Overview of the Java Parallel ImageStreamGang Case Study

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

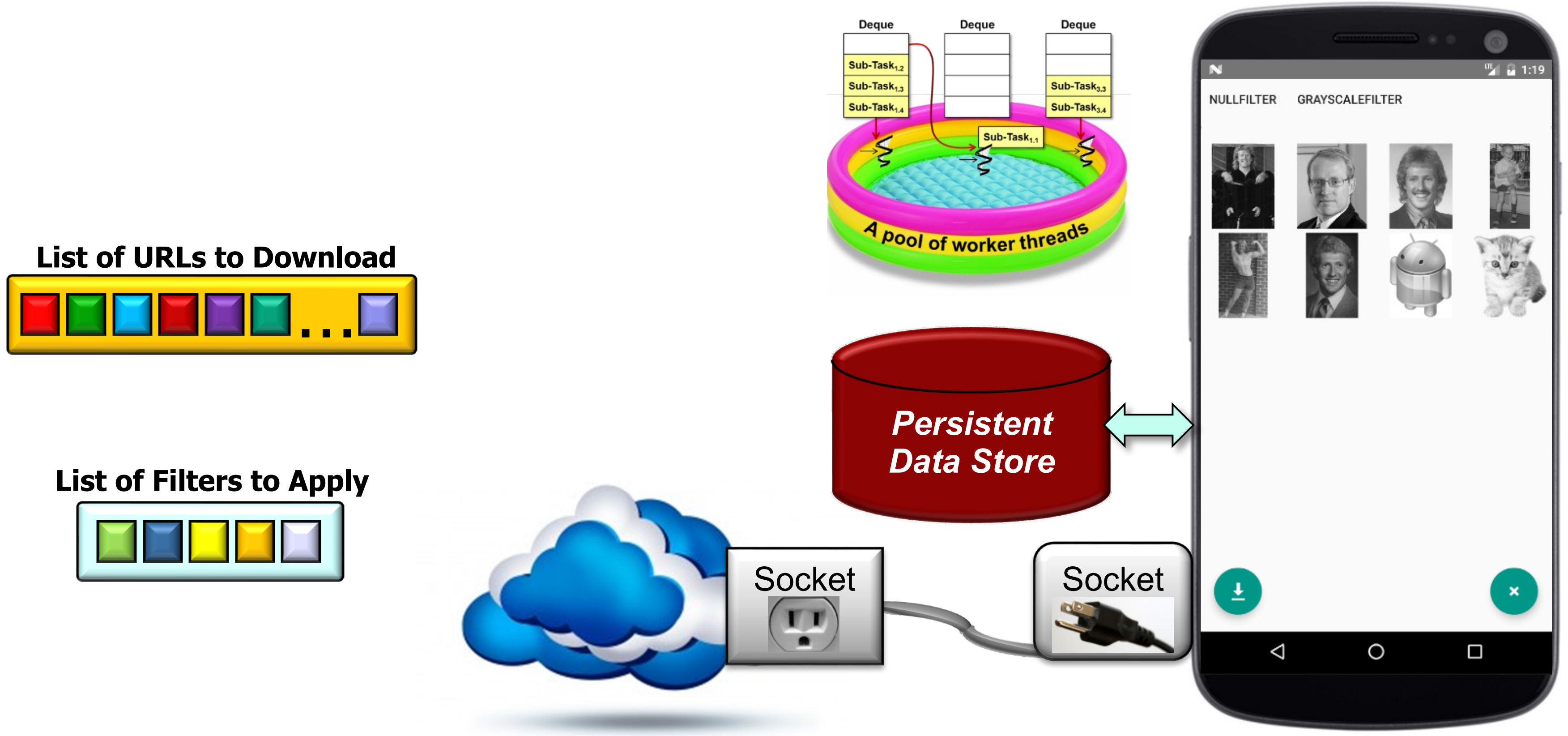
Professor of Computer Science

Institute for Software Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA

Learning Objectives in this Part of the Lesson

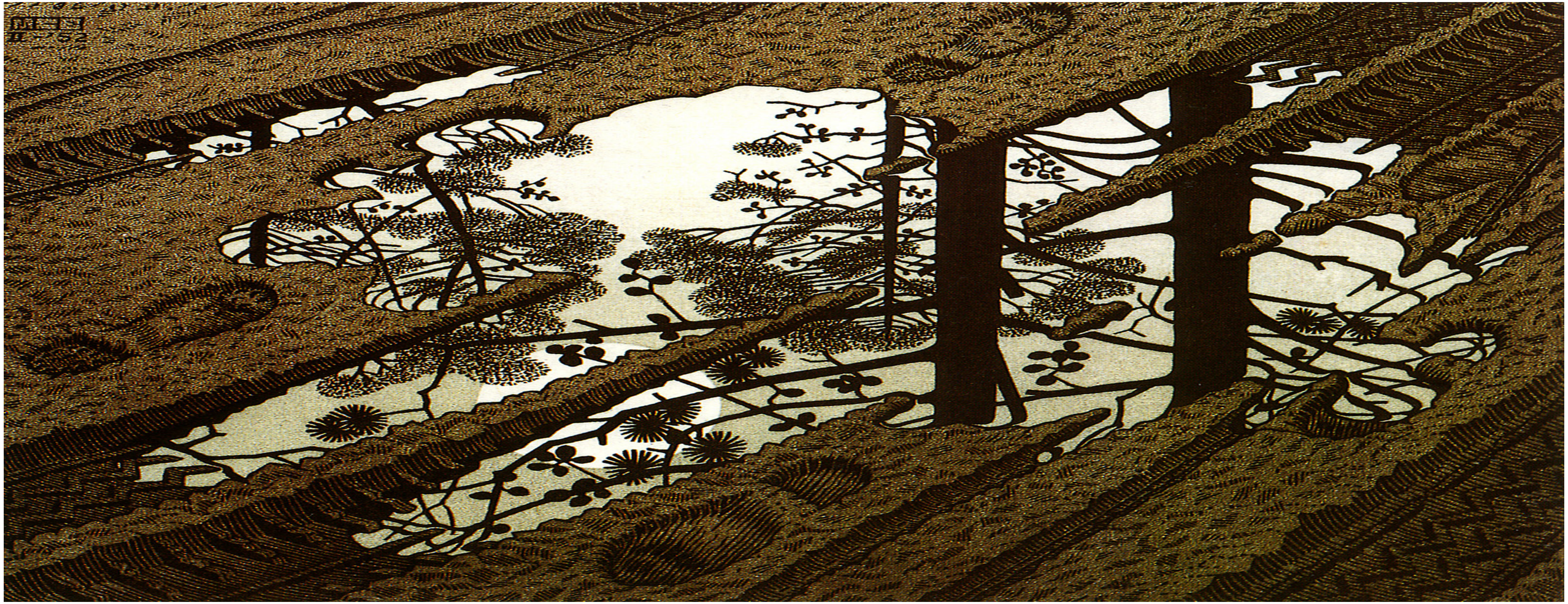
- Understand purpose of the ImageStreamGang app



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

Learning Objectives in this Part of the Lesson

- Understand the structure & functionality of the ImageStreamGang app
  - Be aware of strategies for its OO & functional design & implementation

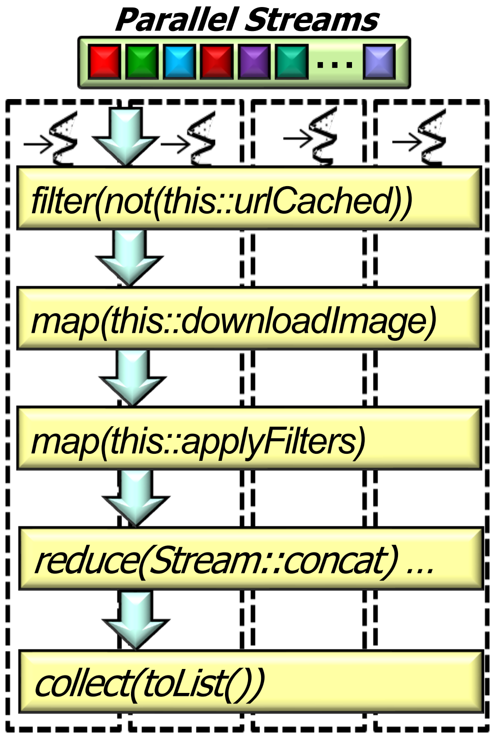


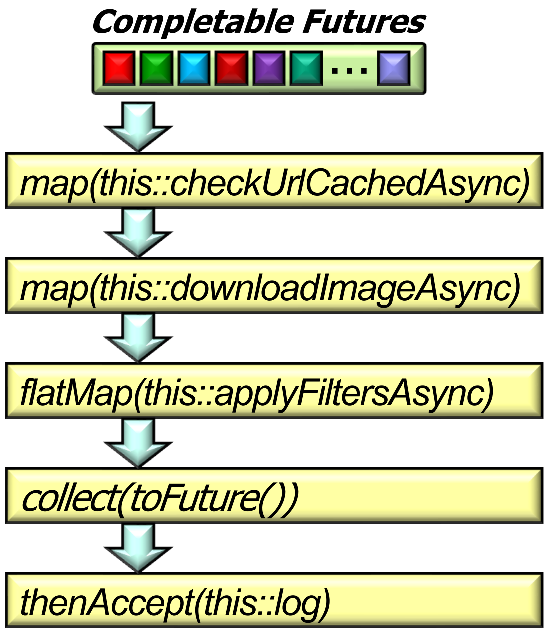
Including pattern-oriented design, data flows, & detailed code walkthroughs

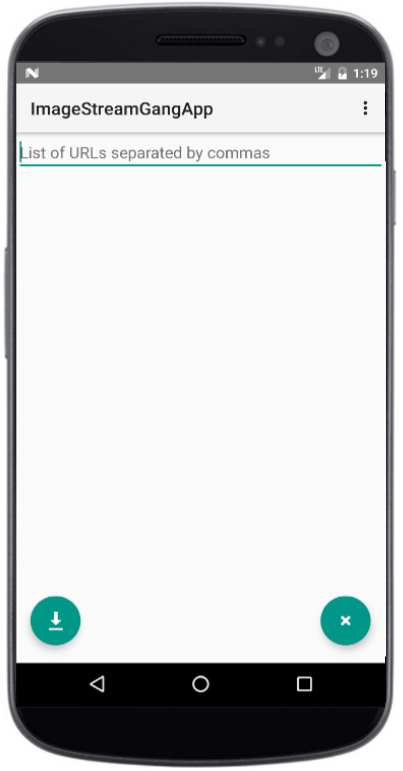
Overview of the Pattern-Oriented ImageStream Gang App

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



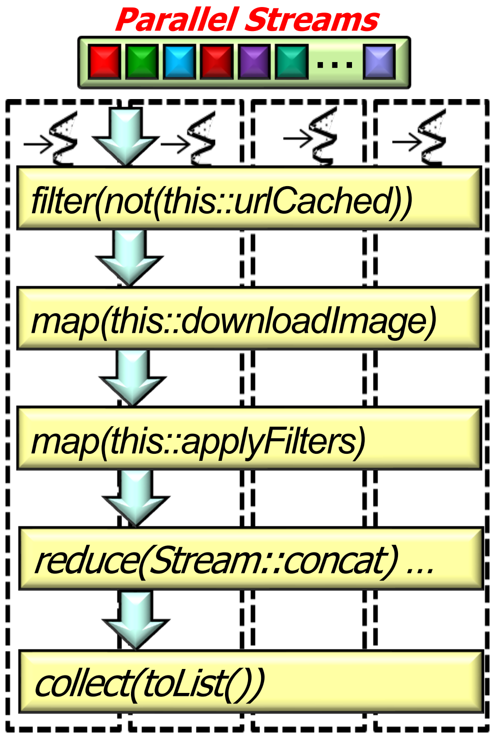

Completable Futures
map(this::checkUrlCachedAsync)
map(this::downloadImageAsync)
flatMap(this::applyFiltersAsync)
collect(toFuture())
thenAccept(this::log)

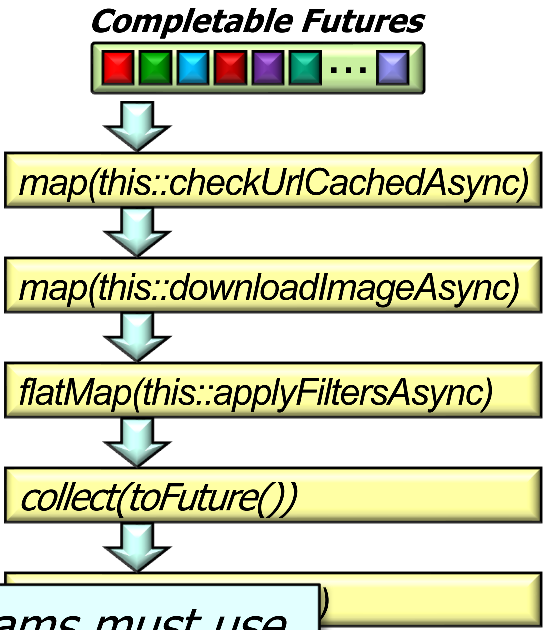


See github.com/douglascraigschmidt/LiveLessons/tree/master/ImageStreamGang

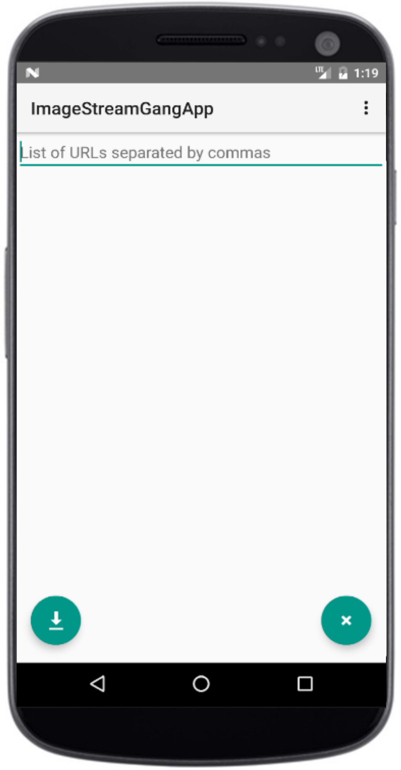
Overview of the Pattern-Oriented ImageStreamGang App

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




Completable Futures
map(this::checkUrlCachedAsync)
map(this::downloadImageAsync)
flatMap(this::applyFiltersAsync)
collect(toFuture())
)

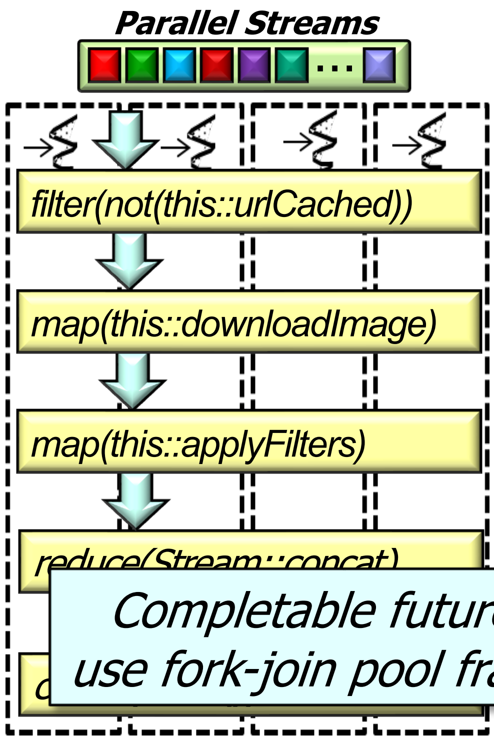
Parallel streams must use fork-join pool framework



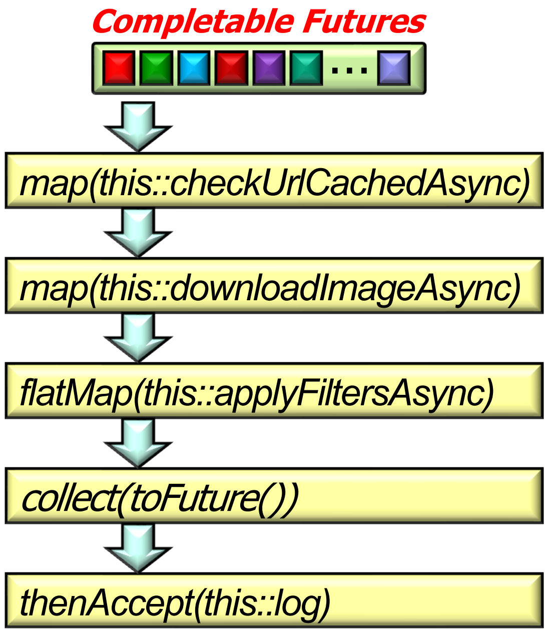
See docs.oracle.com/javase/tutorial/collections/streams/parallelism.html

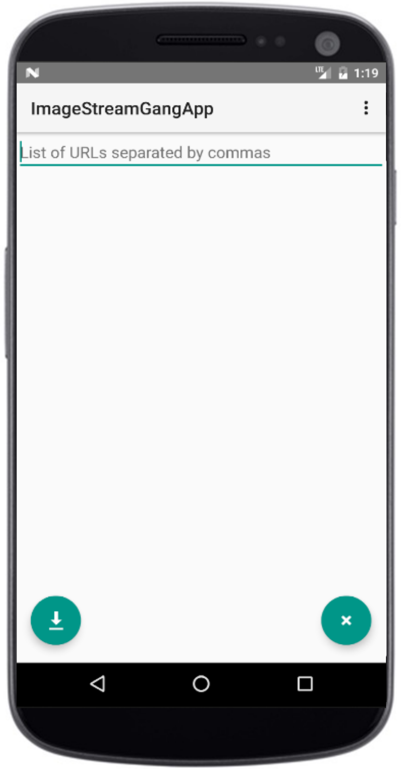
Overview of the Pattern-Oriented ImageStreamGang App

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



Completable futures may use fork-join pool framework

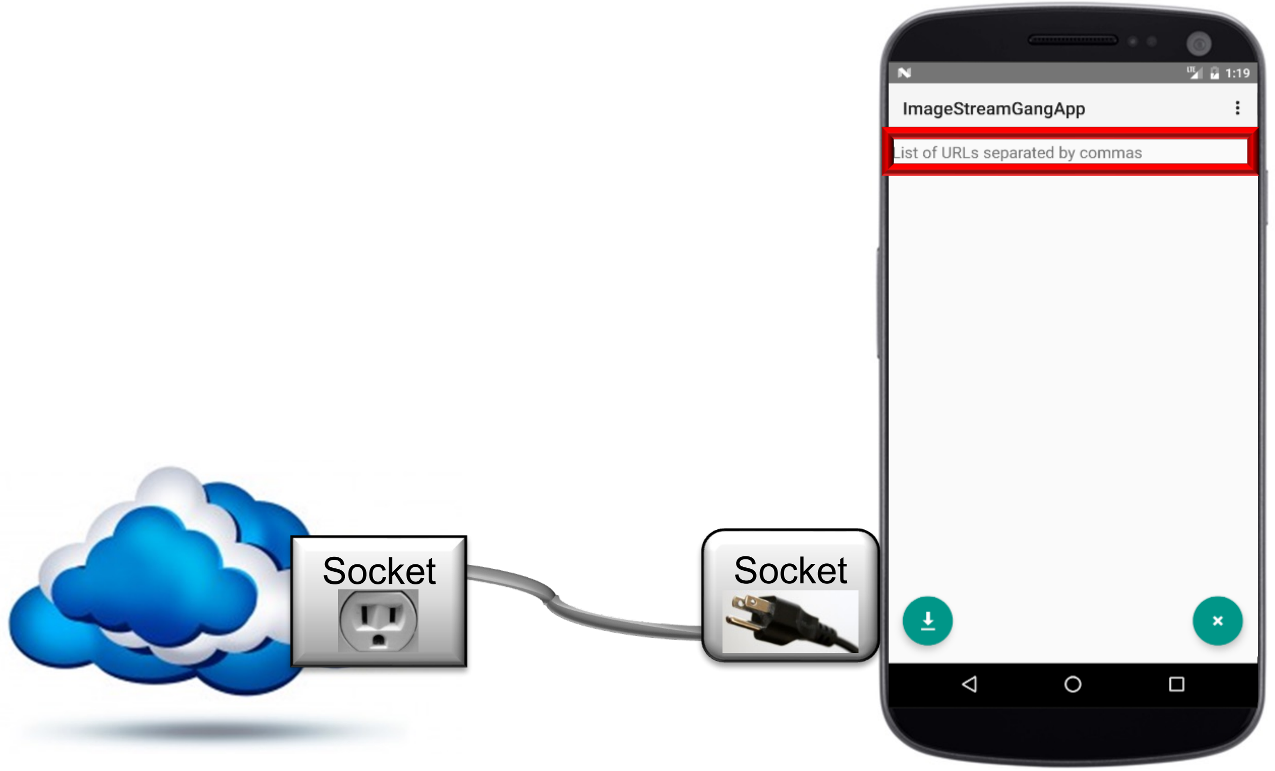

Completable Futures
map(this::checkUrlCachedAsync)
map(this::downloadImageAsync)
flatMap(this::applyFiltersAsync)
collect(toFuture())
thenAccept(this::log)



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

Overview of the Pattern-Oriented ImageStreamGang App

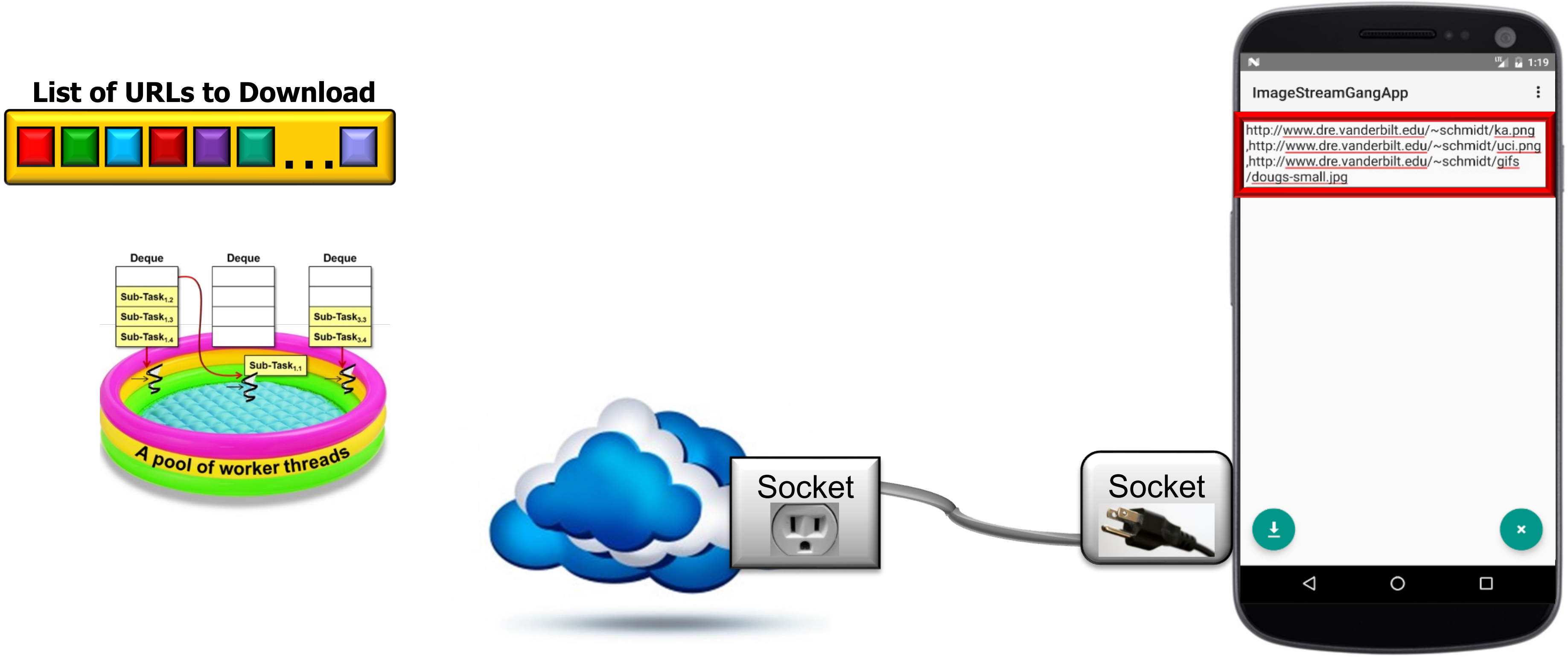
- This app combines streams, completable futures, & reactive streams with the StreamGang framework to download, transform, store, & display images, e.g.



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.

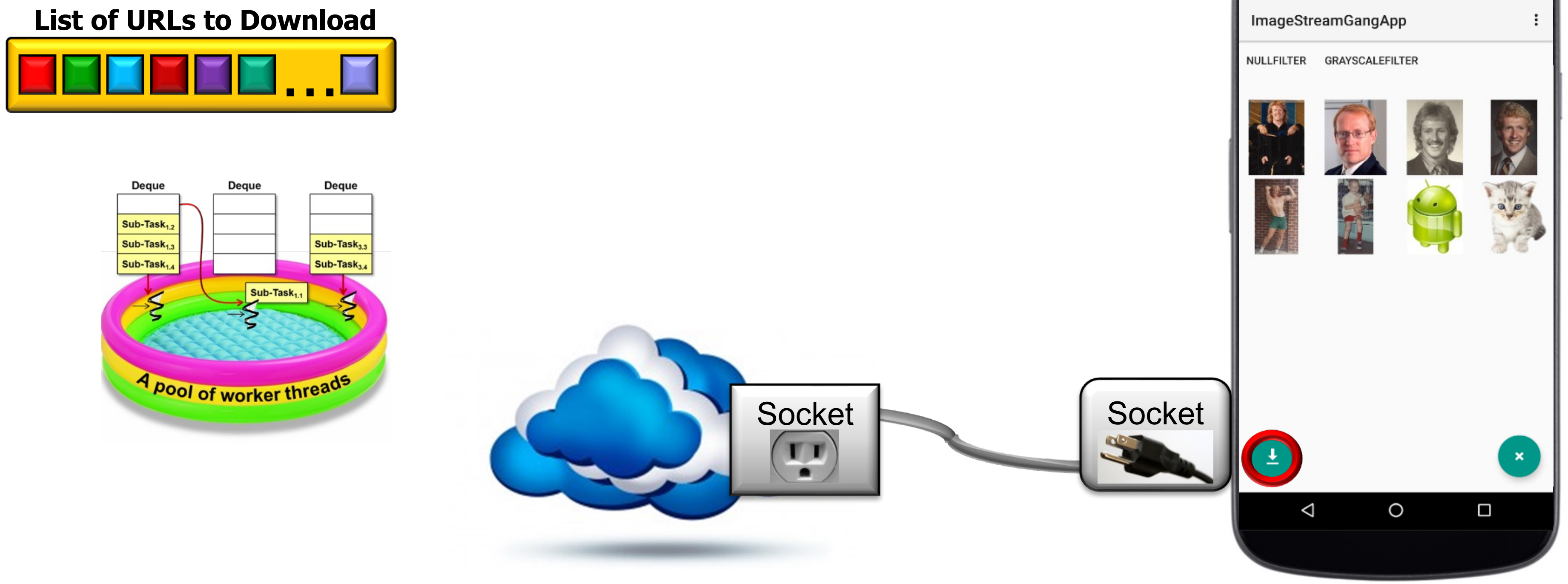


http://www.dre.vanderbilt.edu/~schmidt/ka.png
,http://www.dre.vanderbilt.edu/~schmidt/uci.png
,http://www.dre.vanderbilt.edu/~schmidt/gifs
/dougs-small.jpg

User supplies the list of URLs to download

Overview of the Pattern-Oriented ImageStreamGang App

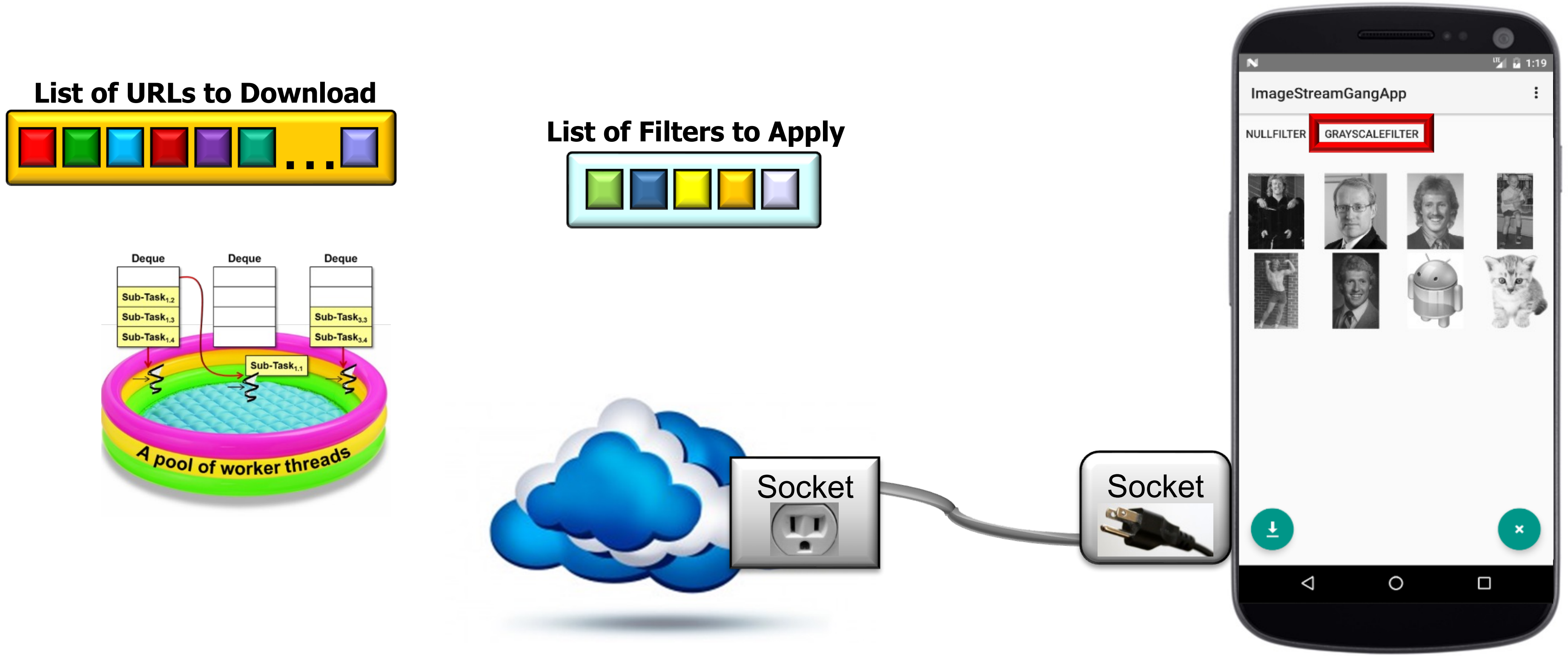
- This app combines streams, completable futures, & reactive streams with the StreamGang framework to download, transform, store, & display images, e.g.



Download images via one or more threads

Overview of the Pattern-Oriented ImageStreamGang App

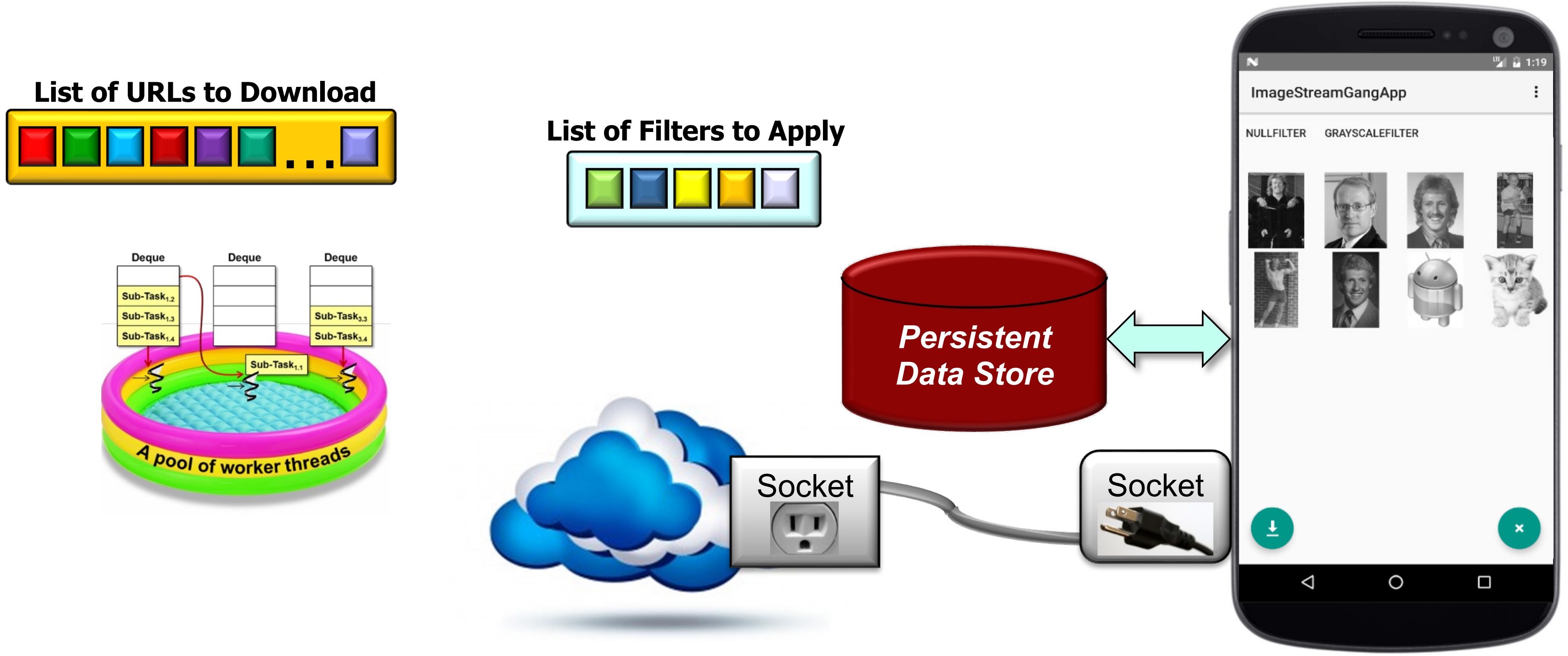
- This app combines streams, completable futures, & reactive streams with the StreamGang framework to download, transform, store, & display images, e.g.



Apply filters to transform downloaded images

Overview of the Pattern-Oriented ImageStreamGang App

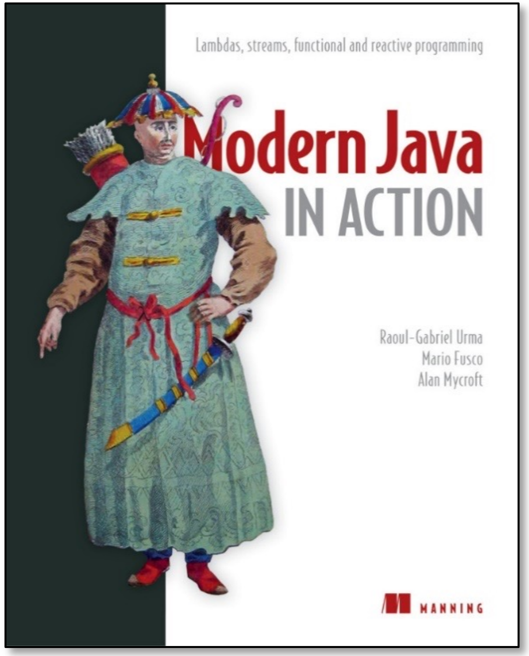
- This app combines streams, completable futures, & reactive streams with the StreamGang framework to download, transform, store, & display images, e.g.

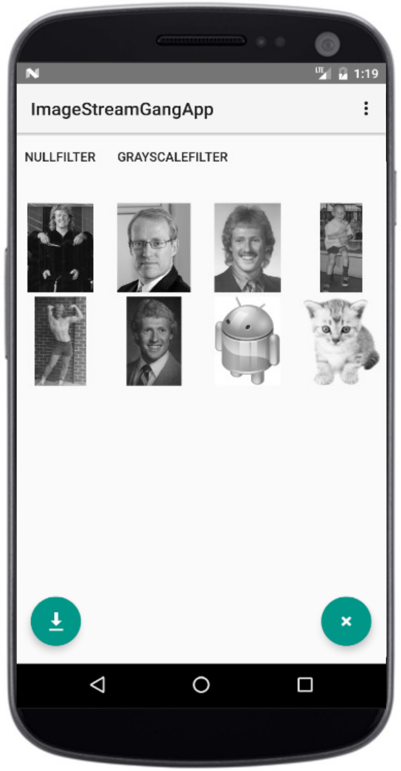


Output filtered images to persistent storage on the local device

Overview of the Pattern-Oriented ImageStreamGang App

- The ImageStreamGang app applies a range of modern Java features



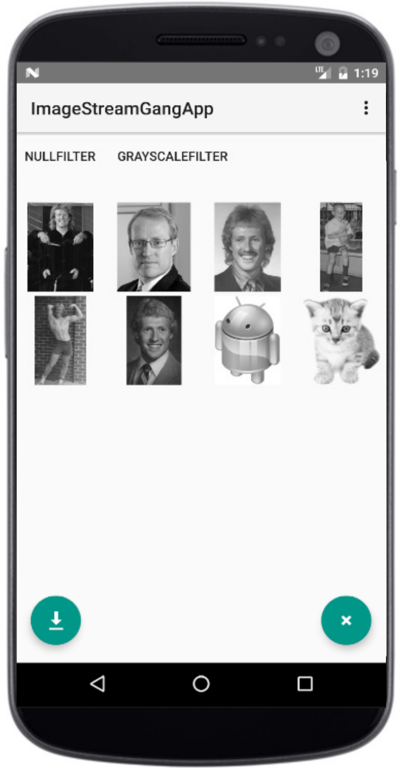


See www.manning.com/books/modern-java-in-action

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)
      .map(this::applyFilters)
      .reduce(Stream::concat)
      .orElse(Stream.empty())
      .collect(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

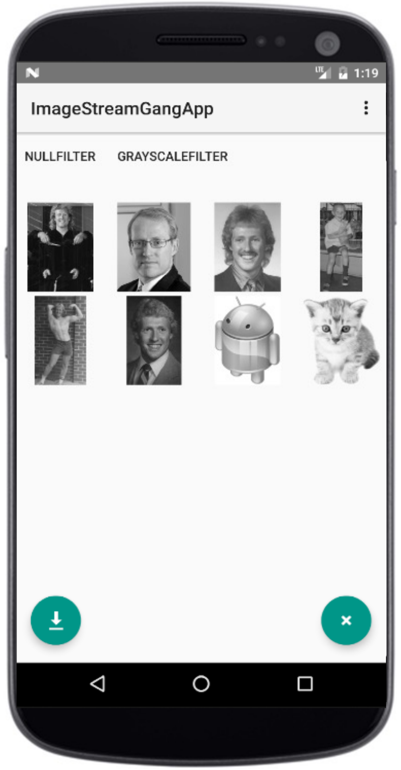
```
getInput()
  .stream()
  .map(this::checkUrlCachedAsync)
  .map(this::downloadImageAsync)
  .flatMap(this::applyFiltersAsync)
  .collect(toFuture())
  .thenAccept
    (stream ->
     log(stream.flatMap(Optional::stream),
         urls.size()))
  .join();
```

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

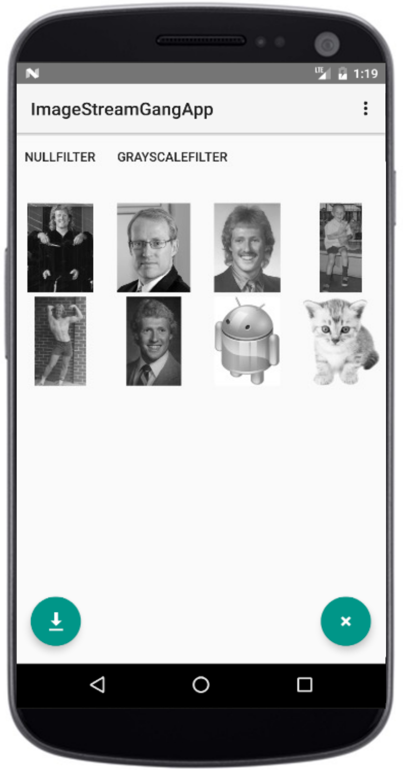
```
ArrayList<Image> filteredImages = Flux
  .fromIterable(urls)
  .flatMap(url -> Flux
               .just(url)
               .subscribeOn(boundedElastic())
               .filter(___ -> !urlCached(url))
               .map(this::blockingDownload)
               .flatMap(this::applyFilters))
  .reduceWith(ArrayList<Image>::new,
                 this::append)
  .block();
```



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



We cover completable futures & reactive streams later

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

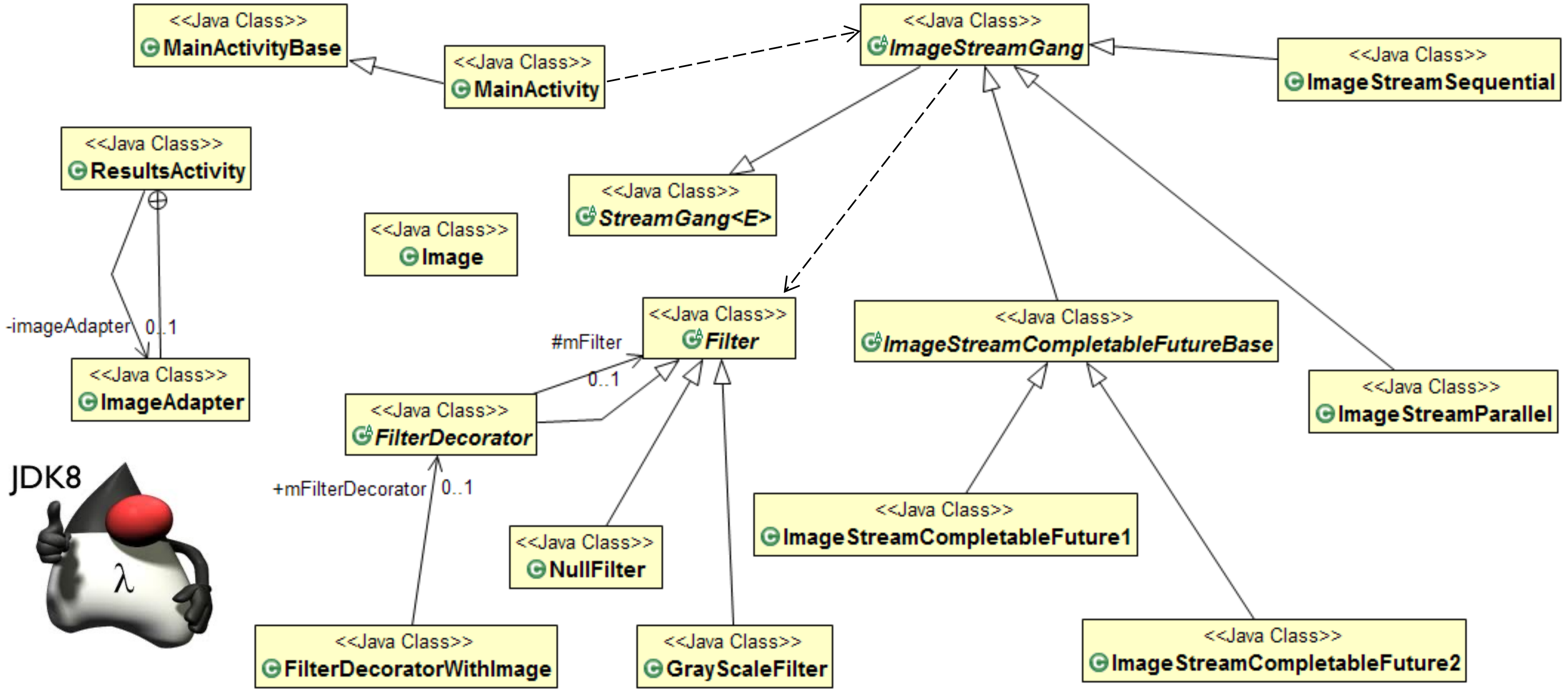
```
Runnable mCompletionHook = new Runnable() {
    public void run() {
    MainActivity.this.runOnUiThread
      (new Runnable() { public void run()
        { goToResultActivity(); } }); }};
```

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

Strategy for Understanding the ImageStreamGang App

Strategy for Understanding the ImageStreamGang App

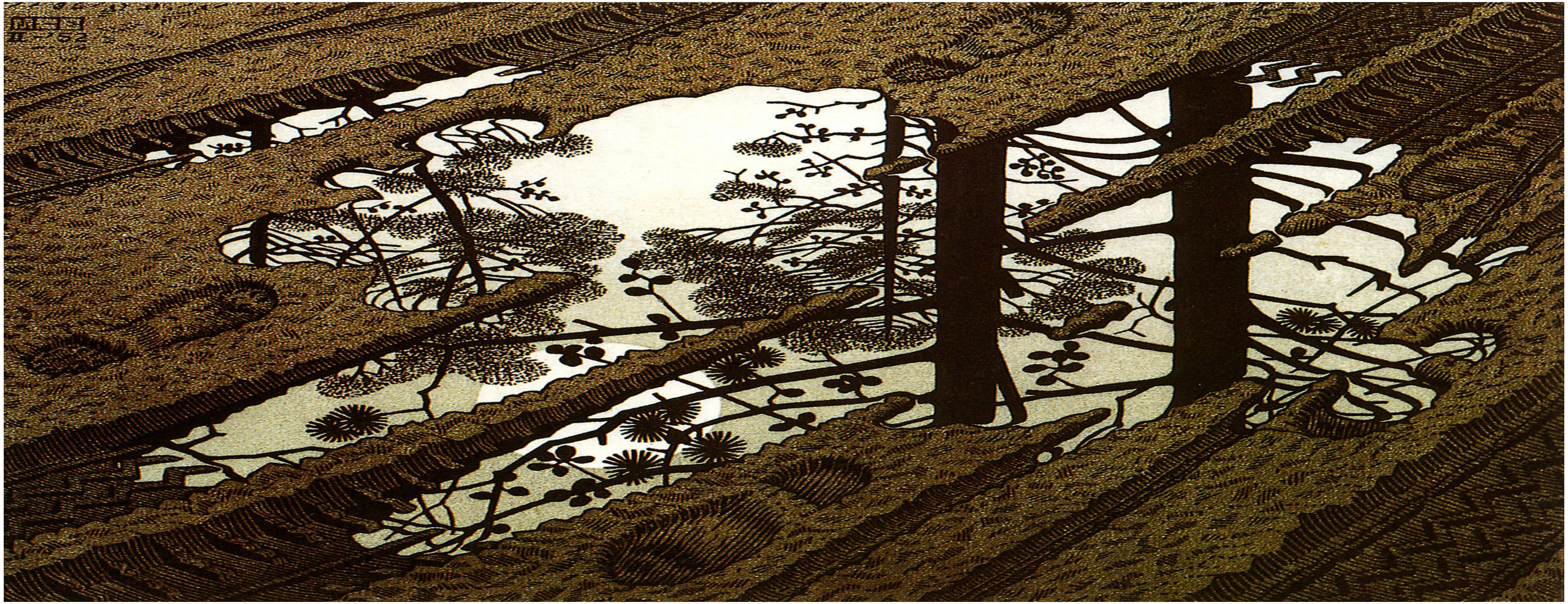
- This app is complicated & contains many classes



<<Java Class>> MainActivityBase
<<Java Class>> MainActivity
<<Java Class>> ImageStreamGang
<<Java Class>> ImageStreamSequential
<<Java Class>> ResultsActivity
<<Java Class>> StreamGang<E>
<<Java Class>> Image
-imageAdapter 0..1
<<Java Class>> ImageAdapter
#mFilter
0..1
<<Java Class>> Filter
<<Java Class>> ImageStreamCompletableFutureBase
<<Java Class>> ImageStreamParallel
<<Java Class>> FilterDecorator
+mFilterDecorator 0..1
JDK8
<<Java Class>> ImageStreamCompletableFuture1
<<Java Class>> NullFilter
<<Java Class>> FilterDecoratorWithImage
<<Java Class>> GrayScaleFilter
<<Java Class>> ImageStreamCompletableFuture2

Strategy for Understanding the ImageStreamGang App

- This app is complicated & contains many classes
  - We therefore analyze it from various perspectives



Including pattern-oriented design, data flows, & detailed code walkthroughs

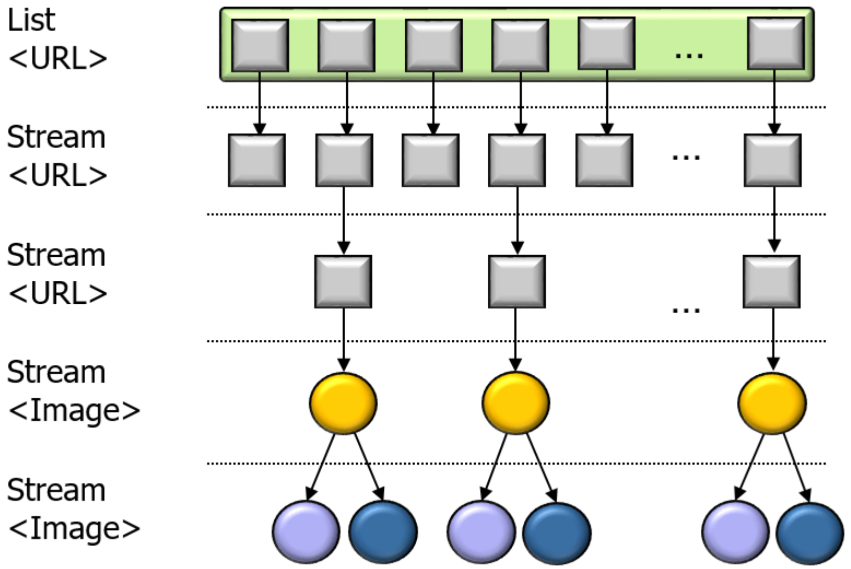
Strategy for Understanding the ImageStreamGang App

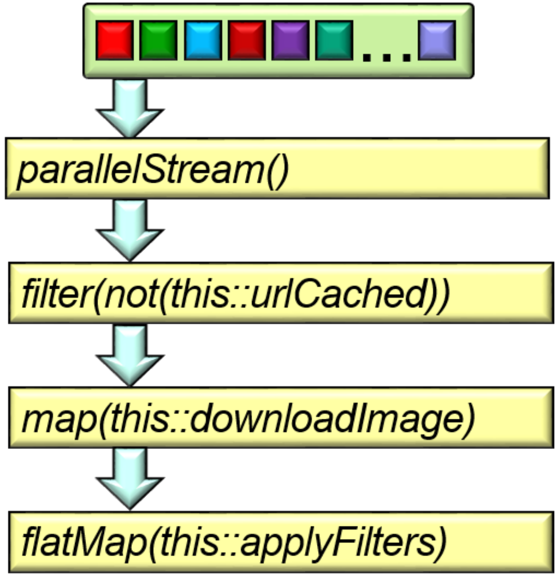
- This app is complicated & contains many classes
  - We therefore analyze it from various perspectives
  - Watch this entire lesson carefully to understand how it all works

livelessons
video instruction from technology experts
Java Concurrency LiveLessons
Addison-Wesley
livelessons
©2015 Pearson, Inc.

Strategy for Understanding the ImageStreamGang App

- This app is complicated & contains many classes
  - We therefore analyze it from various perspectives
  - Watch this entire lesson carefully to understand how it all works
  - Visualize the data flow in a parallel stream





Strategy for Understanding the ImageStreamGang App

- This app is complicated & contains many classes
  - We therefore analyze it from various perspectives
  - Watch this entire lesson carefully to understand how it all works
  - Visualize the data flow in a parallel stream
  - Run/read the code to see how it all works

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

End of Overview of the Java Parallel ImageStreamGang Case Study