ANALYSIS OF SOFTWARE ARCHITECTURE AND DESIGN

Software Architecture and Design
(Group 2)

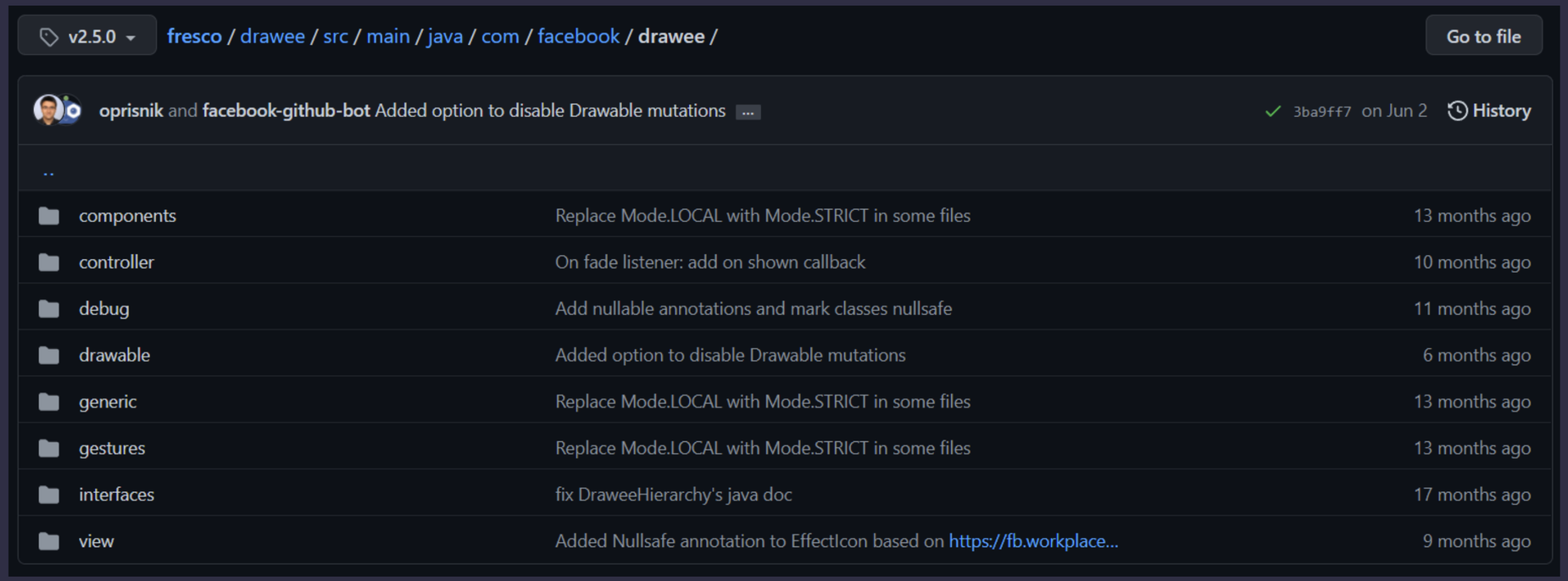
system for displaying images in Android applications. It takes care of image loading and display.
(Version 2.5.0)

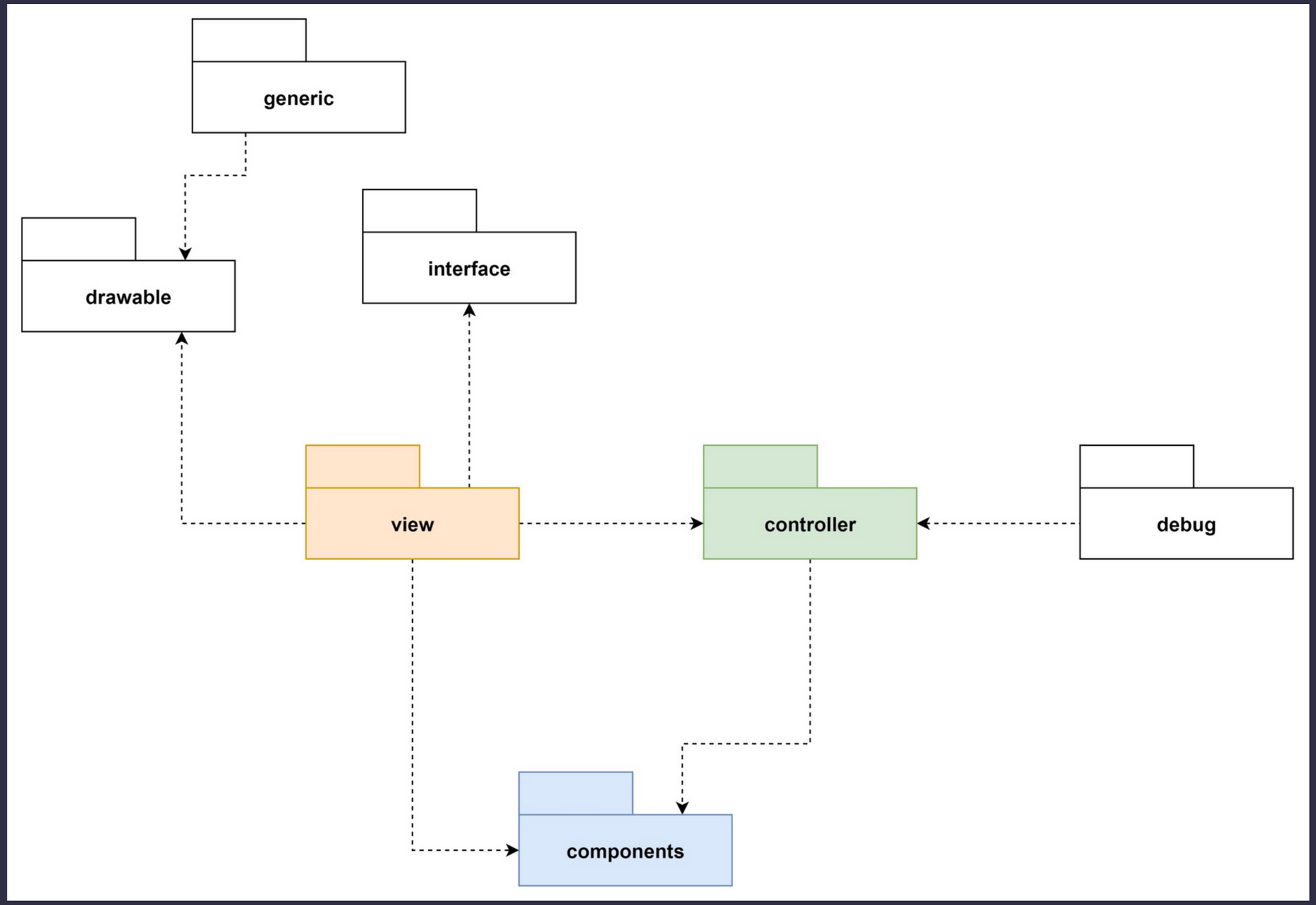
#01
ARCHITECTURE

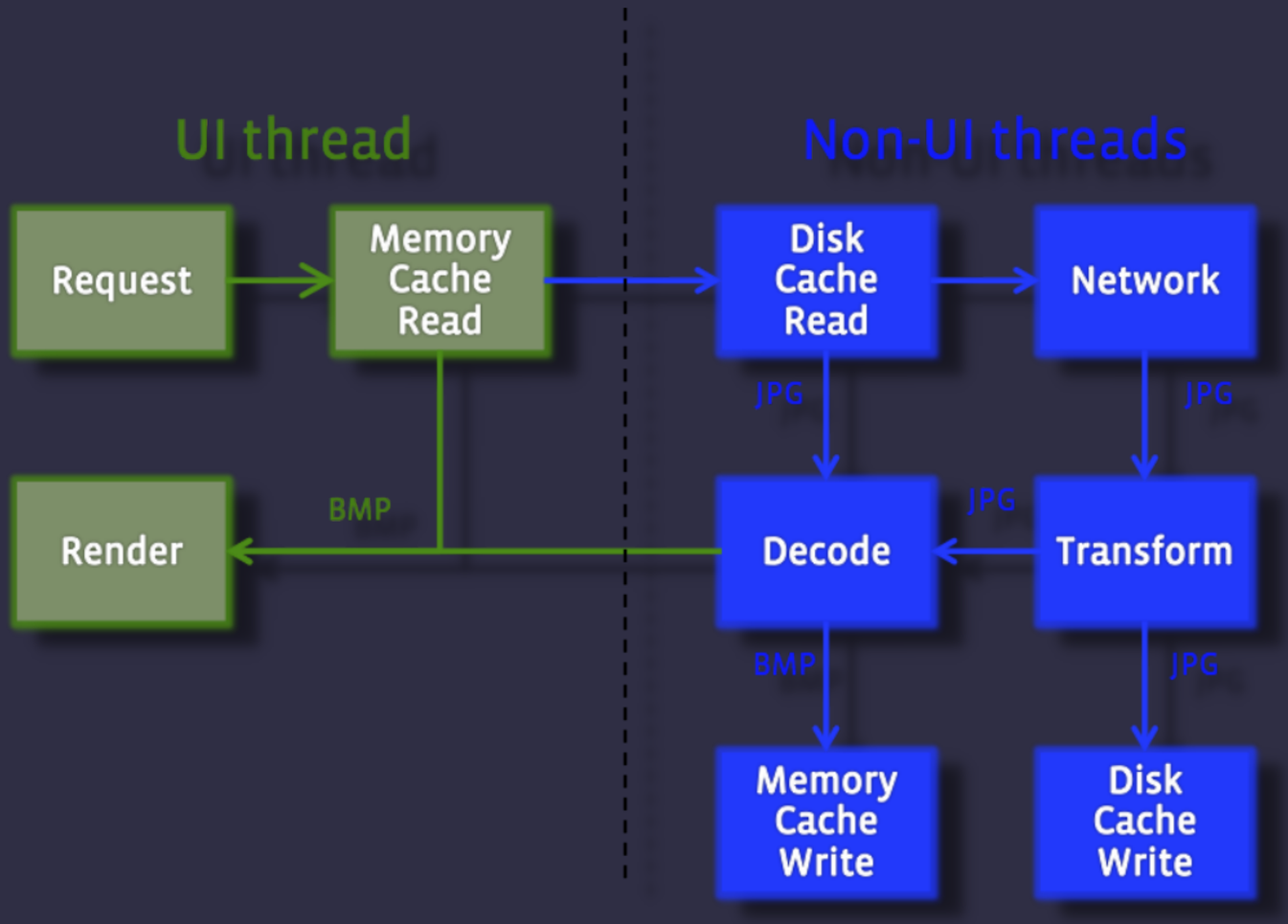
MVC: Module -View - Controller

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)

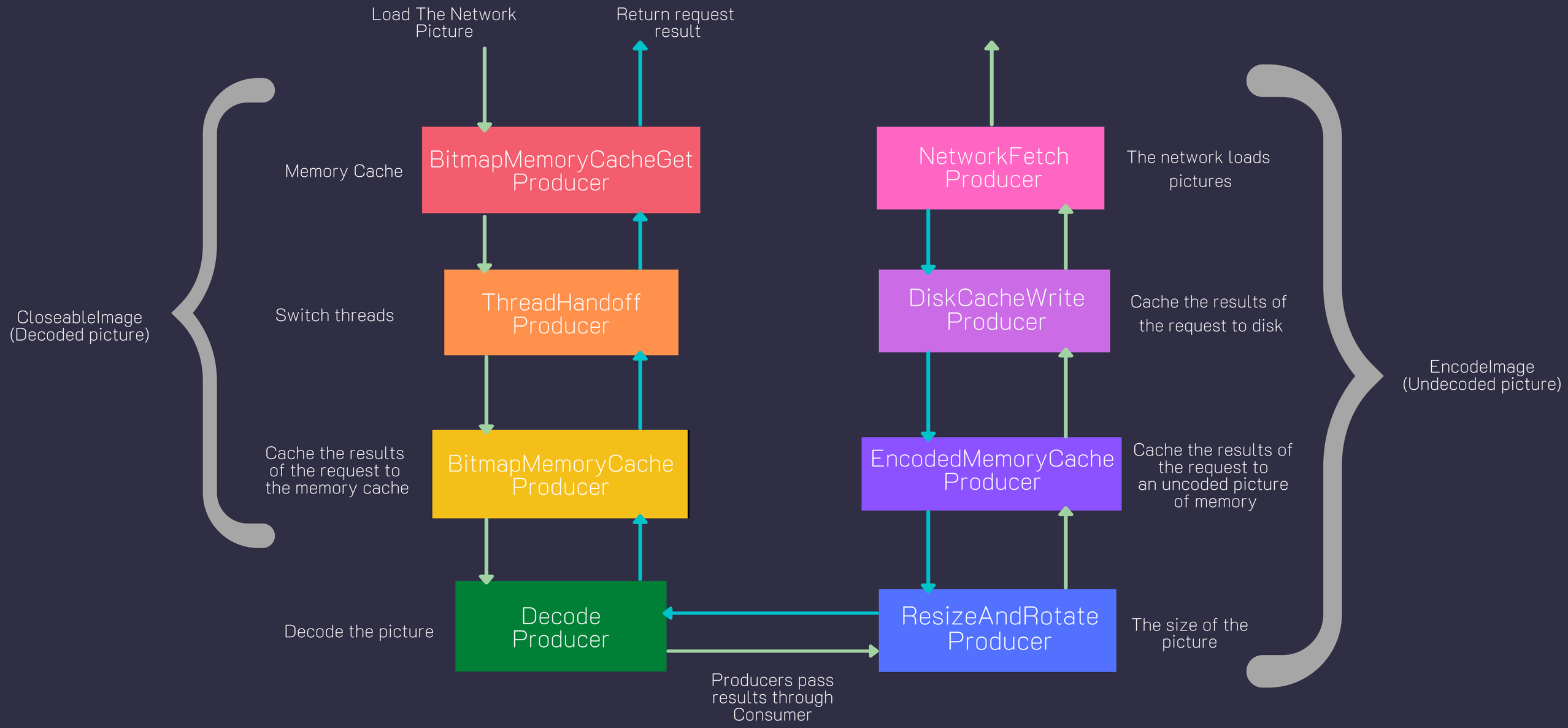


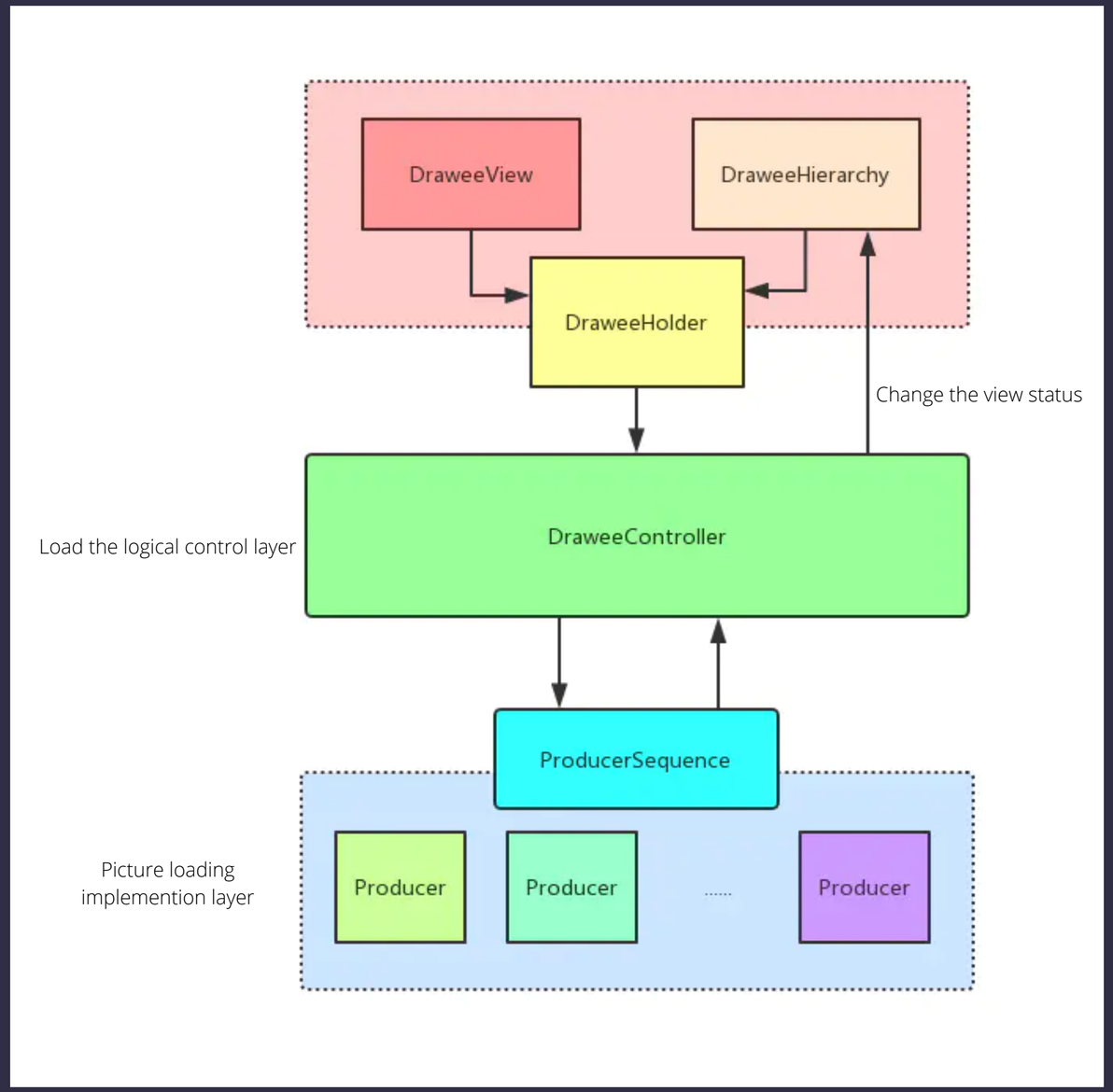

generic
drawable
interface
view
controller
debug
components



Pipes and filter

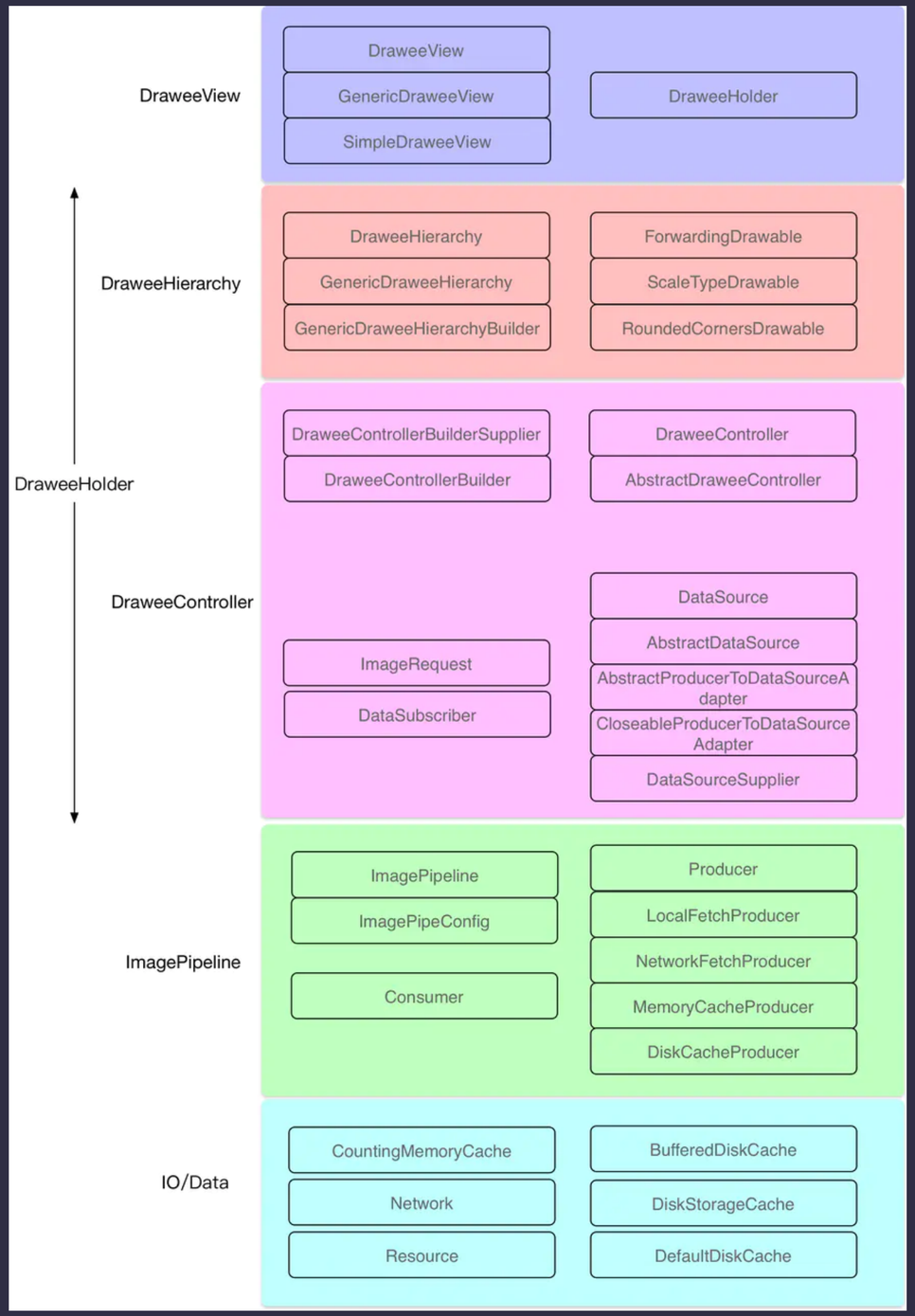
https://frescolib.org/docs/intro-image-pipeline.html

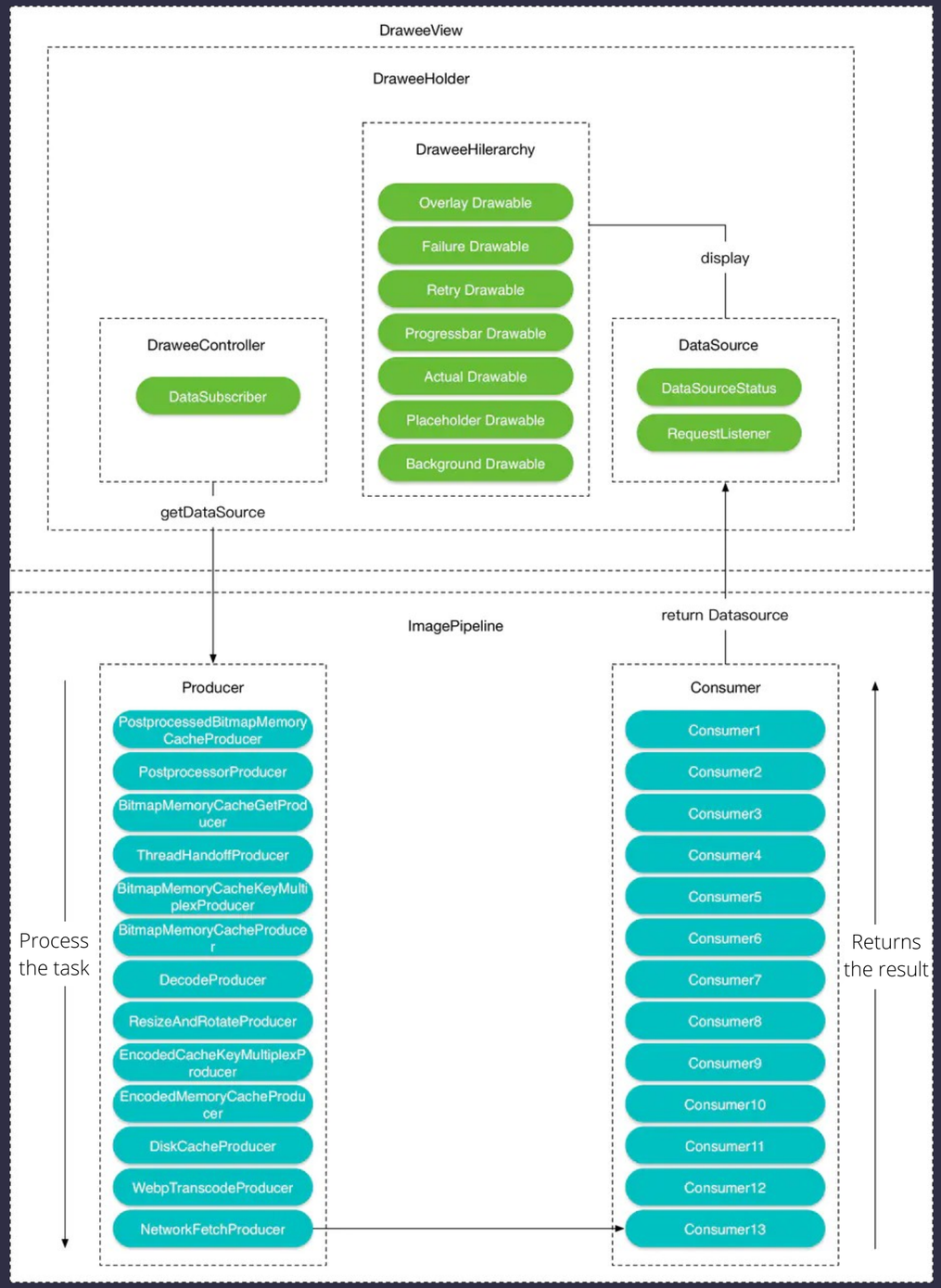

Load The Network Picture
Return request result
Memory Cache
BitmapMemoryCacheGet Producer
Switch threads
ThreadHandoff Producer
Cache the results of the request to the memory cache
BitmapMemoryCache Producer
Decode the picture
Decode Producer
CloseableImage (Decoded picture)
NetworkFetch Producer
The network loads pictures
Cache the results of the request to disk
DiskCacheWrite Producer
Cache the results of the request to an uncoded picture of memory
EncodedMemoryCache Producer
The size of the picture
ResizeAndRotate Producer
EncodeImage (Undecoded picture)
Producers pass results through Consumer



Layer

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


DraweeView
DraweeView
GenericDraweeView
SimpleDraweeView
DraweeHolder
DraweeHierarchy
DraweeHierarchy
GenericDraweeHierarchy
GenericDraweeHierarchyBuilder
ForwardingDrawable
ScaleTypeDrawable
RoundedCornersDrawable
DraweeHolder
DraweeControllerBuilderSupplier
DraweeControllerBuilder
DraweeController
AbstractDraweeController
DraweeController
DataSource
AbstractDataSource
AbstractProducerToDataSourceAdapter
CloseableProducerToDataSourceAdapter
DataSourceSupplier
ImageRequest
DataSubscriber
ImagePipeline
ImagePipeline
ImagePipeConfig
Consumer
Producer
LocalFetchProducer
NetworkFetchProducer
MemoryCacheProducer
DiskCacheProducer
IO/Data
CountingMemoryCache
Network
Resource
BufferedDiskCache
DiskStorageCache
DefaultDiskCache


DraweeView
DraweeHolder
DraweeHilerarchy
Overlay Drawable
Failure Drawable
Retry Drawable
Progressbar Drawable
Actual Drawable
Placeholder Drawable
Background Drawable
display
DraweeController
DataSubscriber
DataSource
DataSourceStatus
RequestListener
getDataSource
ImagePipeline
return Datasource
Producer
PostprocessedBitmapMemoryCacheProducer
PostprocessorProducer
BitmapMemoryCacheGetProducer
ThreadHandoffProducer
BitmapMemoryCacheKeyMultiplexProducer
BitmapMemoryCacheProducer
DecodeProducer
ResizeAndRotateProducer
EncodedCacheKeyMultiplexProducer
EncodedMemoryCacheProducer
DiskCacheProducer
WebpTranscodeProducer
NetworkFetchProducer
Consumer
Consumer1
Consumer2
Consumer3
Consumer4
Consumer5
Consumer6
Consumer7
Consumer8
Consumer9
Consumer10
Consumer11
Consumer12
Consumer13
Process the task
Returns the result

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

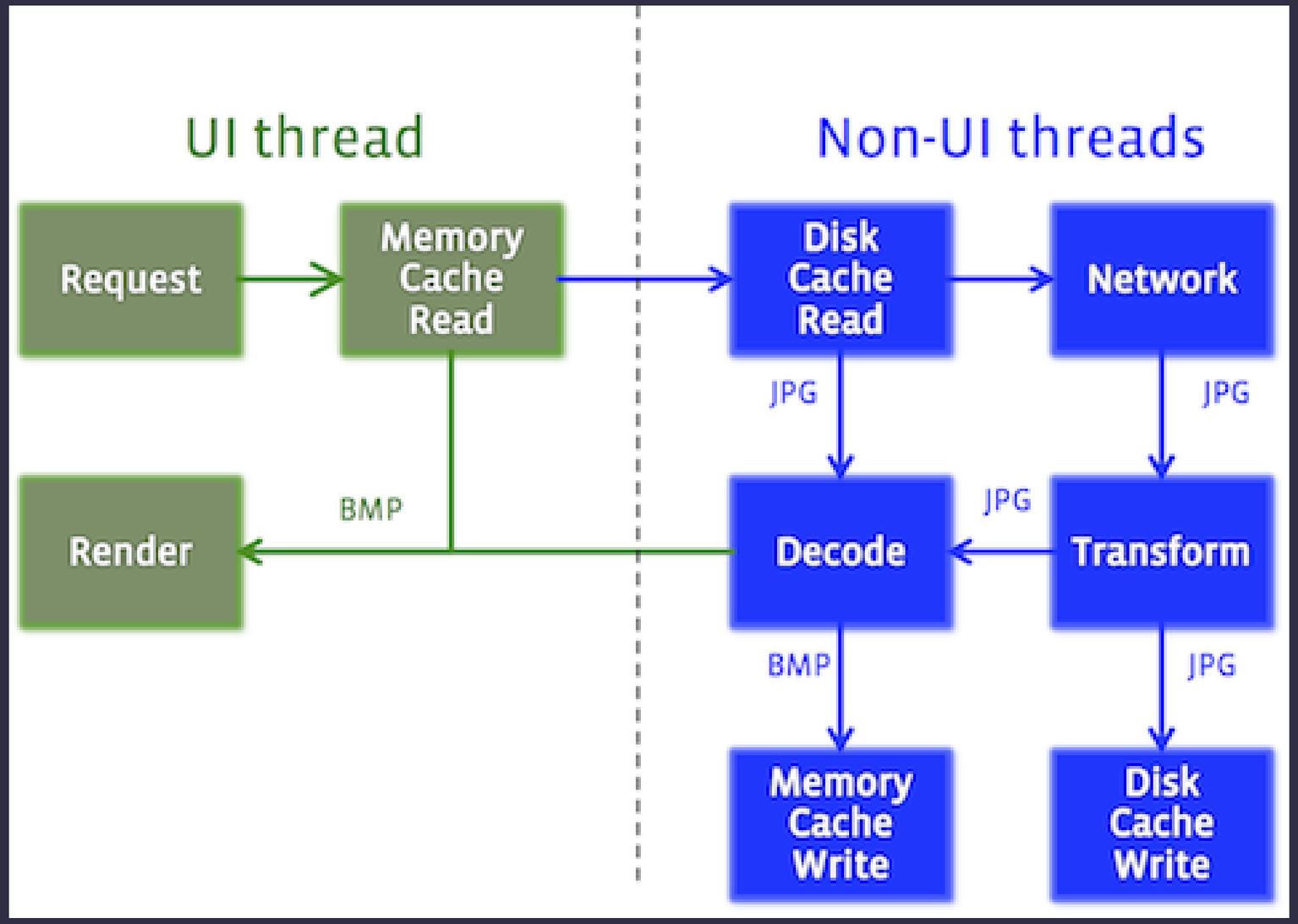
#2 Quality Attributes

- 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

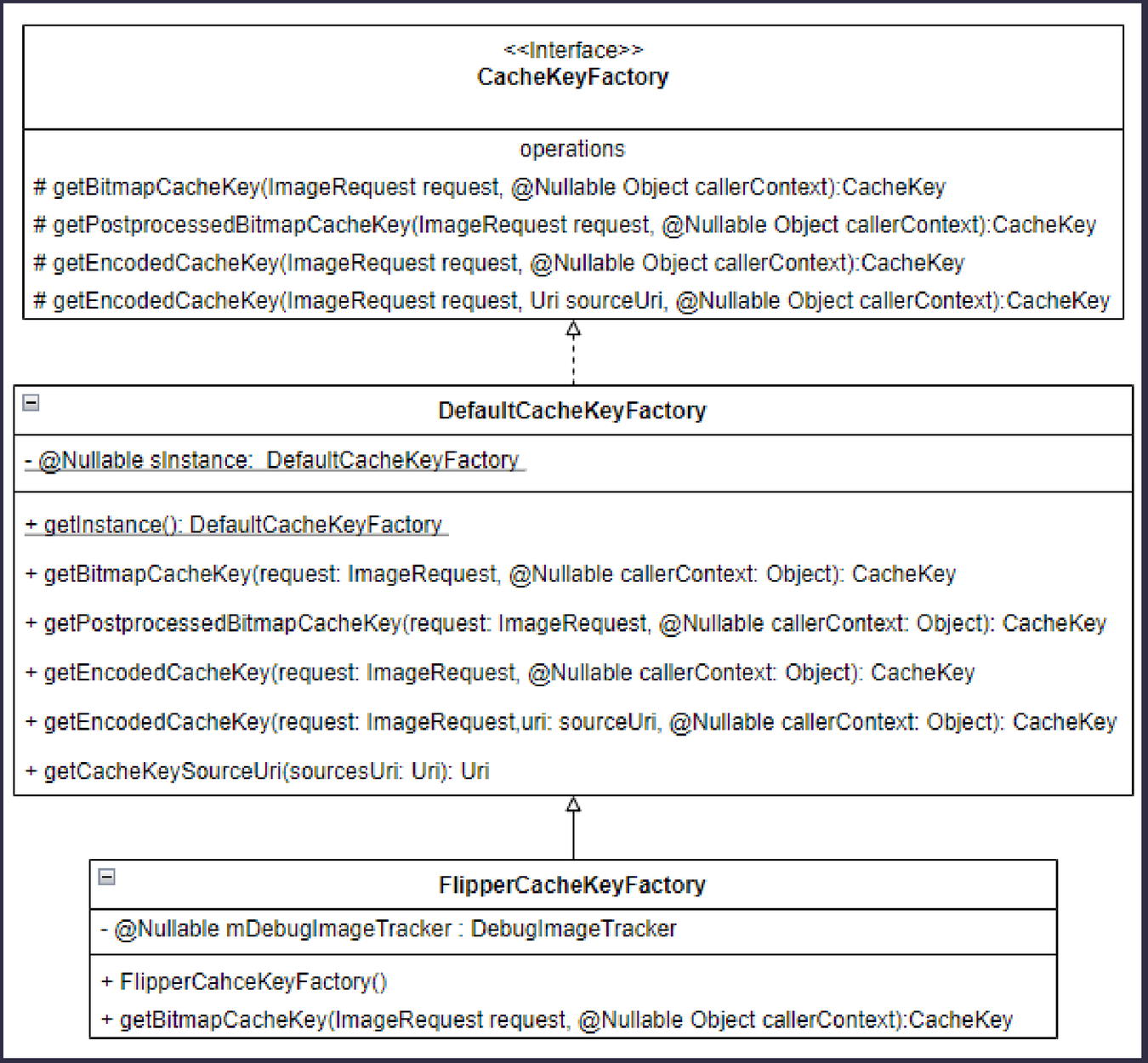
```java
/** Factory methods for creating cache keys for the pipeline. */
public interface CacheKeyFactory {

  /** @return {@link CacheKey} for doing bitmap cache lookups in the pipeline. */
  CacheKey getBitmapCacheKey(ImageRequest request, @Nullable Object callerContext);
```

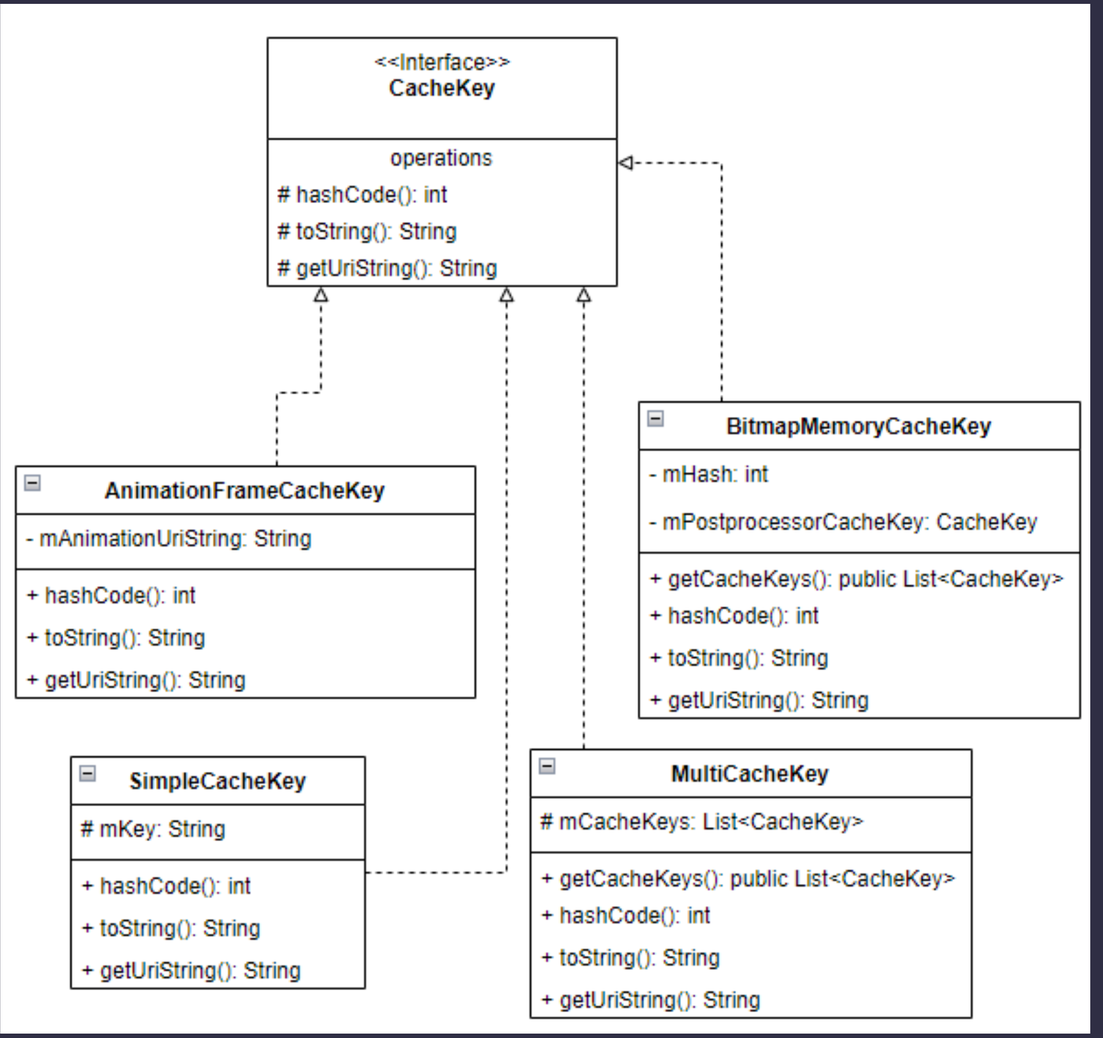
implement

```java
/** Default implementation of {@link CacheKeyFactory}. */
@Nullsafe(Nullsafe.Mode.LOCAL)
public class DefaultCacheKeyFactory implements CacheKeyFactory {
  @Override
  public CacheKey getBitmapCacheKey(ImageRequest request, @Nullable Object callerContext) {
    return new BitmapMemoryCacheKey(
        getCacheKeySourceUri(request.getSourceUri()).toString(),
        request.getResizeOptions(),
        request.getRotationOptions(),
        request.getImageDecodeOptions(),
        null,
        null,
        callerContext);
```

extends

```java
@Nullsafe(Nullsafe.Mode.LOCAL)
public class FlipperCacheKeyFactory extends DefaultCacheKeyFactory {
  @Override
  public CacheKey getBitmapCacheKey(ImageRequest request, @Nullable Object callerContext) {
    CacheKey bitmapCacheKey = super.getBitmapCacheKey(request, callerContext);
    if (mDebugImageTracker != null) {
      mDebugImageTracker.trackImage(request, bitmapCacheKey);
    }
    return bitmapCacheKey;
  }
}
```


<<Interface>>
CacheKeyFactory
operations
getBitmapCacheKey(ImageRequest request, @Nullable Object callerContext):CacheKey
getPostprocessedBitmapCacheKey(ImageRequest request, @Nullable Object callerContext):CacheKey
getEncodedCacheKey(ImageRequest request, @Nullable Object callerContext):CacheKey
getEncodedCacheKey(ImageRequest request, Uri sourceUri, @Nullable Object callerContext):CacheKey
DefaultCacheKeyFactory
- @Nullable sInstance: DefaultCacheKeyFactory
+ getInstance(): DefaultCacheKeyFactory
+ getBitmapCacheKey(request: ImageRequest, @Nullable callerContext: Object): CacheKey
+ getPostprocessedBitmapCacheKey(request: ImageRequest, @Nullable callerContext: Object): CacheKey
+ getEncodedCacheKey(request: ImageRequest, @Nullable callerContext: Object): CacheKey
+ getEncodedCacheKey(request: ImageRequest,uri: sourceUri, @Nullable callerContext: Object): CacheKey
+ getCacheKeySourceUri(sourcesUri: Uri): Uri
FlipperCacheKeyFactory
- @Nullable mDebugImageTracker : DebugImageTracker
+ FlipperCahceKeyFactory()
+ getBitmapCacheKey(ImageRequest request, @Nullable Object callerContext):CacheKey


<<Interface>>
CacheKey
operations
hashCode(): int
toString(): String
getUriString(): String
AnimationFrameCacheKey
- mAnimationUriString: String
+ hashCode(): int
+ toString(): String
+ getUriString(): String
BitmapMemoryCacheKey
- mHash: int
- mPostprocessorCacheKey: CacheKey
+ getCacheKeys(): public List<CacheKey>
+ hashCode(): int
+ toString(): String
+ getUriString(): String
SimpleCacheKey
mKey: String
+ hashCode(): int
+ toString(): String
+ getUriString(): String
MultiCacheKey
mCacheKeys: List<CacheKey>
+ getCacheKeys(): public List<CacheKey>
+ hashCode(): int
+ toString(): String
+ getUriString(): String

Builder

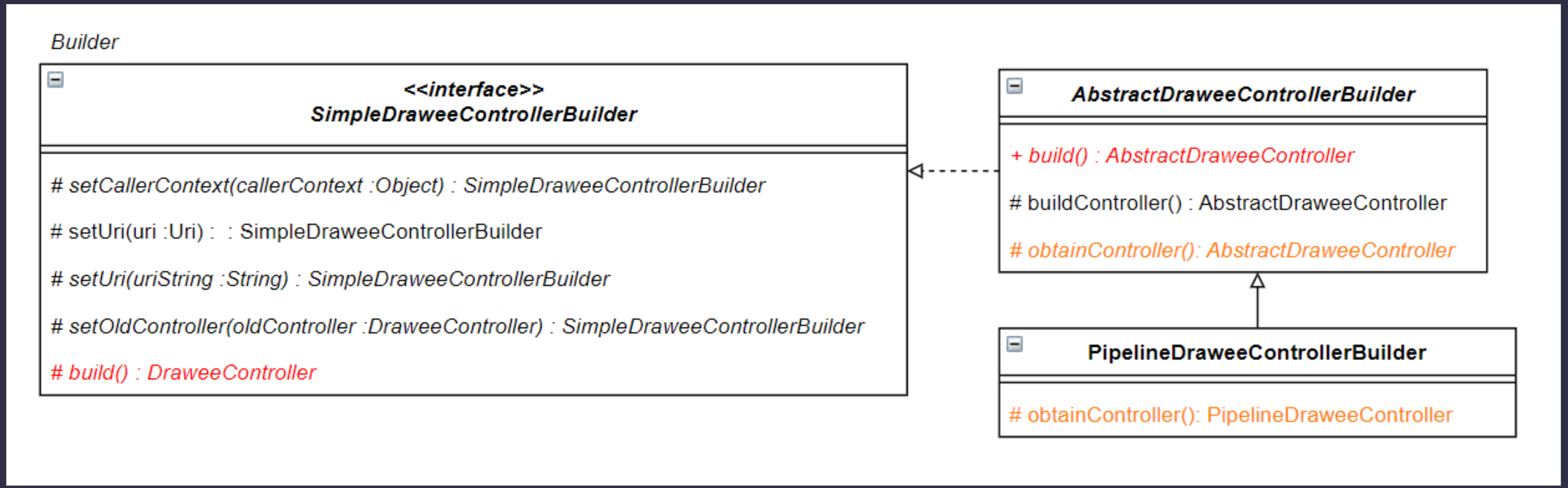
fresco/drawee/src/main/java/com/facebook/drawee/interfaces/SimpleDraweeControllerBuilder.java

CLICK HERE

```java
public interface SimpleDraweeControllerBuilder {

  /** Sets the caller context. */
  SimpleDraweeControllerBuilder setCallerContext(Object callerContext);

  /** Sets the uri. */
  SimpleDraweeControllerBuilder setUri(Uri uri);

  /** Sets the uri from a string. */
  SimpleDraweeControllerBuilder setUri(@Nullable String uriString);

  /** Sets the old controller to be reused if possible. */
  SimpleDraweeControllerBuilder setOldController(@Nullable DraweeController oldController);

  /** Builds the specified controller. */
  DraweeController build();
}
```

```java
public AbstractDraweeController build() {
  validate();                                   // AbstractDraweeControllerBuilder.java

  // 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();
}
```

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


Builder
<<interface>>
SimpleDraweeControllerBuilder
setCallerContext(callerContext :Object) : SimpleDraweeControllerBuilder
setUri(uri :Uri) : : SimpleDraweeControllerBuilder
setUri(uriString :String) : SimpleDraweeControllerBuilder
setOldController(oldController :DraweeController) : SimpleDraweeControllerBuilder
build() : DraweeController
AbstractDraweeControllerBuilder
+ build() : AbstractDraweeController
# buildController() : AbstractDraweeController
# obtainController(): AbstractDraweeController
PipelineDraweeControllerBuilder
obtainController(): PipelineDraweeController

Abstract Factory

PlatformBitmapFactory.java

CLICK HERE

ArtBitmapFactory.java

CLICK HERE

GingerbreadBitmapFactory.java

CLICK HERE

abstract

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

abstract method

extends

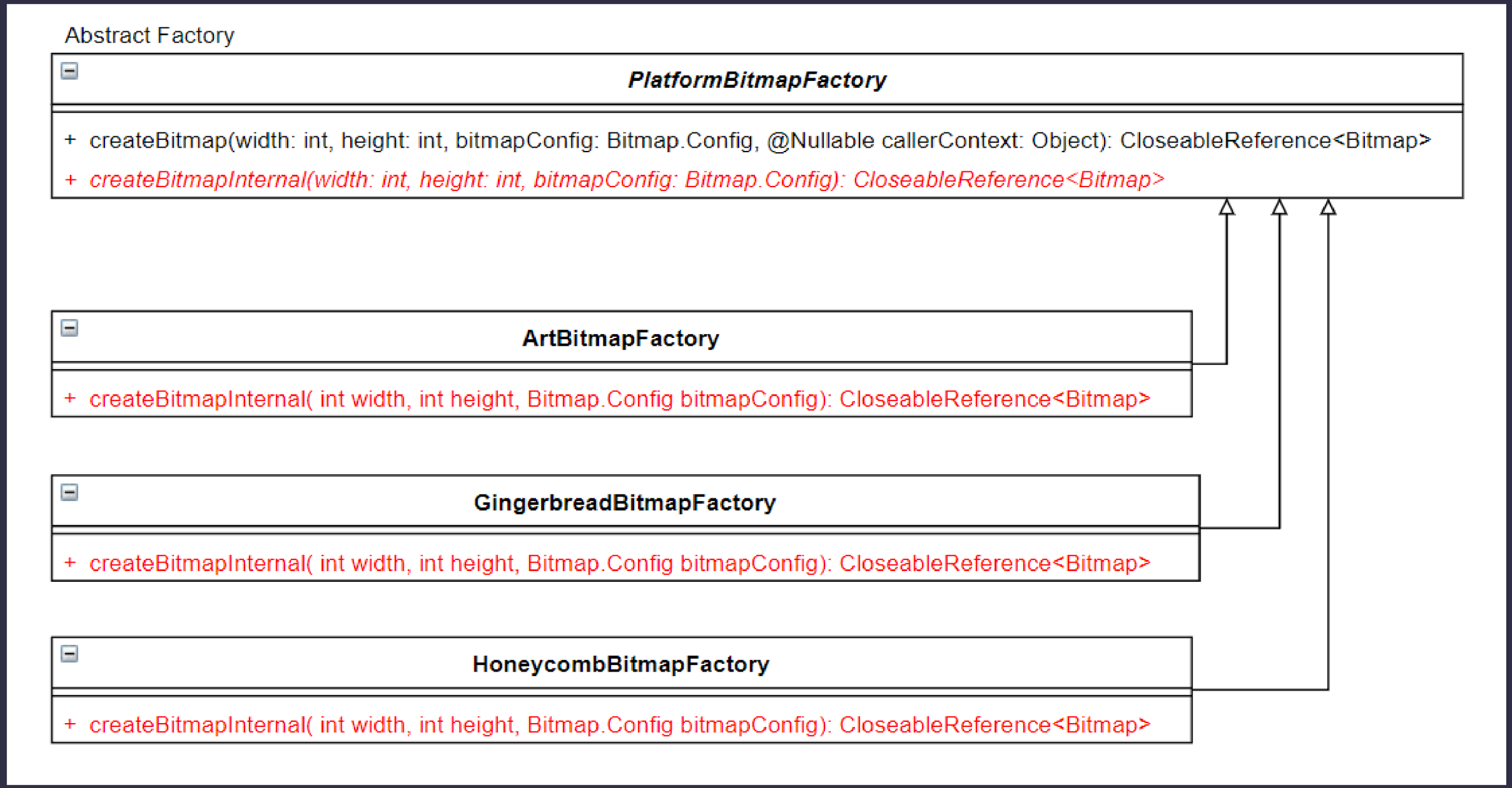
```
24  @ThreadSafe
25  public class ArtBitmapFactory extends PlatformBitmapFactory {
46   @Override
47   public CloseableReference<Bitmap> createBitmapInternal(
48       int width, int height, Bitmap.Config bitmapConfig) {
49     int sizeInBytes = BitmapUtil.getSizeInByteForBitmap(width, height, bitmapConfig);
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);
```

Override

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());
```

Override


Abstract Factory
PlatformBitmapFactory
+ createBitmap(width: int, height: int, bitmapConfig: Bitmap.Config, @Nullable callerContext: Object): CloseableReference<Bitmap>
+ createBitmapInternal(width: int, height: int, bitmapConfig: Bitmap.Config): CloseableReference<Bitmap>
ArtBitmapFactory
+ createBitmapInternal(int width, int height, Bitmap.Config bitmapConfig): CloseableReference<Bitmap>
GingerbreadBitmapFactory
+ createBitmapInternal(int width, int height, Bitmap.Config bitmapConfig): CloseableReference<Bitmap>
HoneycombBitmapFactory
+ createBitmapInternal(int width, int height, Bitmap.Config bitmapConfig): CloseableReference<Bitmap>

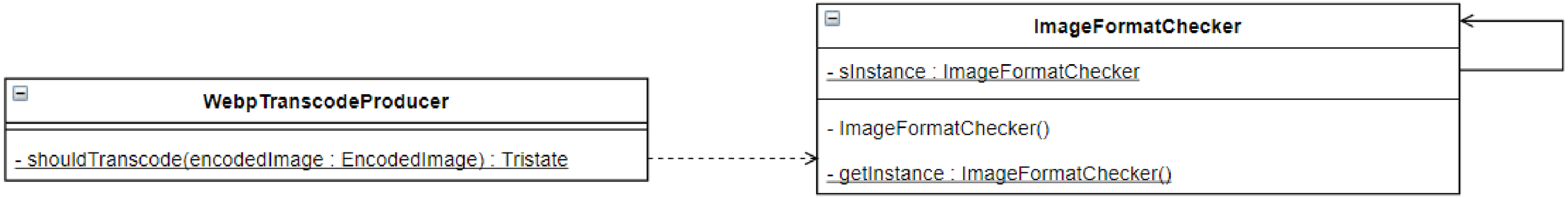
Singleton

fresco/imagepipeline-base/src/main/java/com/facebook/imageformat/ImageFormatChecker.java

CLICK HERE

```java
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();
  }
```

```java
public static synchronized ImageFormatChecker getInstance() {
  if (sInstance == null) {
    sInstance = new ImageFormatChecker();
  }
  return sInstance;
}
```


WebpTranscodeProducer
- shouldTranscode(encodedImage : EncodedImage) : Tristate
ImageFormatChecker
- sInstance : ImageFormatChecker
- ImageFormatChecker()
- getInstance : ImageFormatChecker()

MEMBERS

62010318	62010922	62010722	62010713	62010434	62010710
ทศภณ สิงสันจิตร	สหทัศน์ ลีวัฒนา	ภูมิภัทร เชื้อลื้อ	ภูฟ้า จันทรพัฒน์	ธีรัช เหล่าสุขสกุล	ภูดินันท์ เพ็ญสุข