

ANALYSIS OF SOFTWARE ARCHITECTURE AND DESIGN

Software Architecture and Design
(Group 2)



system for displaying images in Android applications. It
takes care of image loading and display.
(Version 2.5.0)



#01

ARCHITECTURE

MVC: Module –View – Controller

fresco/drawee/src/main/java/com/facebook/drawee/

- Drawees are spaces in which images are rendered. These are made up of three components, like a Model-View-Controller (MVC) framework. (reference: <https://frescolib.org/docs/concepts.html>)

v2.5.0 ▾

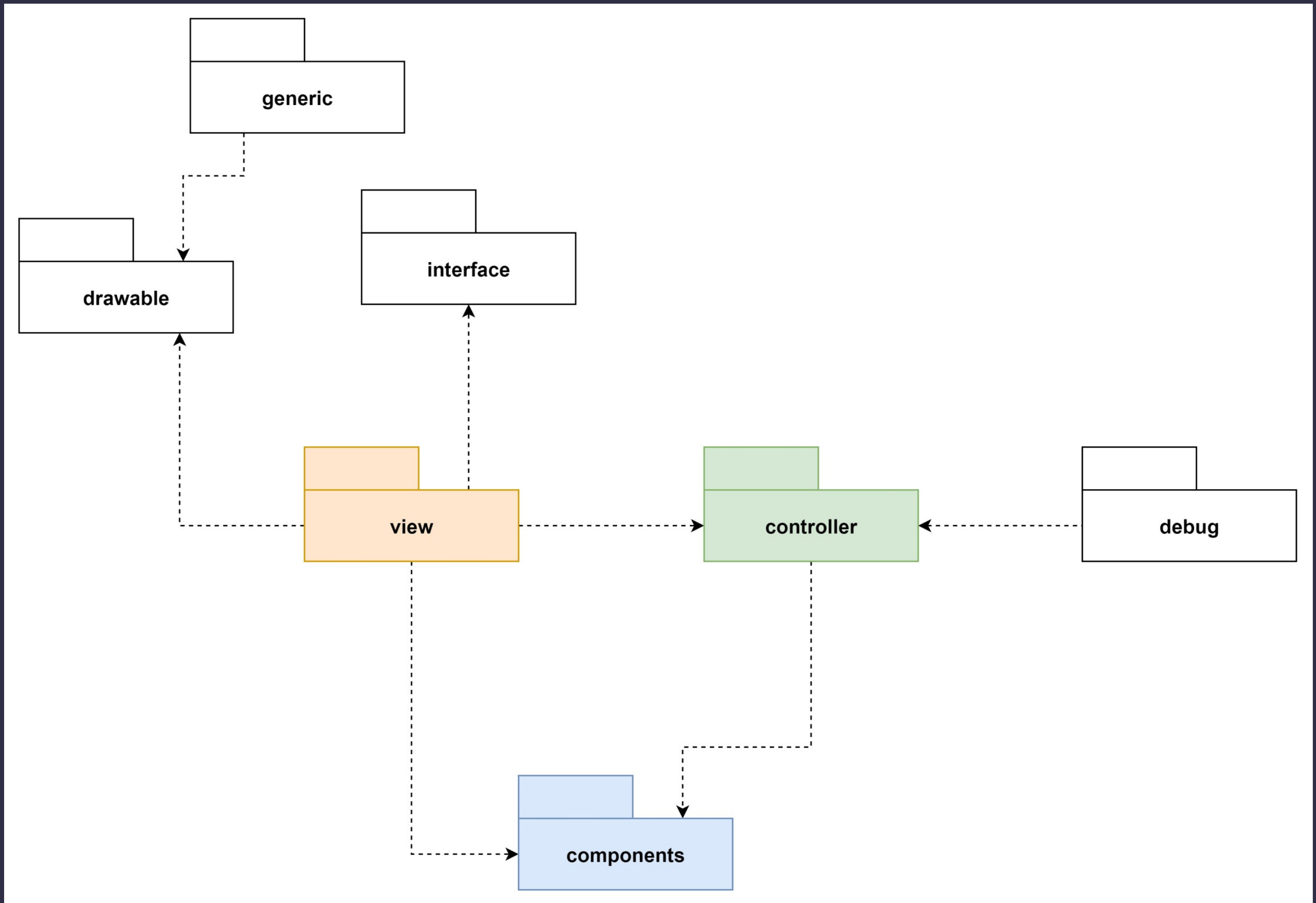
fresco / drawee / src / main / java / com / facebook / drawee /

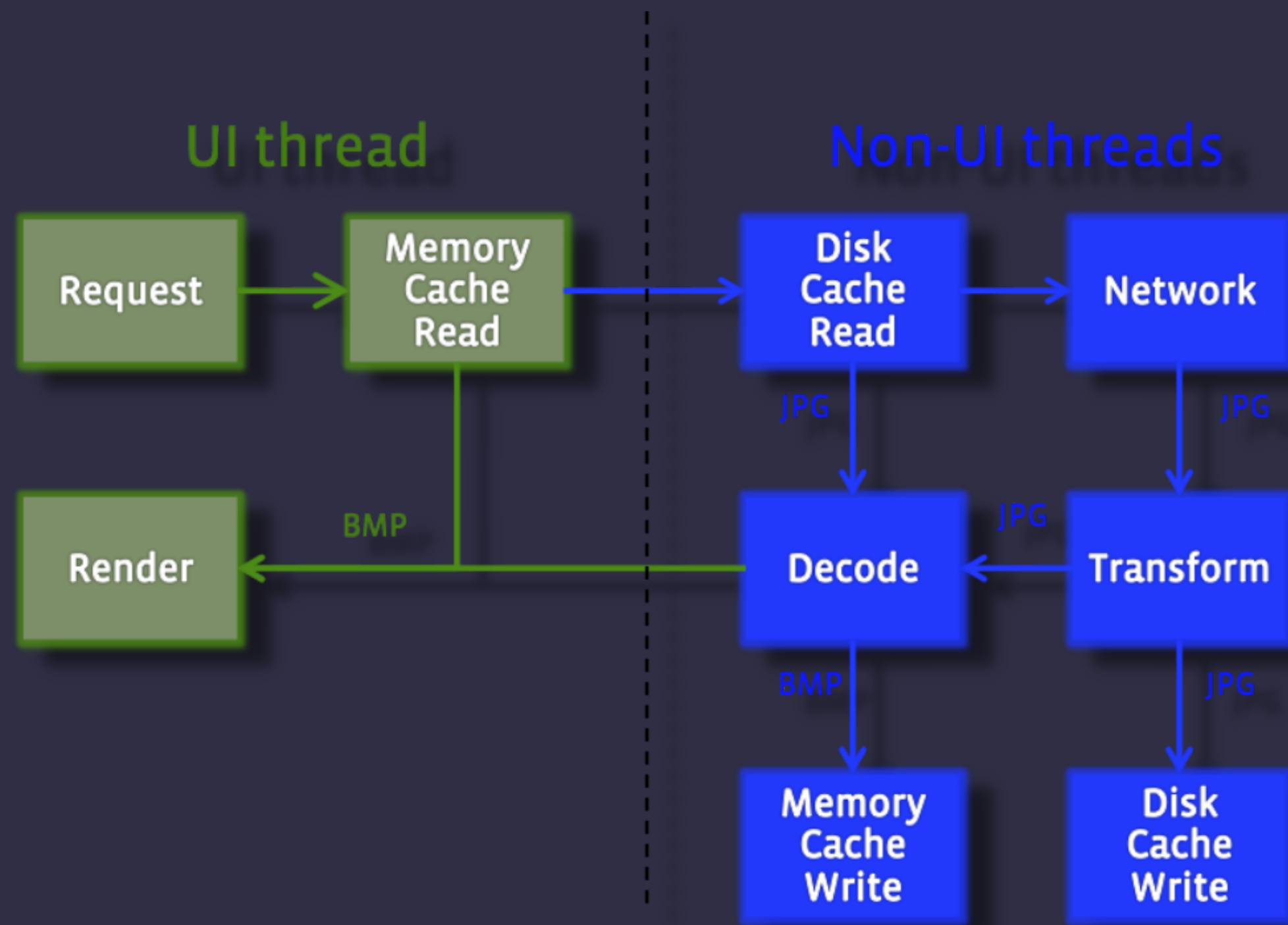
Go to file

 oprisnik and facebook-github-bot Added option to disable Drawable mutations ...

✓ 3ba9ff7 on Jun 2 [History](#)

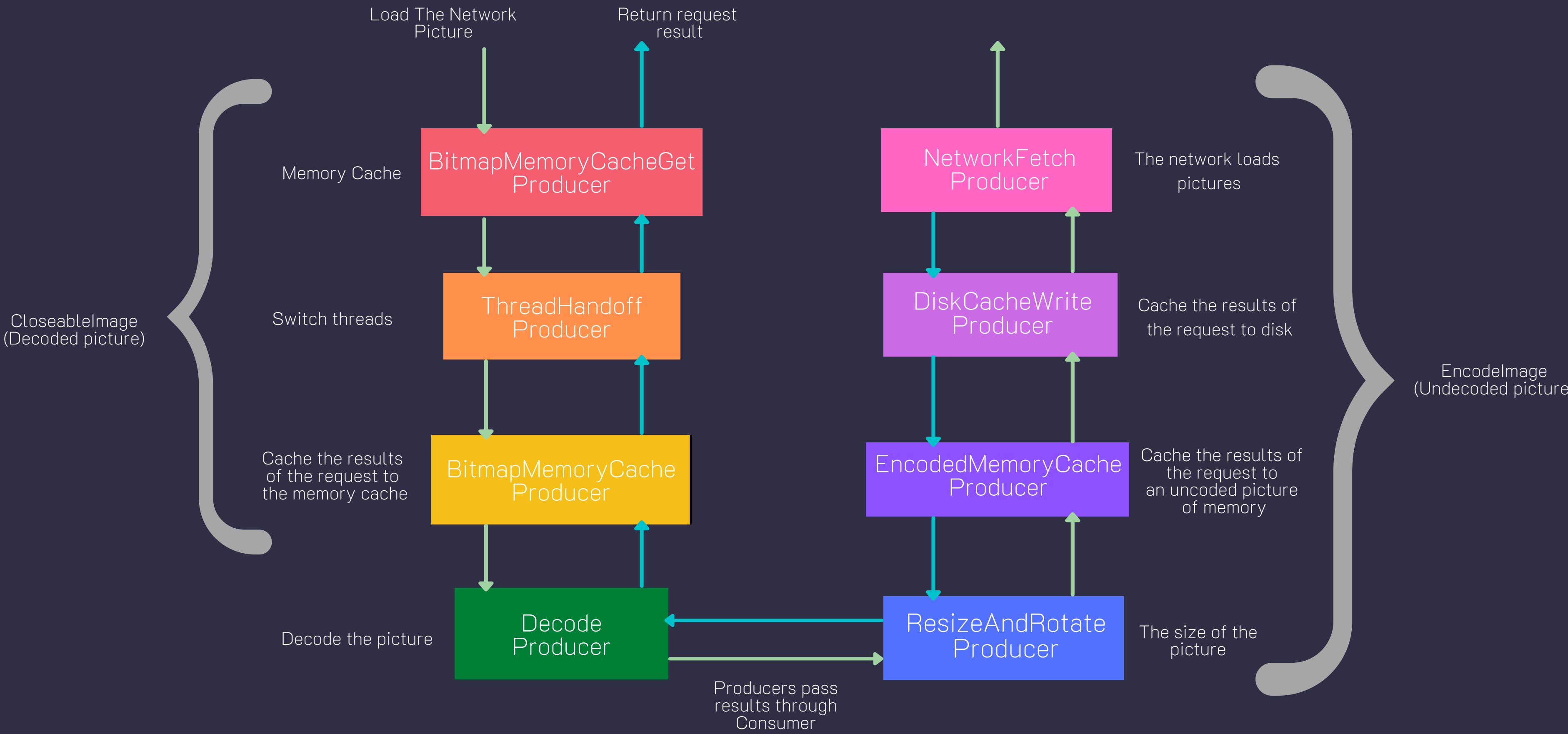
| | | |
|--|--|---------------|
| .. | | |
|  components | Replace Mode.LOCAL with Mode.STRICT in some files | 13 months ago |
|  controller | On fade listener: add on shown callback | 10 months ago |
|  debug | Add nullable annotations and mark classes nullsafe | 11 months ago |
|  drawable | Added option to disable Drawable mutations | 6 months ago |
|  generic | Replace Mode.LOCAL with Mode.STRICT in some files | 13 months ago |
|  gestures | Replace Mode.LOCAL with Mode.STRICT in some files | 13 months ago |
|  interfaces | fix DraweeHierarchy's java doc | 17 months ago |
|  view | Added Nullsafe annotation to EffectIcon based on https://fb.workplace... | 9 months ago |

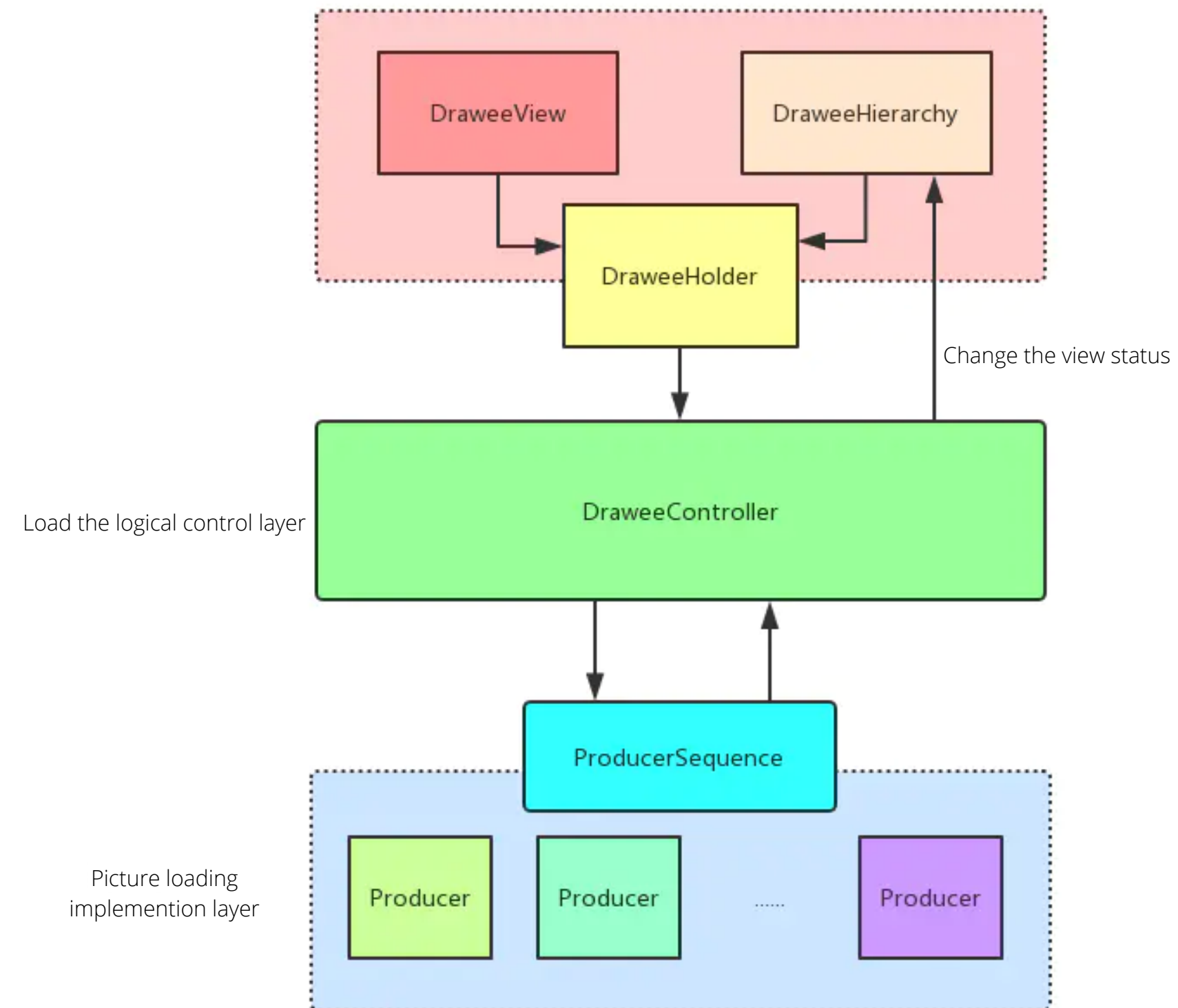




Pipes and filter

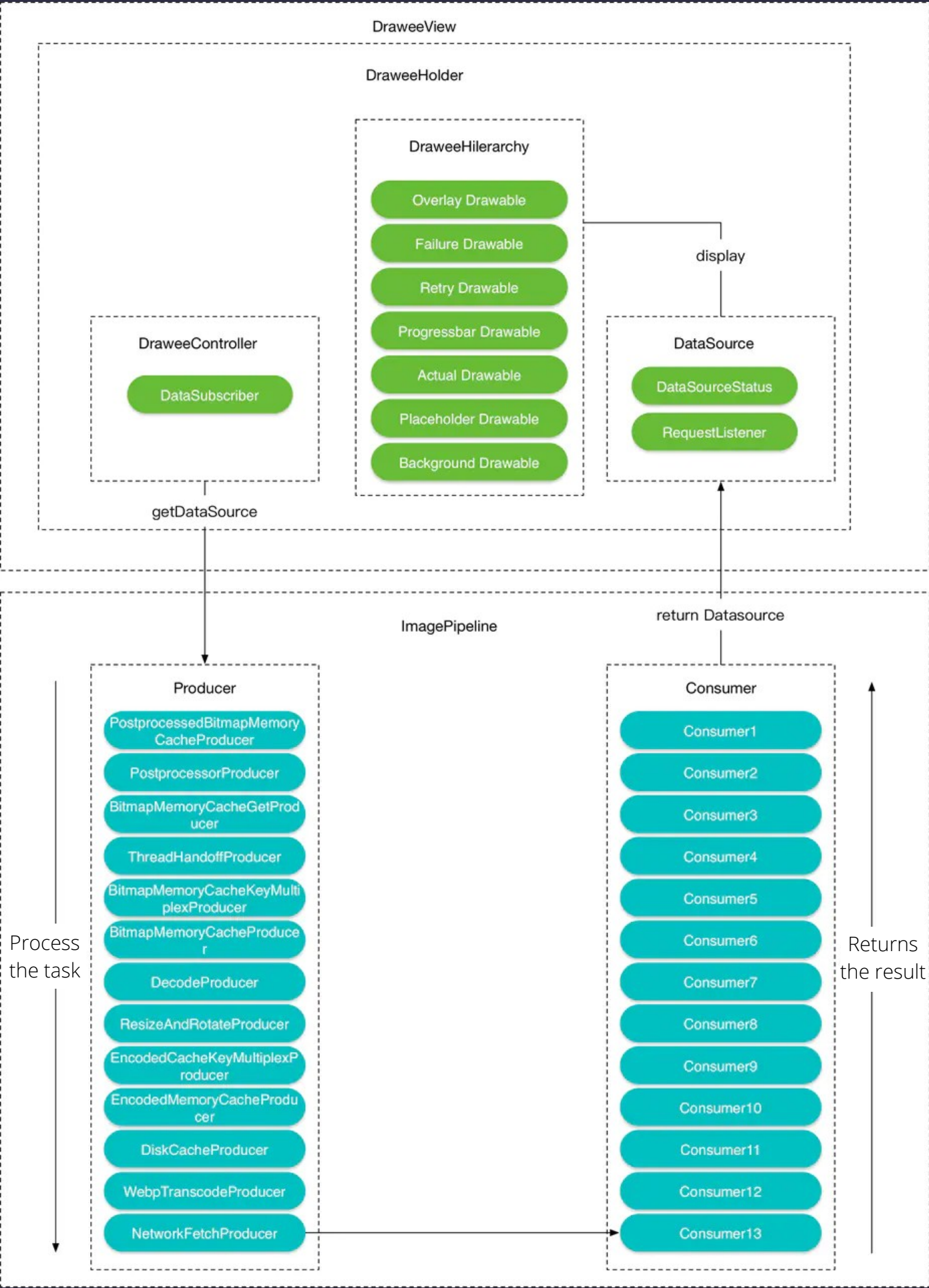
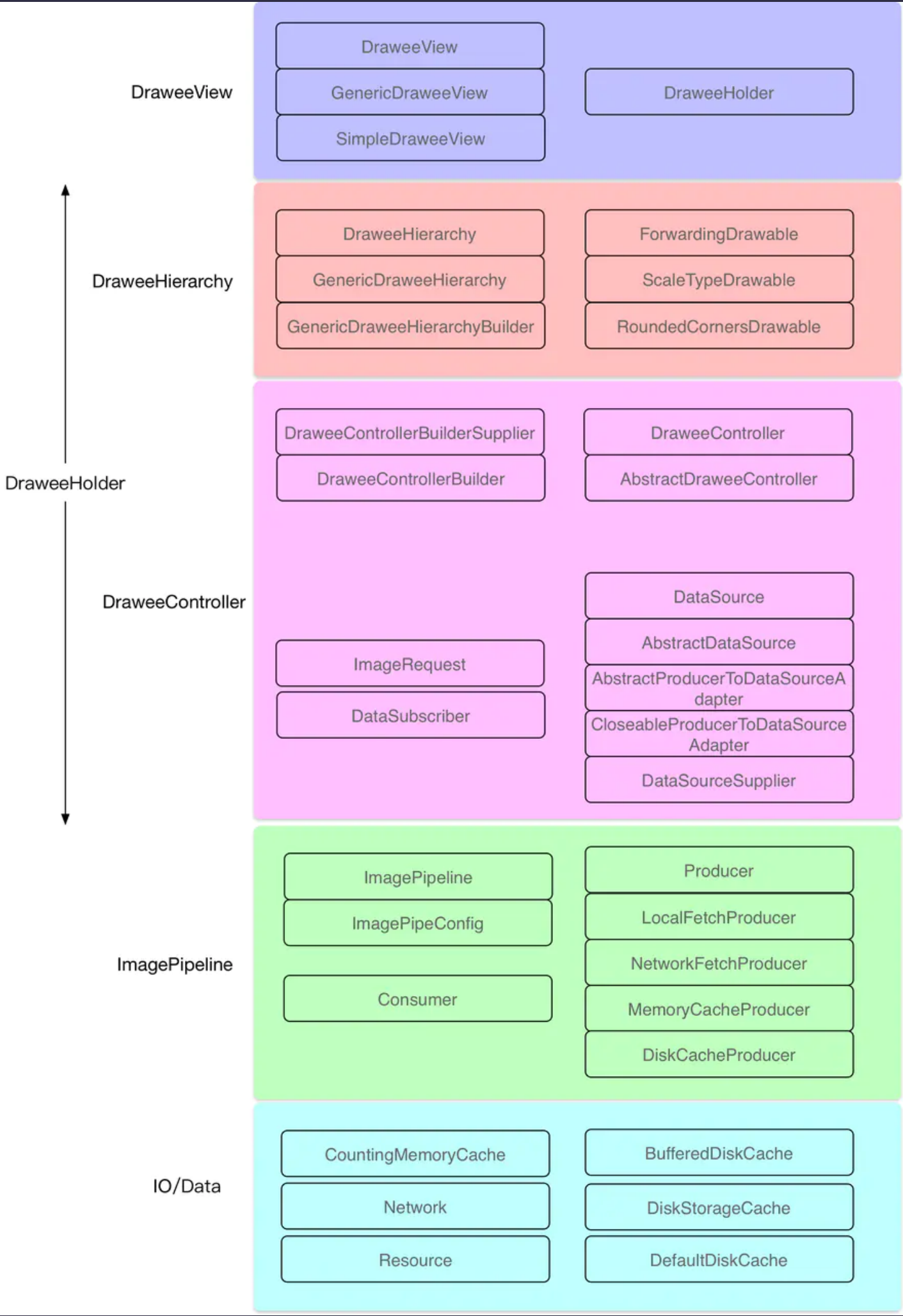
<https://frescolib.org/docs/intro-image-pipeline.html>





Layer

Ref : <https://programmerall.com/article/58202074767/>



Quality Attribute

#1 Quality Attributes

- On Android 4.x and lower, Images aren't stored in the Java heap, but in the ashmem heap. Intermediate byte buffers are also stored in the native heap. This leaves a lot more memory available for applications to use. It reduces the risk of OutOfMemoryErrors.

Ref : <https://frescolib.org/docs/caching.html>

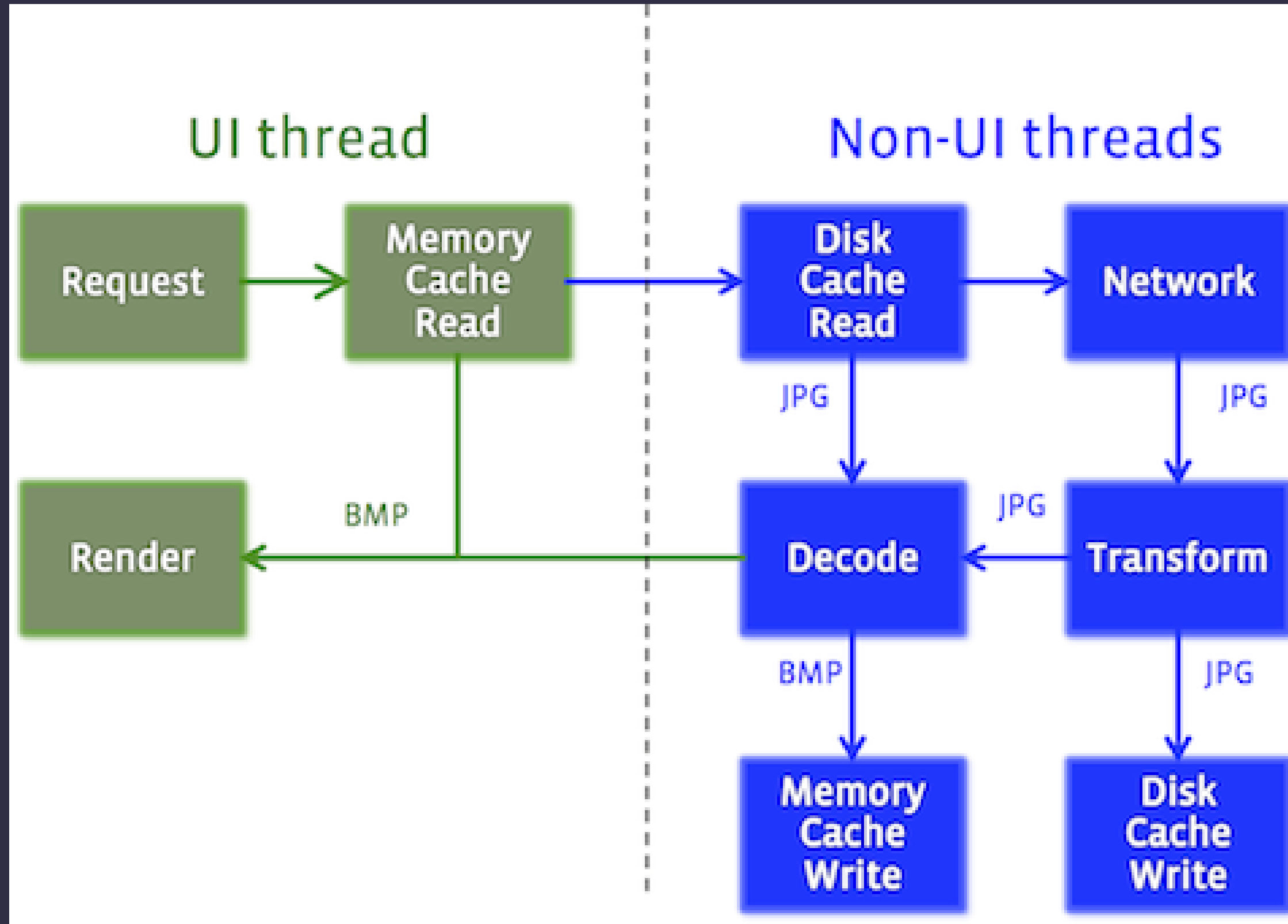
QA : Compatibility, Availability

#2 Quality Attributes

- Fresco memory cache is in three-levels:
 - Ready images for display or post-processing are decoded with the help of Bitmap.
 - Compressed images are stored in the original state within the memory with the help of an encoded memory cache.
 - Local storage stores compressed images in their original state.

With the help of the image pipeline class, Fresco manages the cache which provides the capability to check the existence of images in the cache, get the cached image and even delete them.

QA : Performance Efficiency



<https://frescolib.org/docs/intro-image-pipeline.html>

3# Quality Attributes

- A lot of functionality(streaming, drawing tools, memory management, etc)
- GIF support
- WebP and Animated Webp support
- Animated GIFs and WebPs can be challenging for apps. Each frame is a large Bitmap, and each animation is a series of frames. Fresco takes care of loading and disposing of frames and managing their memory.

Ref : <https://frescolib.org/docs/progressive-jpegs.html>

QA : Compatibility

Weakness

- The addition of layers adds up-front cost and complexity to a system.
- Layers contribute a performance penalty.
- Having large numbers of independent filters can add substantial amounts of computational overhead.



- Weak : Performance, Cost and Complexity, Scalability

#02

DESIGN PATTERN & UML

Factory Method

CacheKeyFactory.java

CLICK HERE



DefaultCacheKeyFactory.java

CLICK HERE



FlipperCacheKeyFactory.java

CLICK HERE



interface

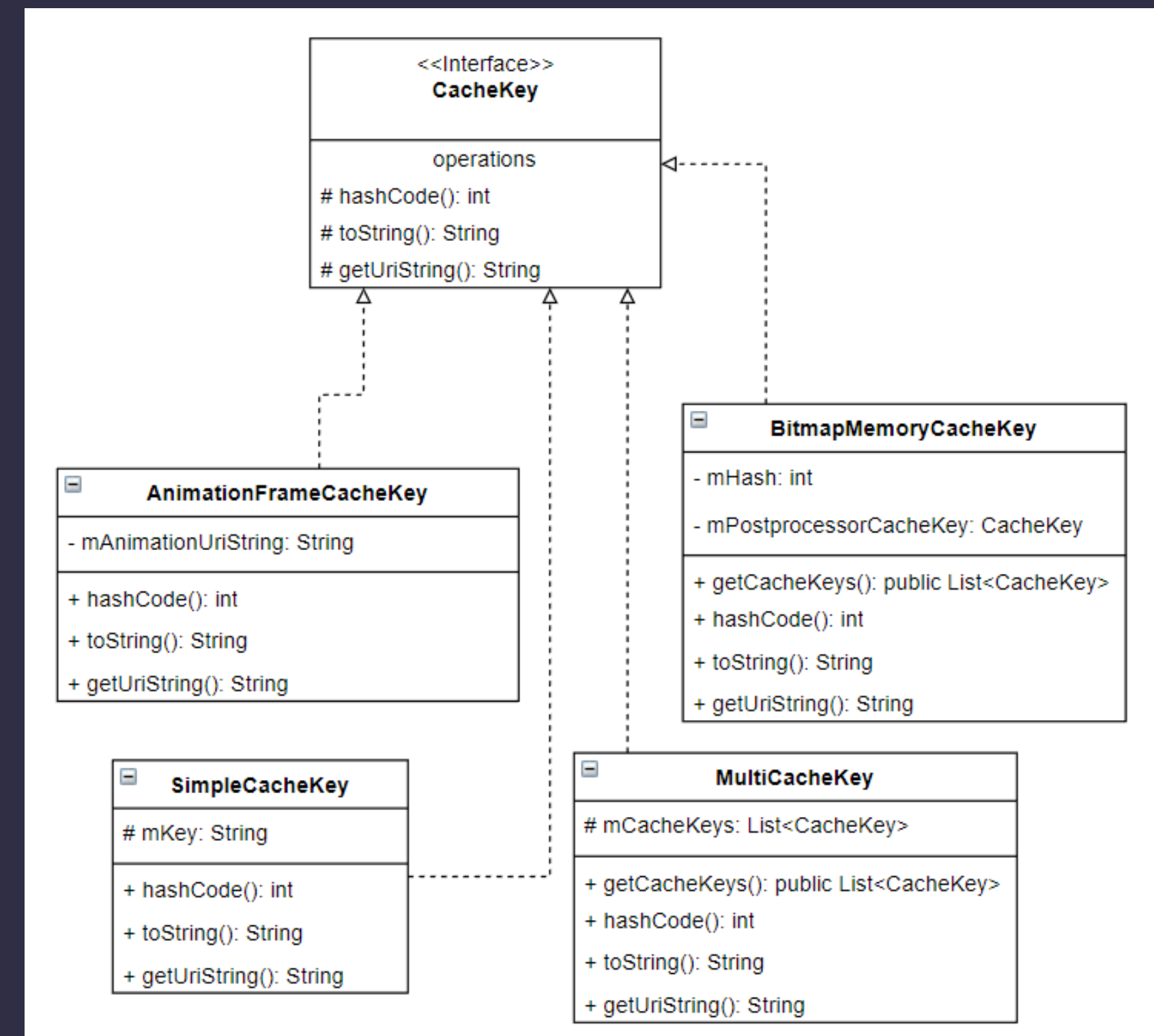
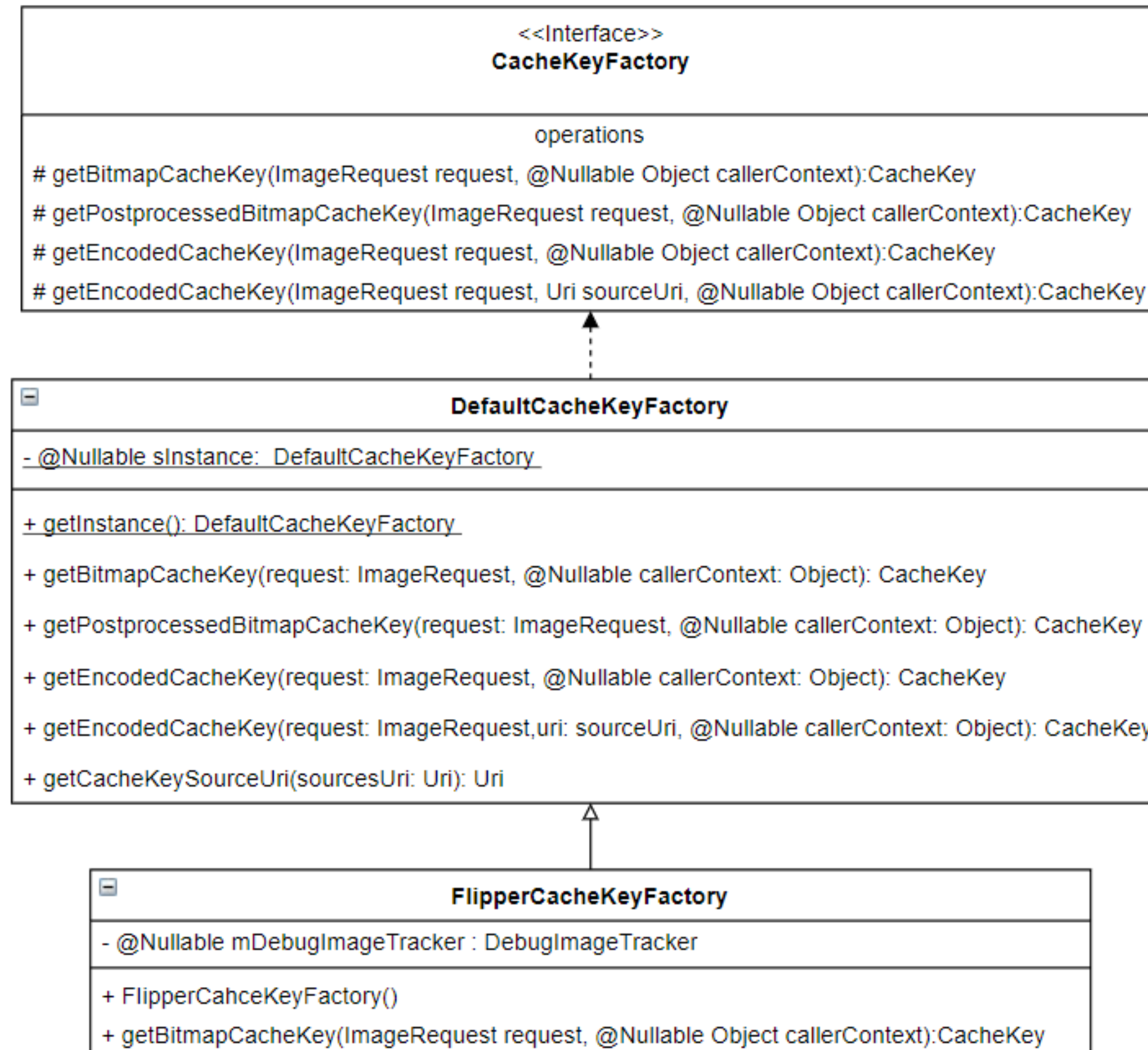
```
15  /** Factory methods for creating cache keys for the pipeline. */
16  public interface CacheKeyFactory {
17
18      /** @return {@link CacheKey} for doing bitmap cache lookups in the pipeline. */
19      CacheKey getBitmapCacheKey(ImageRequest request, @Nullable Object callerContext);
```

implement

```
18  /** Default implementation of {@link CacheKeyFactory}. */
19  @Nullable(Nullable.Mode.LOCAL)
20  public class DefaultCacheKeyFactory implements CacheKeyFactory {
33      @Override
34      public CacheKey getBitmapCacheKey(ImageRequest request, @Nullable Object callerContext) {
35          return new BitmapMemoryCacheKey(
36              getCacheKeySourceUri(request.getSourceUri()).toString(),
37              request.getResizeOptions(),
38              request.getRotationOptions(),
39              request.getImageDecodeOptions(),
40              null,
41              null,
42              callerContext);
```

extends

```
16  @Nullable(Nullable.Mode.LOCAL)
17  public class FlipperCacheKeyFactory extends DefaultCacheKeyFactory {
25      @Override
26      public CacheKey getBitmapCacheKey(ImageRequest request, @Nullable Object callerContext) {
27          CacheKey bitmapCacheKey = super.getBitmapCacheKey(request, callerContext);
28          if (mDebugImageTracker != null) {
29              mDebugImageTracker.trackImage(request, bitmapCacheKey);
30          }
31          return bitmapCacheKey;
32      }
33  }
```



Builder

fresco/drawee/src/main/java/com/facebook/drawee/interfaces/SimpleDraweeControllerBuilder.java

CLICK HERE



```
16 public interface SimpleDraweeControllerBuilder {
17
18     /** Sets the caller context. */
19     SimpleDraweeControllerBuilder setCallerContext(Object callerContext);
20
21     /** Sets the uri. */
22     SimpleDraweeControllerBuilder setUri(Uri uri);
23
24     /** Sets the uri from a string. */
25     SimpleDraweeControllerBuilder setUri(@Nullable String uriString);
26
27     /** Sets the old controller to be reused if possible. */
28     SimpleDraweeControllerBuilder setOldController(@Nullable DraweeController oldController);
29
30     /** Builds the specified controller. */
31     DraweeController build();
32 }
```

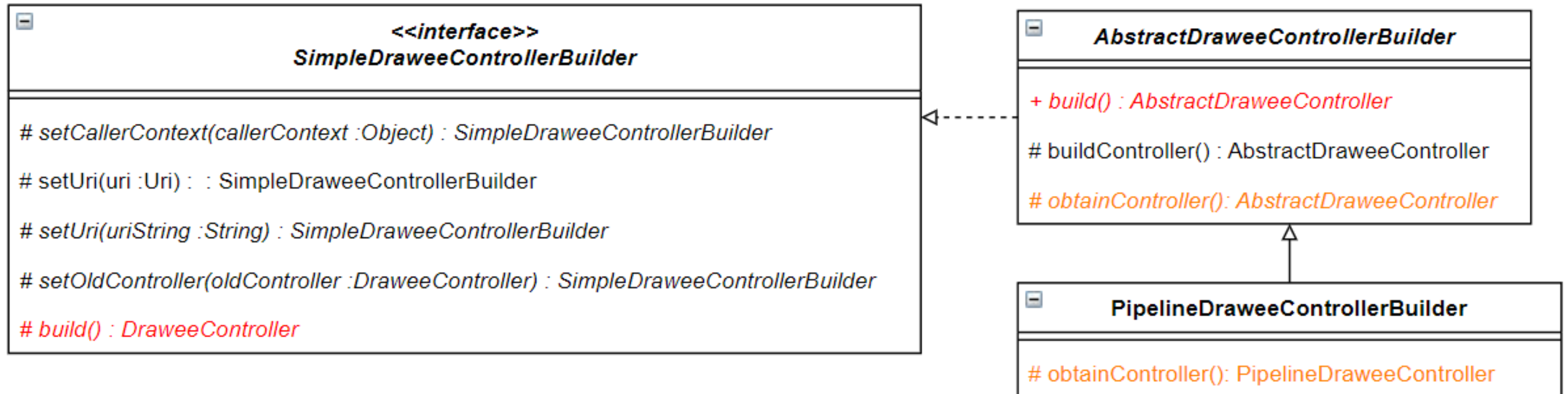
```
public AbstractDraweeController build() {
    validate();

    // if only a low-res request is specified, treat it as a final request.
    if (mImageRequest == null && mMultiImageRequests == null && mLowResImageRequest != null) {
        mImageRequest = mLowResImageRequest;
        mLowResImageRequest = null;
    }

    return buildController();
}
```

```
protected abstract AbstractDraweeController obtainController();
```

Builder



Abstract Factory

PlatformBitmapFactory.java

CLICK HERE



ArtBitmapFactory.java

CLICK HERE



GingerbreadBitmapFactory.java

CLICK HERE



abstract

```
24 @Nullable(Nullable.Mode.STRICT)
25 public abstract class PlatformBitmapFactory {
698     public abstract CloseableReference<Bitmap> createBitmapInternal(
699         int width, int height, Bitmap.Config bitmapConfig);
```

abstract method

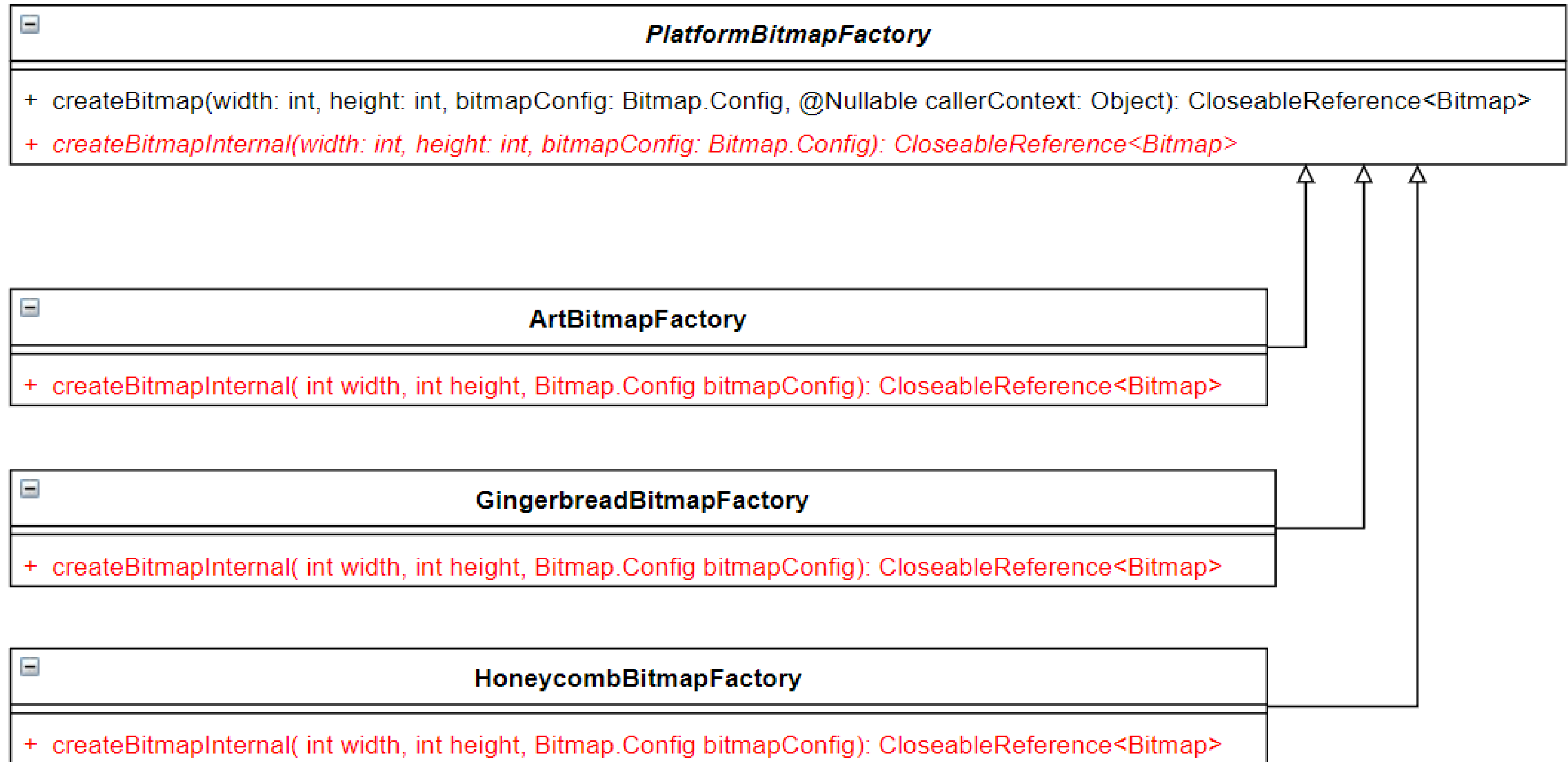
extends

```
24 @ThreadSafe
25 public class ArtBitmapFactory extends PlatformBitmapFactory {
46     @Override
47     public CloseableReference<Bitmap> createBitmapInternal(
48         int width, int height, Bitmap.Config bitmapConfig) {
49         int sizeInBytes = BitmapUtil.getSizeInByteForBitmap(width, height, bitmapConfig);
50         Bitmap bitmap = mBitmapPool.get(sizeInBytes);
51         Preconditions.checkArgument(
52             bitmap.getAllocationByteCount()
53                 >= width * height * BitmapUtil.getPixelSizeForBitmapConfig(bitmapConfig));
54         bitmap.reconfigure(width, height, bitmapConfig);
55         return mCloseableReferenceFactory.create(bitmap, mBitmapPool);
```

extends

```
16 public class GingerbreadBitmapFactory extends PlatformBitmapFactory {
28     @Override
29     public CloseableReference<Bitmap> createBitmapInternal(
30         int width, int height, Bitmap.Config bitmapConfig) {
31         Bitmap bitmap = Bitmap.createBitmap(width, height, bitmapConfig);
32         return CloseableReference.of(bitmap, SimpleBitmapReleaser.getInstance());
```

Abstract Factory



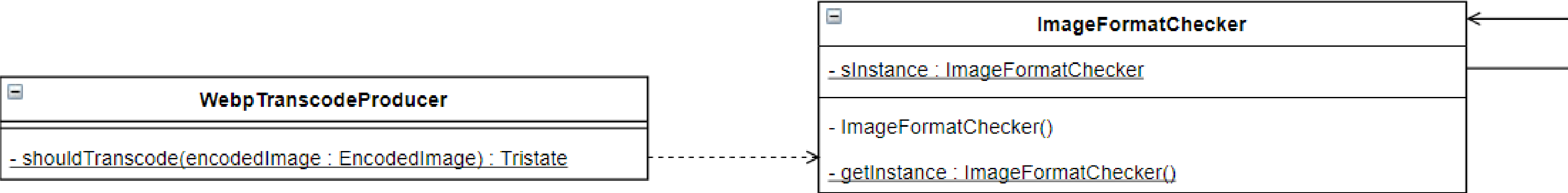
Singleton

fresco/samples/showcase/src/main/java/com/facebook/fresco/samples/showcase/misc/DebugOverlaySupplierSingleton.java

CLICK HERE



```
public class ImageFormatChecker {  
  
    private static ImageFormatChecker sInstance;  
  
    private int mMaxHeaderLength;  
  
    @Nullable private List<ImageFormat.FormatChecker> mCustomImageFormatCheckers;  
  
    private final DefaultImageFormatChecker mDefaultFormatChecker = new DefaultImageFormatChecker();  
  
    private ImageFormatChecker() {  
        updateMaxHeaderLength();  
    }  
  
    public static synchronized ImageFormatChecker getInstance() {  
        if (sInstance == null) {  
            sInstance = new ImageFormatChecker();  
        }  
        return sInstance;  
    }  
}
```



MEMBERS



62010318
ทศภณ สิงสันจิตร

62010922
สหทัศน์ ลีวัฒนา

62010722
ภูมิภัทร เชื้อลื้อ

62010713
ภูฟ้า จันทรพัฒน์

62010434
ธีรัช เหล่าสุขสกุล

62010710
ภูตินันท์ เพ็ญสุข