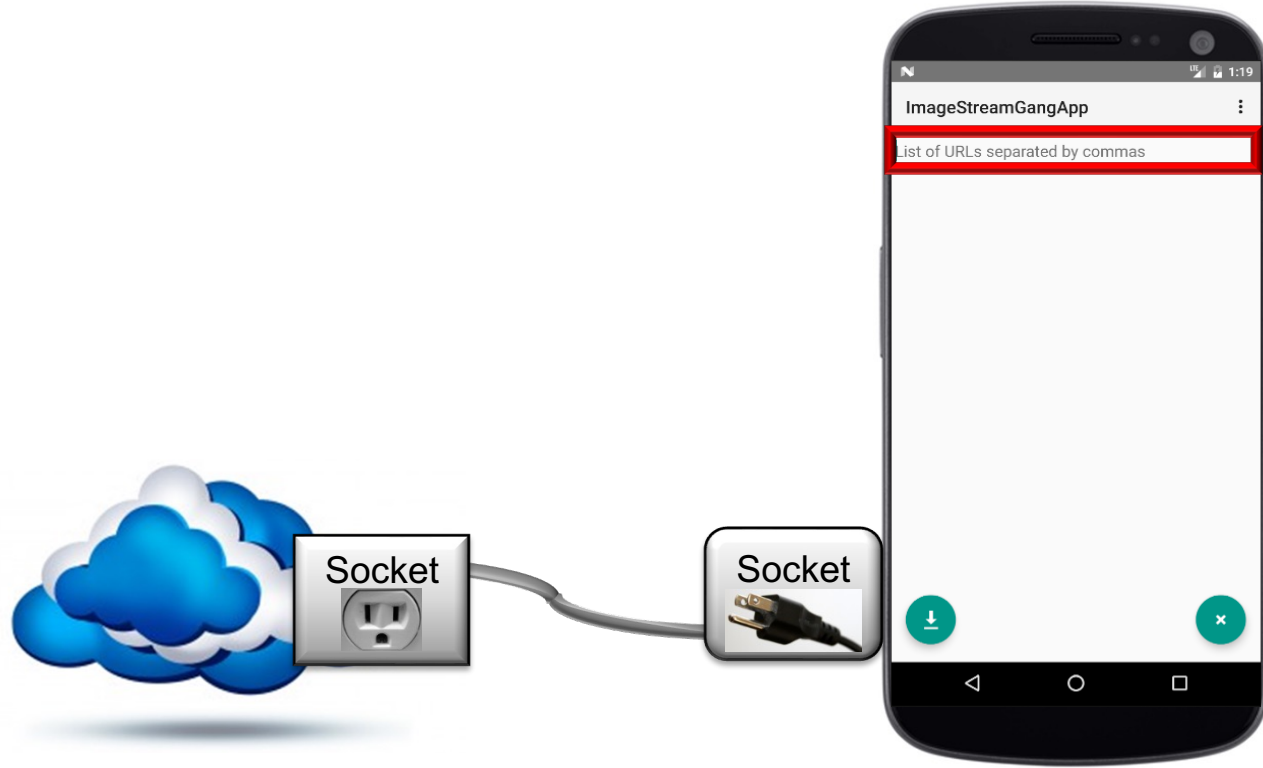
- This app combines streams, completable futures, & reactive streams with the StreamGang framework to download, transform, store, & display images



See www.nurkiewicz.com/2013/05/java-8-definitive-guide-to.html

Overview of the Pattern-Oriented ImageStreamGang App

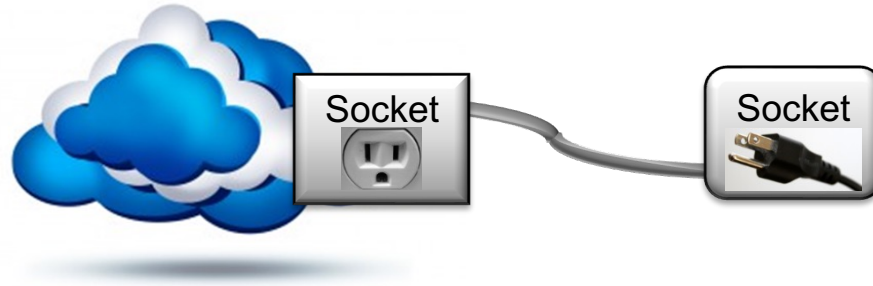
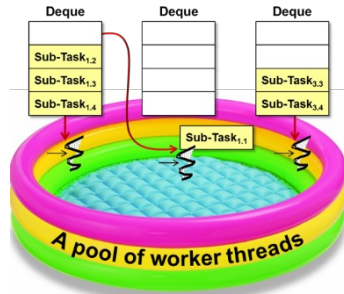
- This app combines streams, completable futures, & reactive streams with the StreamGang framework to download, transform, store, & display images, e.g.



Prompt user for list of URLs to download

Overview of the Pattern-Oriented ImageStreamGang App

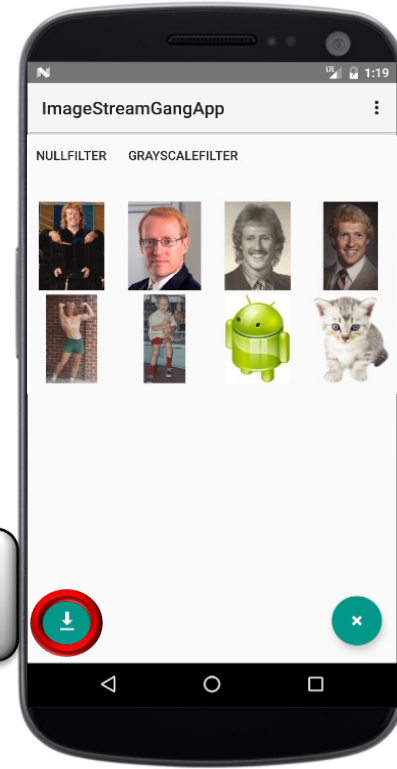
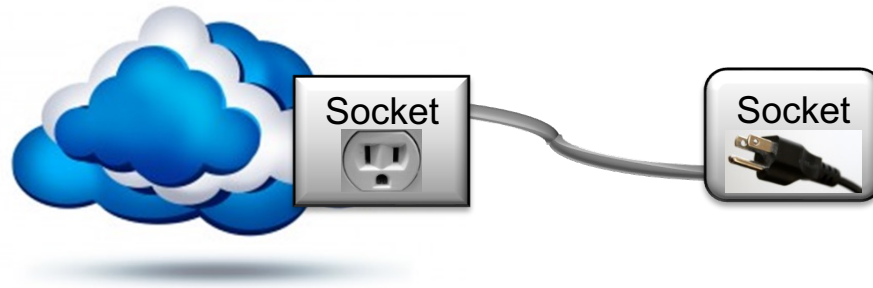
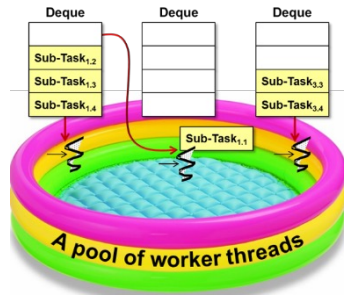
- This app combines streams, completable futures, & reactive streams with the StreamGang framework to download, transform, store, & display images, e.g.



User supplies the list of URLs to download

Overview of the Pattern-Oriented ImageStreamGang App

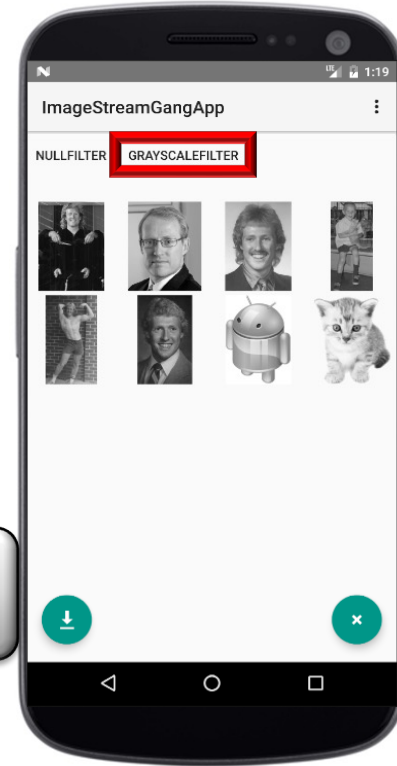
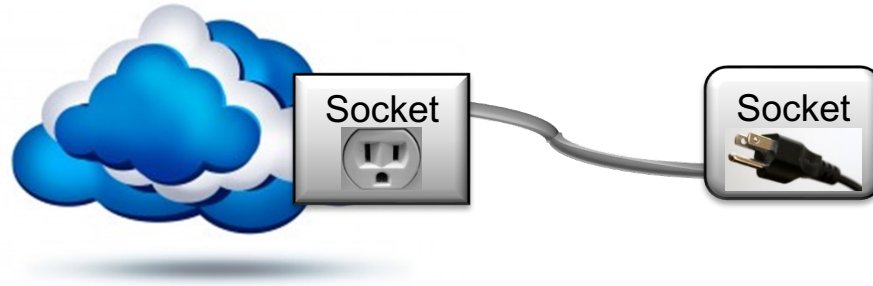
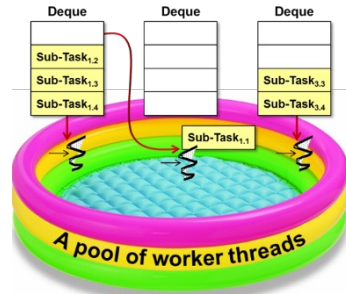
- This app combines streams, completable futures, & reactive streams with the StreamGang framework to download, transform, store, & display images, e.g.



Download images via one or more threads

Overview of the Pattern-Oriented ImageStreamGang App

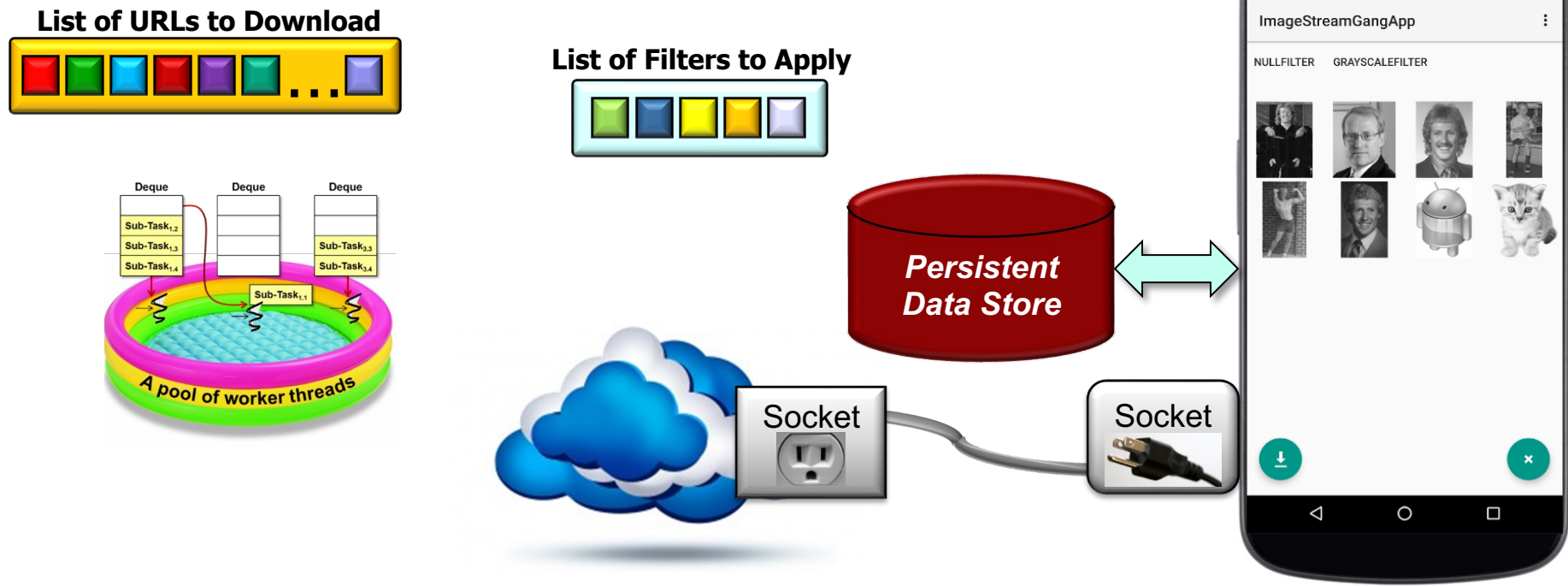
- This app combines streams, completable futures, & reactive streams with the StreamGang framework to download, transform, store, & display images, e.g.



Apply filters to transform downloaded images

Overview of the Pattern-Oriented ImageStreamGang App

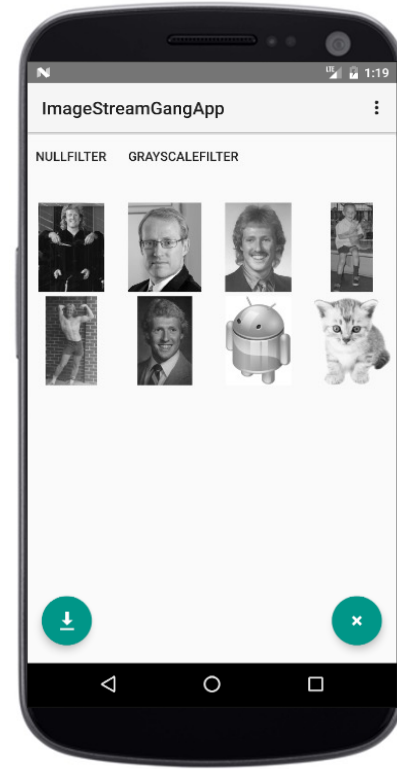
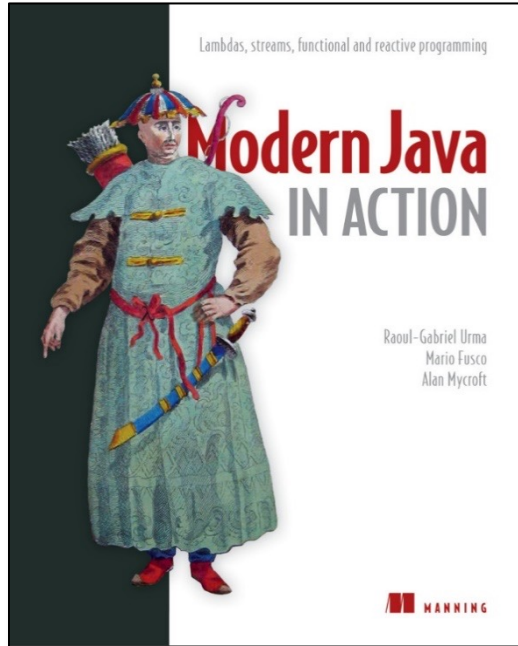
- This app combines streams, completable futures, & reactive streams with the StreamGang framework to download, transform, store, & display images, e.g.



Output filtered images to persistent storage on the local device

Overview of the Pattern-Oriented ImageStreamGang App

- The ImageStreamGang app applies a range of modern Java features

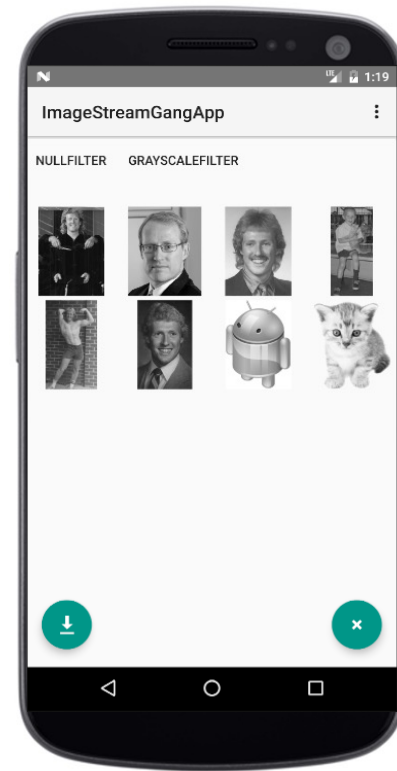


See www.manning.com/books/modern-java-in-action

Overview of the Pattern-Oriented ImageStreamGang App

- The ImageStreamGang app applies a range of modern Java features, e.g.
- Sequential & parallel streams

```
List<Image> filteredImages =  
    getInput()  
        .parallelStream()  
        .filter(not(this::urlCached))  
        .map(this::downloadImage)  
        .mapMulti(this::applyFilters)  
        .toList();
```

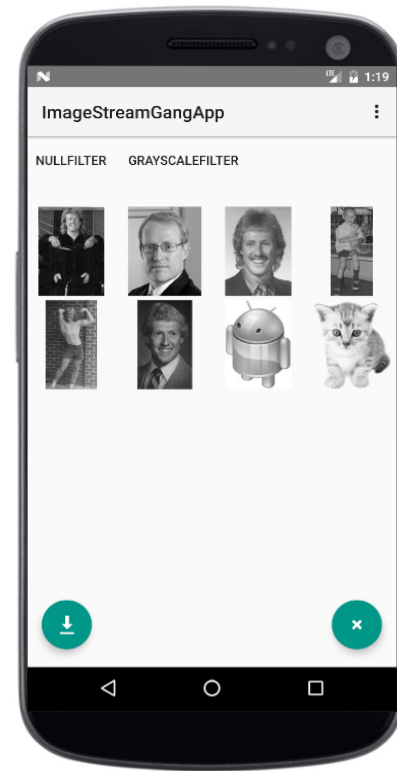


We'll cover parallel streams now

Overview of the Pattern-Oriented ImageStreamGang App

- The ImageStreamGang app applies a range of modern Java features, e.g.
 - Sequential & parallel streams
- Completable futures

```
getInput()  
    .stream()  
    .map(this::checkUrlCachedAsync)  
    .map(this::downloadImageAsync)  
    .flatMap(this::applyFiltersAsync)  
    .collect(toFuture())  
    .thenAccept  
        (stream ->  
            log(stream.flatMap(Optional::stream) ,  
                urls.size()))  
    .join();
```



Overview of the Pattern-Oriented ImageStreamGang App

- The ImageStreamGang app applies a range of modern Java features, e.g.
 - Sequential & parallel streams
 - Completable futures & reactive streams

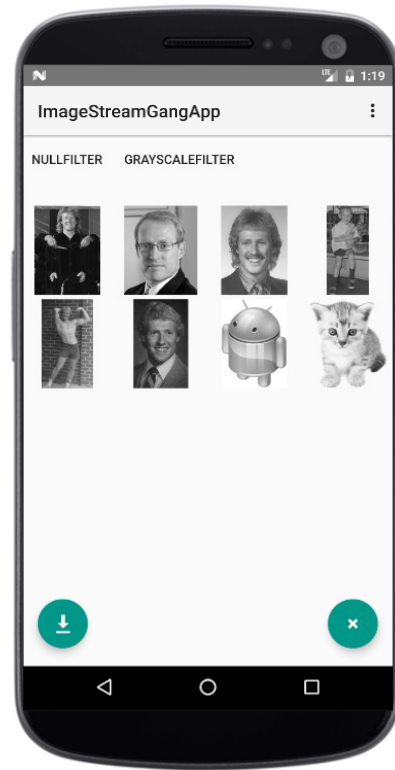


Project
Reactor



Flowable

```
.fromIterable(getInput())  
.parallel()  
.compose(commonPoolFlowable())  
.filter(url -> !urlCached(url))  
.map(this::blockingDownload)  
.flatMap(this::applyFilters)  
.sequential()  
.collect(toList())  
.blockingSubscribe(filteredImages -> ...);
```

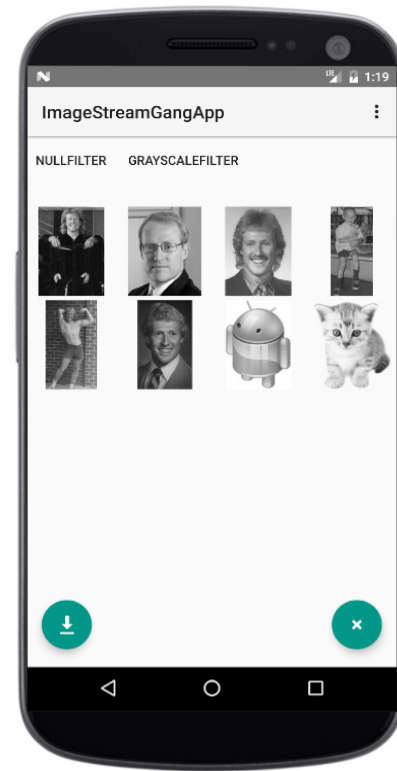


See www.baeldung.com/rx-java & projectreactor.io

Overview of the Pattern-Oriented ImageStreamGang App

- The ImageStreamGang app applies a range of modern Java features, e.g.
 - Sequential & parallel streams
 - Completable futures & reactive streams

COMING SOON

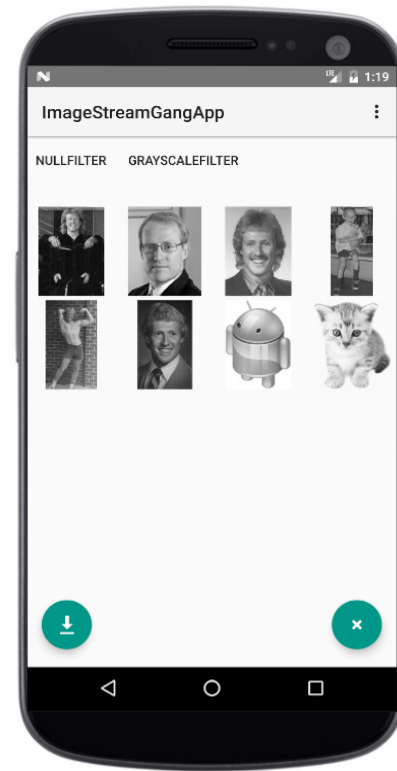


We cover completable futures & reactive streams shortly

Overview of the Pattern-Oriented ImageStreamGang App

- The ImageStreamGang app applies a range of modern Java features, e.g.
 - Sequential & parallel streams
 - Completable futures & reactive streams
 - Lambda expressions & method references

```
Runnable mCompletionHook =  
    () -> MainActivity.this.runOnUiThread  
        (this::goToResultActivity);
```



We covered these foundational Java features earlier

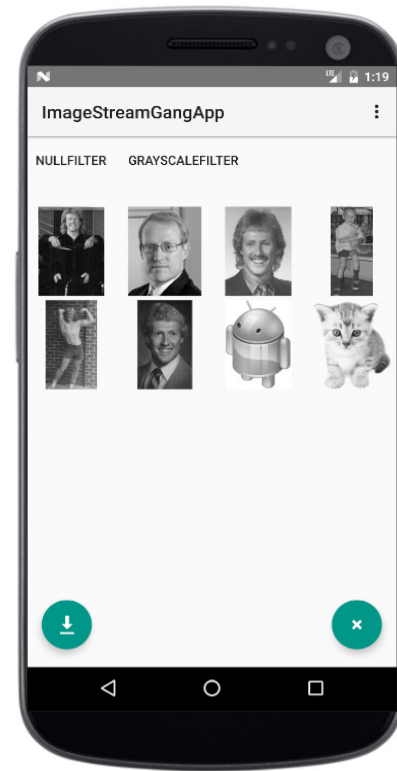
Overview of the Pattern-Oriented ImageStreamGang App

- The ImageStreamGang app applies a range of modern Java features, e.g.
 - Sequential & parallel streams
 - Completable futures & reactive streams
 - Lambda expressions & method references

```
Runnable mCompletionHook =  
    () -> MainActivity.this.runOnUiThread  
        (this::goToResultActivity);
```

versus

```
Runnable mCompletionHook = new Runnable() {  
    public void run() {  
        MainActivity.this.runOnUiThread  
            (new Runnable() { public void run()  
                { goToResultActivity(); } });  
    }  
};
```

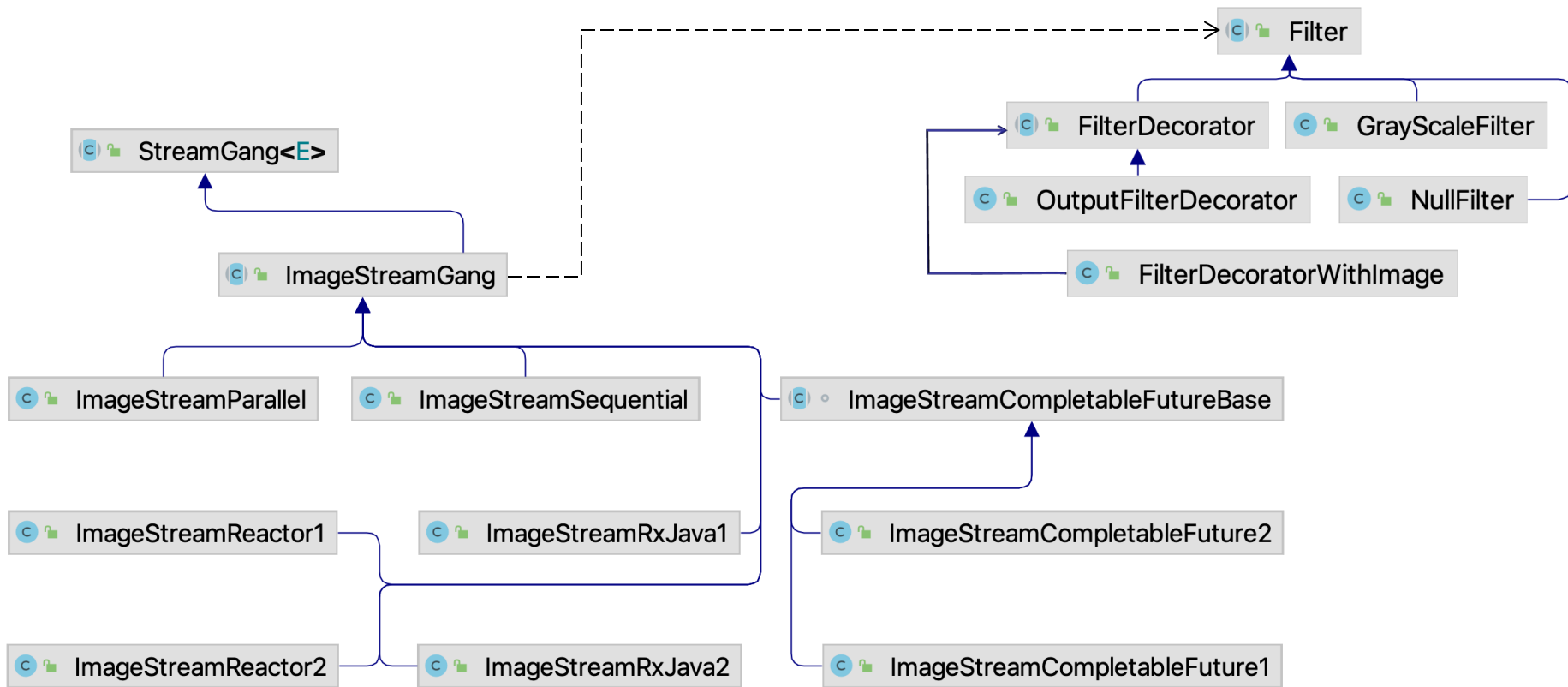


See stackoverflow.com/questions/16397027/whats-the-harm-in-using-anonymous-class

Strategy for Understanding the ImageStreamGang App

Strategy for Understanding the ImageStreamGang App

- This app is complicated & contains many classes



See github.com/douglasraigschmidt/LiveLessons/tree/master/ImageStreamGang

End of Overview of the Java Parallel ImageStreamGang Case Study