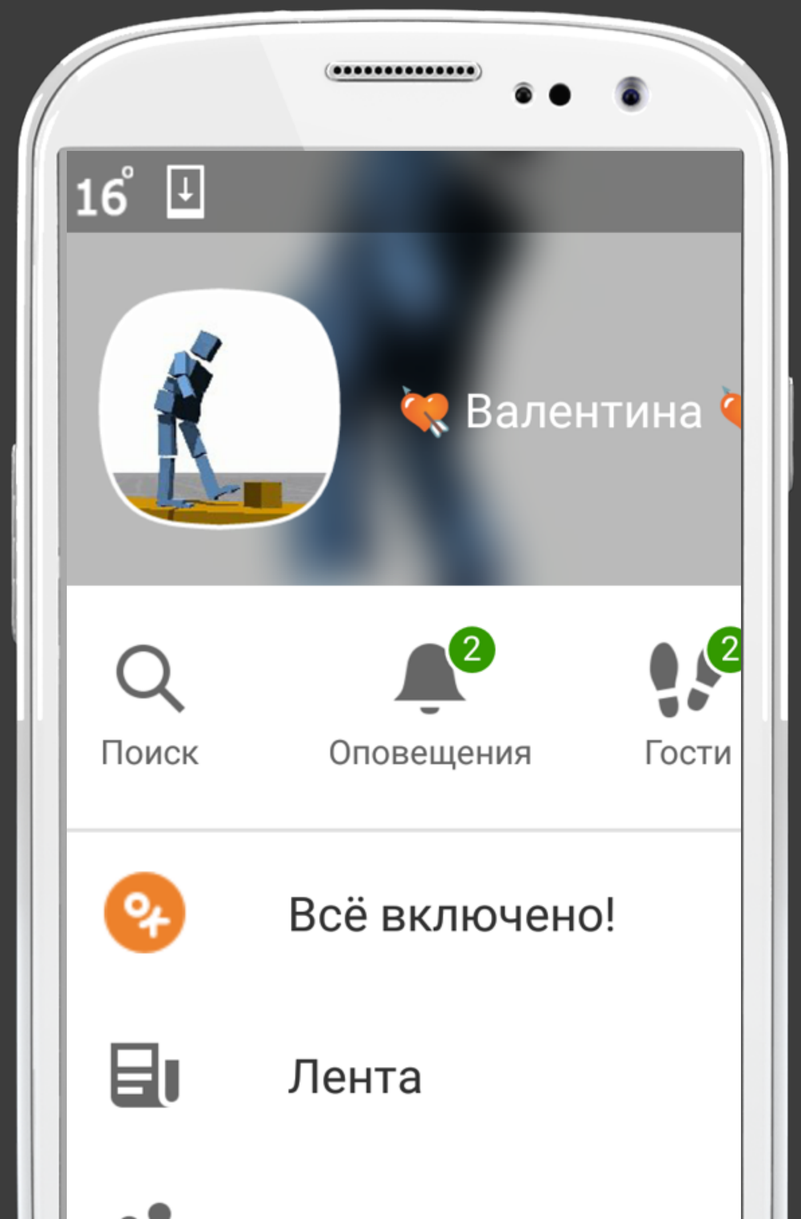
О Fresco и картинках в целом

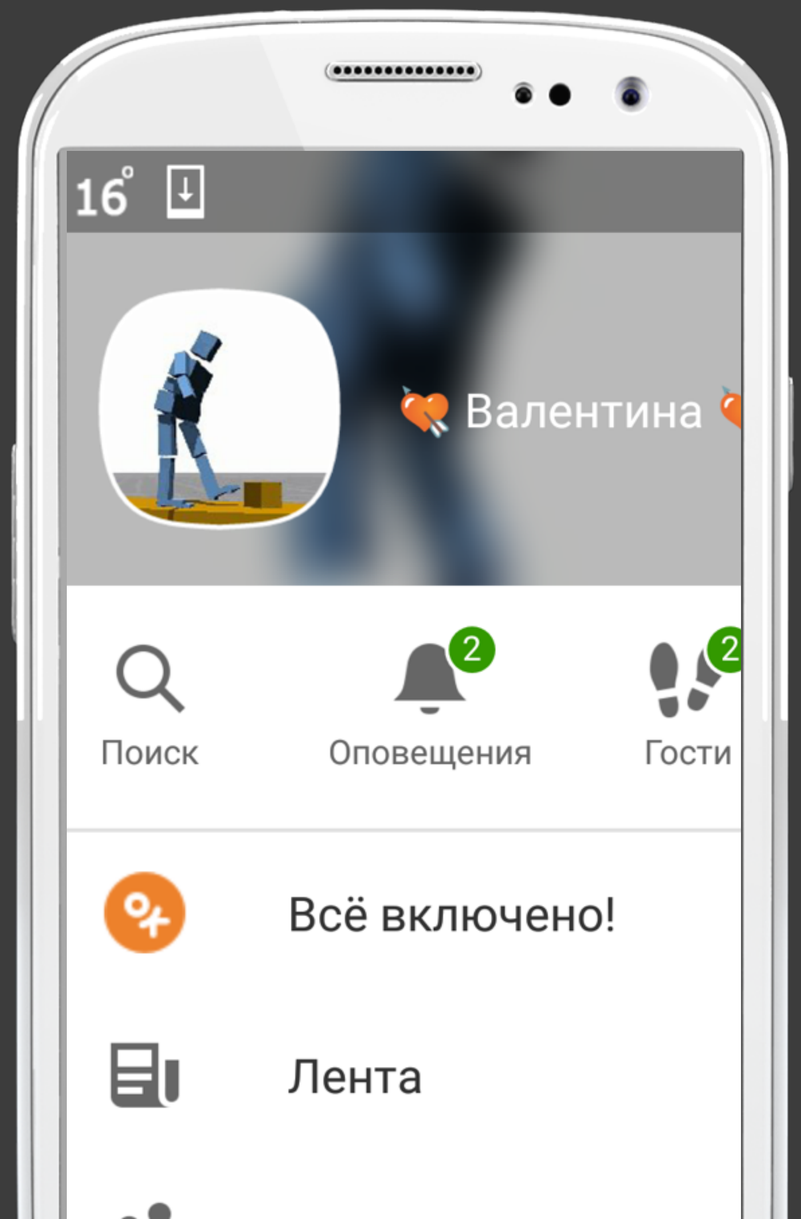
Зачем?

Показать картинку это просто?



Да, нужно только:

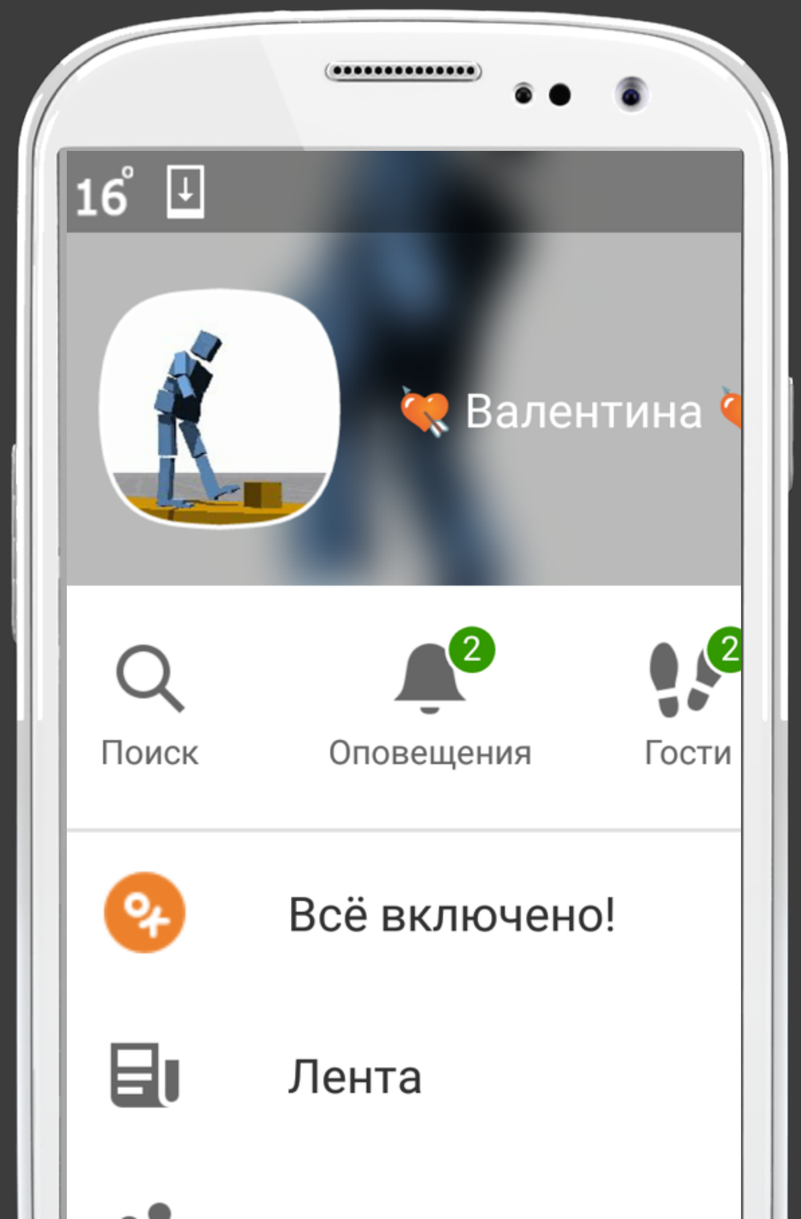
Показать картинку это просто?



Да, нужно только:

- Запросить картинку

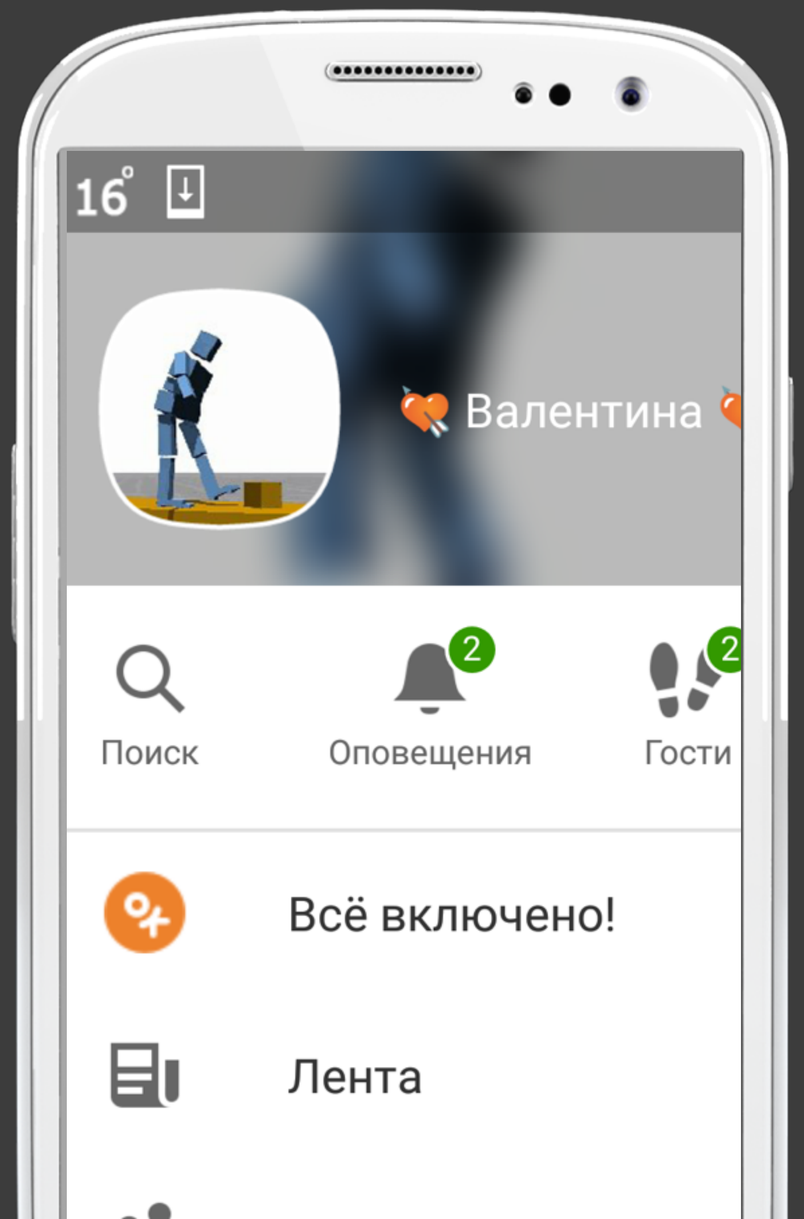
Показать картинку это просто?



Да, нужно только:

- Запросить картинку
- Проверить кеш

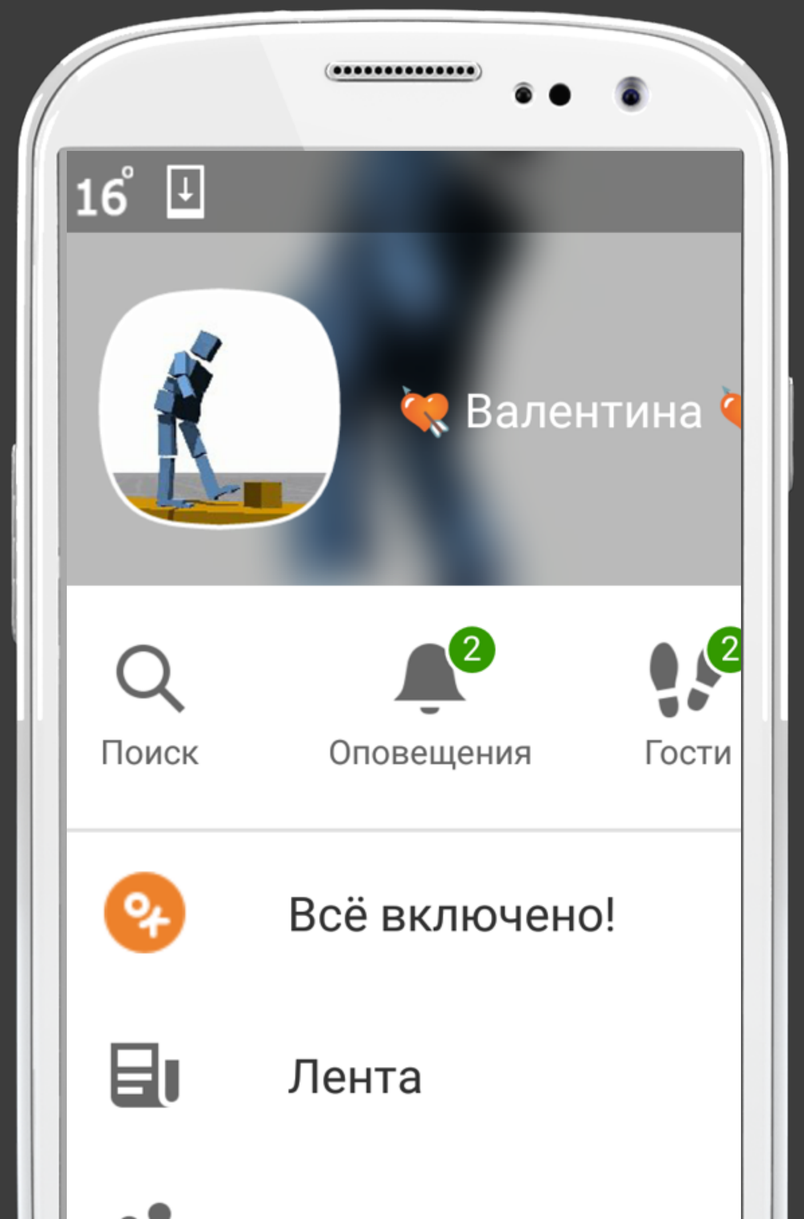
Показать картинку это просто?



Да, нужно только:

- Запросить картинку
- Проверить кеш
- Скачать картинку

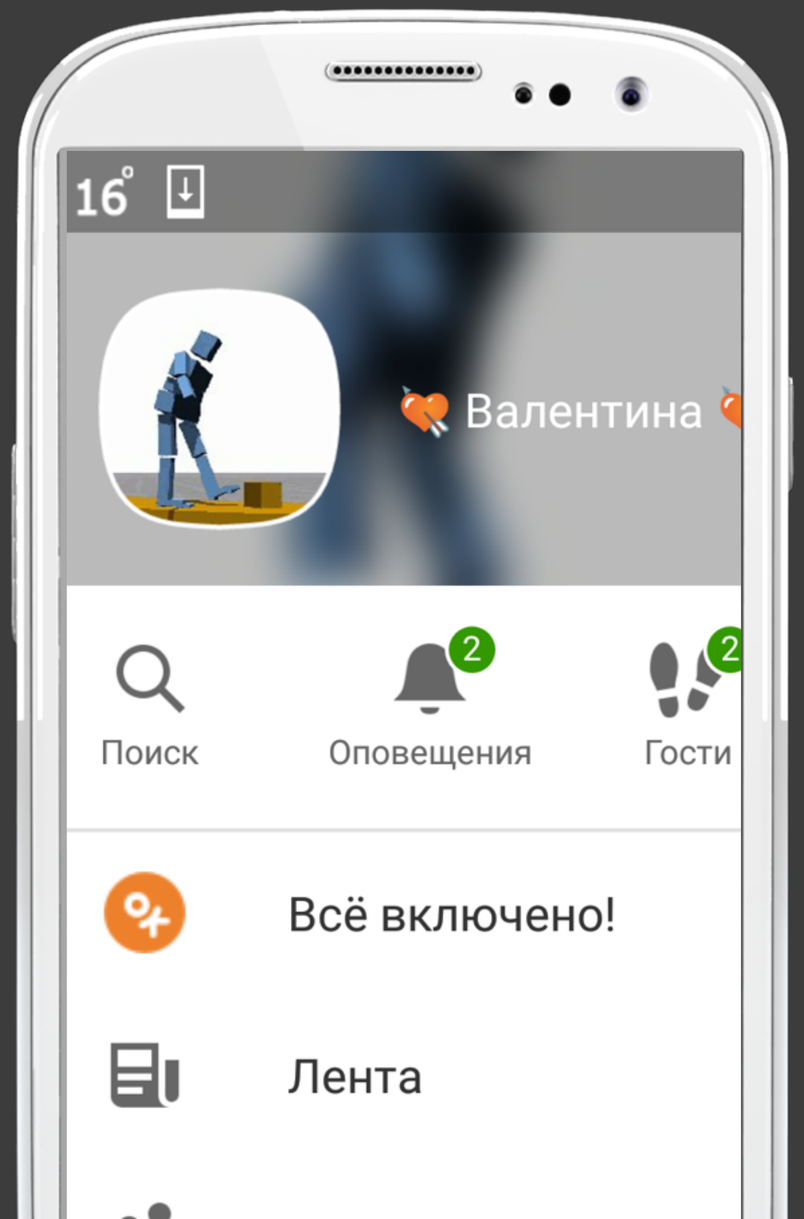
Показать картинку это просто?



Да, нужно только:

- Запросить картинку
- Проверить кеш
- Скачать картинку
- Положить её в кеш

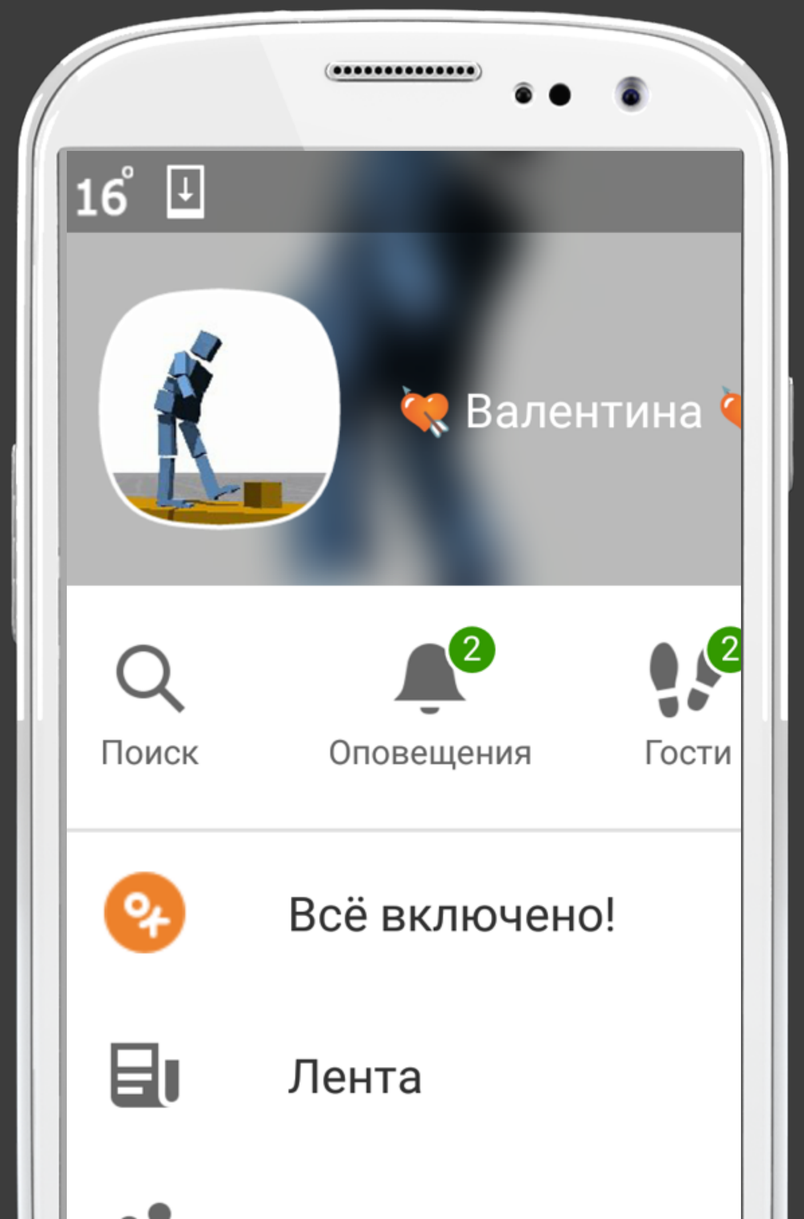
Показать картинку это просто?



Да, нужно только:

- Запросить картинку
- Проверить кеш
- Скачать картинку
- Положить её в кеш
- Модифицировать её

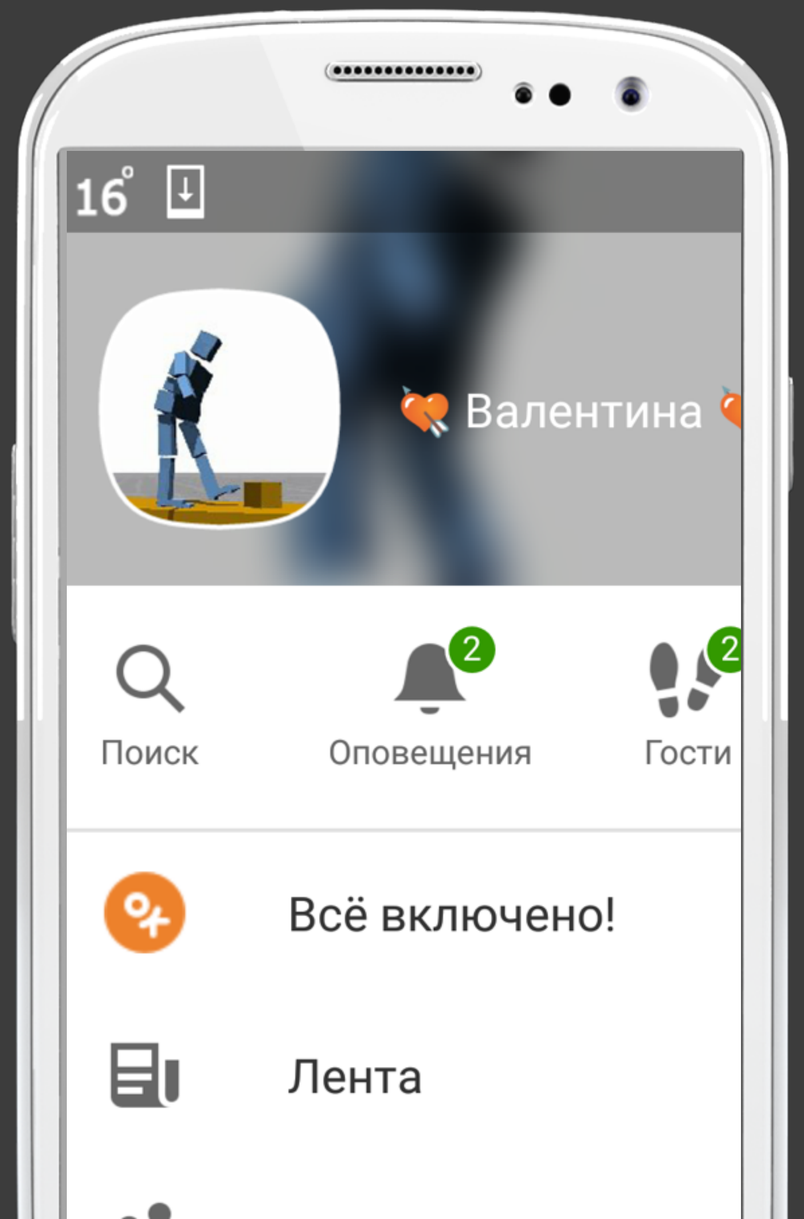
Показать картинку это просто?



Да, нужно только:

- Запросить картинку
- Проверить кеш
- Скачать картинку
- Положить её в кеш
- Модифицировать её
- Опять положить в кеш

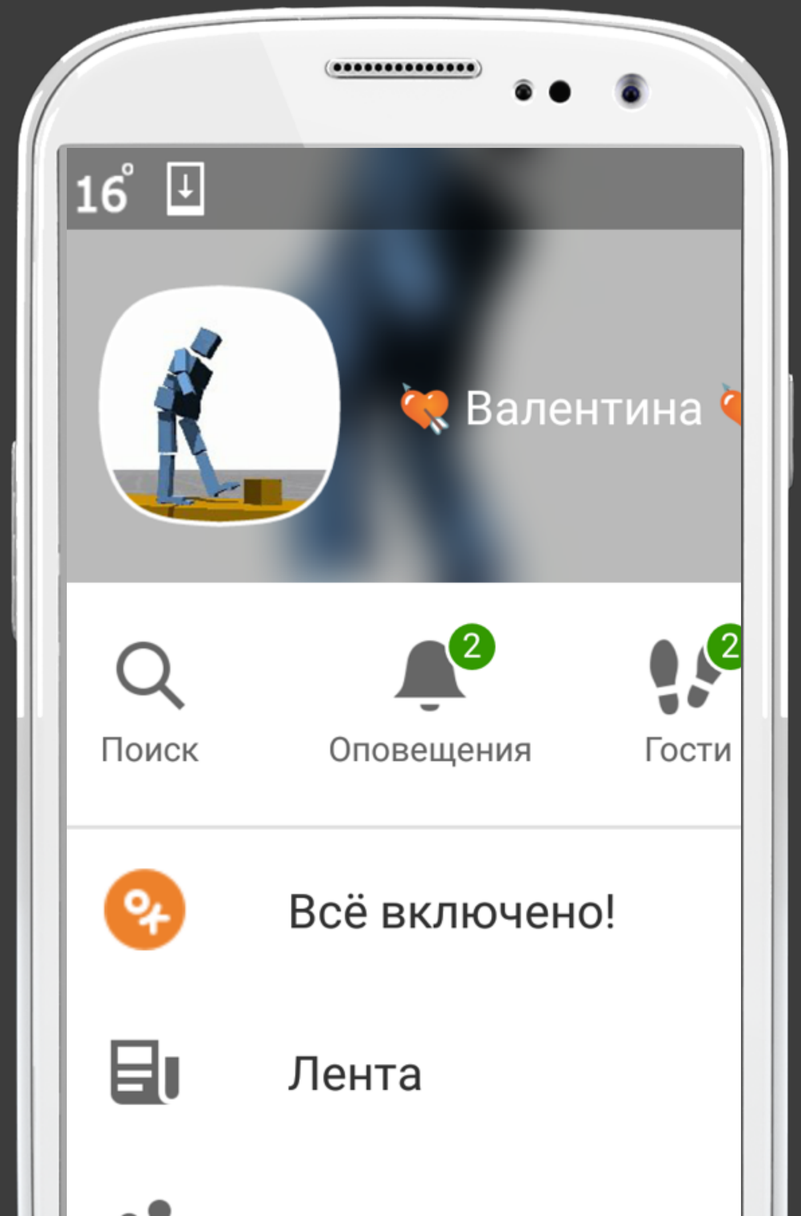
Показать картинку это просто?



Да, нужно только:

- Запросить картинку
- Проверить кеш
- Скачать картинку
- Положить её в кеш
- Модифицировать её
- Опять положить в кеш
- Отобразить картинку

Показать картинку это просто?



Да, нужно только:

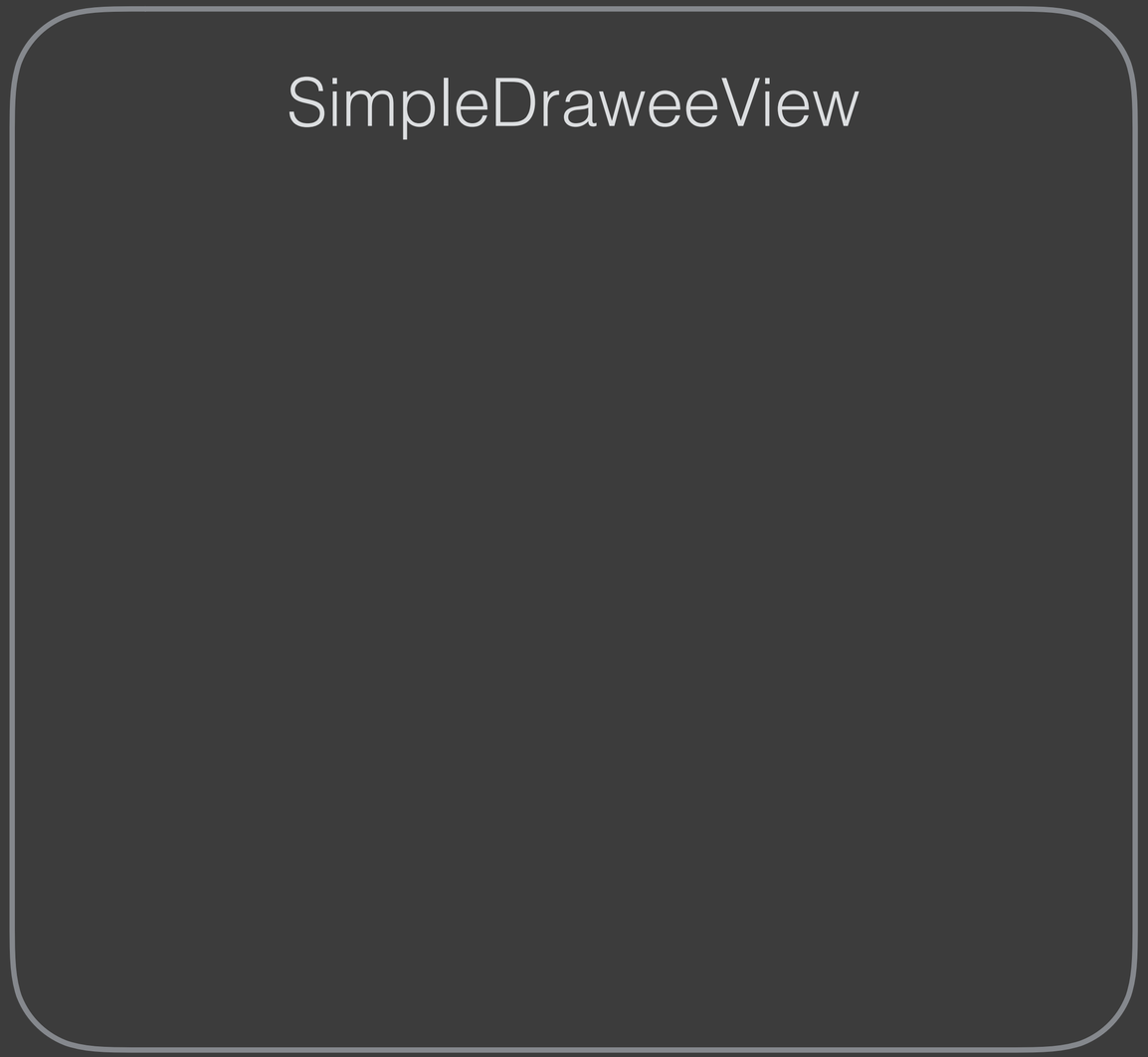
- Запросить картинку
- Проверить кеш
- Скачать картинку
- Положить её в кеш
- Модифицировать её
- Опять положить в кеш
- Отобразить картинку
- Переиспользовать память

Что делать дальше?

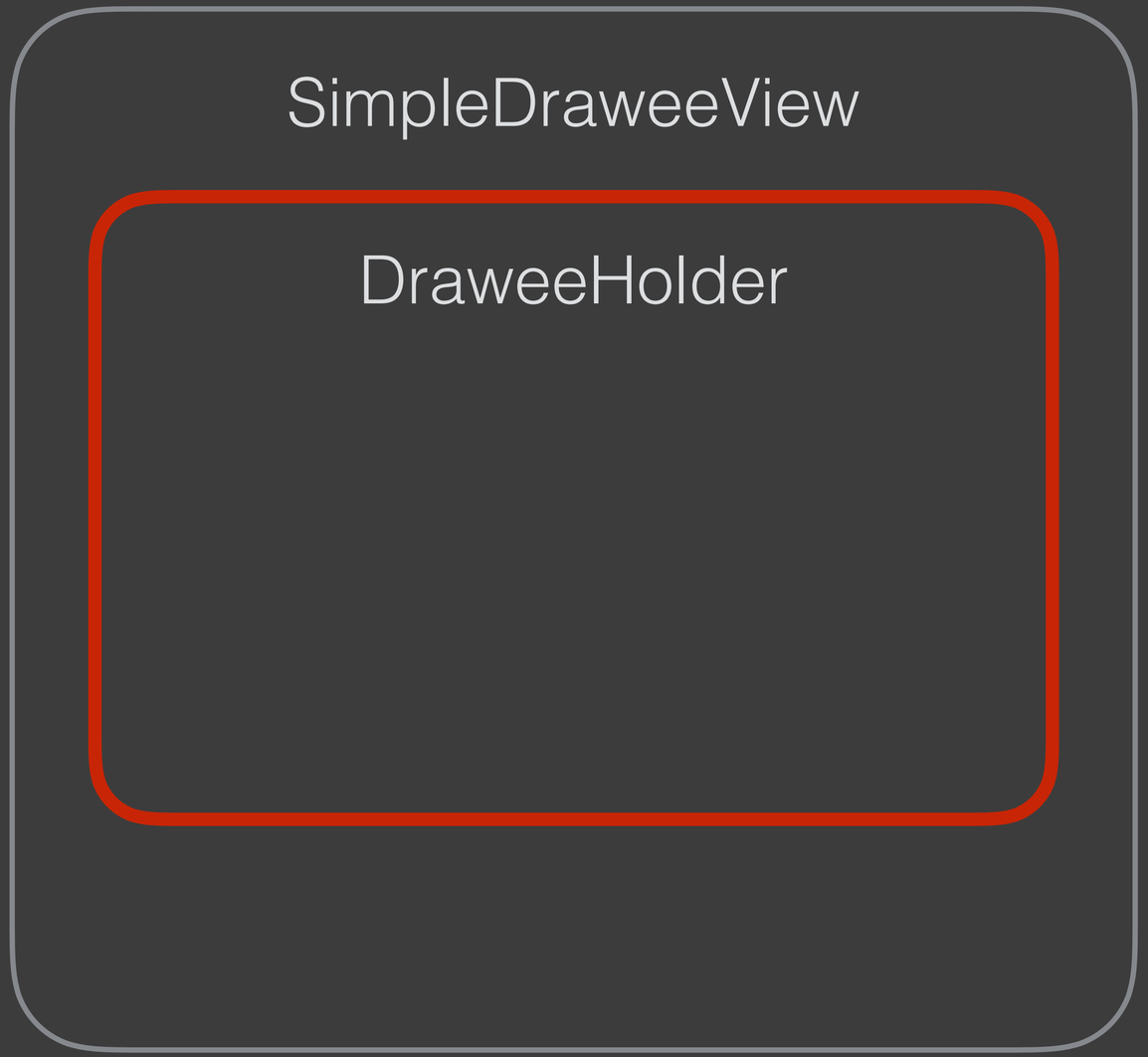
Fresco

Fresco 

О Fresco и картинках в целом



О Fresco и картинках в целом



О Fresco и картинках в целом

SimpleDraweeView

DraweeHolder

О Fresco и картинках в целом

SimpleDraweeView

DraweeHolder

DraweeHierarchy

О Fresco и картинках в целом

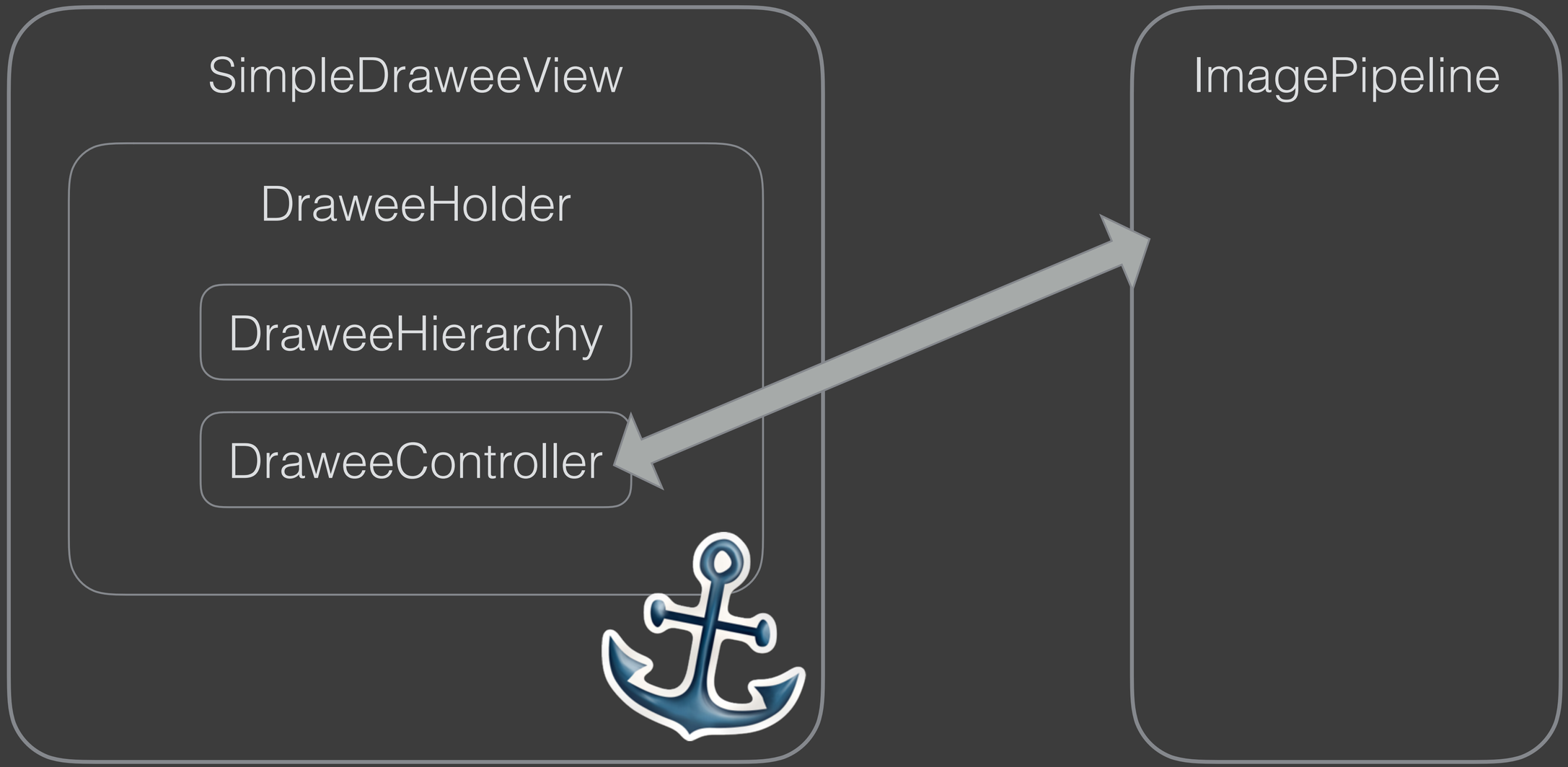
SimpleDraweeView

DraweeHolder

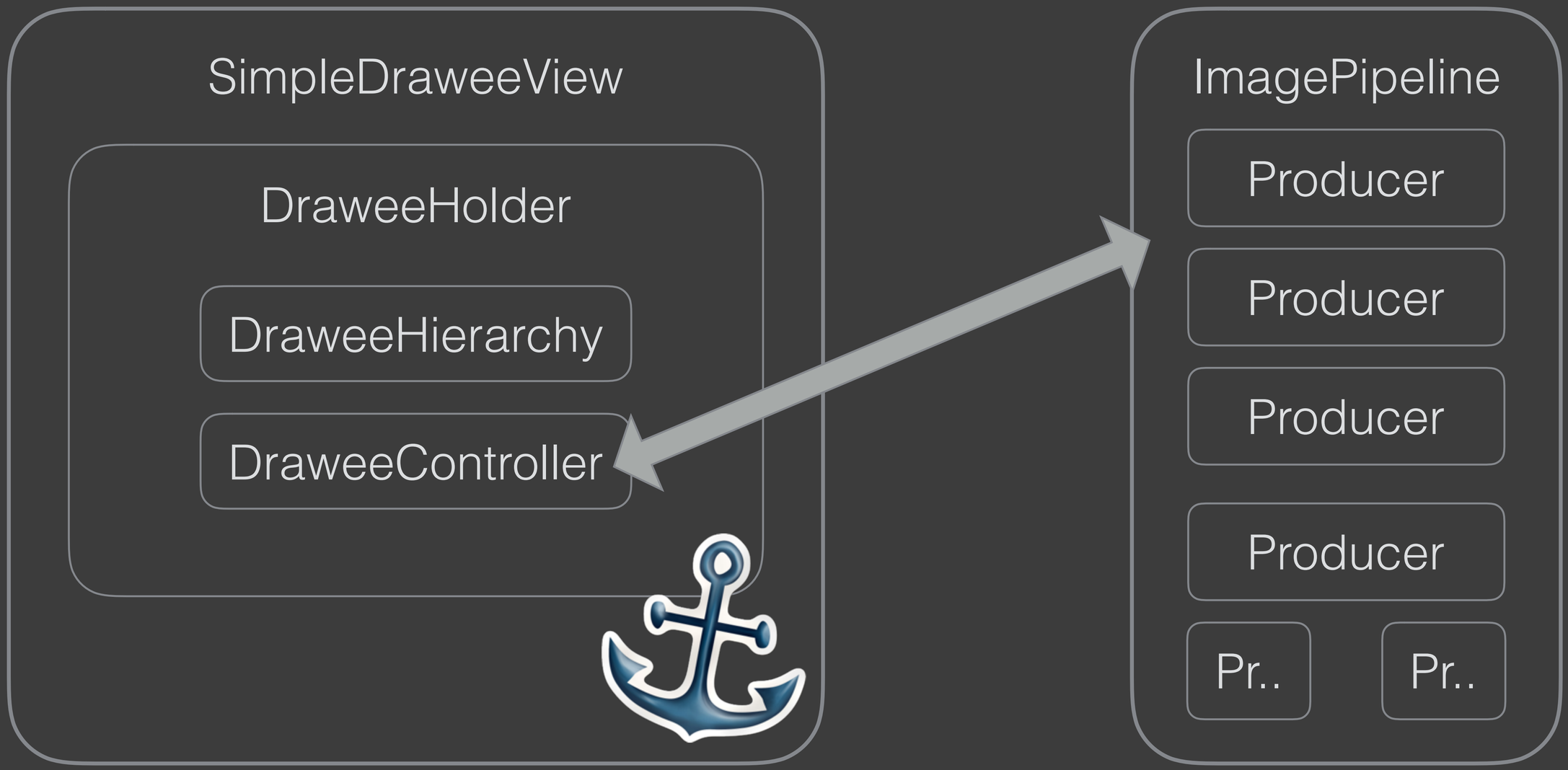
DraweeHierarchy

DraweeController

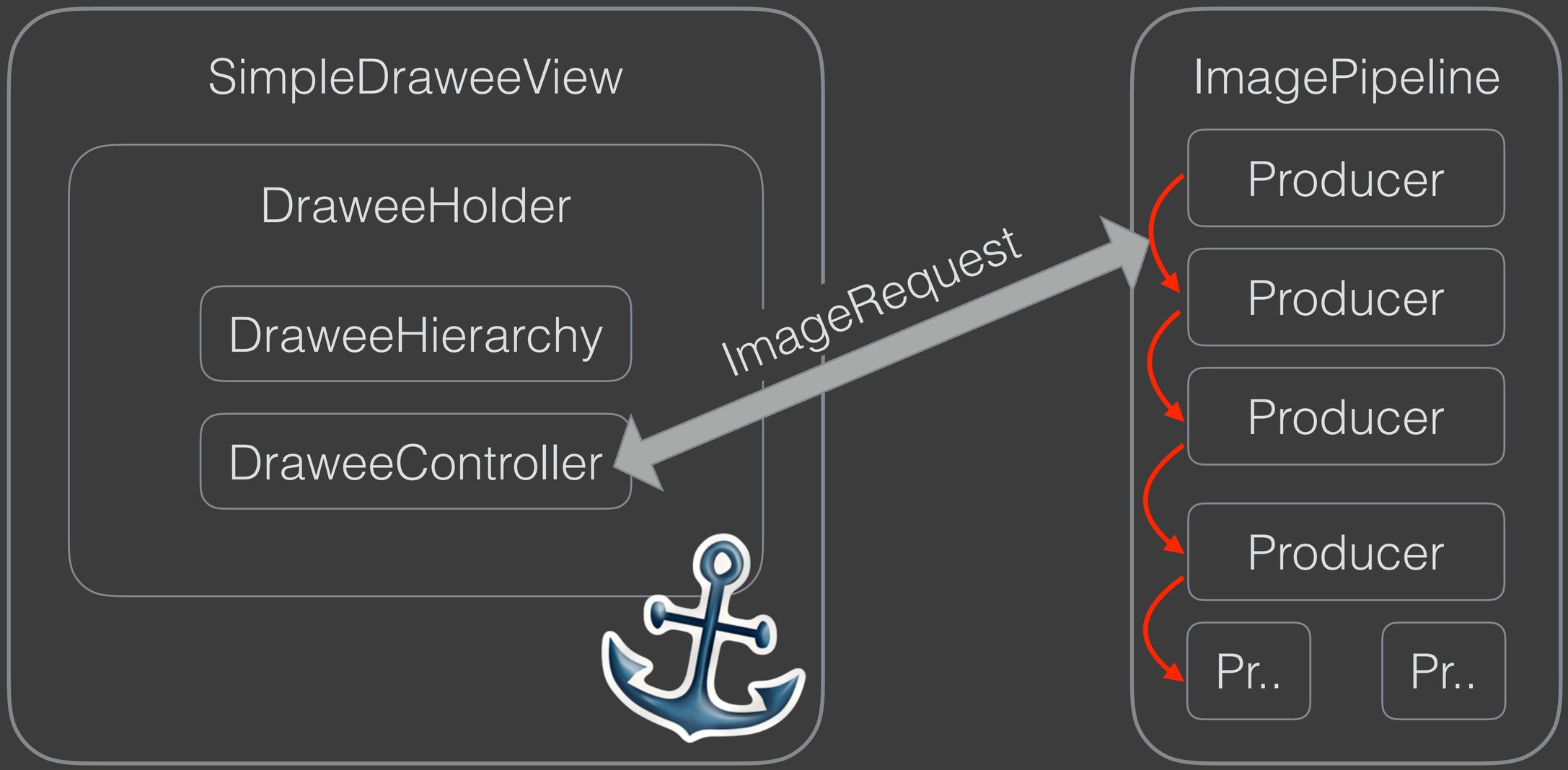
О Fresco и картинках в целом



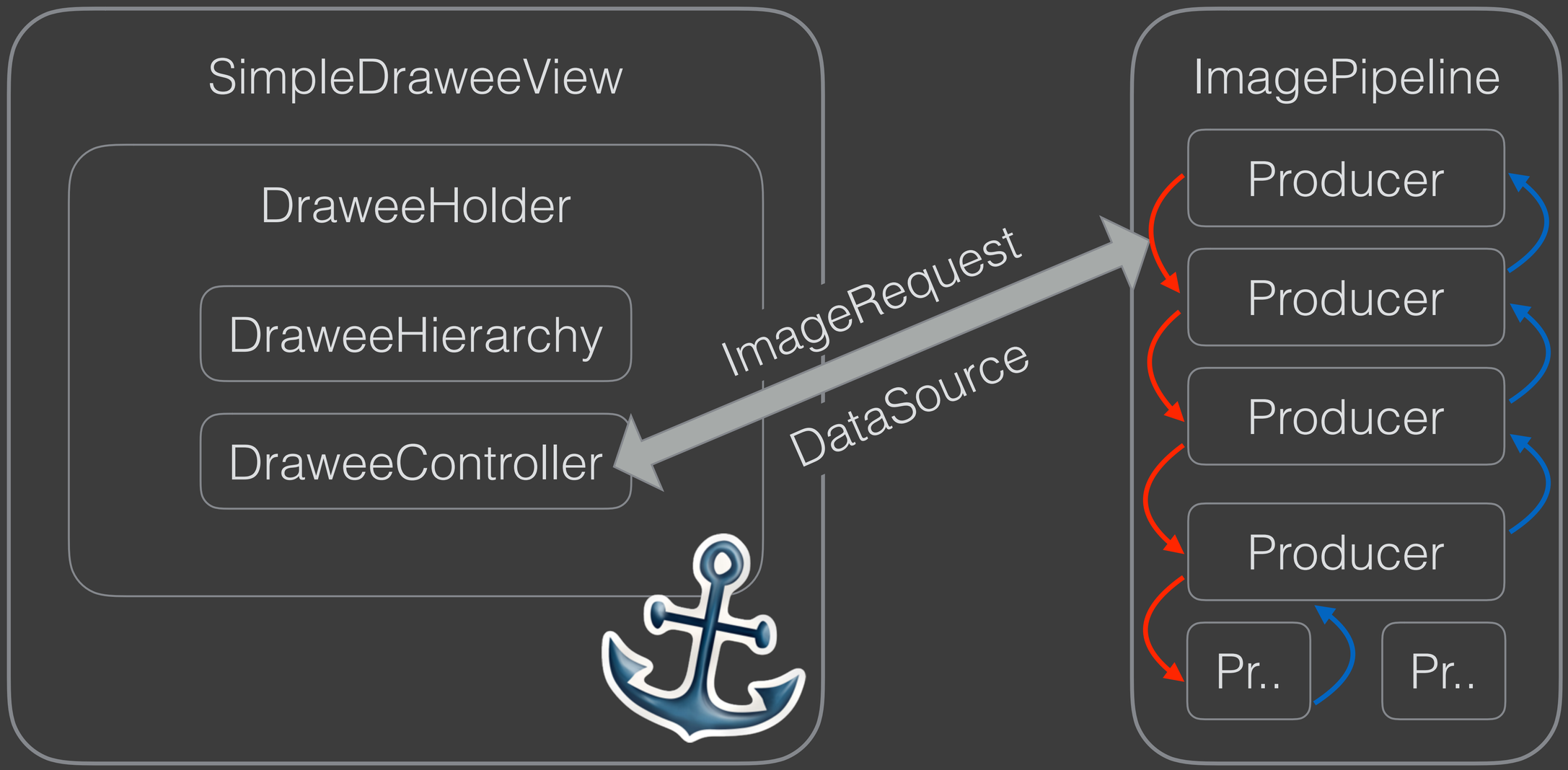
О Fresco и картинках в целом



О Fresco и картинках в целом



О Fresco и картинках в целом



В путь!

- Оптимальное использование памяти

О Fresco и картинках в целом

```
public static PlatformDecoder buildPlatformDecoder(
  // ...
  if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
  // ...
    return new ArtDecoder(/* ... */);
  } else {
    if (Build.VERSION.SDK_INT < Build.VERSION_CODES.KITKAT) {
      return new GingerbreadPurgeableDecoder(/* ... */);
    } else {
      return new KitKatPurgeableDecoder(/* ... */);
    }
  }
}
```

О Fresco и картинках в целом

```
public static PlatformDecoder buildPlatformDecoder(
  // ...
  if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
  // ...
    return new ArtDecoder(/* ... */);
  } else {
    if (Build.VERSION.SDK_INT < Build.VERSION_CODES.KITKAT) {
      return new GingerbreadPurgeableDecoder(/* ... */);
    } else {
      return new KitKatPurgeableDecoder(/* ... */);
    }
  }
}
```

О Fresco и картинках в целом

```
public static PlatformDecoder buildPlatformDecoder(
  // ...
  if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
  // ...
    return new ArtDecoder(/* ... */);
  } else {
    if (Build.VERSION.SDK_INT < Build.VERSION_CODES.KITKAT) {
      return new GingerbreadPurgeableDecoder(/* ... */);
    } else {
      return new KitKatPurgeableDecoder(/* ... */);
    }
  }
}
```

Демо с ashmem

О Fresco и картинках в целом

```
static void Bitmaps_pinBitmap(
    JNIEnv* env,
    jclass clazz,
    jobject bitmap) {
  UNUSED(clazz);
  int rc = AndroidBitmap_lockPixels(env, bitmap, 0);
  if (rc != ANDROID_BITMAP_RESULT_SUCCESS) {
    safe_throw_exception(env, "Failed to pin Bitmap");
  }
}
```

О Fresco и картинках в целом

О Fresco и картинках в целом

- addReference()
- deleteReference()

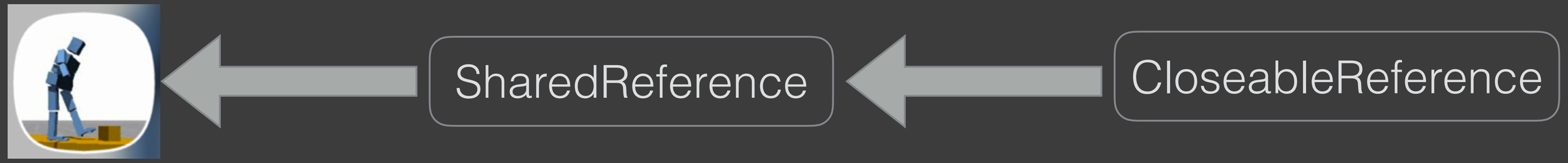
О Fresco и картинках в целом

SharedReference

CloseableReference

CloseableReference.*closeSafely*(ref)
CloseableReference.*cloneOrNull*(ref)

О Fresco и картинках в целом



1. Вызывающий владеет ссылкой

О Fresco и картинках в целом

SharedReference

CloseableReference

1. Вызывающий владеет ссылкой
2. Вызывающий должен закрыть ссылку перед выходом из скоупа

О Fresco и картинках в целом

SharedReference

CloseableReference

1. Вызывающий владеет ссылкой
2. Вызывающий должен закрыть ссылку перед выходом из скоупа
3. Никто кроме владельца не должен закрывать ссылку

О Fresco и картинках в целом

SharedReference

CloseableReference

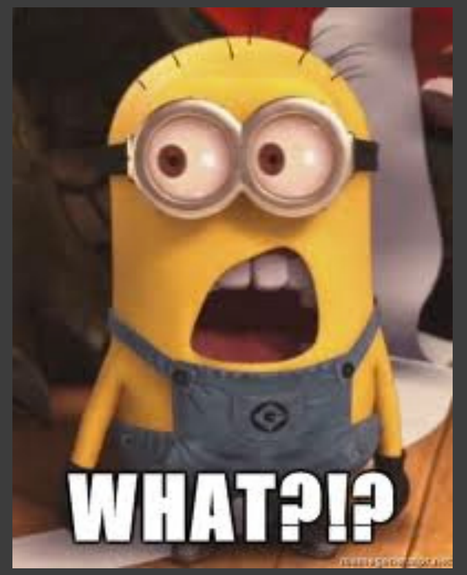
1. Вызывающий владеет ссылкой
2. Вызывающий должен закрыть ссылку перед выходом из скоупа
3. Никто кроме владельца не должен закрывать ссылку
4. Вызывающий должен всегда клонировать ссылку перед присвоением

О Fresco и картинках в целом

SharedReference

CloseableReference

1. Вызывающий владеет ссылкой
2. Вызывающий должен закрыть ссылку перед выходом из скоупа
3. Никто кроме владельца не должен закрывать ссылку
4. Вызывающий должен всегда клонировать ссылку перед присвоением



О Fresco и картинках в целом

SharedReference

CloseableReference

О Fresco и картинках в целом

SharedReference

CloseableReference

SimpleDraweeView

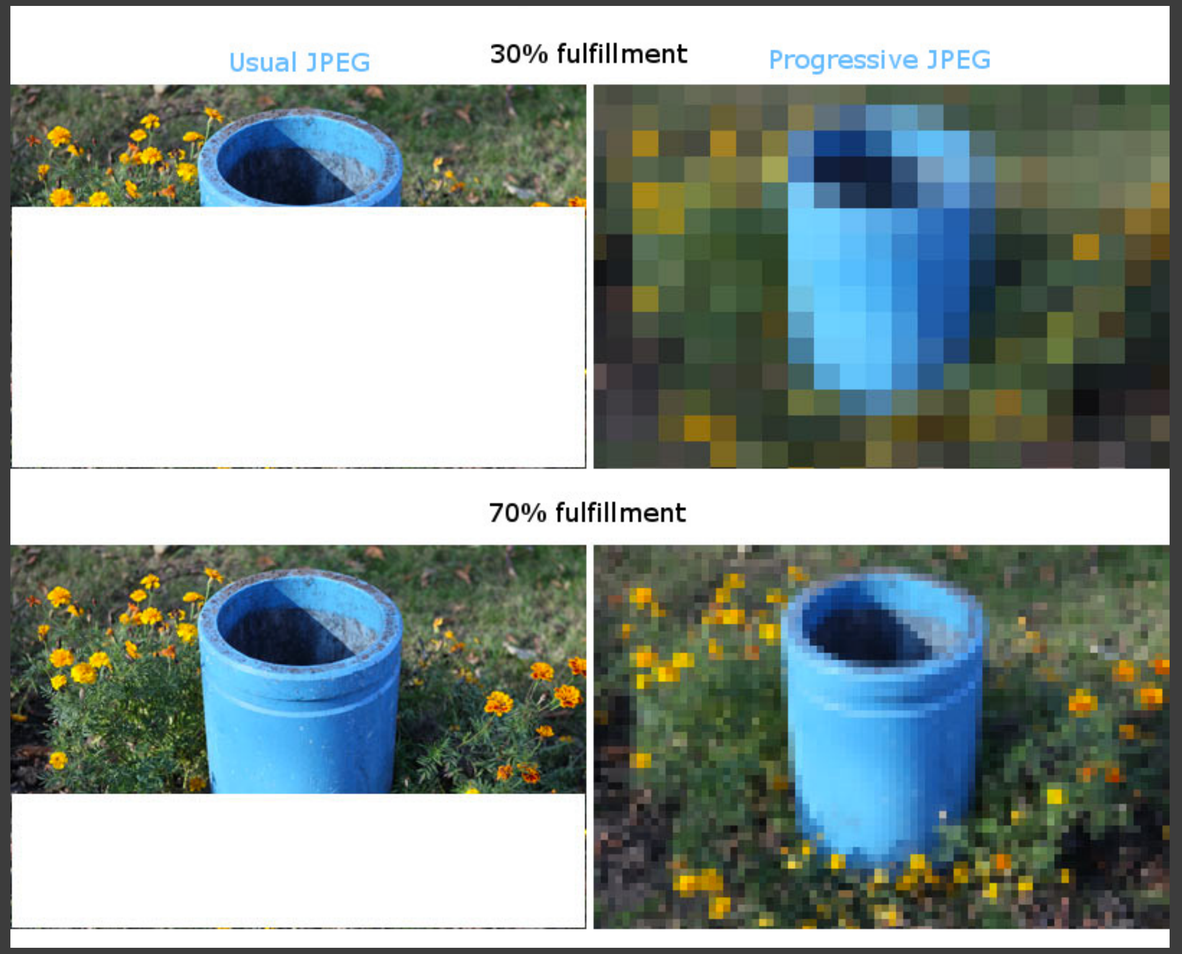
```
draweeHolder.onAttach()

draweeHolder.onDetach()
```

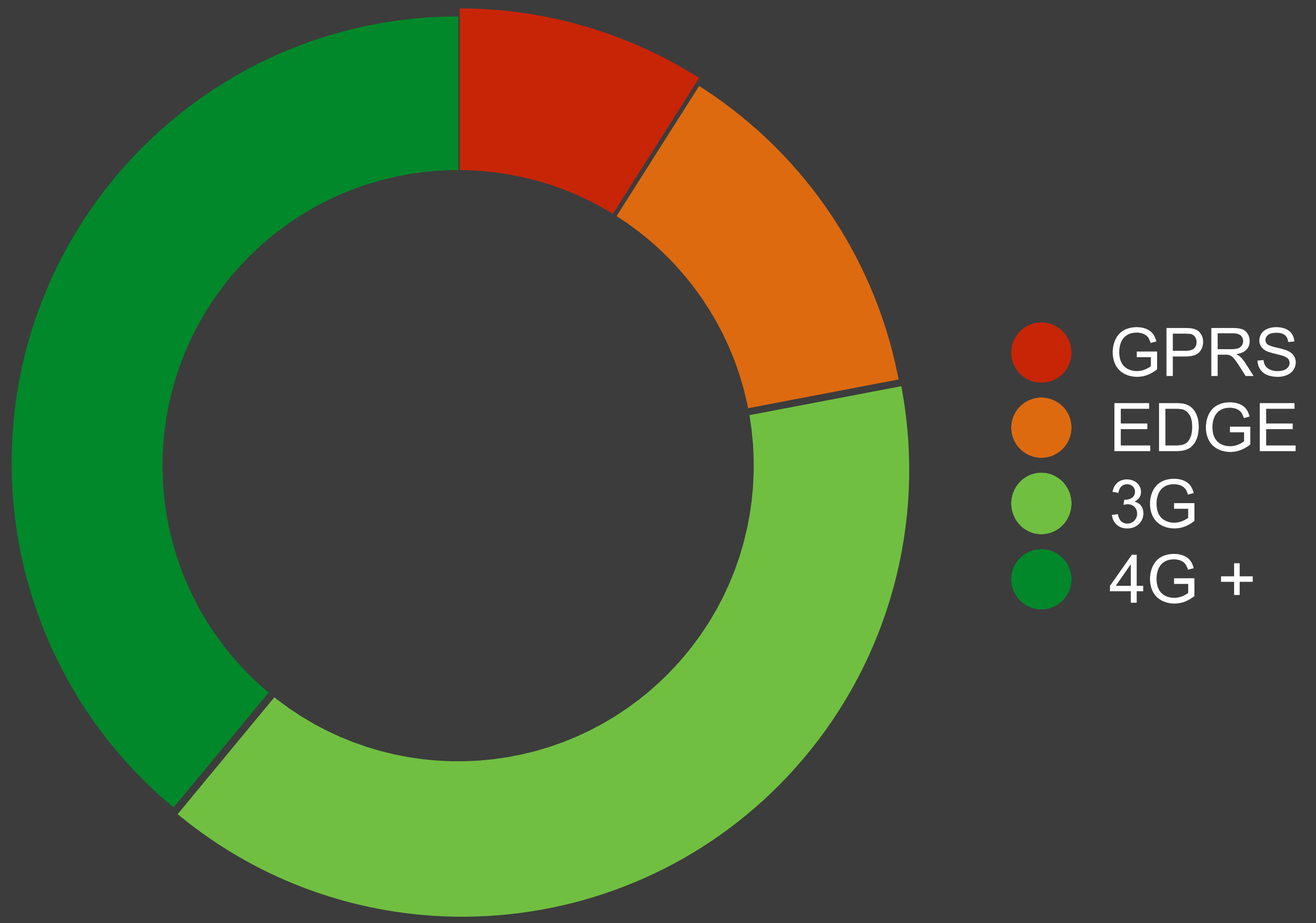
О Fresco и картинках в целом

- Оптимальное использование памяти
- Прогрессивный JPEG

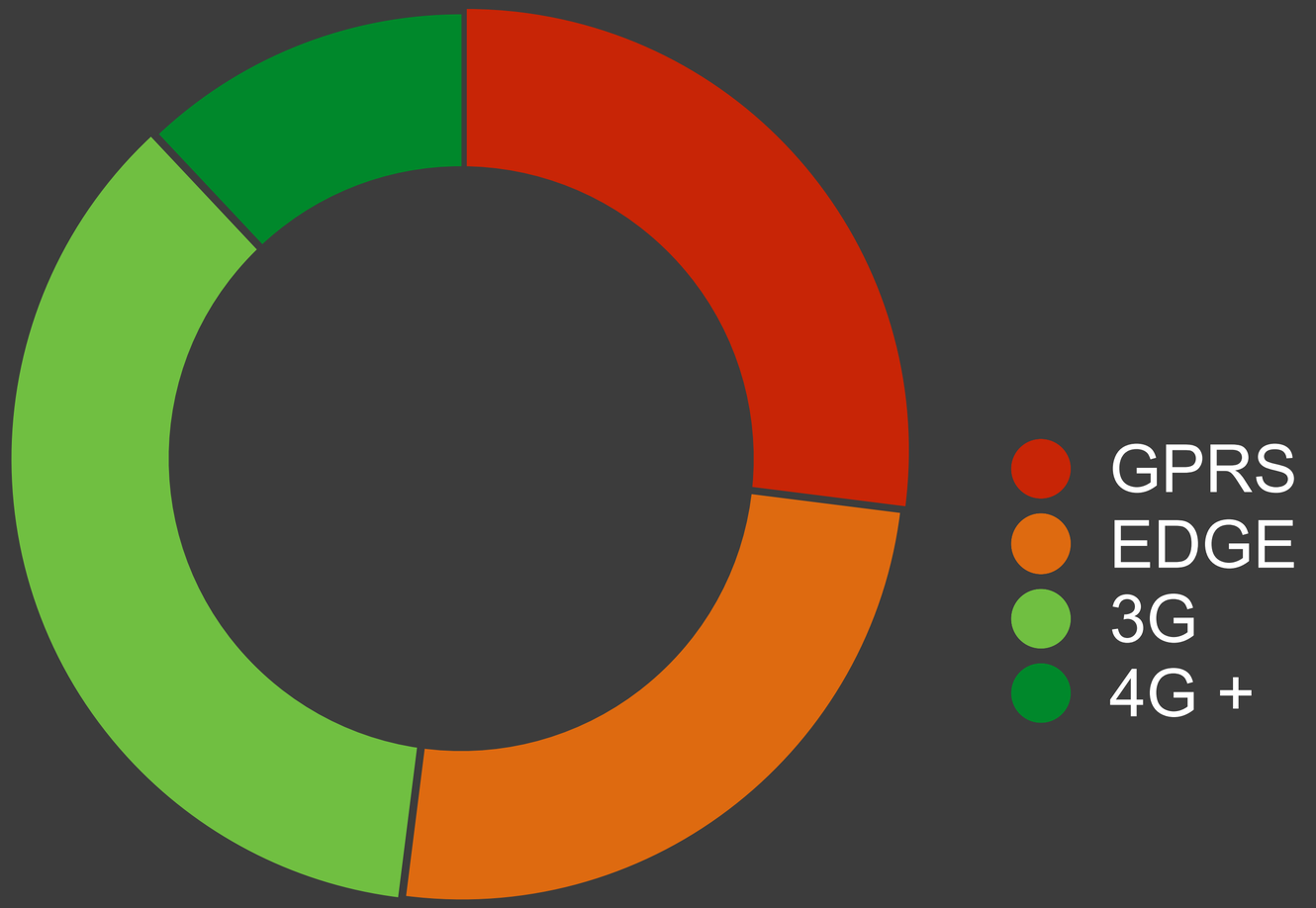
Usual JPEG vs Progressive JPEG



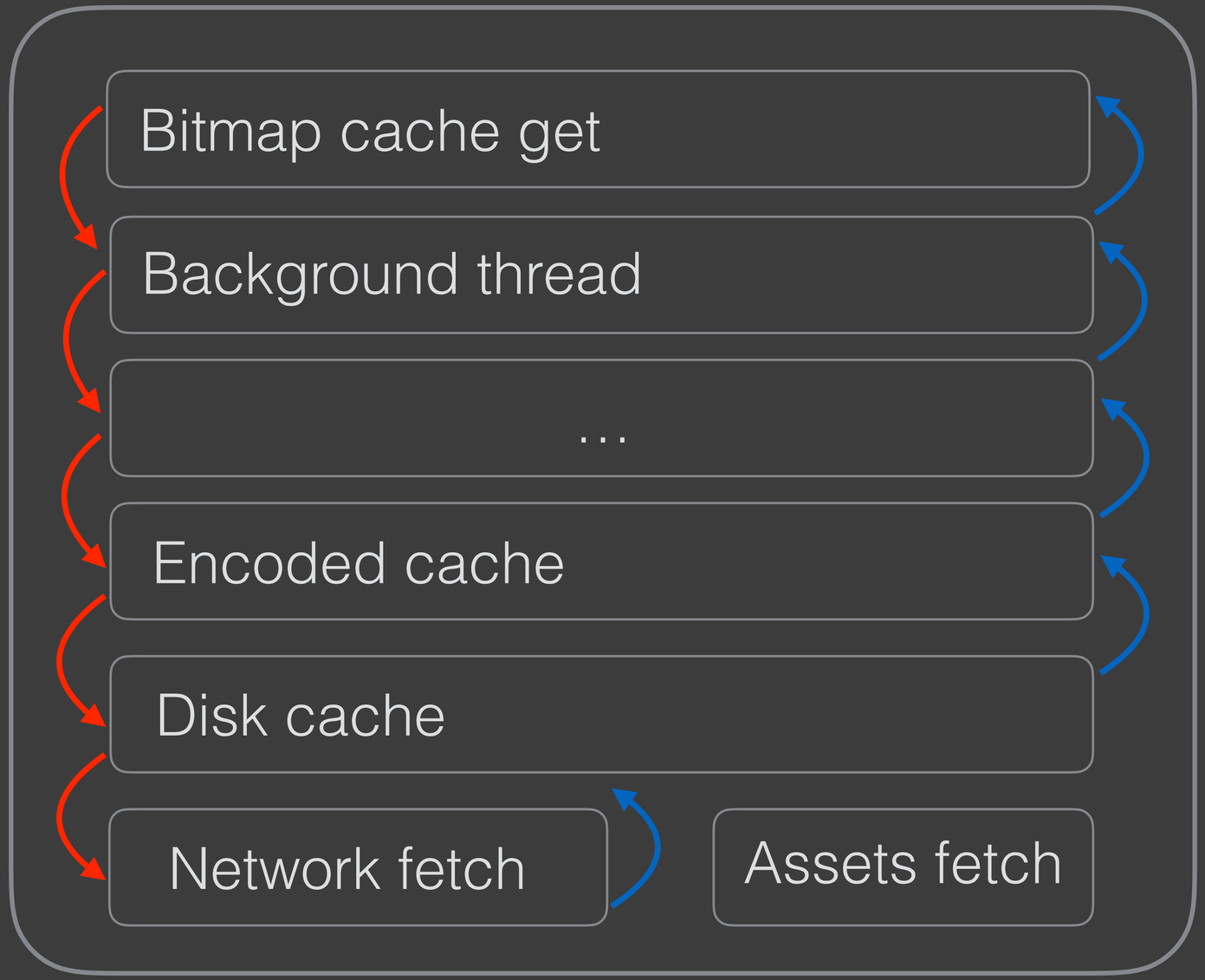
Скорость интернета в России



Скорость интернета в средней Азии



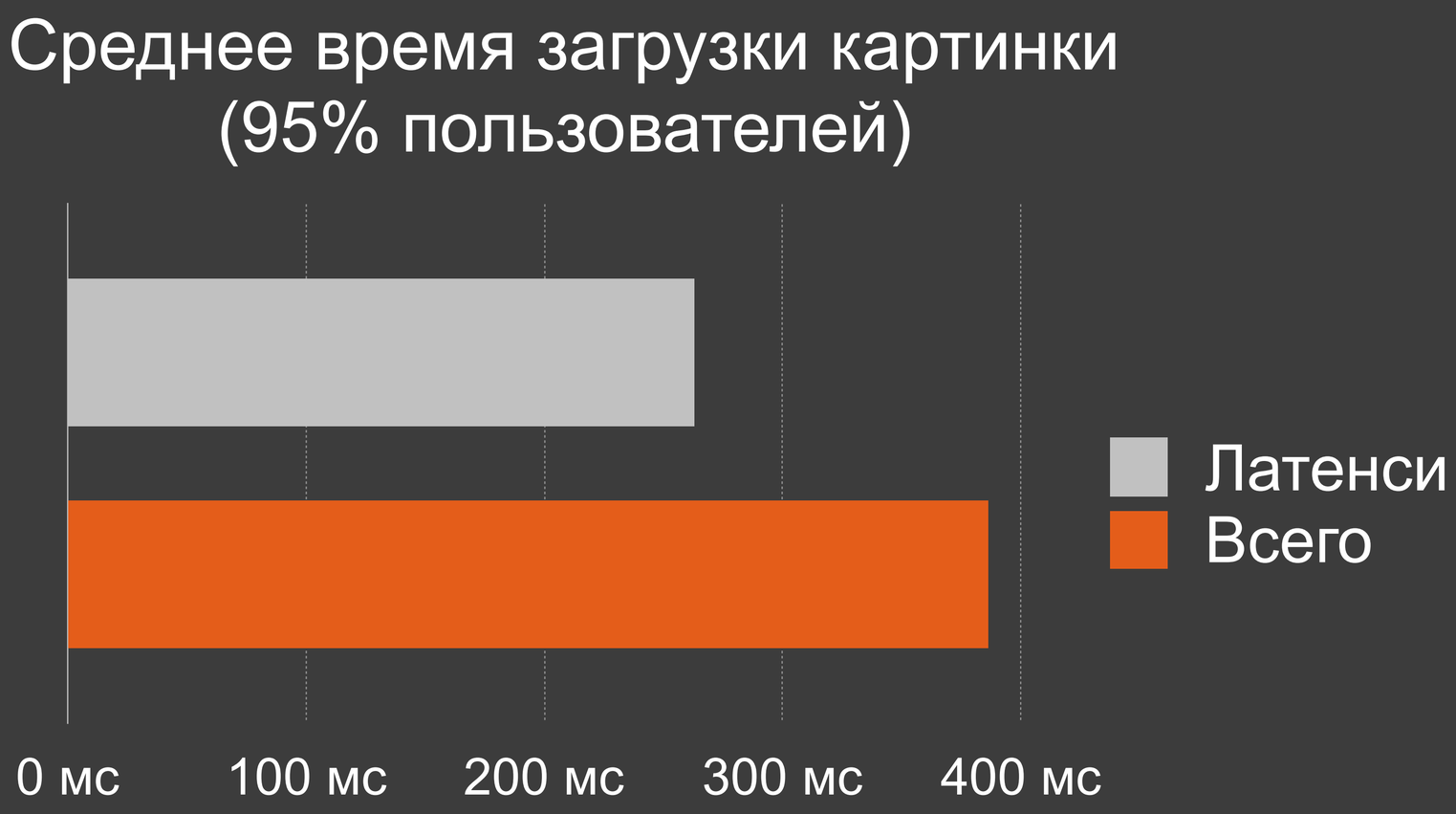
ImagePipeline



О Fresco и картинках в целом

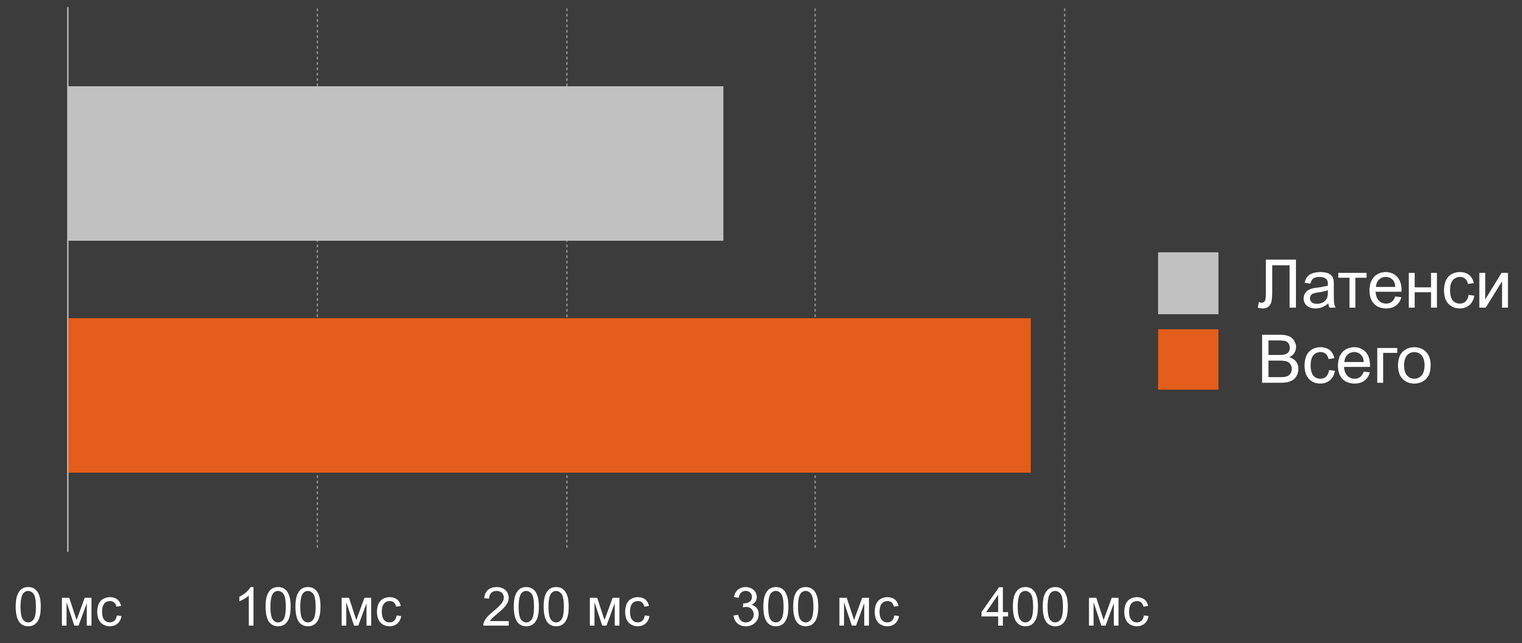
- Оптимальное использование памяти
- Прогрессивный JPEG
- Многоуровневое кеширование

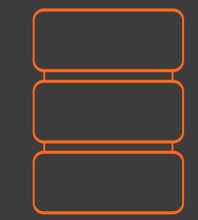
О Fresco и картинках в целом



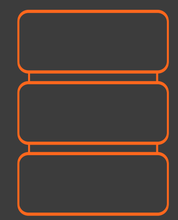
О Fresco и картинках в целом

Среднее время загрузки картинки (95% пользователей)





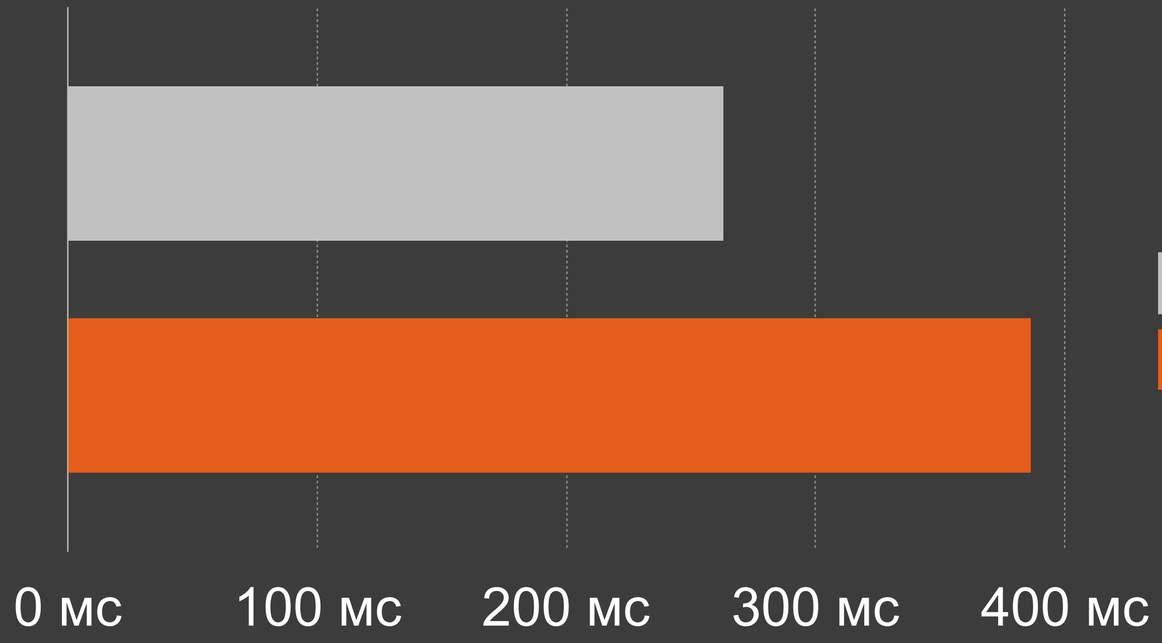
Disc cache

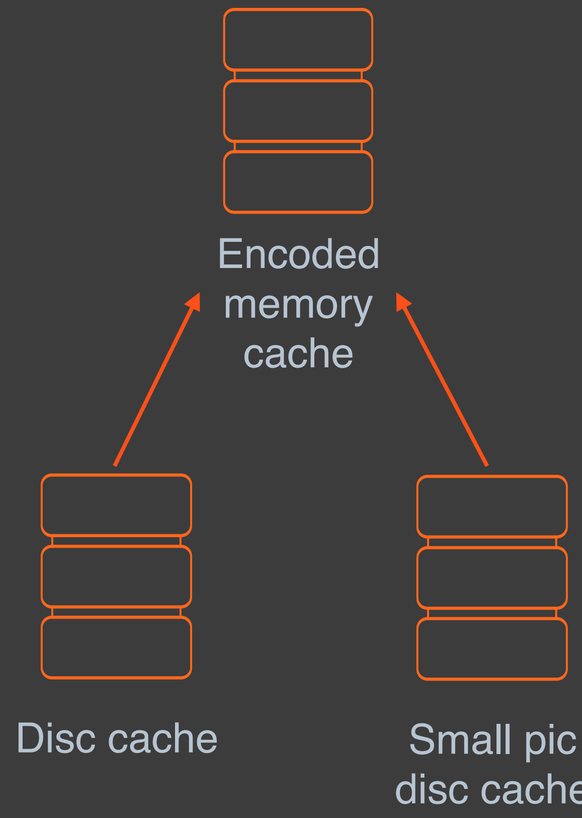


Small pic disc cache

О Fresco и картинках в целом

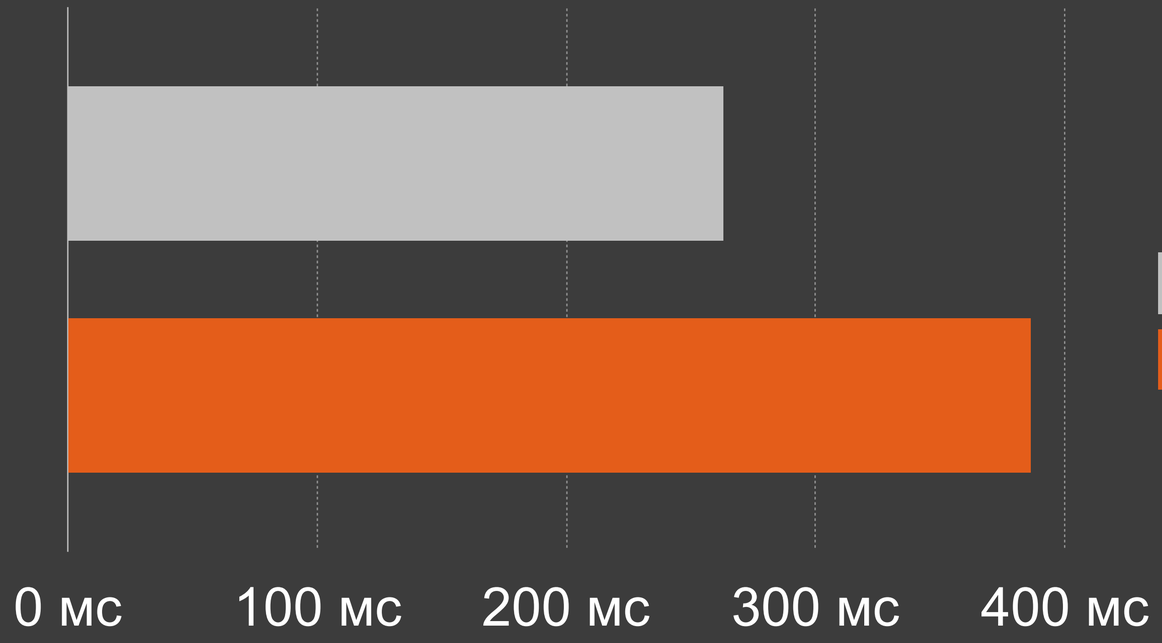
Среднее время загрузки картинки (95% пользователей)

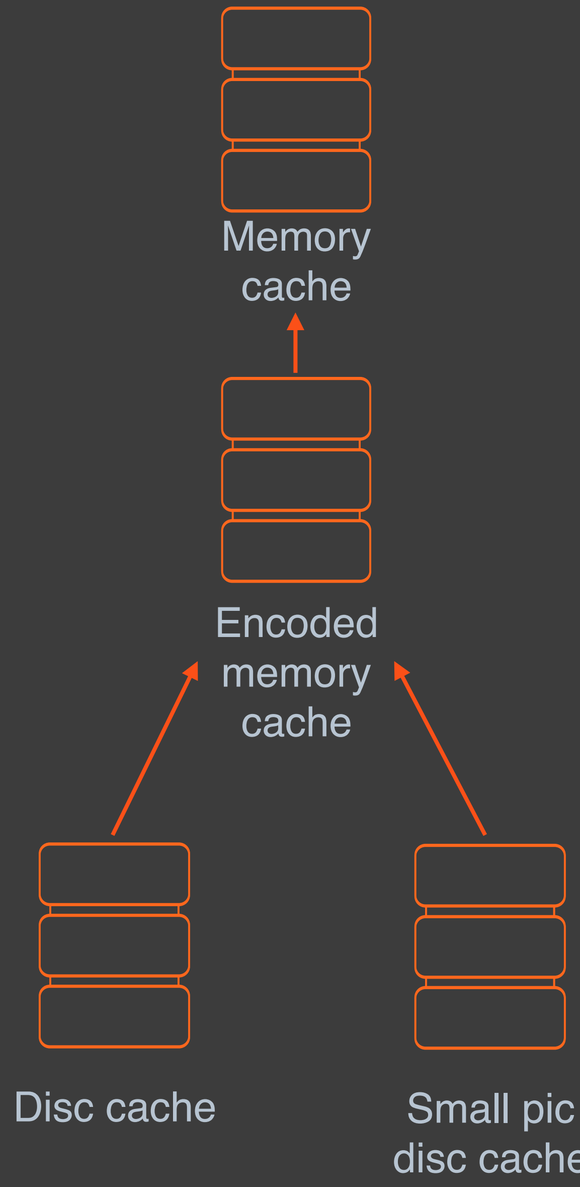




О Fresco и картинках в целом

Среднее время загрузки картинки (95% пользователей)





О Fresco и картинках в целом

- Оптимальное использование памяти
- Прогрессивный JPEG
- Многоуровневое кеширование
- Постобработка изображений

О Fresco и картинках в целом

```
draweeView.setController(Fresco.newDraweeControllerBuilder()
        .setImageRequest(
                ImageRequestBuilder.newBuilderWithSource(okLogoUri)
                .setPostprocessor(new BlurPostprocessor(7))
                .build())
        .build());
```

О Fresco и картинках в целом

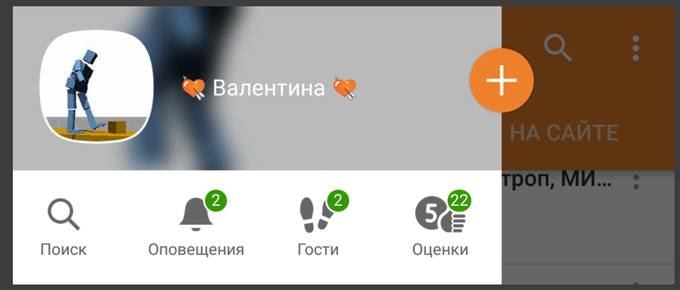
```
public class BlurPostprocessor extends BasePostprocessor {

    private final float blurRadius;

    public BlurPostprocessor(float blurRadius) {
        this.blurRadius = blurRadius;
    }

    @Override
    public void process(Bitmap bitmap) {
        RsBlur.blur(bitmap, blurRadius);
    }

    @Override
    public CacheKey getPostprocessorCacheKey() {
        return new SimpleCacheKey(Float.toString(blurRadius));
    }
}
```



О Fresco и картинках в целом

- Оптимальное использование памяти
- Прогрессивный JPEG
- Многоуровневое кеширование
- Постобработка изображений
- Гибкость

О Fresco и картинках в целом

Fresco

О Fresco и картинках в целом

Fresco

Библиотека
mp4 (gif)

О Fresco и картинках в целом

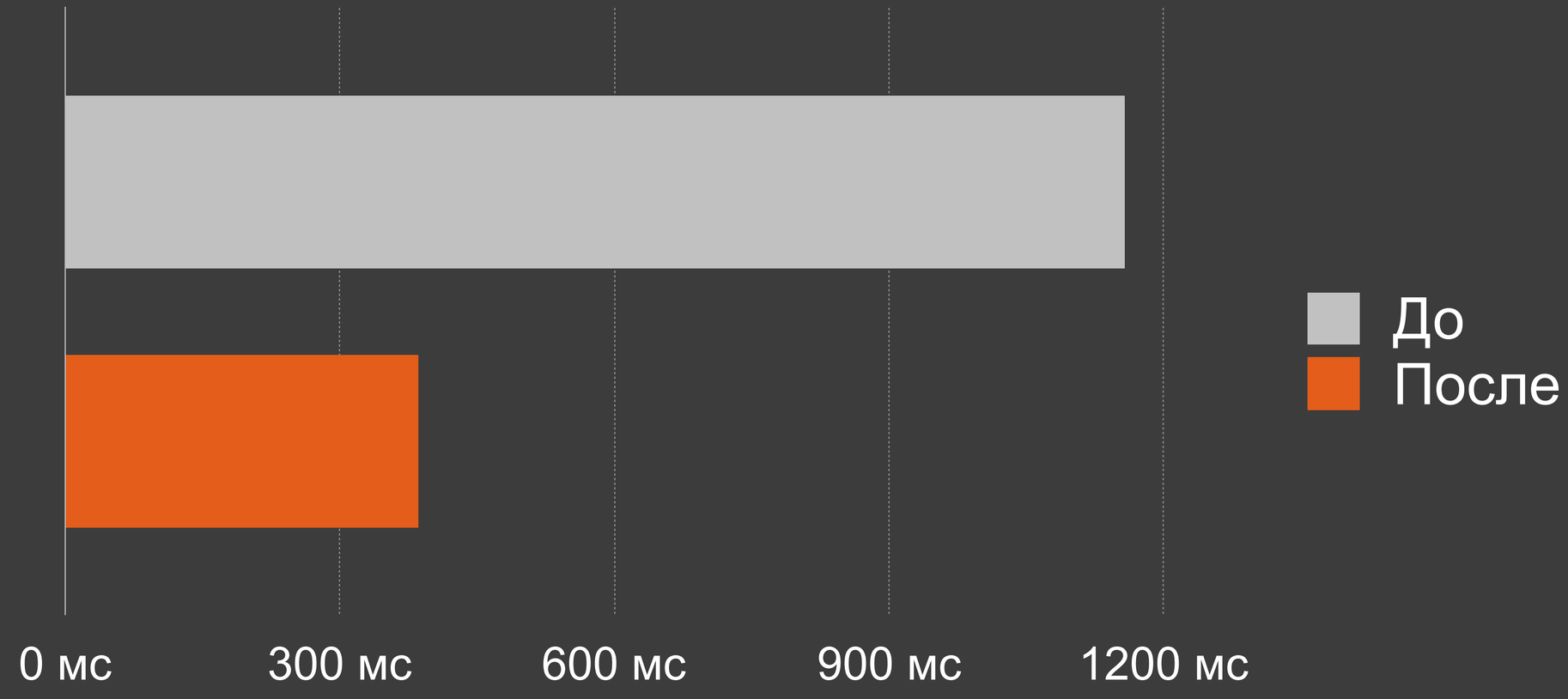
Fresco

Библиотека
mp4 (gif)

Спрайты
(подарки)

О Fresco и картинках в целом

О Fresco и картинках в целом



Кеширование

- *http://dg56.mycdn.me/image?id=835239547121&bid=835239547121&t=0&plc=API&viewToken=S9HOP6ny0qgJvsGR6Kj4VA&aid=25662464&tkn=*NSzQatr0vku0pAljrocb1EZKxWE*

Кеширование

- *http://dg56.mycdn.me/image?**id**=8352395471 21&**bid**=8352395471 21&**t**=0&**plc**=API&**viewToken**=S9HOP6ny0qgJvsGR6Kj4VA&**aid**=25662464&**tkn**=*NSzQatr0vku0pAljrocb1EZKxWE*

Кеширование

- *http://dg56.mycdn.me/image?id=8352395471 21&**bid**=8352395471 21&**t**=0&plc=API&viewToken=S9HOP6ny0qgJvsGR6Kj4VA&aid=25662464&tkn=*NSzQatr0vku0pAljrocb1EZKxWE*

Кеширование

```
public class OdklCacheKeyFactory extends DefaultCacheKeyFactory {
    public Uri getCacheKeySourceUri(Uri sourceUri) {
        if (!OdklUtils.isOdklHost(sourceUri)) { return sourceUri; }

        String bid = imageUri.getQueryParameter("bid");
        String t = imageUri.getQueryParameter("t");

        return Uri.parse("ok-image-cache:bid=" + bid + "&t=" + t);
    }
}
```

О Fresco и картинках в целом

Android: Resize a large bitmap file to scaled output file

157

I have a large bitmap (say 3888x2592) in a file. Now, I want to resize that bitmap to 800x533 and save it to another file. I normally would scale the bitmap by calling `Bitmap.createBitmap` method but it needs a source bitmap as the first argument, which I can't provide because loading the original image into a Bitmap object would of course exceed the memory (see here, for example).

92

I also can't read the bitmap with, for example, `BitmapFactory.decodeFile(file, options)`, providing a `BitmapFactory.Options.inSampleSize`, because I want to resize it to an exact width and height. Using `inSampleSize` would resize the bitmap to 972x648 (if I use `inSampleSize=4`) or to 778x518 (if I use `inSampleSize=5`, which isn't even a power of 2).

О Fresco и картинках в целом

120

No. I'd love for someone to correct me, but I accepted the load/resize approach you tried as a compromise.

Here are the steps for anyone browsing:

1. Calculate the maximum possible `inSampleSize` that still yields an image larger than your target.
2. Load the image using `BitmapFactory.decodeFile(file, options)`, passing inSampleSize as an option.
3. Resize to the desired dimensions using `Bitmap.createScaledBitmap()`.

О Fresco и картинках в целом

YES

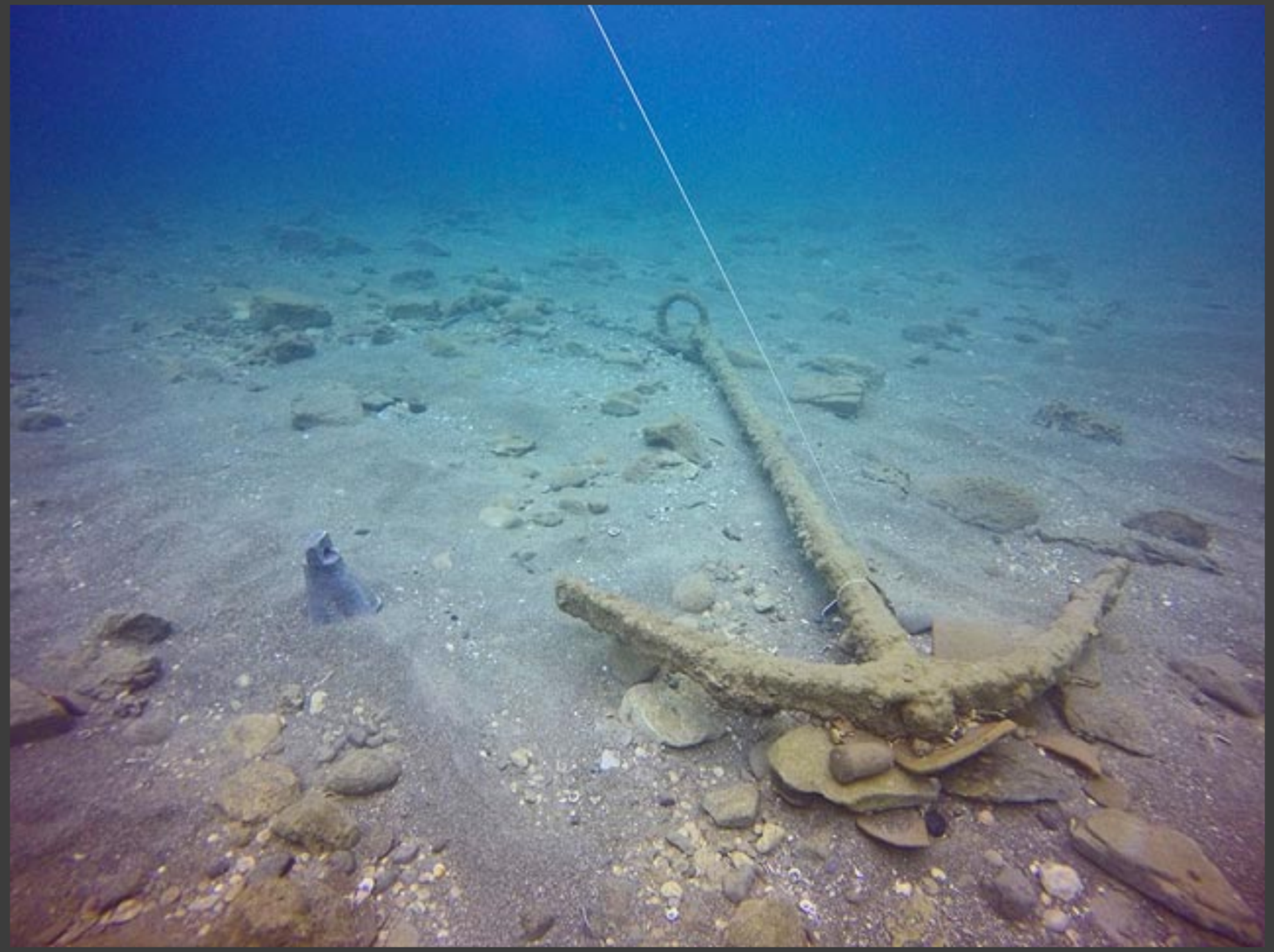
YES

```
JpegTranscoder.transcodeJpeg(final InputStream inputStream,
                             final OutputStream outputStream,
                             final int rotationAngle,
                             final int scaleNumerator,
                             final int quality)
```

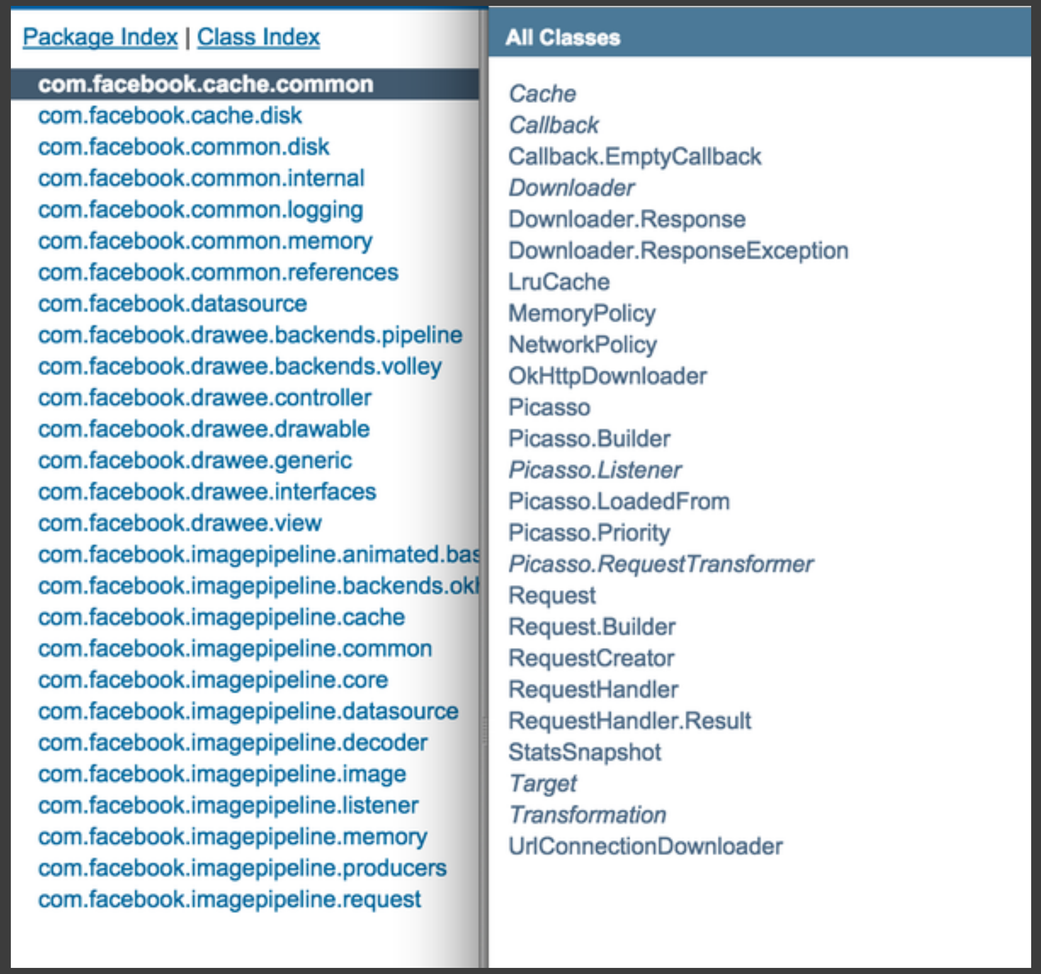
USE IT!

```
ImageRequest request = ImageRequestBuilder.newBuilderWithSource(uri)
        .setResizeOptions(new ResizeOptions(width, height)).build();
```

О Fresco и картинках в целом



Fresco vs Picasso



Package Index | Class Index

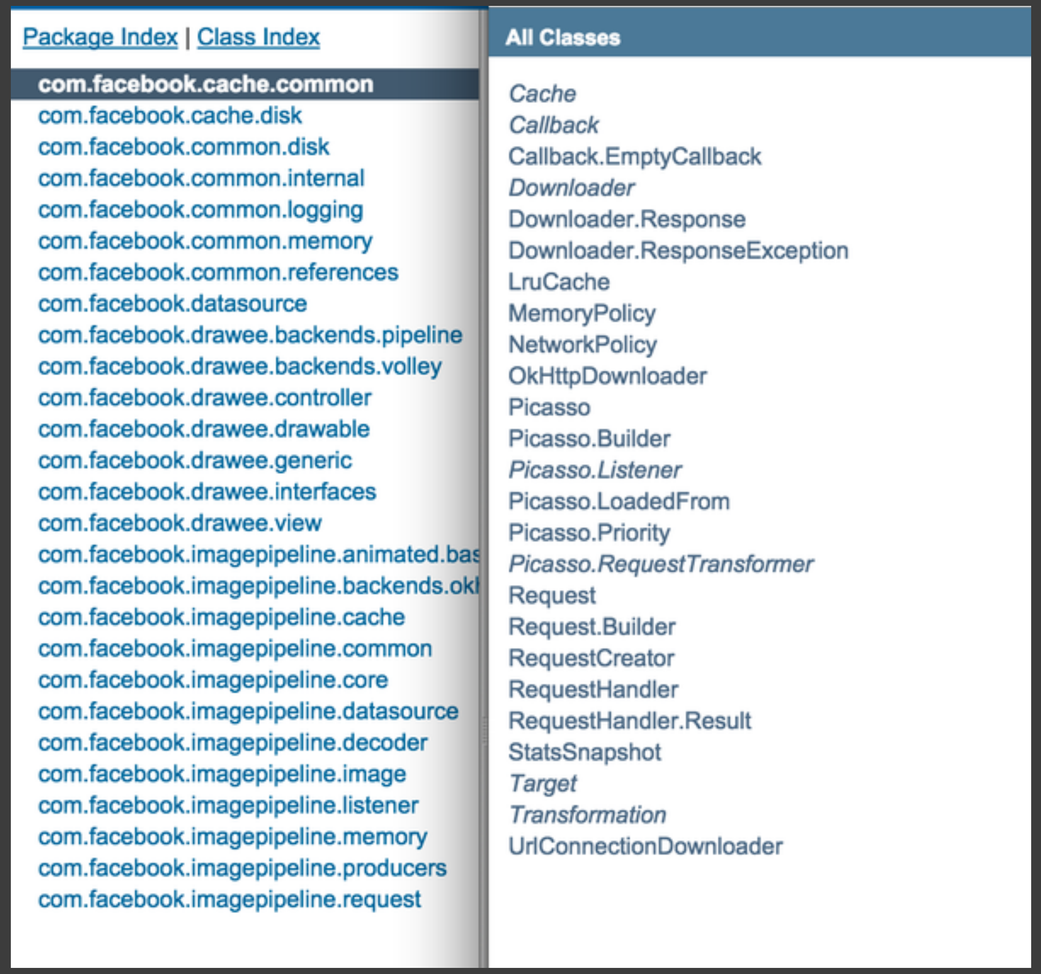
com.facebook.cache.common
com.facebook.cache.disk
com.facebook.common.disk
com.facebook.common.internal
com.facebook.common.logging
com.facebook.common.memory
com.facebook.common.references
com.facebook.datasource
com.facebook.drawee.backends.pipeline
com.facebook.drawee.backends.volley
com.facebook.drawee.controller
com.facebook.drawee.drawable
com.facebook.drawee.generic
com.facebook.drawee.interfaces
com.facebook.drawee.view
com.facebook.imagepipeline.animated.bas
com.facebook.imagepipeline.backends.okl
com.facebook.imagepipeline.cache
com.facebook.imagepipeline.common
com.facebook.imagepipeline.core
com.facebook.imagepipeline.datasource
com.facebook.imagepipeline.decoder
com.facebook.imagepipeline.image
com.facebook.imagepipeline.listener
com.facebook.imagepipeline.memory
com.facebook.imagepipeline.producers
com.facebook.imagepipeline.request

All Classes

Cache
Callback
Callback.EmptyCallback
Downloader
Downloader.Response
Downloader.ResponseException
LruCache
MemoryPolicy
NetworkPolicy
OkHttpDownloader
Picasso
Picasso.Builder
Picasso.Listener
Picasso.LoadedFrom
Picasso.Priority
Picasso.RequestTransformer
Request
Request.Builder
RequestCreator
RequestHandler
RequestHandler.Result
StatsSnapshot
Target
Transformation
UrlConnectionDownloader

Fresco vs Picasso

24 пакета



24 класса

Миграция

Простой случай

```
public class FrescoApplication extends Application {
    @Override
    public void onCreate() {
        super.onCreate();
        Fresco.initialize(this);
    }
}
```

Простой случай

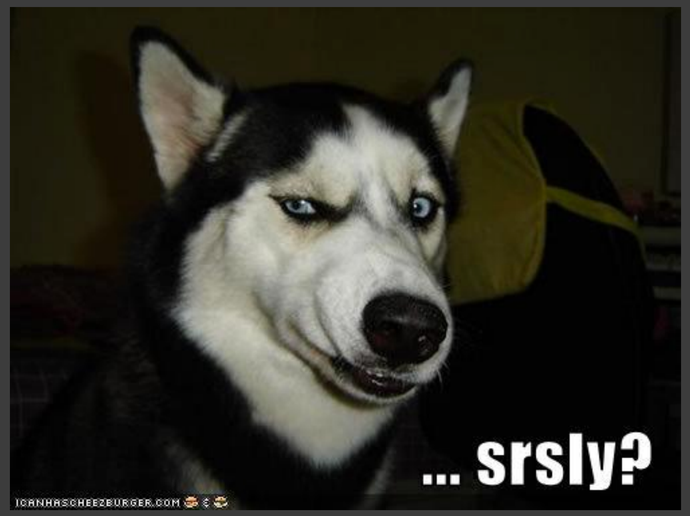
```
public class FrescoApplication extends Application {
    @Override
    public void onCreate() {
        super.onCreate();
        Fresco.initialize(this);
    }
}
```

```
<ImageView
    android:id="@+id/my_image_view"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    />
```

Простой случай

```
public class FrescoApplication extends Application {
    @Override
    public void onCreate() {
        super.onCreate();
        Fresco.initialize(this);
    }
}
```

```
<ImageView
    android:id="@+id/my_image_view"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    />
```



Простой случай

```
public class FrescoApplication extends Application {
    @Override
    public void onCreate() {
        super.onCreate();
        Fresco.initialize(this);
    }
}
```

```
<com.facebook.drawee.view.SimpleDraweeView
    android:id="@+id/my_image_view"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    />
```

Простой случай

```
public class FrescoApplication extends Application {
    @Override
    public void onCreate() {
        super.onCreate();
        Fresco.initialize(this);
    }
}
```

```
<com.facebook.drawee.view.SimpleDraweeView
    android:id="@+id/my_image_view"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    fresco:viewAspectRatio="1.33"
    />
```

Простой случай

```
public class FrescoApplication extends Application {
    @Override
    public void onCreate() {
        super.onCreate();
        Fresco.initialize(this);
    }
}
```

```
<com.facebook.drawee.view.SimpleDraweeView
    android:id="@+id/my_image_view"
    android:layout_width="130dp"
    android:layout_height="130dp"
    />
```

Простой случай

```
public class FrescoApplication extends Application {
    @Override
    public void onCreate() {
        super.onCreate();
        Fresco.initialize(this);
    }
}
```

```
<com.facebook.drawee.view.SimpleDraweeView
    android:id="@+id/my_image_view"
    android:layout_width="130dp"
    android:layout_height="130dp"
    />
```

```
SimpleDraweeView draweeView = (SimpleDraweeView) findViewById(R.id.my_image_view);
draweeView.setImageURI(YOUR_URI);
```

Сложный случай

```
public class MyCustomView extends View {

    @Override
    protected void onDraw(Canvas canvas) {
        canvas.drawBitmap(bitmap, 0, 0, null);
    }
}
```

Сложный случай

```
public class MyCustomView extends View {

    @Override
    protected void onDraw(Canvas canvas) {
        canvas.drawBitmap(bitmap, 0, 0, null);
    }
}
```

Сложный случай

```
private init() {
    holder = DraweeHolder.create(hierarchy, getContext())
    holder.getTopLevelDrawable().setCallback(this);
}
```

Сложный случай

```
@Override
protected void onDetachedFromWindow() {
    super.onDetachedFromWindow();
    holder.onDetach();
}

@Override
public void onStartTemporaryDetach() {
    super.onStartTemporaryDetach();
    holder.onDetach();
}
```

```
@Override
protected void onAttachedToWindow() {
    super.onAttachedToWindow();
    holder.onAttach();
}

@Override
public void onFinishTemporaryDetach() {
    super.onFinishTemporaryDetach();
    holder.onAttach();
}
```

Сложный случай

```
@Override
protected void onDraw(Canvas canvas) {
    Drawable drawable = holder.getTopLevelDrawable();
    drawable.setBounds(
            getPaddingLeft(),
            getPaddingTop(),
            getWidth() - getPaddingRight(),
            getHeight() - getPaddingBottom());
    drawable.draw(canvas);
}
```

Сложный случай

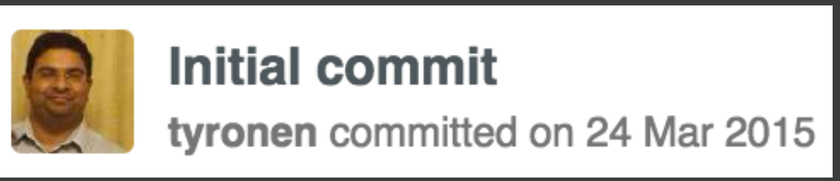
```
@Override
protected void onDraw(Canvas canvas) {
    Drawable drawable = holder.getTopLevelDrawable();
    drawable.setBounds(
            getPaddingLeft(),
            getPaddingTop(),
            getWidth() - getPaddingRight(),
            getHeight() - getPaddingBottom());
    drawable.draw(canvas);
}

@Override
protected boolean verifyDrawable(Drawable who) {
    return holder.getTopLevelDrawable() == who || super.verifyDrawable(who);
}
```

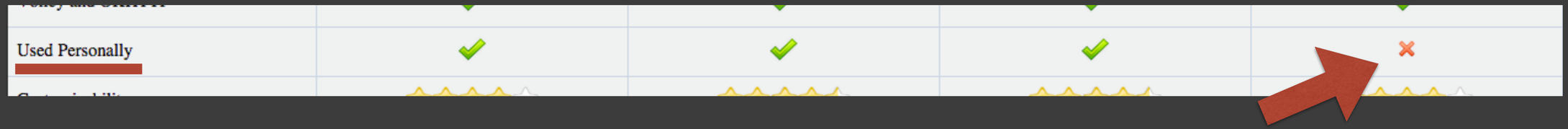
Сложный случай

```
public void setUri(Uri uri, Uri lowResUri) {
    holder.setController(Fresco.newDraweeControllerBuilder()
        .setRetainImageOnFailure(true)
        .setLowResImageRequest(ImageRequest.fromUri(lowResUri))
        .setImageRequest(ImageRequest.fromUri(uri))
        .setOldController(holder.getController())
        .build()
    );
}
```

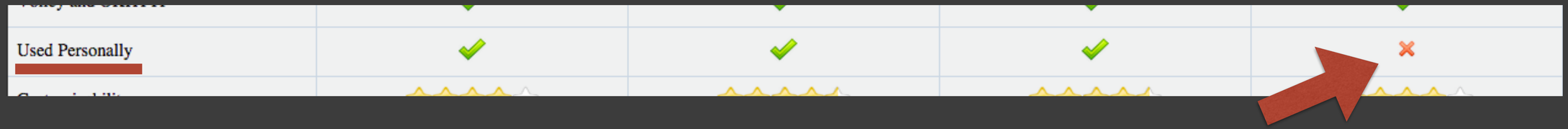
О Fresco и картинках в целом



Комментарии про Fresco

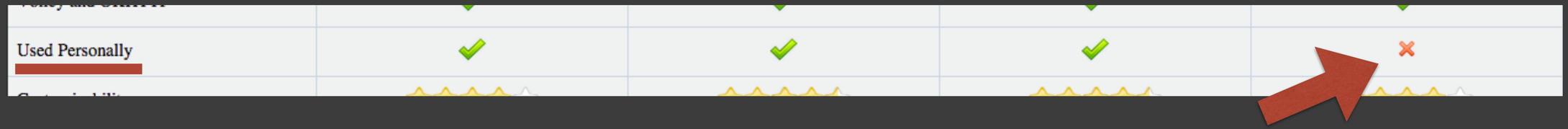


Комментарии про Fresco



Picasso is an easy to use image loader same goes for Imageloader. Fresco Is different approach to image loading haven't used it yet but it looks too me like more like a solution for

Комментарии про Fresco



Picasso is an easy to use image loader same goes for Imageloader. Fresco Is different approach to image loading haven't used it yet but it looks too me like more like a solution for

Fresco was out of the benchmark because for the project I was running the test, we didn't want to refactor our layouts (because of the Drawee view).

Спасибо за внимание!