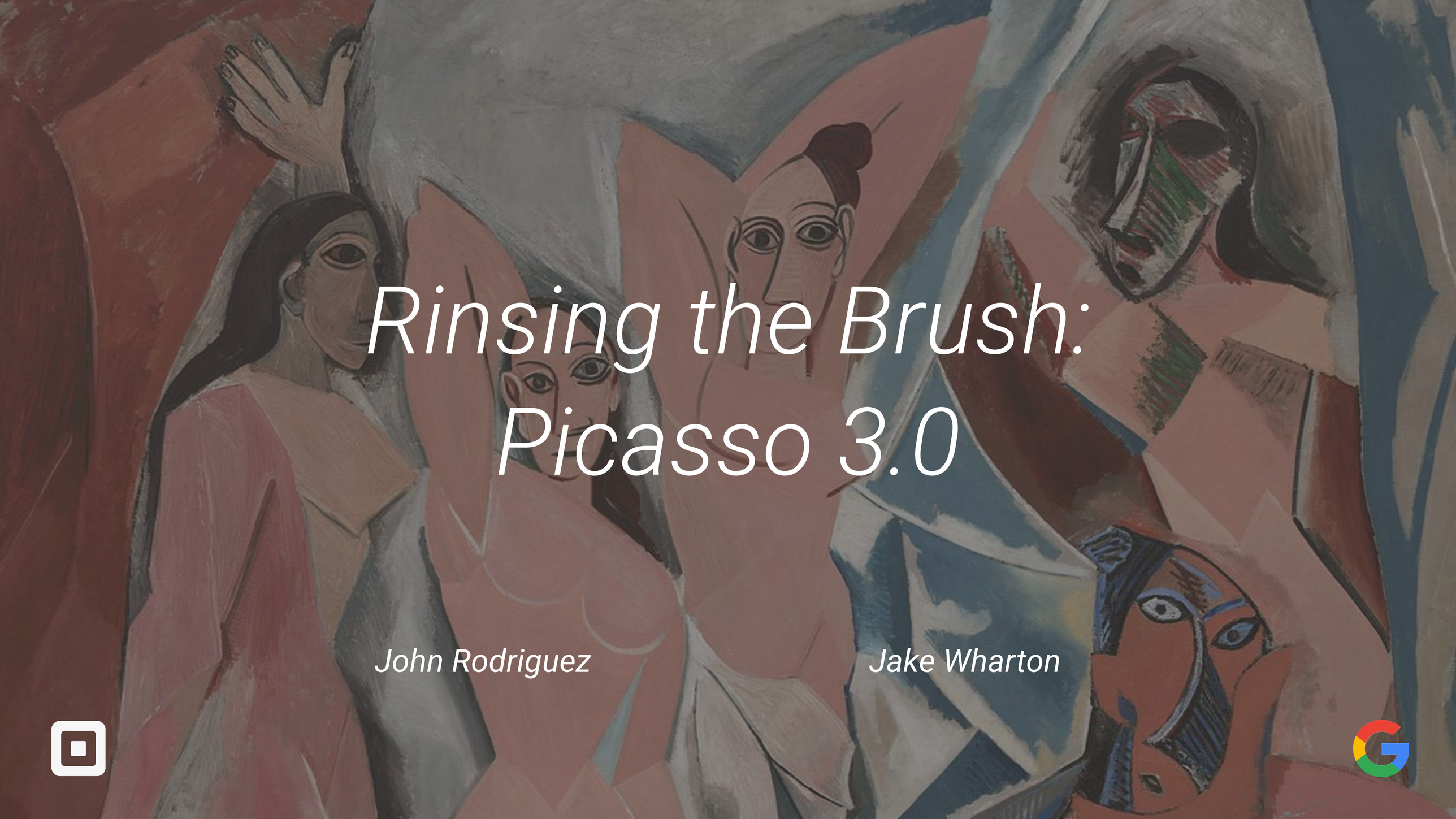


Rinsing the Brush: Picasso 3.0

John Rodriguez

Jake Wharton

square / picasso

Watch 972 Star 15,285 Fork 3,816

Code Issues 129 Pull requests 17 Insights

Releases Tags

picasso-parent-1.0...

dea2e24

1.0.0

JakeWharton released this on Aug 30, 2013 · 980 commits to master since this release

Assets

Source code (zip)

Source code (tar.gz)

Initial release.

Square Engineering Follow
The official account for @Square Engineering.
May 14, 2013 · 2 min read

Enhance Your Application Using Picasso

A fluent image downloading and caching library for Android.

Written by D. Koutsogiorgas and Jake Wharton.

Today, we would like to introduce and open source Picasso, our solution for image downloading and caching on Android.

Picasso aims to be fast and simple to use—often requiring only one line of code.

```
Picasso.with(context).load("http://example.com/logo.png").into(imageView);
```

Today, we would like to introduce and open source Picasso, our solution for image downloading and caching on Android.

Picasso aims to be fast and simple to use—often requiring only one line of code.

```
Picasso.with(context).load("http://example.com/logo.png").into(imageView);
```

And that's it! You can use this for downloading a single image or inside of an adapter's getView method to download many. Picasso automatically handles caching, recycling, and displaying the final bitmap into the target view.

Picasso also allows you to transform images.

```
Picasso.with(context)
    .load("http://example.com/logo.png")
    .resize(100, 100)
```

square / picasso

Watch 972 Star 15,285 Fork 3,816

Code Issues 129 Pull requests 17 Insights

Releases Tags

Since on Feb 1, 2014 ••• Show 5 newer tags

Latest release

picasso-parent-2....

6d63733

2.0.0

JakeWharton released this on Aug 30, 2013 · 764 commits to master since this release

Assets

Source code (zip)

Source code (tar.gz)

square / picasso

Watch 972 Star 15,285 Fork 3,816

Code Issues 129 Pull requests 17 Insights

Releases Tags

on Mar 20, 2015
picasso-parent-2.5.2 ···
6930ff1 zip tar.gz

on Mar 19, 2015
picasso-parent-2.5.1 ···
f0b1737 zip tar.gz

on Feb 6, 2015
picasso-parent-2.5.0 ···
a98d695 zip tar.gz

on Nov 4, 2014
picasso-parent-2.4.0 ···
4a8cf5b zip tar.gz

3 years later

3.x Goals

3.x Goals

- Android P

3.x Goals

- Android P
- OkHttp 2.x => 3.x

3.x Goals

- Android P
- OkHttp 2.x => 3.x
- Okio integration

3.x Goals

- Android P
- OkHttp 2.x => 3.x
- Okio integration
- Improvements

3.x Goals

- Android P
- OkHttp 2.x => 3.x
- Okio integration
- Improvements

ImageDecoder

```
public final class ImageDecoder
extends Object implements AutoCloseable
```

java.lang.Object
↳ android.graphics.ImageDecoder

Class for decoding images as `Bitmap`s or `Drawable`s.

Summary

Nested classes	
`@interface`	`ImageDecoder.Error`
`class`	`ImageDecoder.ImageInfo` Contains information about the encoded image.
`class`	`ImageDecoder.IncompleteException` Thrown if the provided data is incomplete.

ImageDecoder


```
public final class ImageDecoder
extends Object implements AutoCloseable
```

java.lang.Object

↳ android.graphics.ImageDecoder

Class for decoding images as `Bitmap`s or `Drawable`s.

Summary

Nested classes	
@interface	ImageDecoder.Error
class	ImageDecoder.ImageInfo

```
Bitmap decodeStreamPreP(Request r, BufferedSource bs) {
  BitmapFactory.Options options = createBitmapOptions(request);

  if (requiresInSampleSize(options)) {
    InputStream stream = new SourceBufferingInputStream(bs);
    BitmapFactory.decodeStream(stream, null, options);

    calculateInSampleSize(r.targetWidth, r.targetHeight, options, r);
  }

  return BitmapFactory.decodeStream(bs.inputStream(), null, options);
}
```

```
Bitmap decodeStreamPreP(Request r, BufferedSource bs) {
  BitmapFactory.Options options = createBitmapOptions(request);

  if (requiresInSampleSize(options)) {
    InputStream stream = new SourceBufferingInputStream(bs);
    BitmapFactory.decodeStream(stream, null, options);

    calculateInSampleSize(r.targetWidth, r.targetHeight, options, r);
  }

  return BitmapFactory.decodeStream(bs.inputStream(), null, options);
}
```

Class for decoding images as `Bitmap`s or `Drawable`s.

Summary

Nested classes	
`@interface`	`ImageDecoder.Error`
`class`	`ImageDecoder.ImageInfo` Contains information about the encoded image.
`class`	`ImageDecoder.IncompleteException` Thrown if the provided data is incomplete.
`interface`	`ImageDecoder.OnHeaderDecodedListener` Optional listener supplied to `decodeDrawable(ImageDecoder.Source)` or `decodeBitmap(ImageDecoder.Source)`.
`interface`	`ImageDecoder.OnPartialImageListener`

```
Bitmap decodeStreamP(Request r, BufferedSource bs) {
  ImageDecoder.Source imageSource =
      ImageDecoder.createSource(ByteBuffer.wrap(bs.readByteArray()));

  return ImageDecoder.decodeBitmap(
    imageSource,

    new OnHeaderDecodedListener() {
      void onHeaderDecoded(ImageDecoder d, ImageInfo info, Source src) {
        if (r.hasSize()) {
          d.setTargetSize(r.targetWidth, r.targetHeight);
        }
      }
    });
}
```

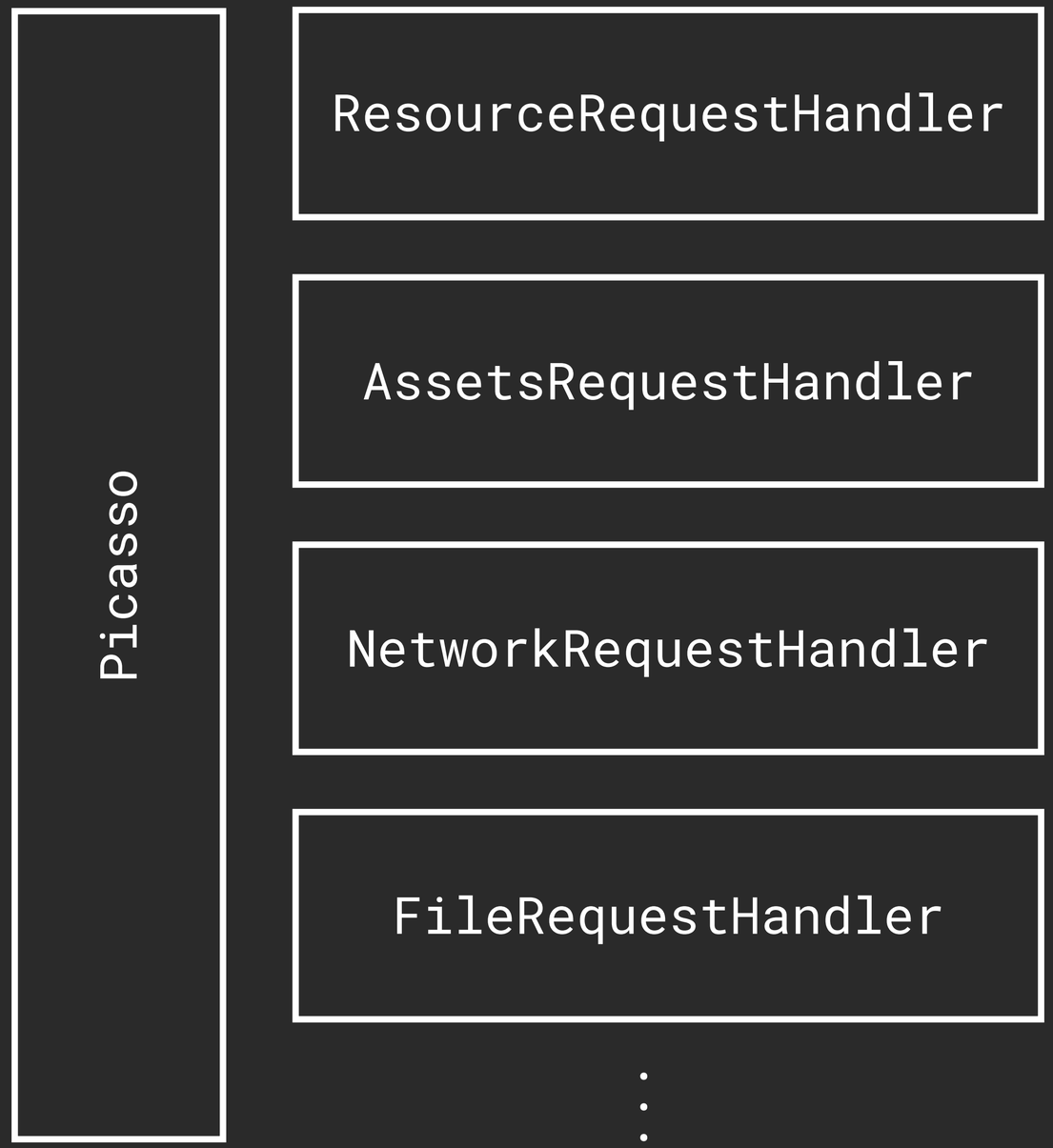
```
Bitmap decodeStreamP(Request r, BufferedSource bs) {
  ImageDecoder.Source imageSource =
      ImageDecoder.createSource(ByteBuffer.wrap(bs.readByteArray()));

  return ImageDecoder.decodeBitmap(
    imageSource,

    new OnHeaderDecodedListener() {
      void onHeaderDecoded(ImageDecoder d, ImageInfo info, Source src) {
        if (r.hasSize()) {
          d.setTargetSize(r.targetWidth, r.targetHeight);
        }
      }
    });
}
```

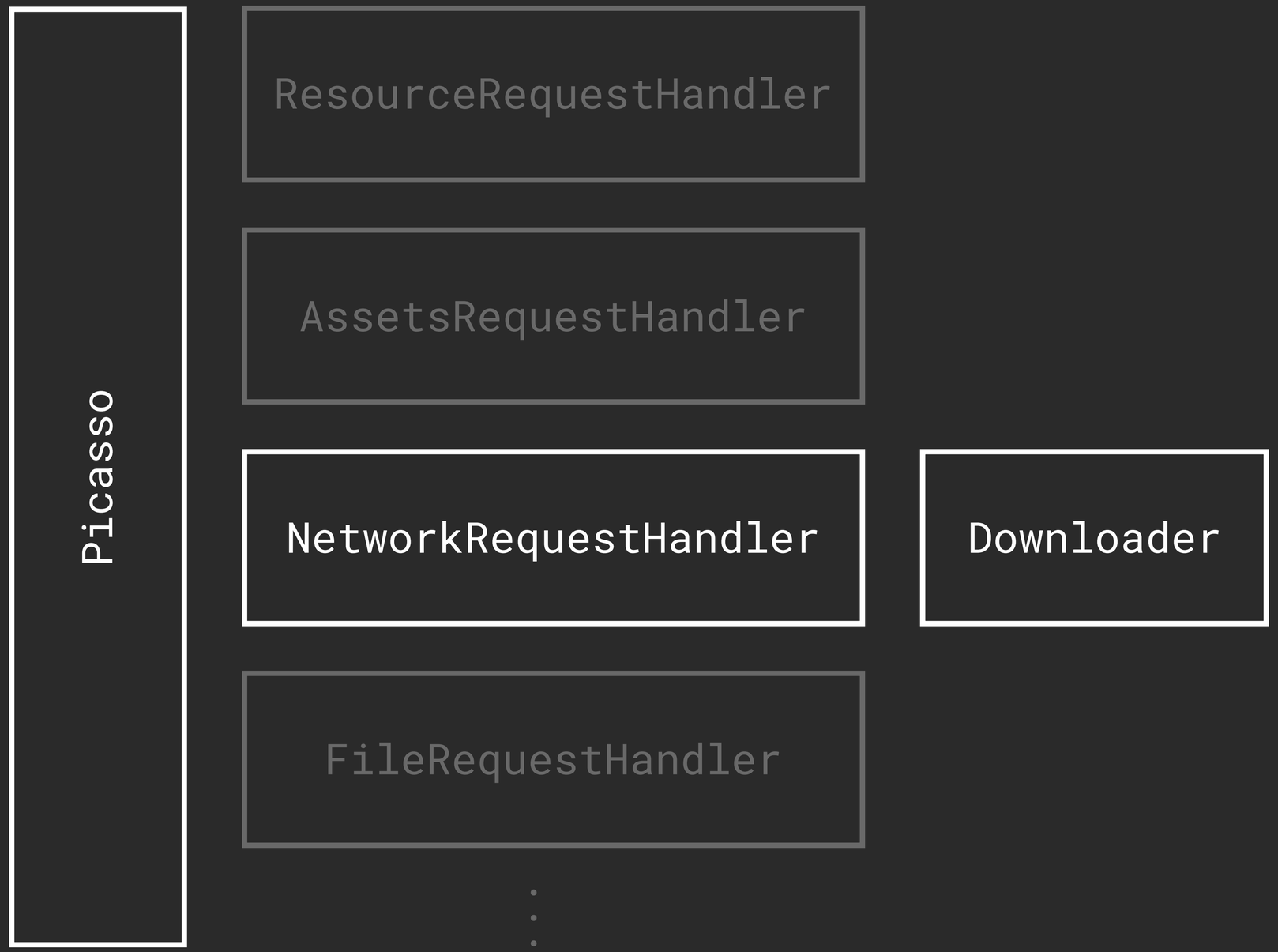
```
Bitmap decodeStreamP(Request r, BufferedSource bs) {
  ImageDecoder.Source imageSource =
      ImageDecoder.createSource(ByteBuffer.wrap(bs.readByteArray()));

  return ImageDecoder.decodeBitmap(
    imageSource,

    new OnHeaderDecodedListener() {
      void onHeaderDecoded(ImageDecoder d, ImageInfo info, Source src) {
        if (r.hasSize()) {
          d.setTargetSize(r.targetWidth, r.targetHeight);
        }
      }
    });
}
```

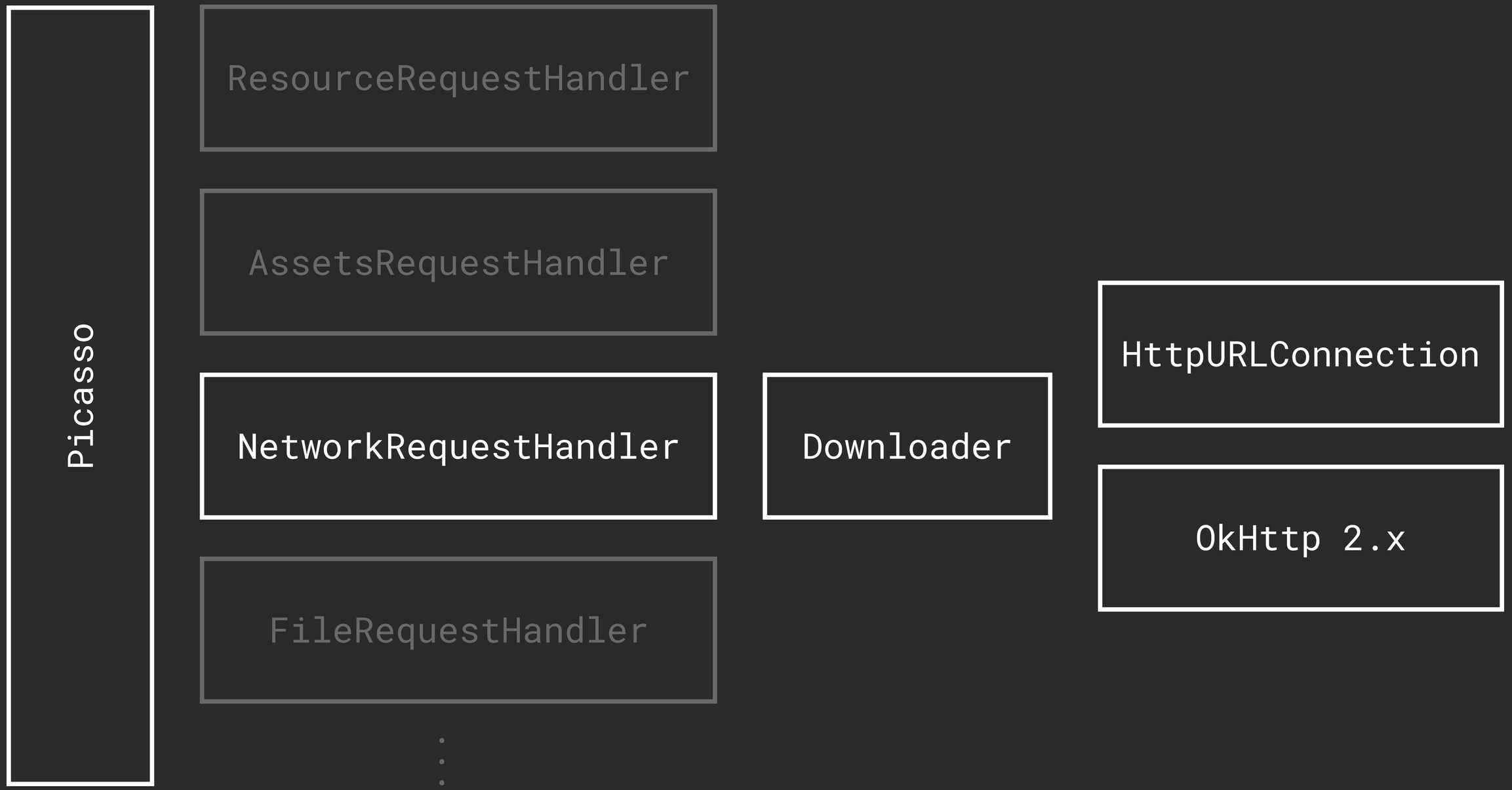
3.x Goals

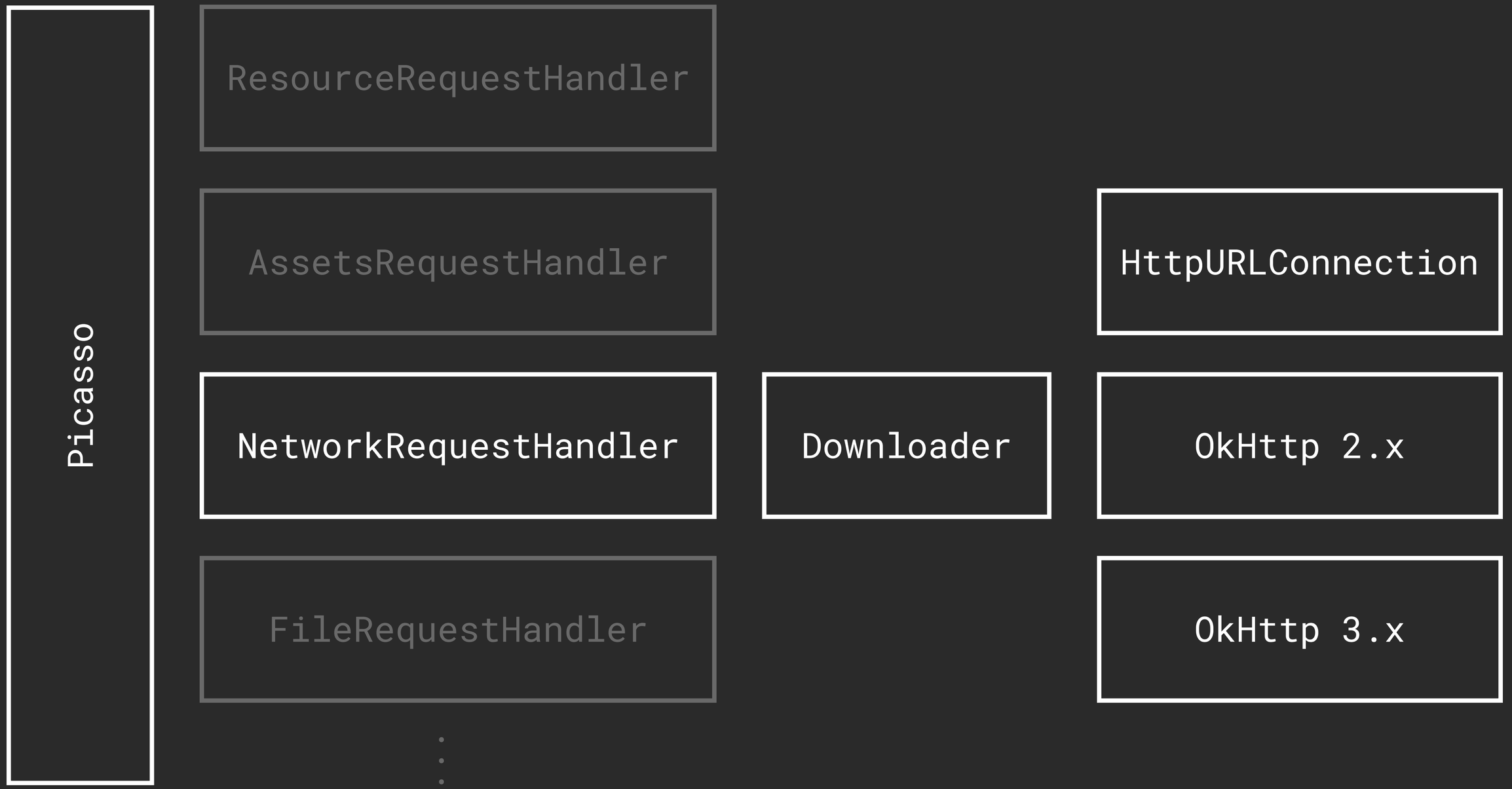
- Android P
- OkHttp 2.x => 3.x
- Okio integration
- Improvements

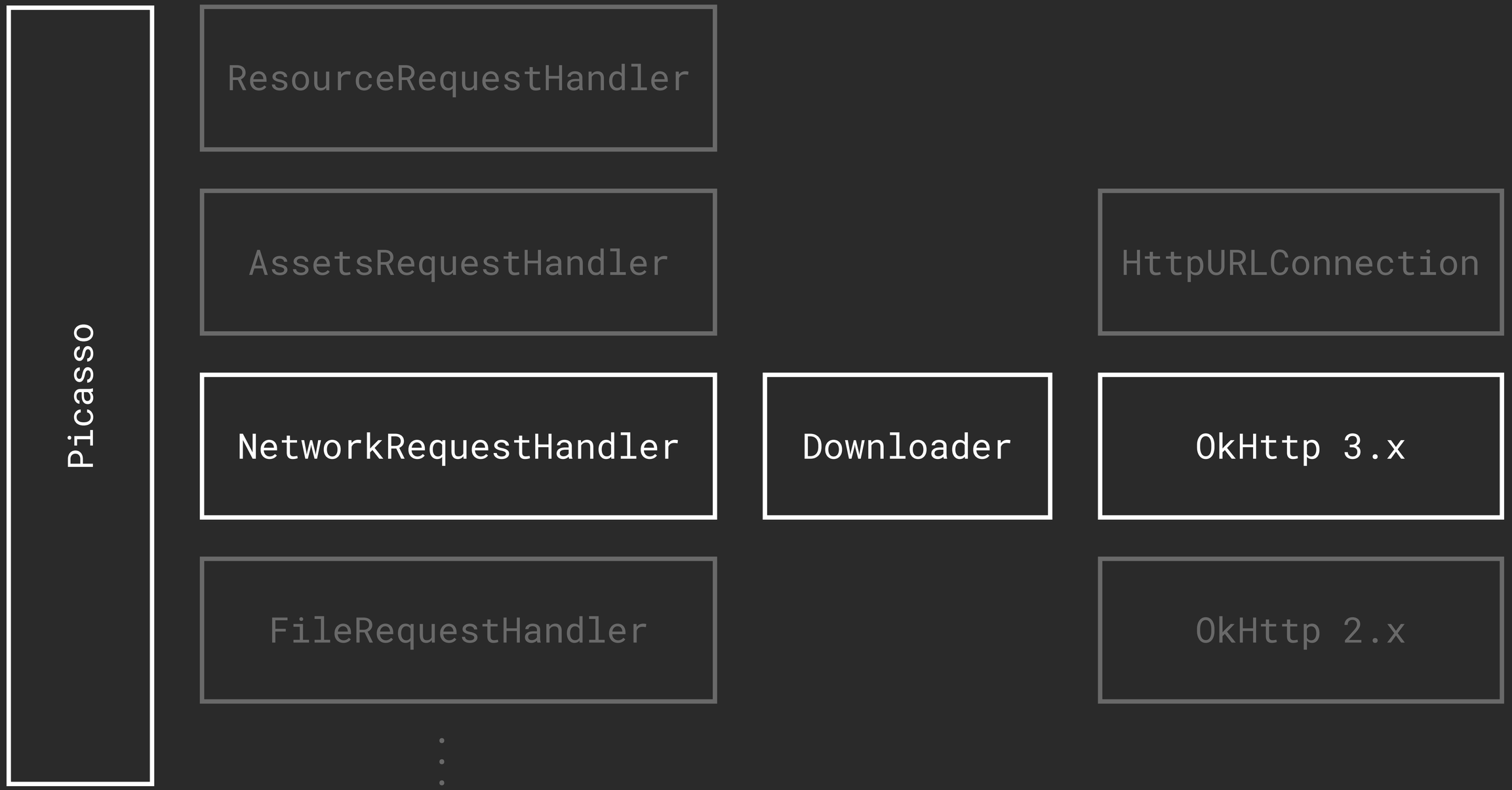
Picasso

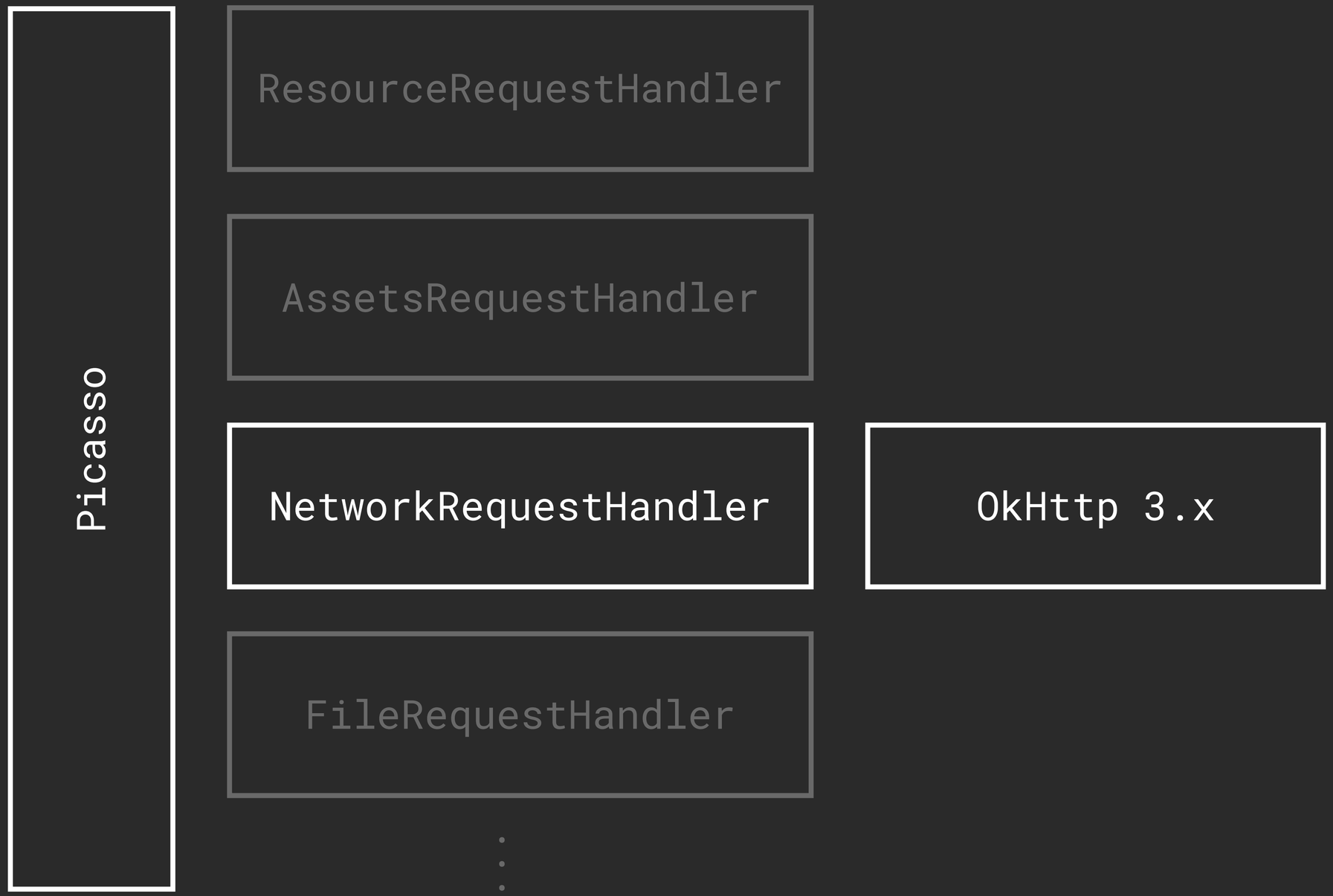

Picasso
ResourceRequestHandler
AssetsRequestHandler
NetworkRequestHandler
FileRequestHandler

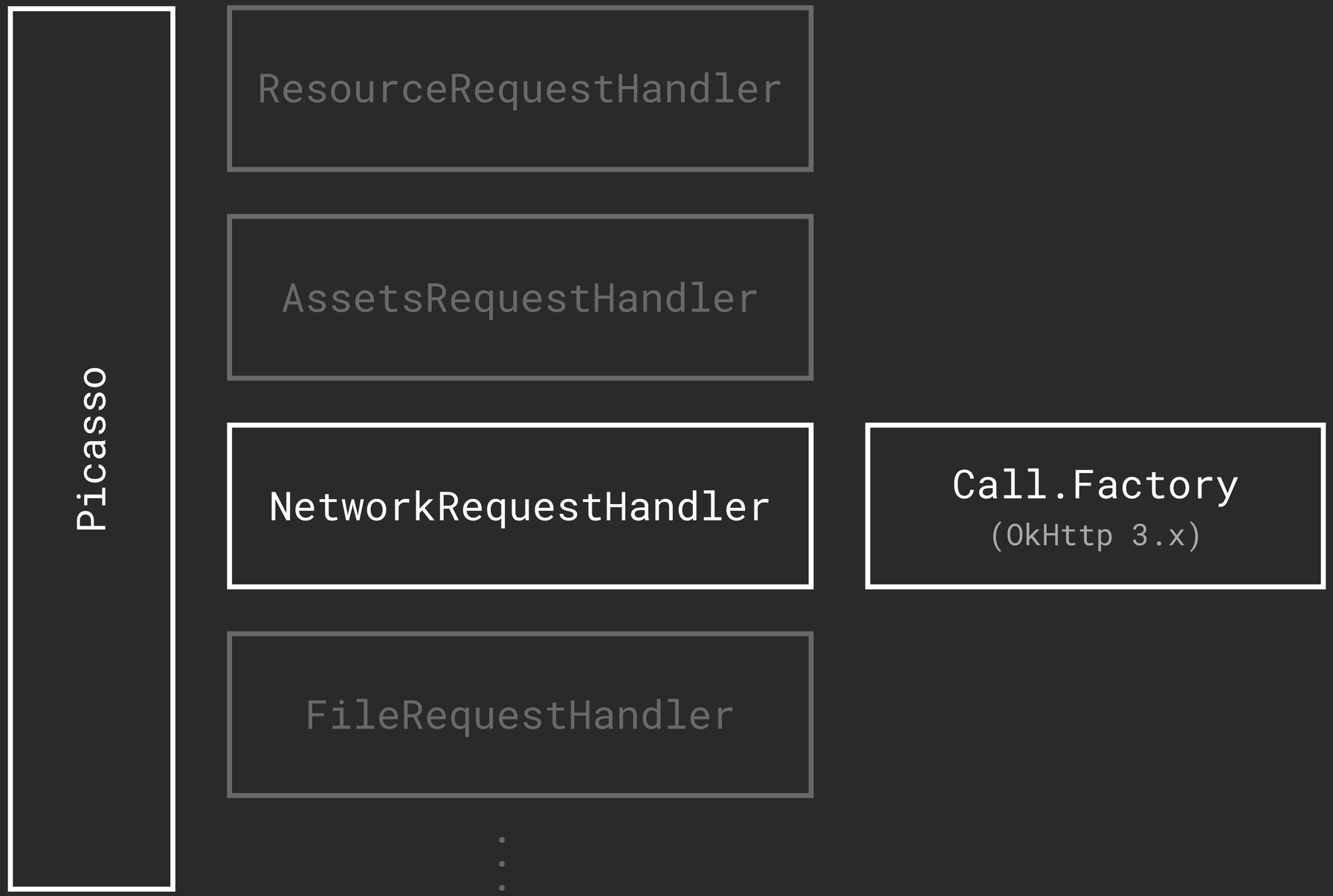

Picasso
ResourceRequestHandler
AssetsRequestHandler
NetworkRequestHandler
FileRequestHandler
Downloader

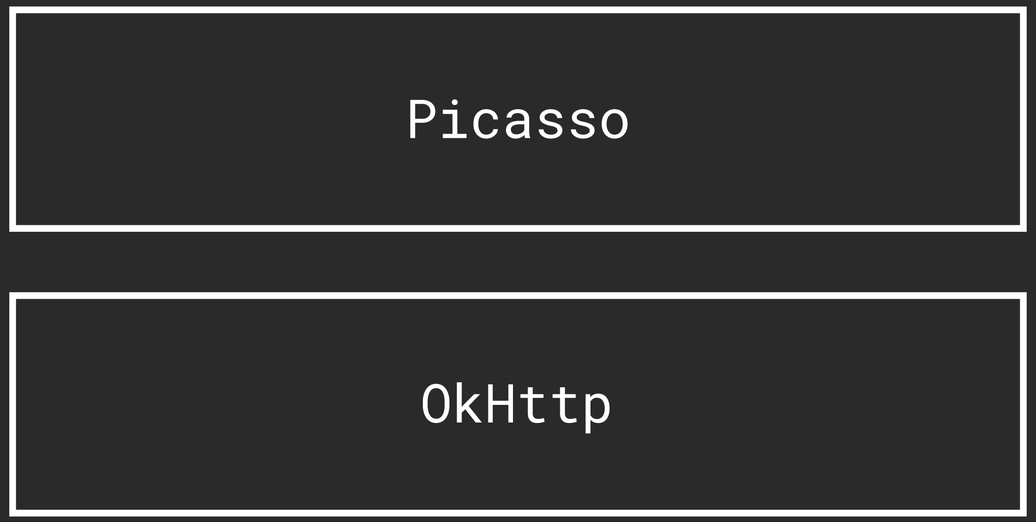

Picasso
ResourceRequestHandler
AssetsRequestHandler
NetworkRequestHandler
FileRequestHandler
Downloader
HttpURLConnection
OkHttp 2.x

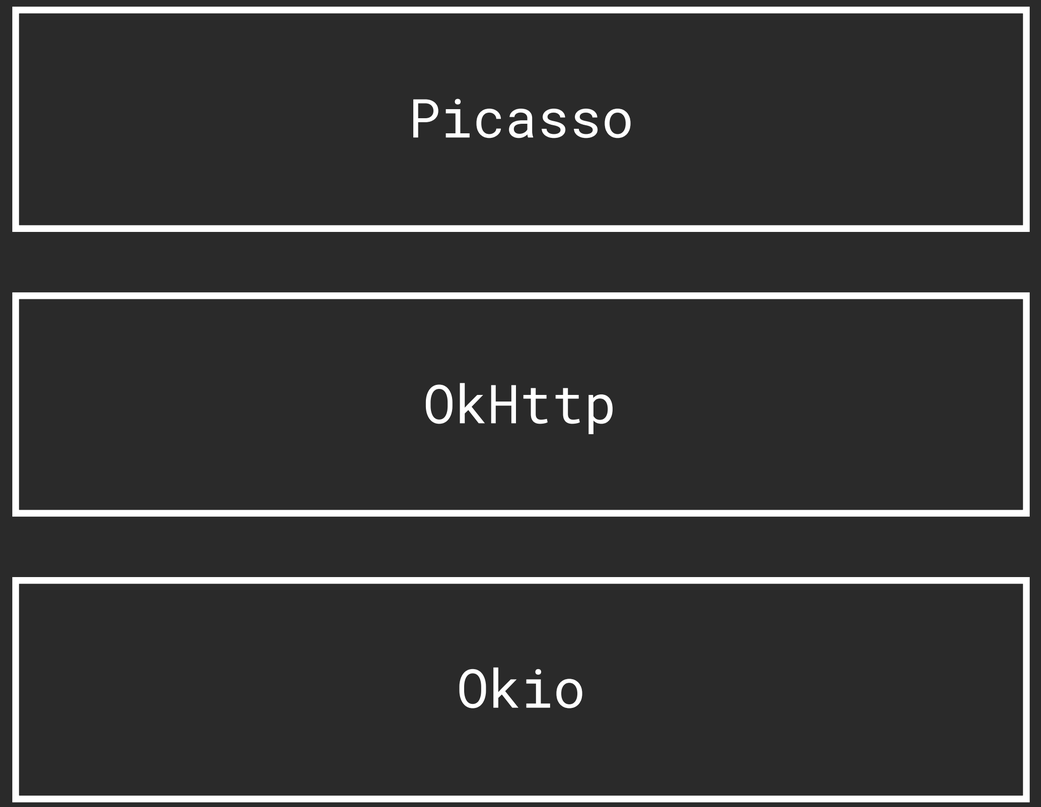

Picasso
ResourceRequestHandler
AssetsRequestHandler
NetworkRequestHandler
FileRequestHandler
Downloader
HttpURLConnection
OkHttp 2.x
OkHttp 3.x


Picasso
ResourceRequestHandler
AssetsRequestHandler
NetworkRequestHandler
FileRequestHandler
Downloader
HttpURLConnection
OkHttp 3.x
OkHttp 2.x


Picasso
ResourceRequestHandler
AssetsRequestHandler
NetworkRequestHandler
OkHttp 3.x
FileRequestHandler

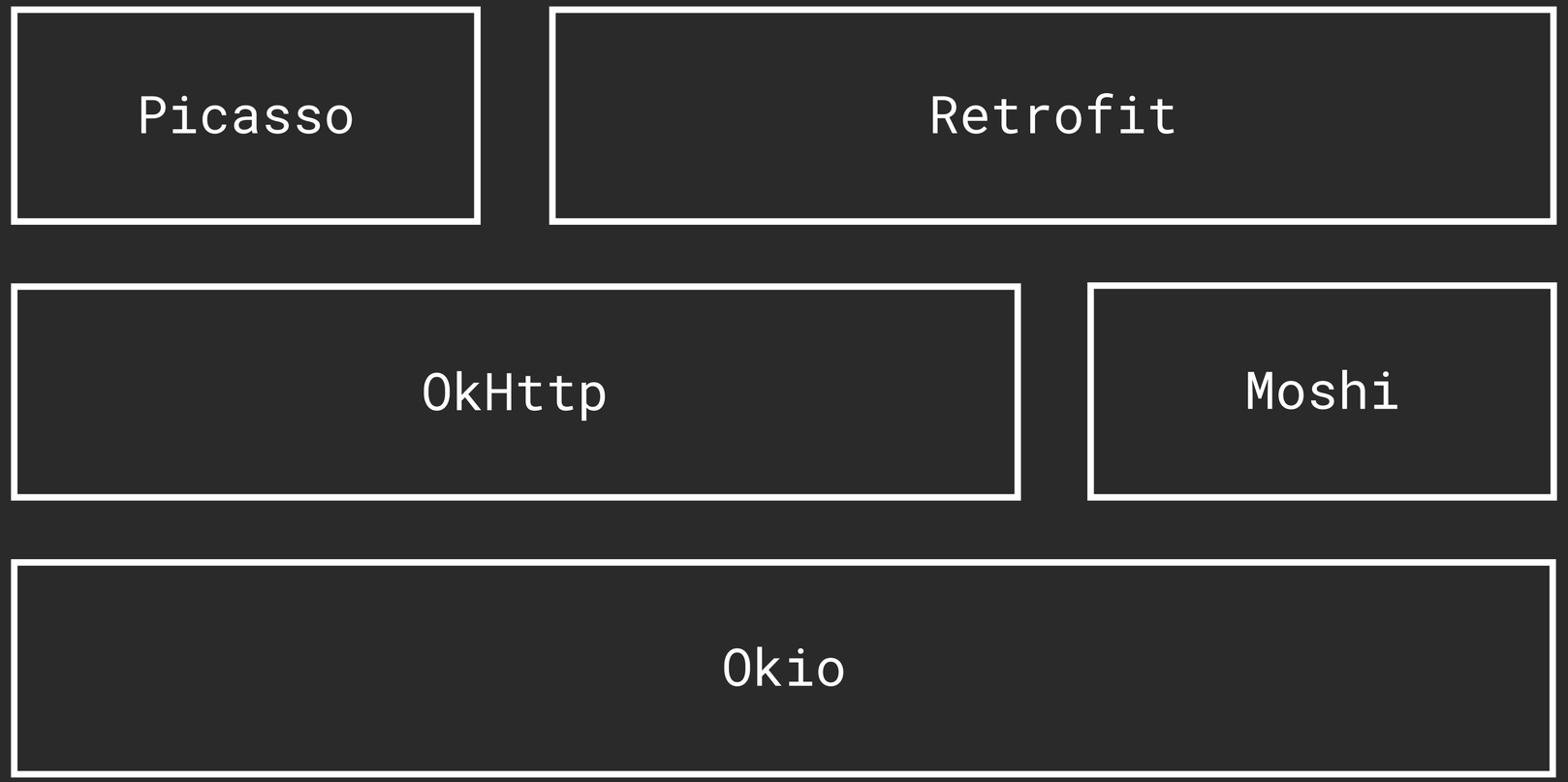

Picasso
ResourceRequestHandler
AssetsRequestHandler
NetworkRequestHandler
FileRequestHandler
Call.Factory
(OkHttp 3.x)

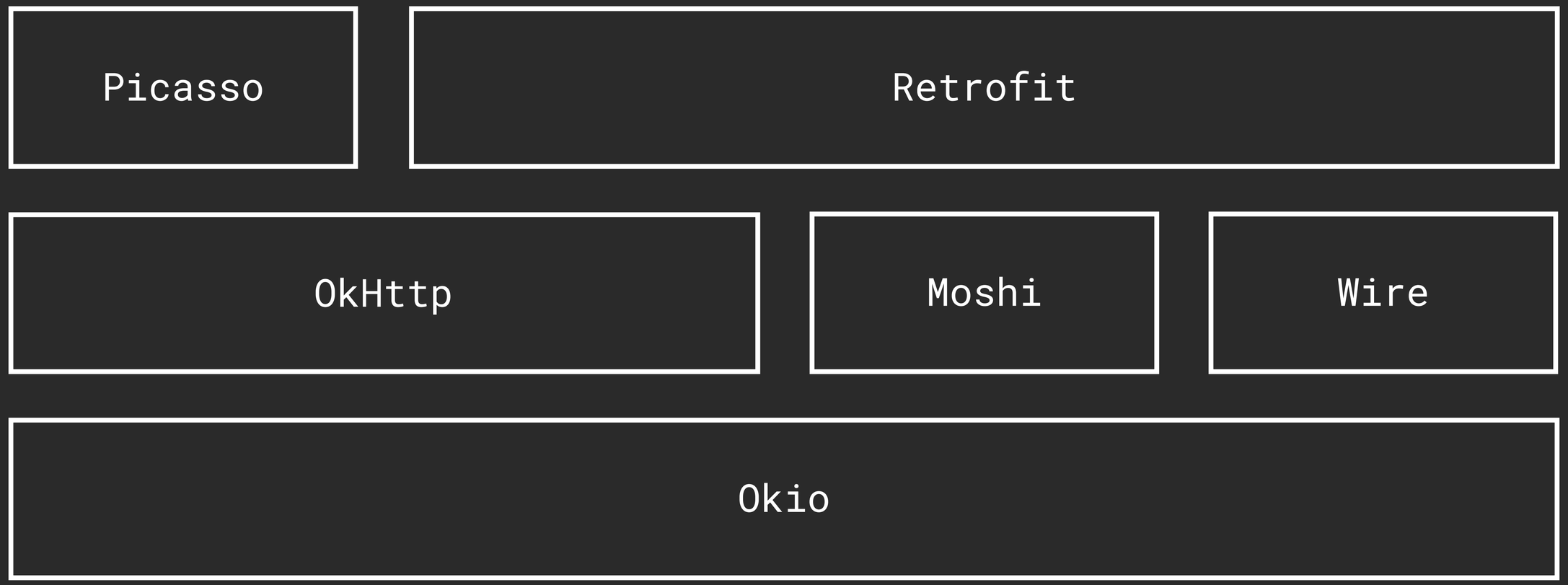

Picasso
OkHttp


Picasso
OkHttp
Okio


Picasso
Retrofit
OkHttp
Okio


Picasso
Retrofit
OkHttp
Moshi
Okio


Picasso
Retrofit
OkHttp
Moshi
Okio


Picasso
Retrofit
OkHttp
Moshi
Wire
Okio

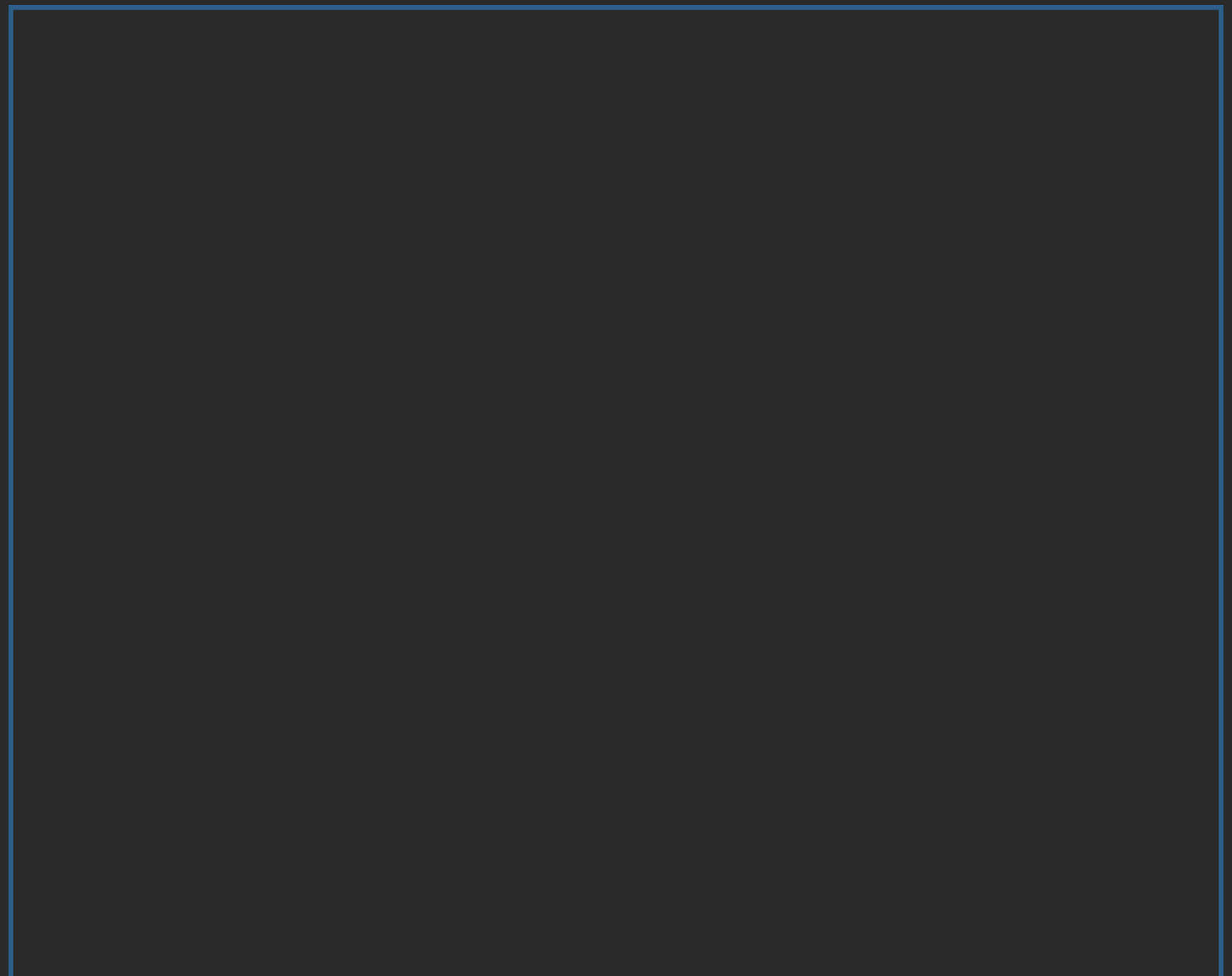

Picasso
Retrofit
OkHttp
Moshi
Wire
Okio

3.x Goals

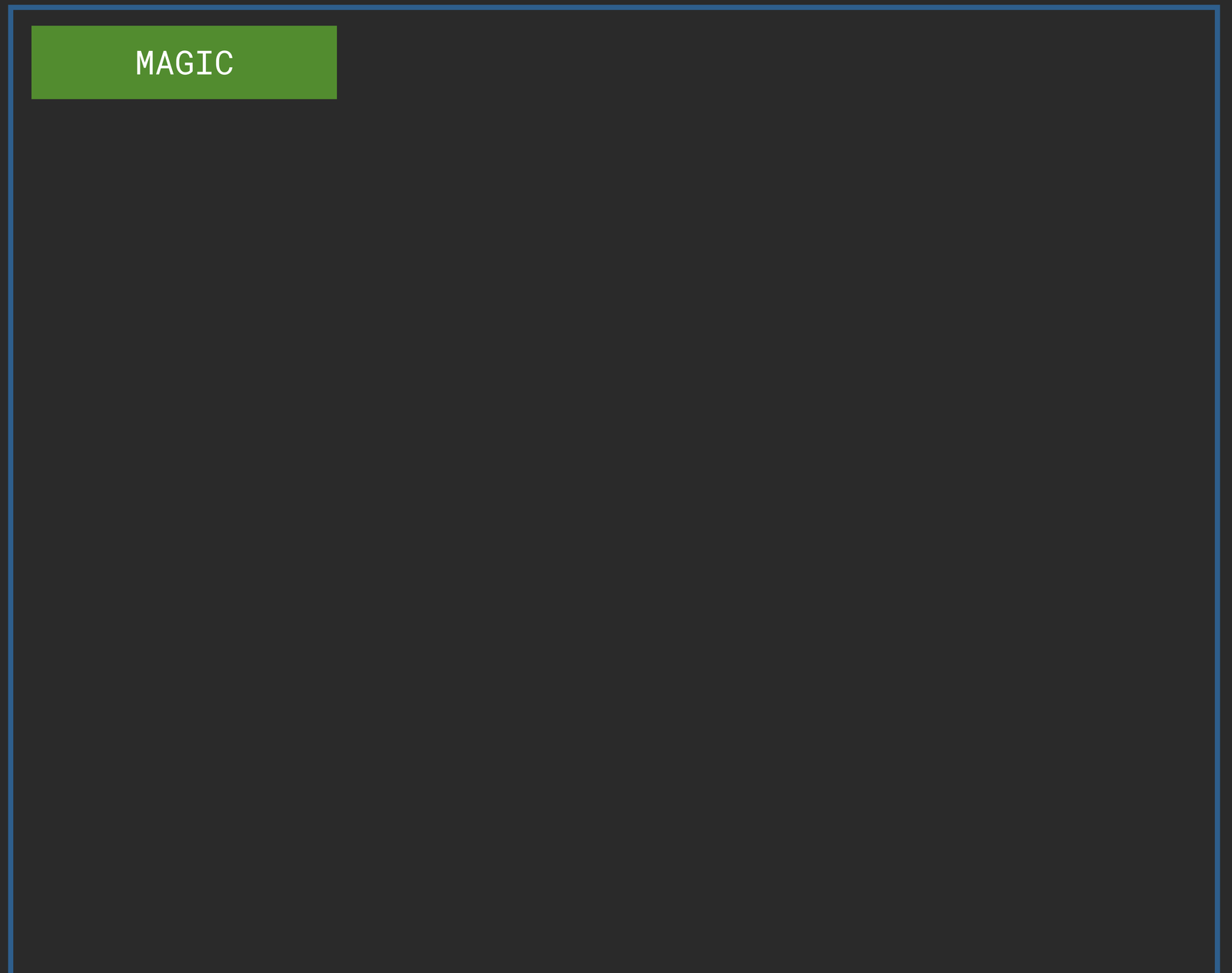
- Android P
- OkHttp 2.x => 3.x
- Okio integration
- Improvements

```
java.io.IOException: Cannot reset
    at com.squareup.picasso.MarkableInputStream.reset(MarkableInputStream.java:99)
    at com.squareup.picasso.BitmapHunter.decodeStream(BitmapHunter.java:140)
    at com.squareup.picasso.BitmapHunter.hunt(BitmapHunter.java:217)
    at com.squareup.picasso.BitmapHunter.run(BitmapHunter.java:159)
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:457)
    at java.util.concurrent.FutureTask.run(FutureTask.java:266)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1162)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:636)
    at java.lang.Thread.run(Thread.java:764)
    at com.squareup.picasso.Utils$PicassoThread.run(Utils.java:411)
```

nightmare_shrek.png

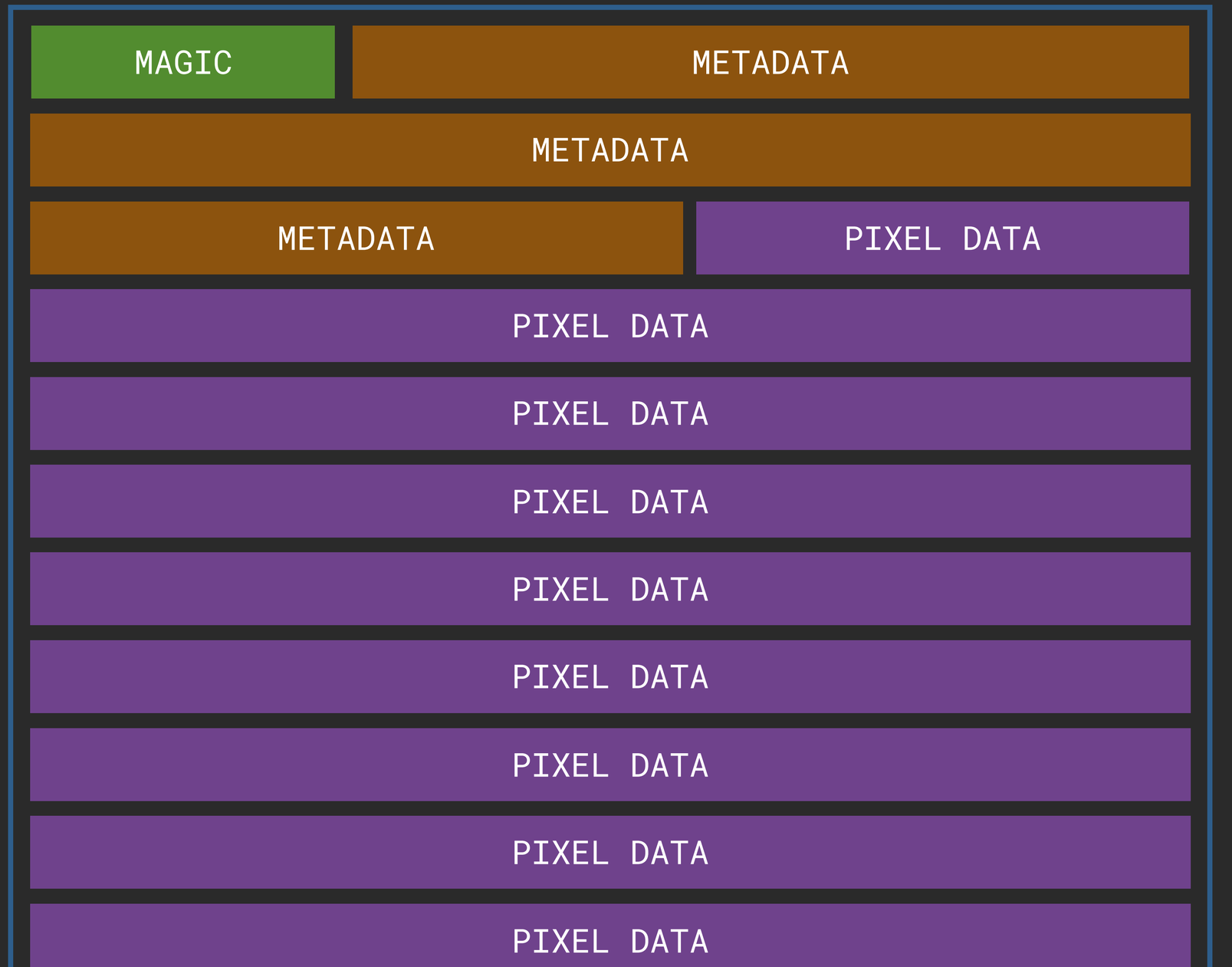


nightmare_shrek.png



nightmare_shrek.png

nightmare_shrek.png




MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA


MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA

MAGIC

METADATA

METADATA

METADATA

PIXEL DATA

PIXEL DATA

PIXEL DATA

PIXEL DATA

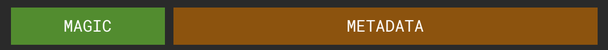
PIXEL DATA

PIXEL DATA

PIXEL DATA

PIXEL DATA

PIXEL DATA


MAGIC
METADATA


METADATA
METADATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA

MAGIC	METADATA	METADATA

METADATA	PIXEL DATA
PIXEL DATA	
PIXEL DATA	
PIXEL DATA	
PIXEL DATA	
PIXEL DATA	
PIXEL DATA	
PIXEL DATA	
PIXEL DATA	

MAGIC | METADATA | METADATA | METADATA

PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA

InputStream

MAGIC | METADATA | METADATA | METADATA

PIXEL DATA

PIXEL DATA

PIXEL DATA

PIXEL DATA

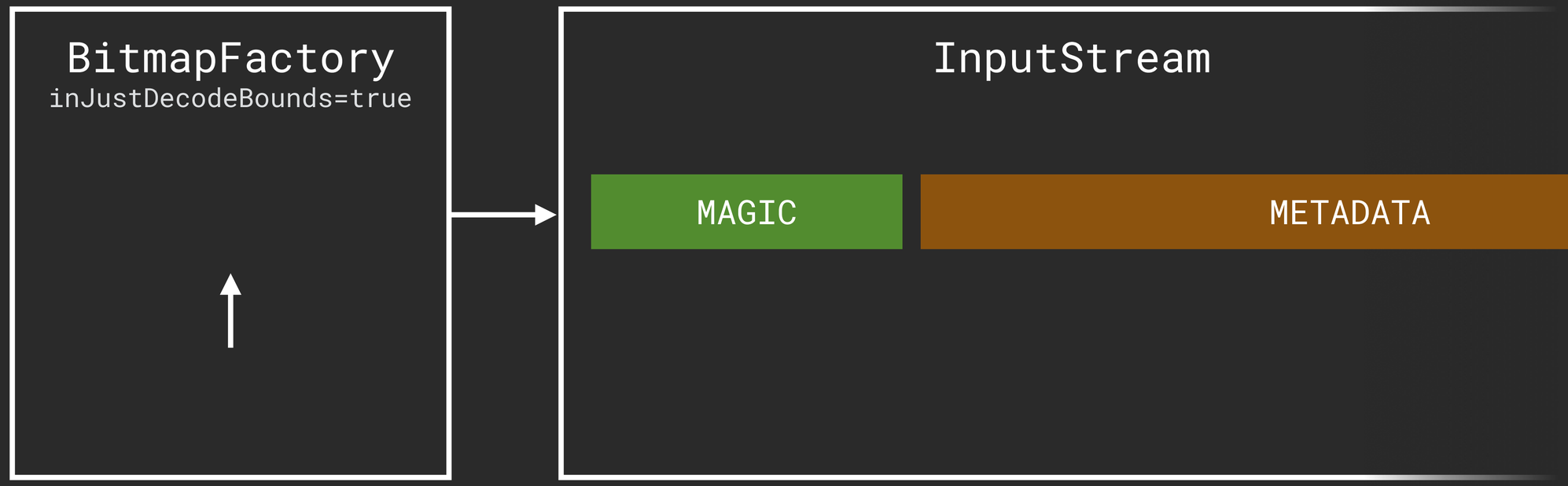
PIXEL DATA

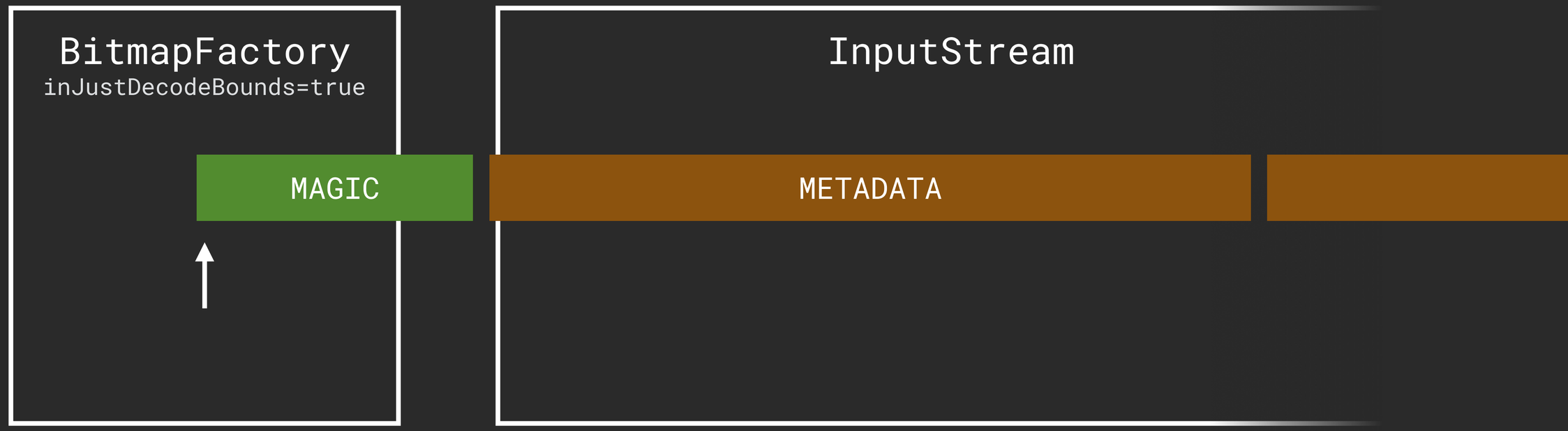
PIXEL DATA

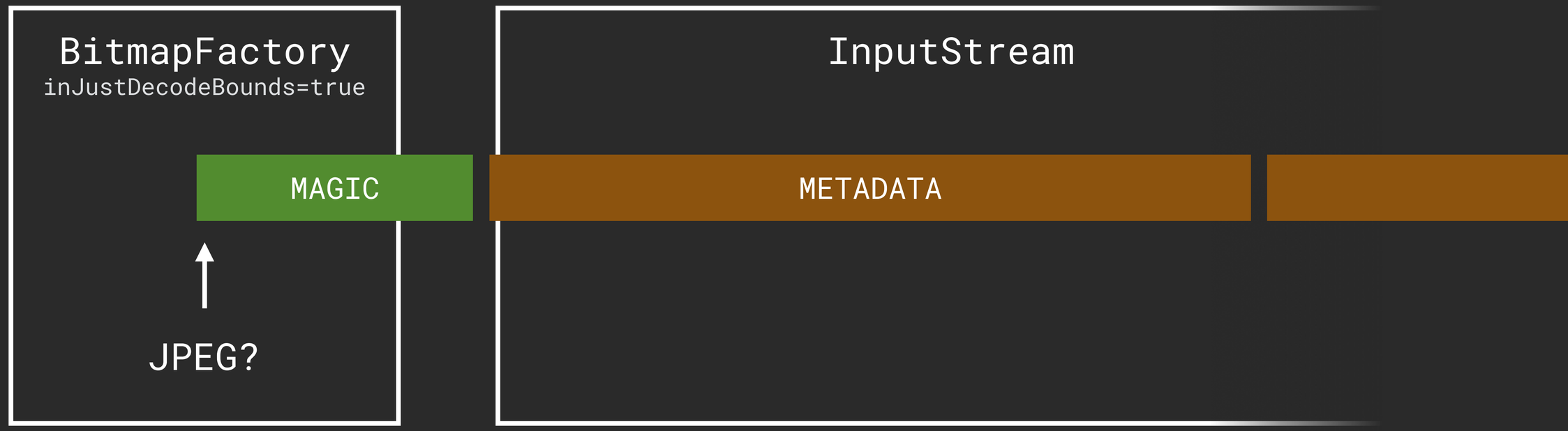
PIXEL DATA

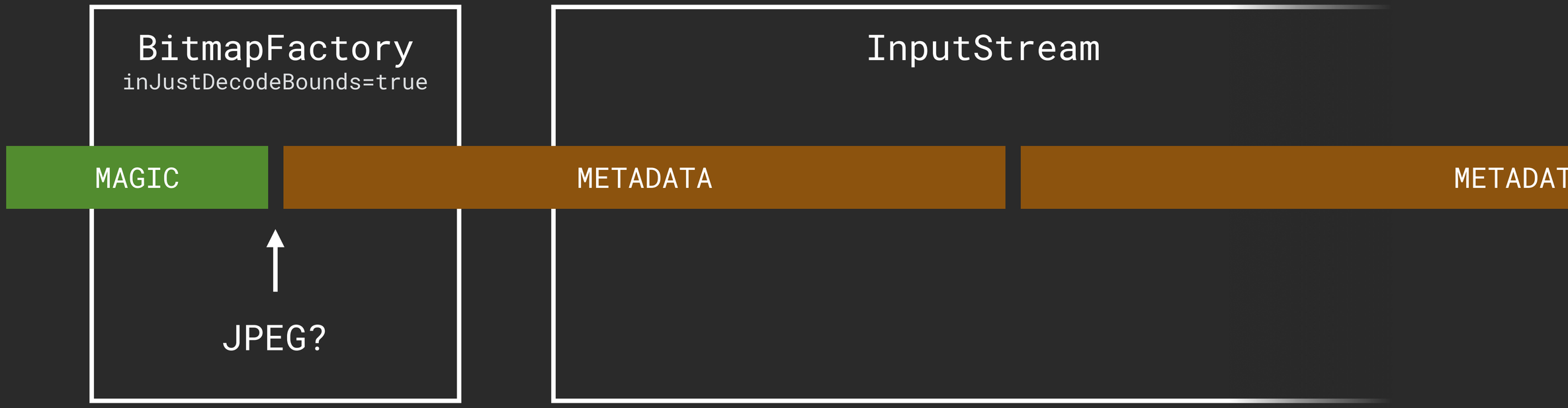
PIXEL DATA

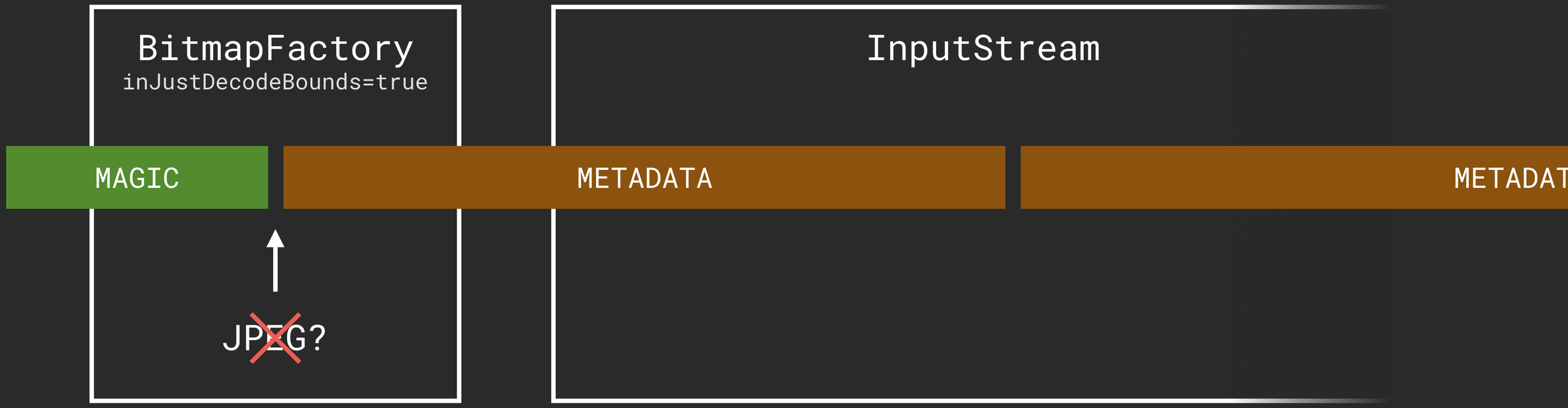
PIXEL DATA

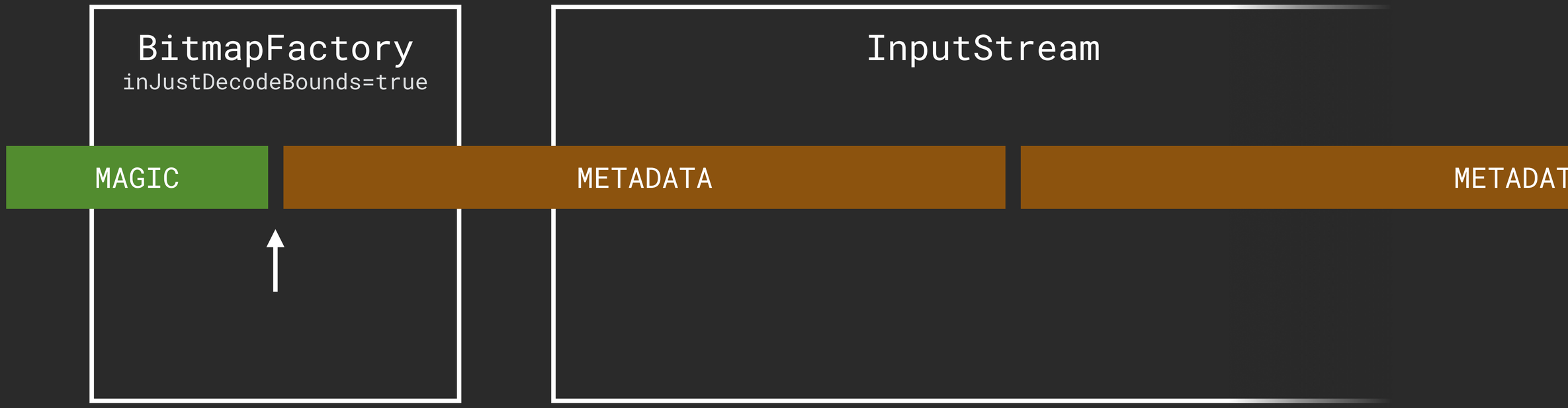

BitmapFactory
inJustDecodeBounds=true
InputStream
MAGIC
METADATA

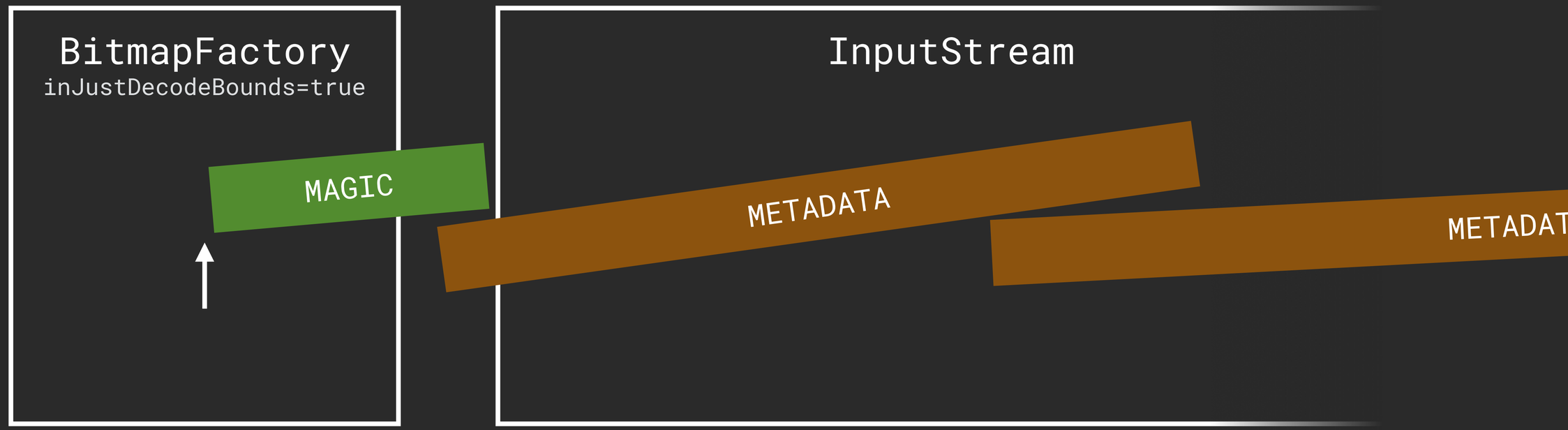

BitmapFactory
inJustDecodeBounds=true
InputStream
MAGIC
METADATA

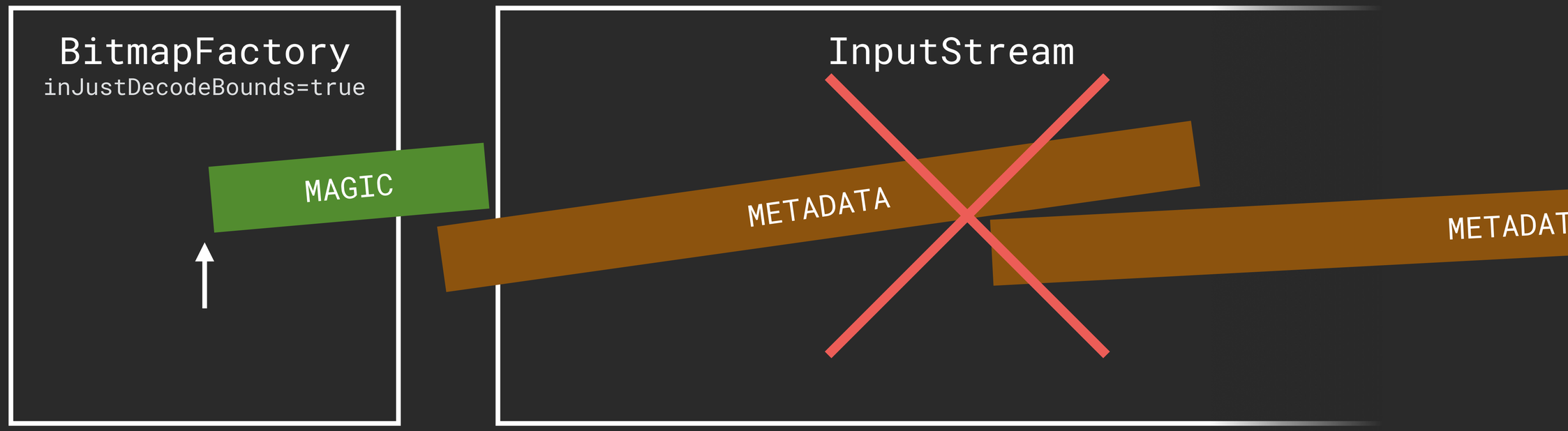

BitmapFactory
inJustDecodeBounds=true
InputStream
MAGIC
METADATA
JPEG?

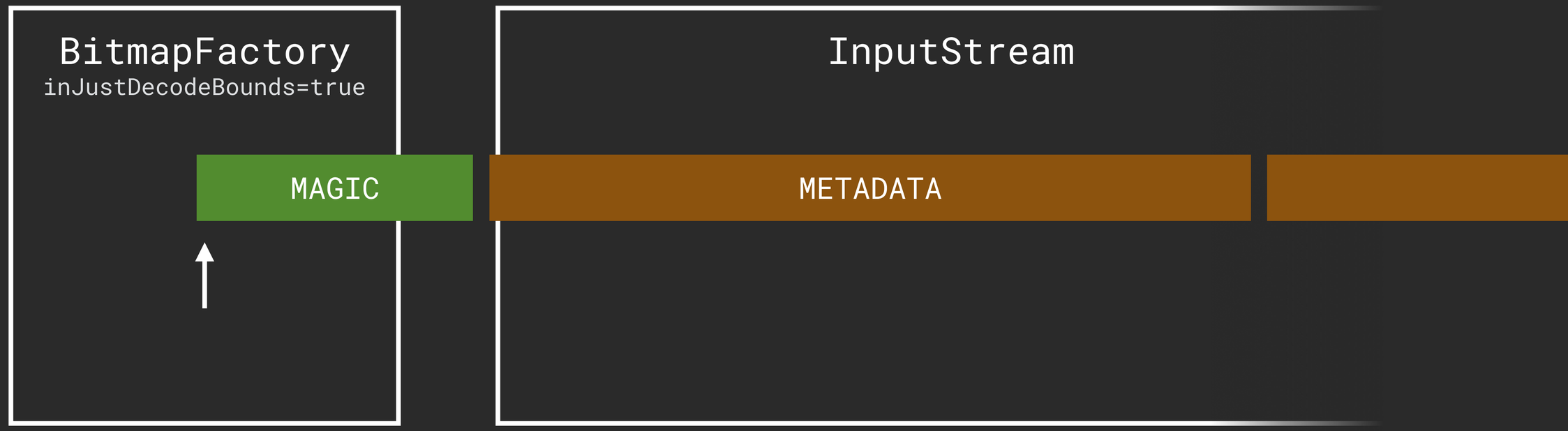

BitmapFactory
inJustDecodeBounds=true
InputStream
MAGIC
METADATA
METADAT
JPEG?

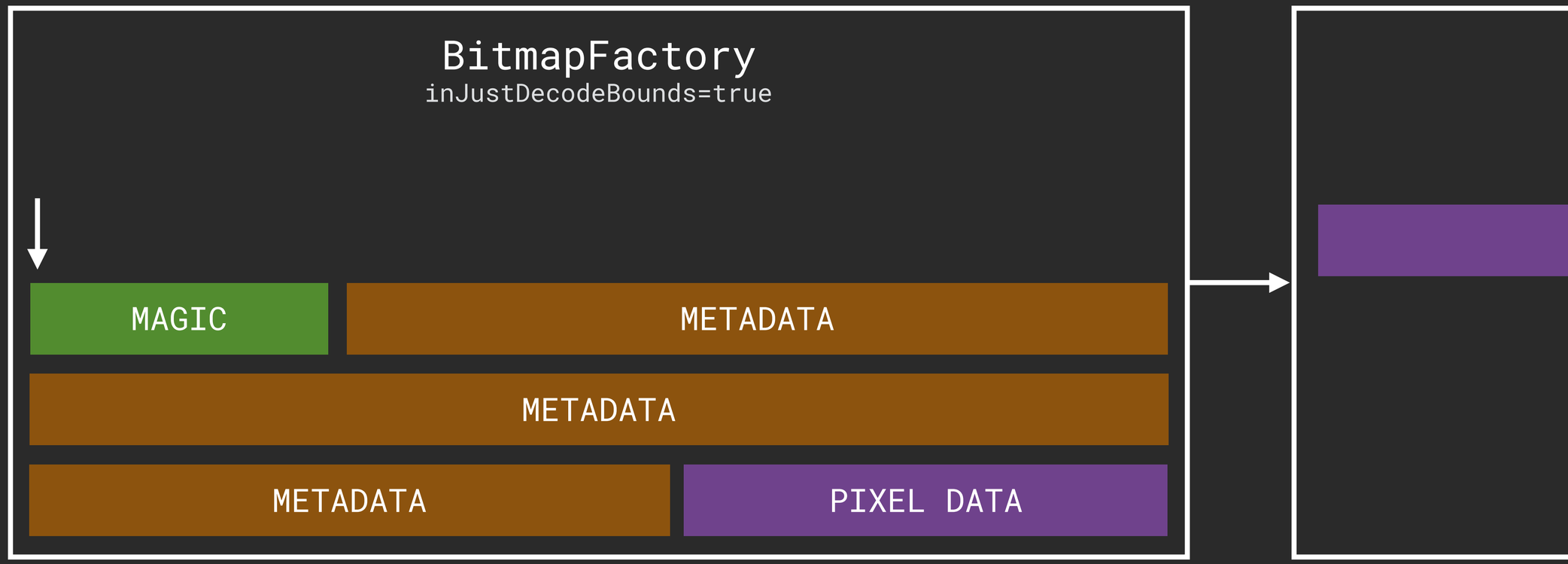

BitmapFactory
inJustDecodeBounds=true
InputStream
MAGIC
METADATA
METADAT
JPEG?

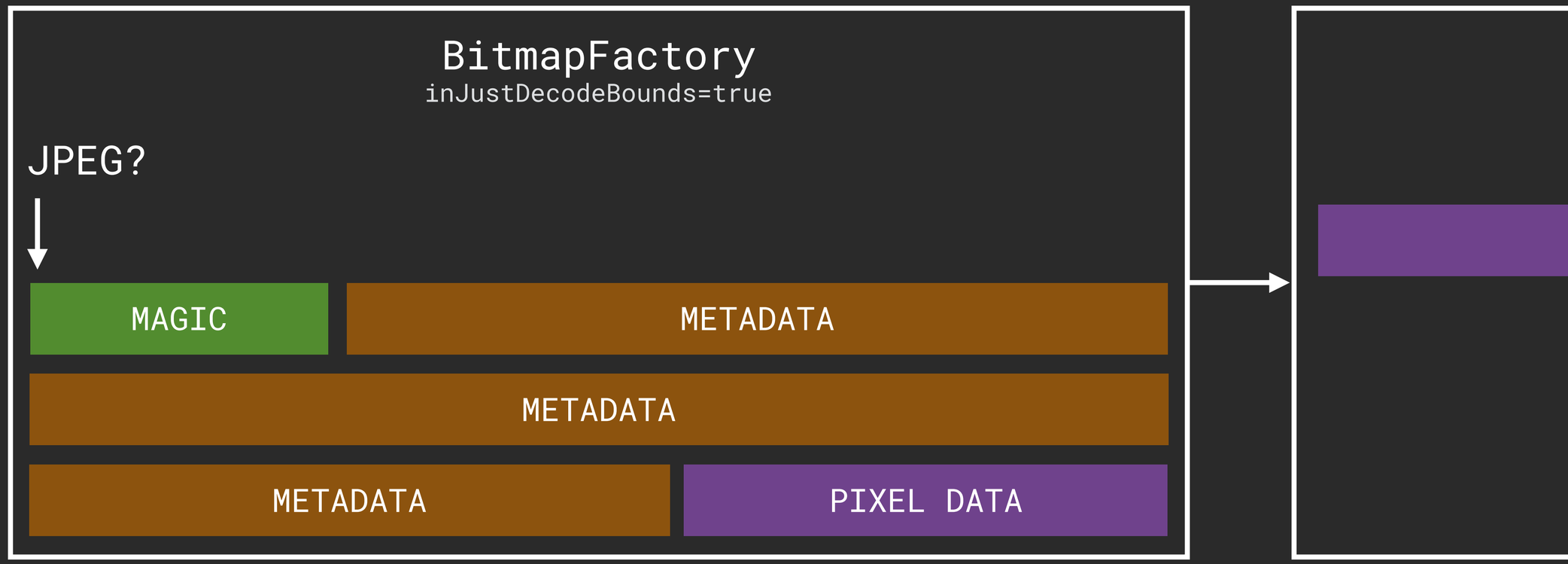

BitmapFactory
inJustDecodeBounds=true
InputStream
MAGIC
METADATA
METADAT

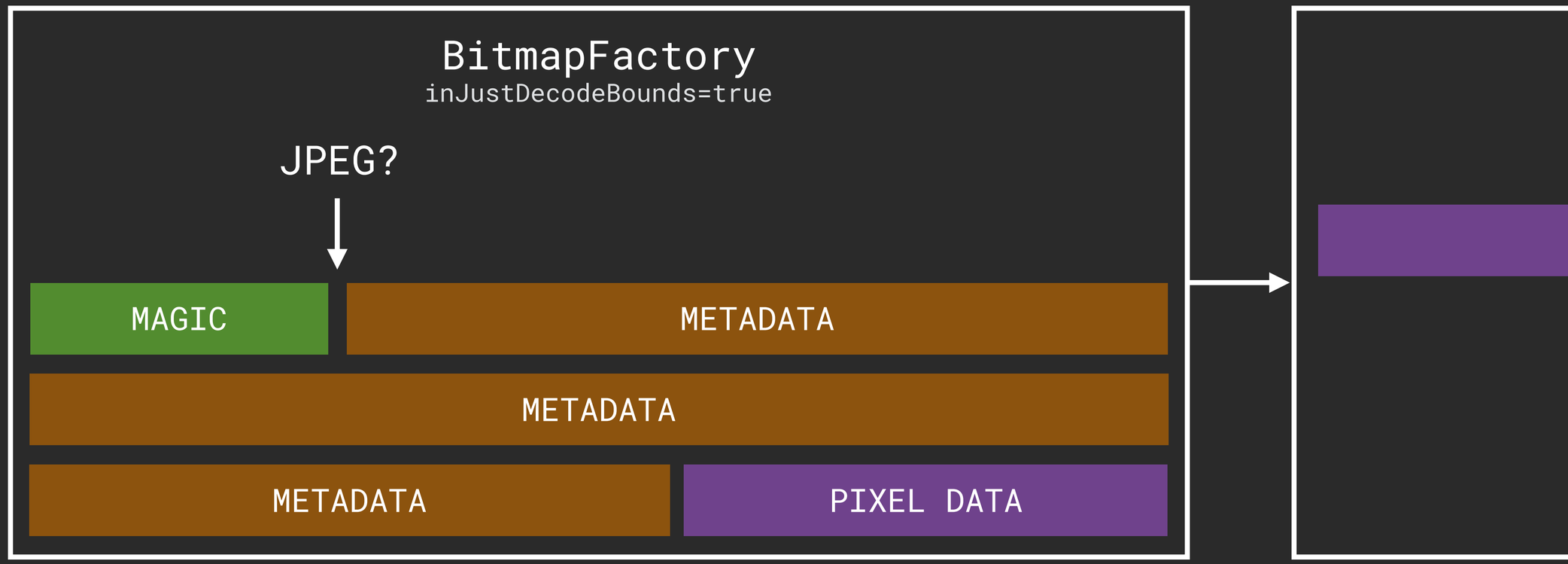

BitmapFactory
inJustDecodeBounds=true
MAGIC
InputStream
METADATA
METADAT

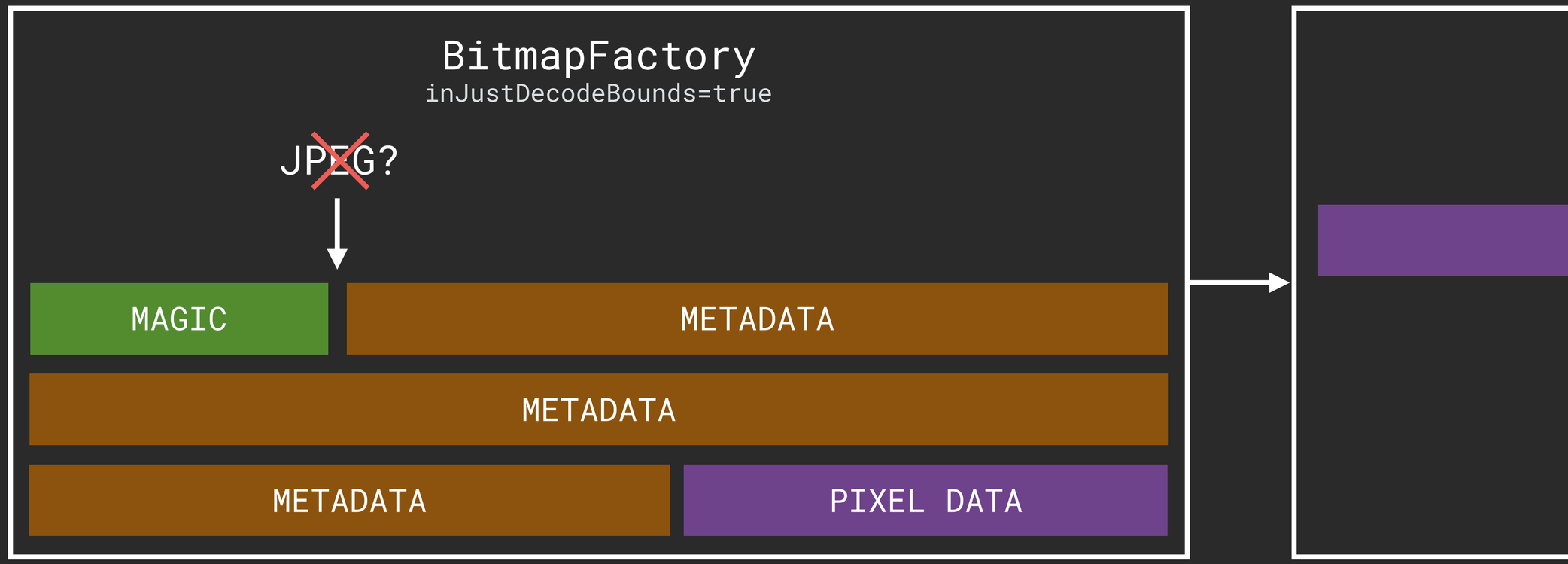

BitmapFactory
inJustDecodeBounds=true
MAGIC
InputStream
METADATA
METADAT

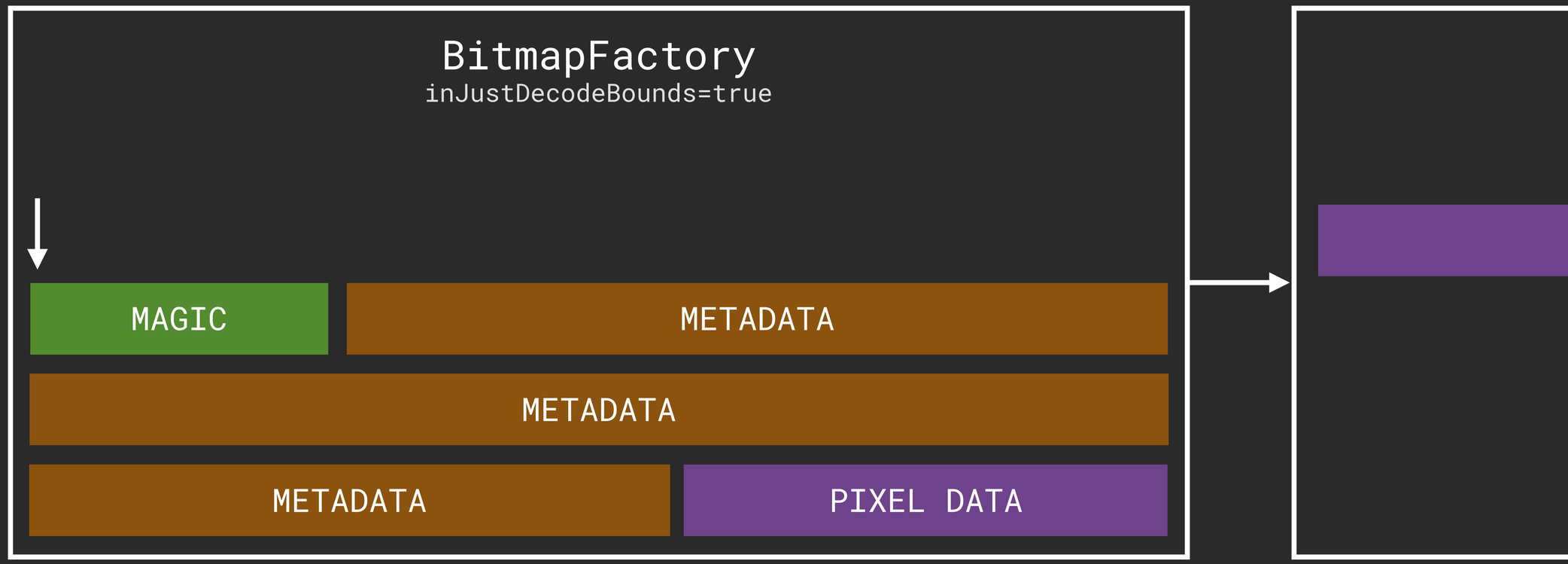

BitmapFactory
inJustDecodeBounds=true
InputStream
MAGIC
METADATA

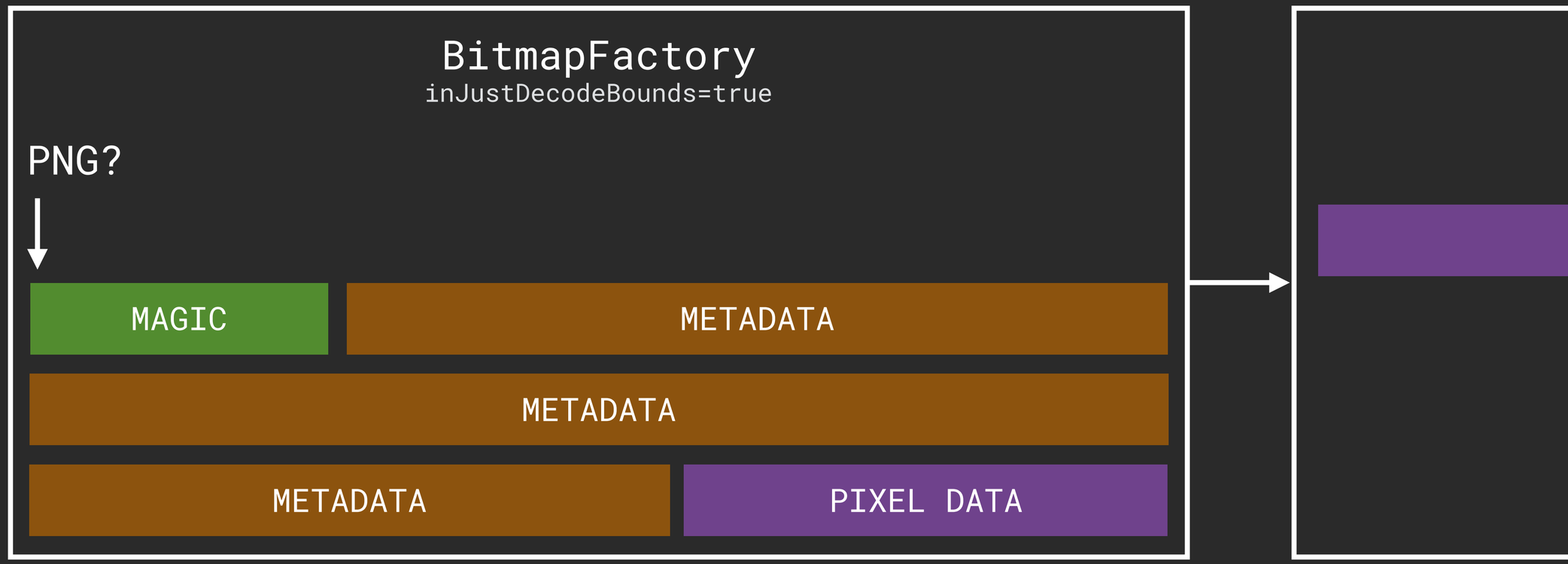

BitmapFactory
inJustDecodeBounds=true
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

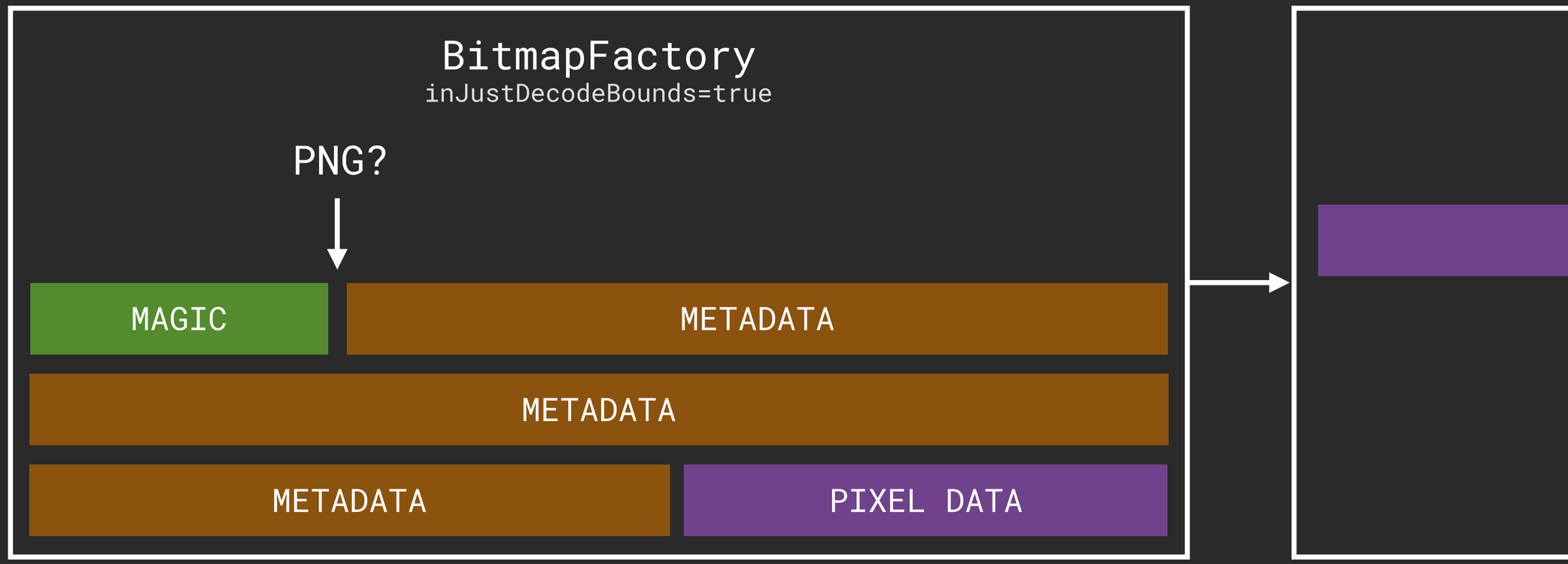

BitmapFactory
inJustDecodeBounds=true
JPEG?
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

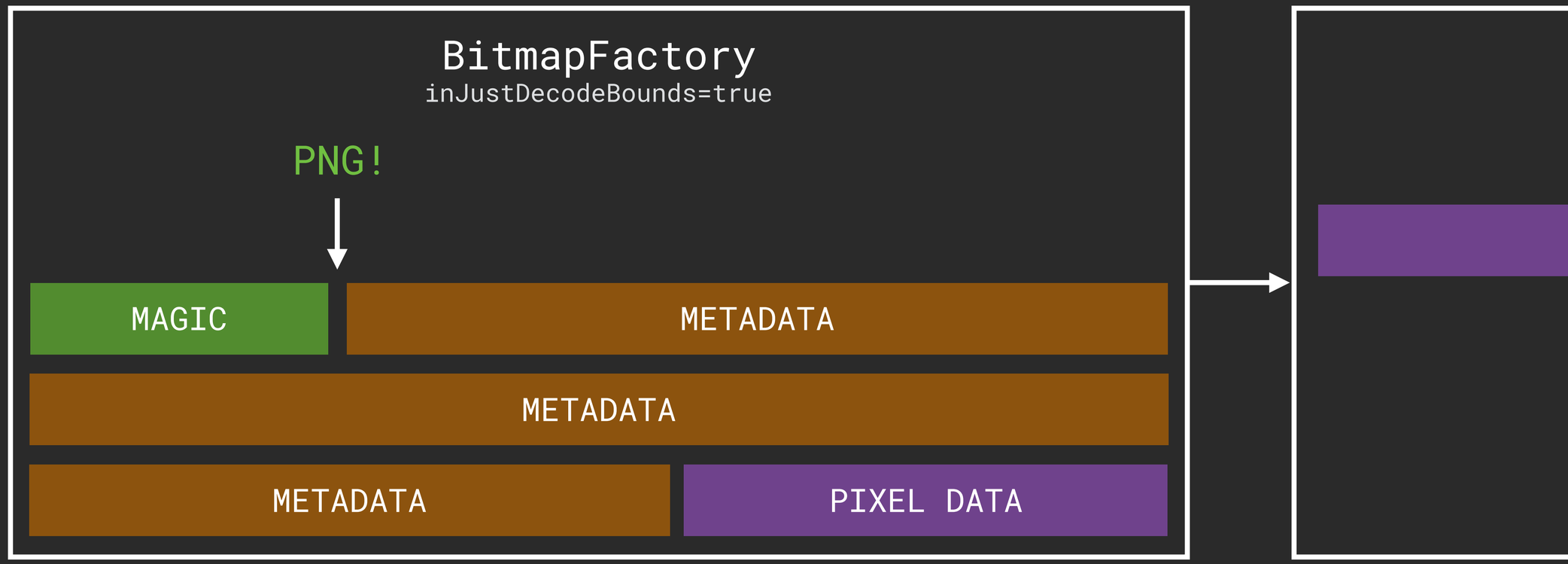

BitmapFactory
inJustDecodeBounds=true
JPEG?
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

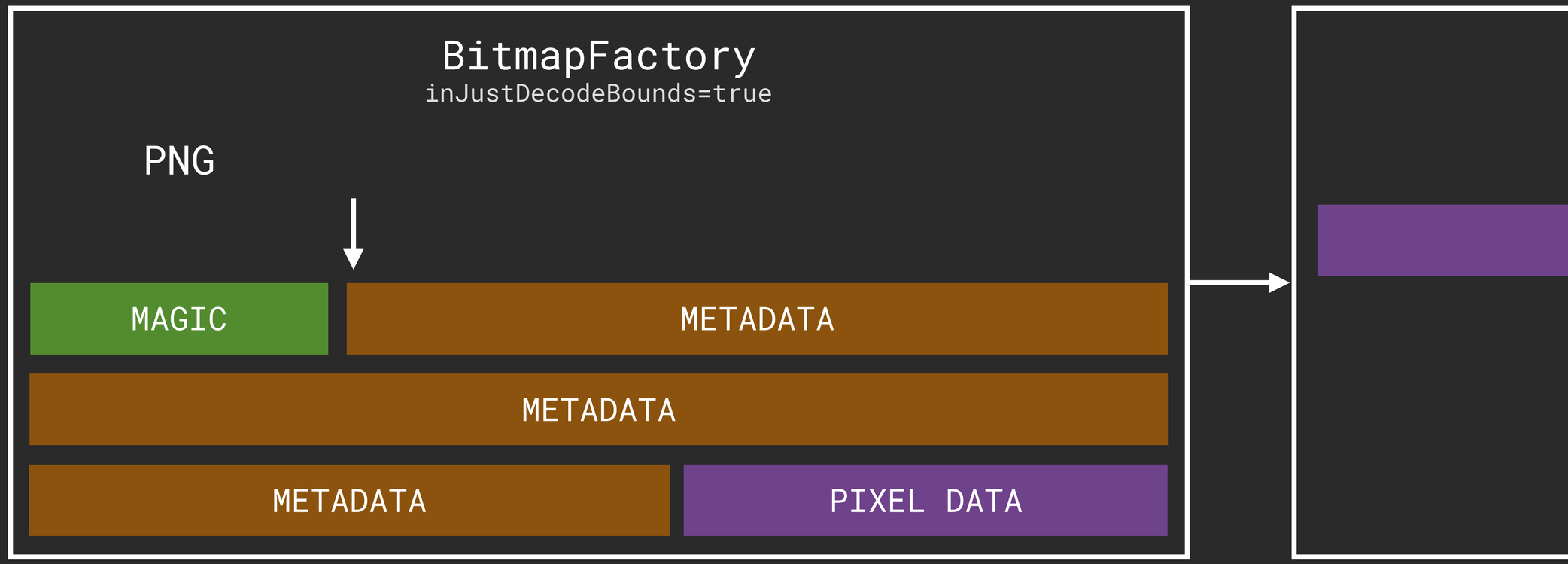

BitmapFactory
inJustDecodeBounds=true
JPEG?
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

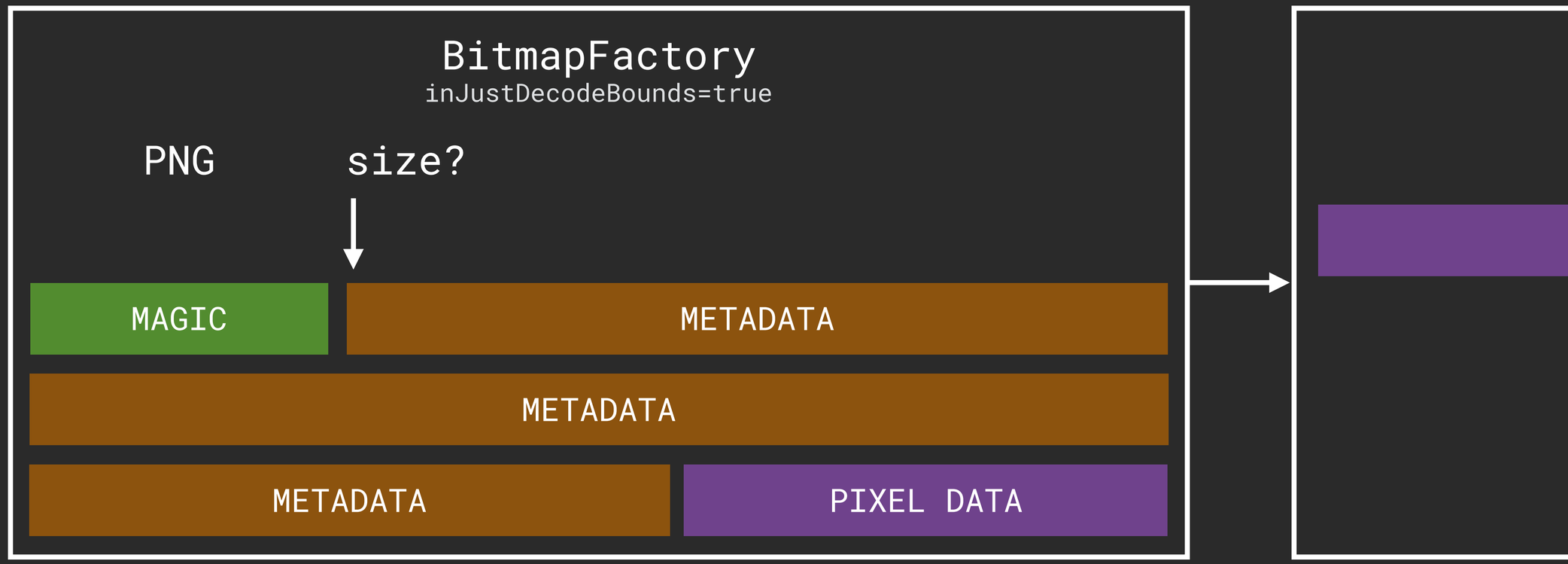

BitmapFactory
inJustDecodeBounds=true
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

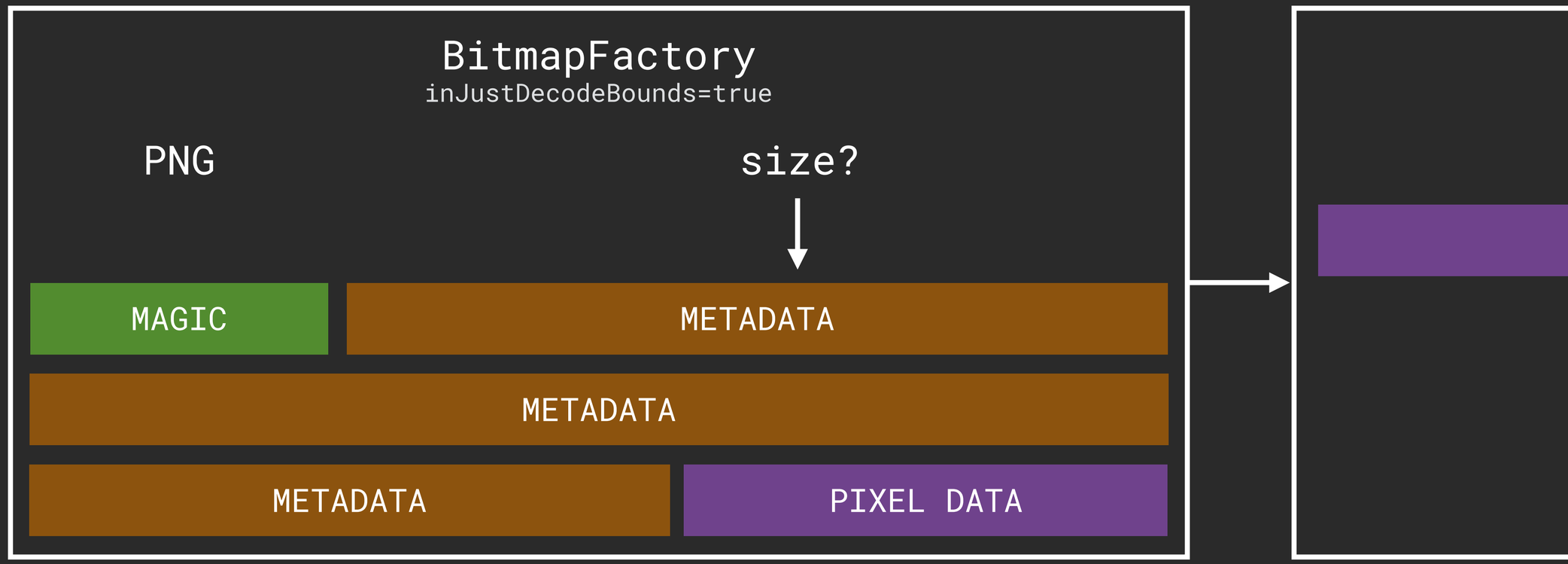

BitmapFactory
inJustDecodeBounds=true
PNG?
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

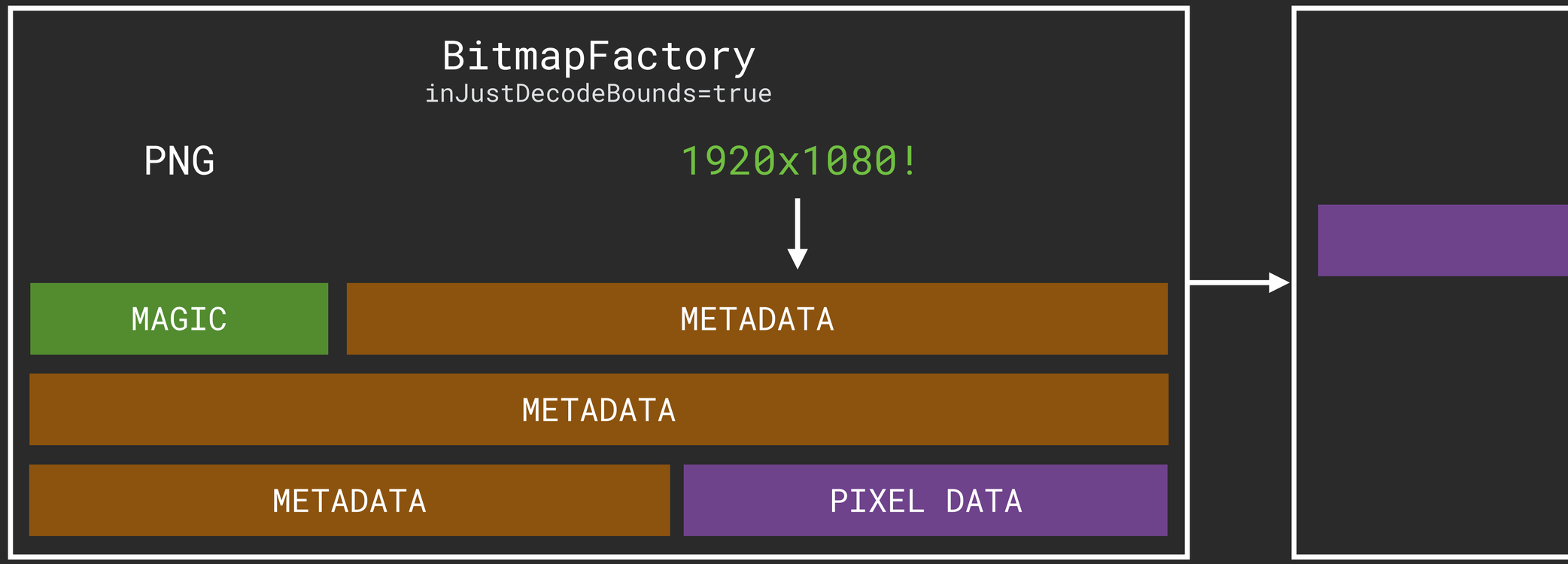

BitmapFactory
inJustDecodeBounds=true
PNG?
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

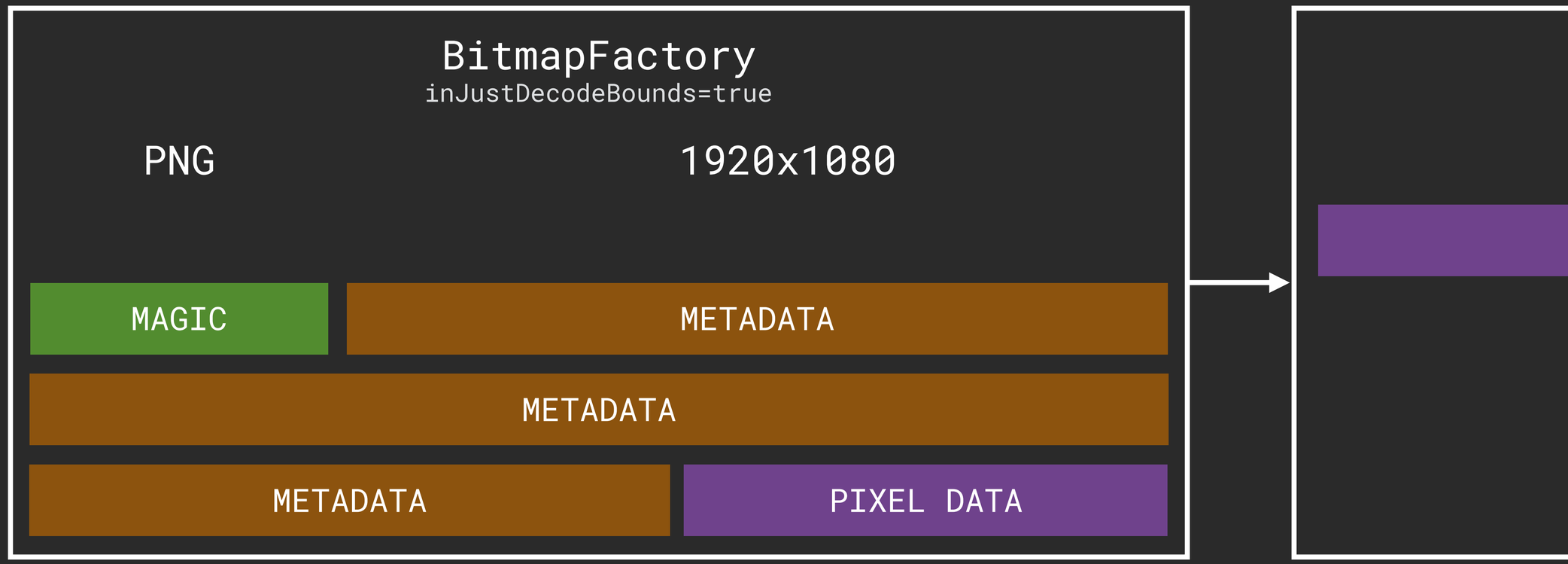

BitmapFactory
inJustDecodeBounds=true
PNG!
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA


BitmapFactory
inJustDecodeBounds=true
PNG
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

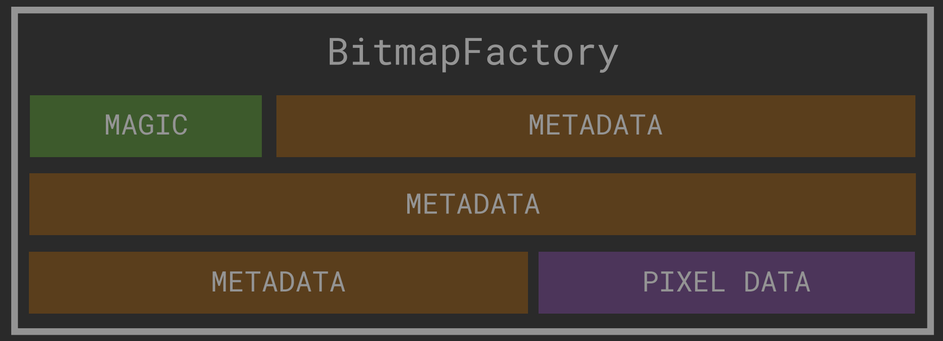

BitmapFactory
inJustDecodeBounds=true
PNG
size?
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA


BitmapFactory
inJustDecodeBounds=true
PNG
size?
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

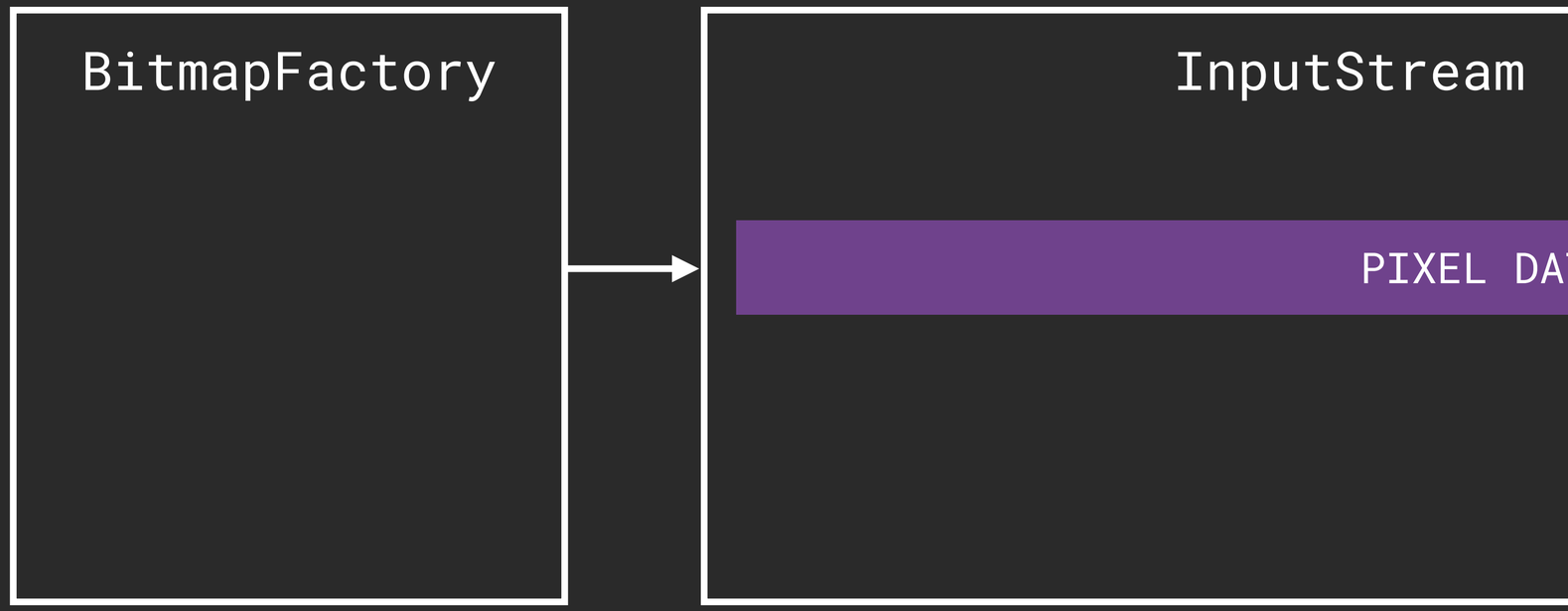

BitmapFactory
inJustDecodeBounds=true
PNG
1920x1080!
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

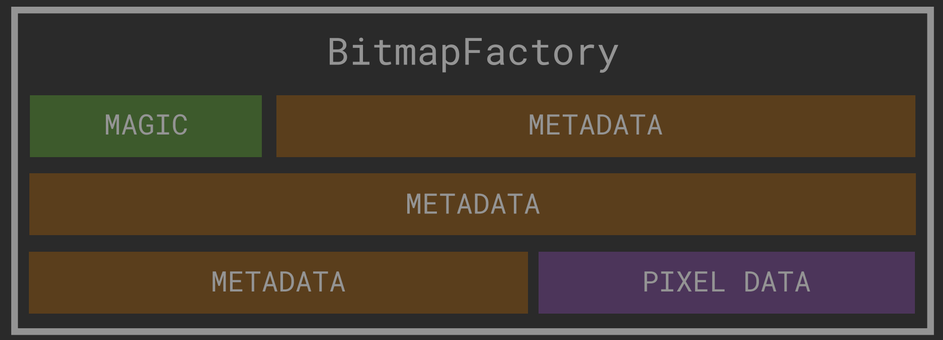

BitmapFactory
inJustDecodeBounds=true
PNG
1920x1080
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

1920x1080

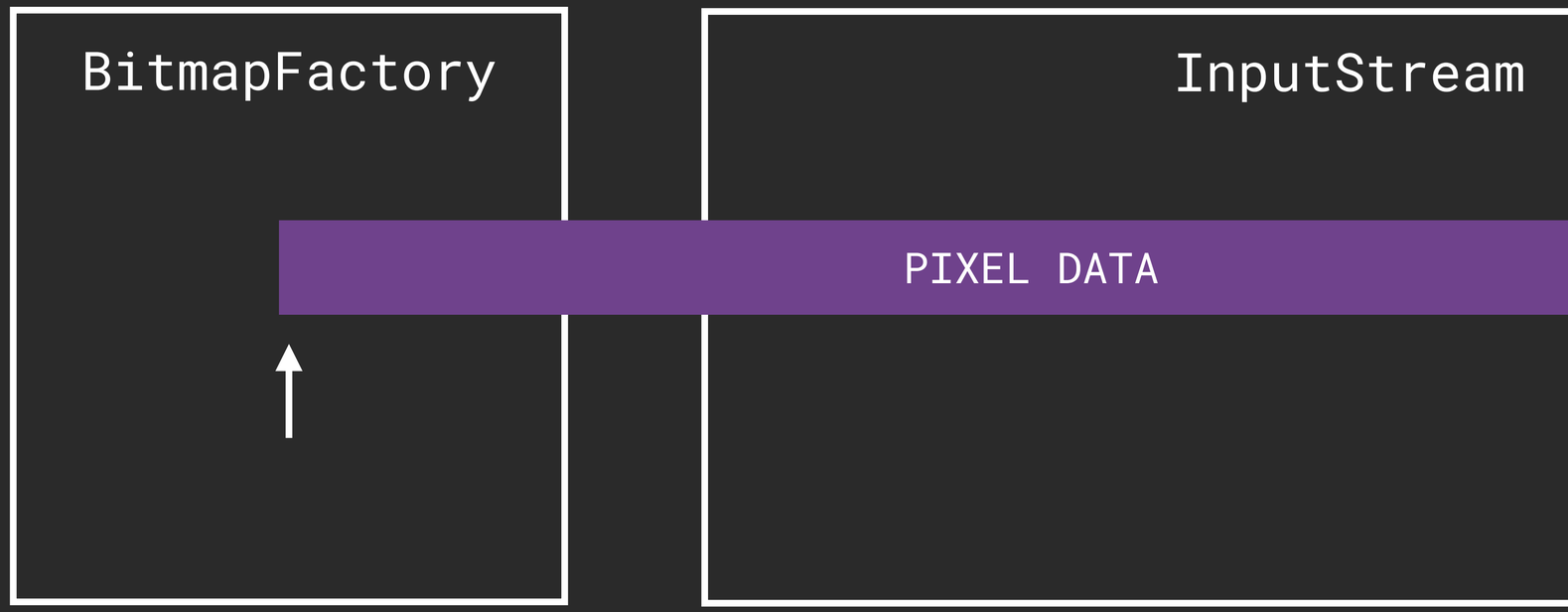

BitmapFactory
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

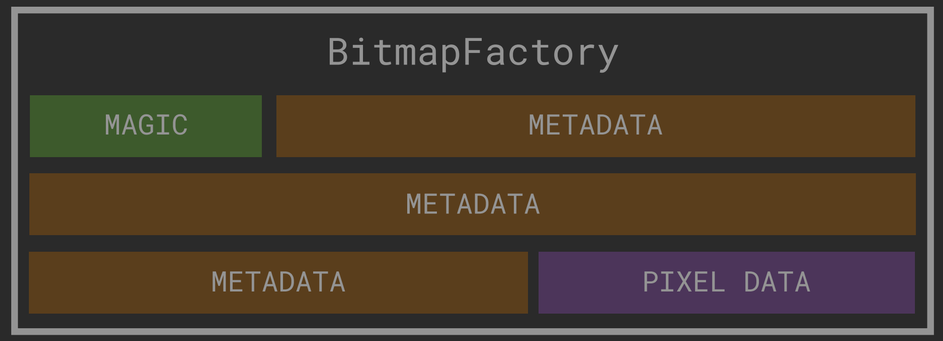
1920x1080


BitmapFactory
InputStream
PIXEL DA

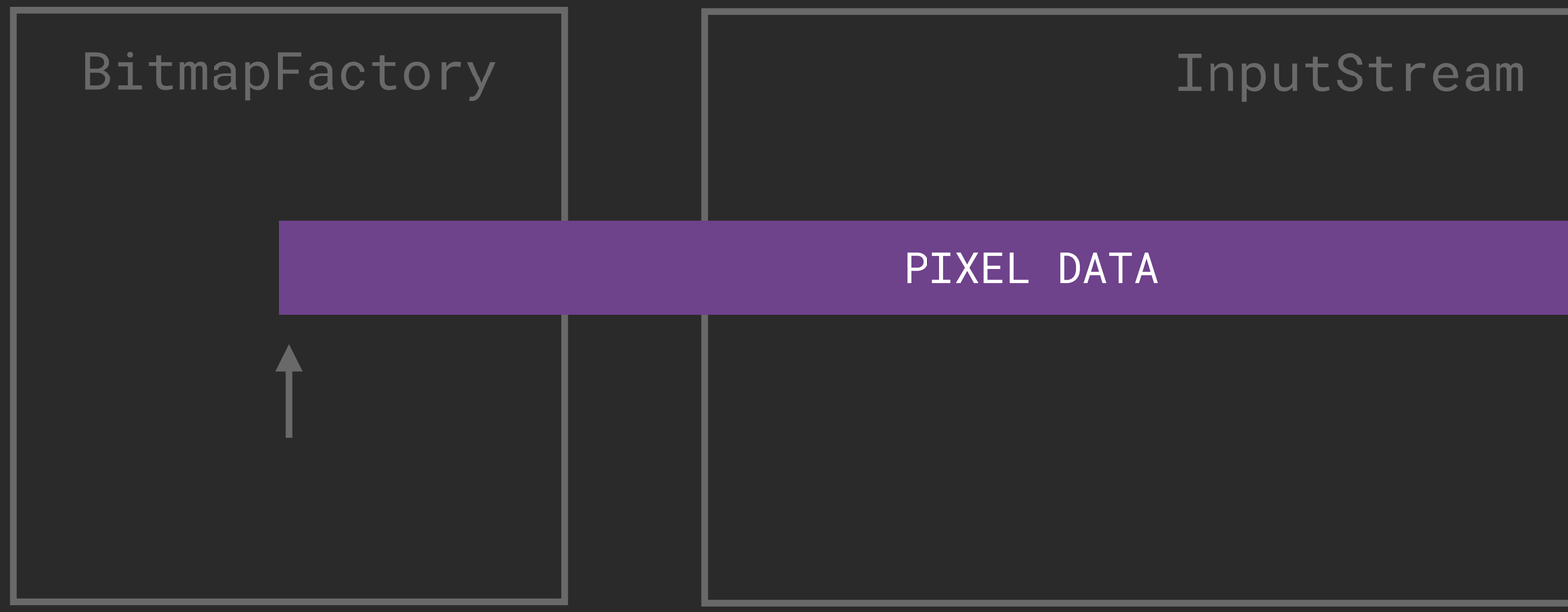

BitmapFactory
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

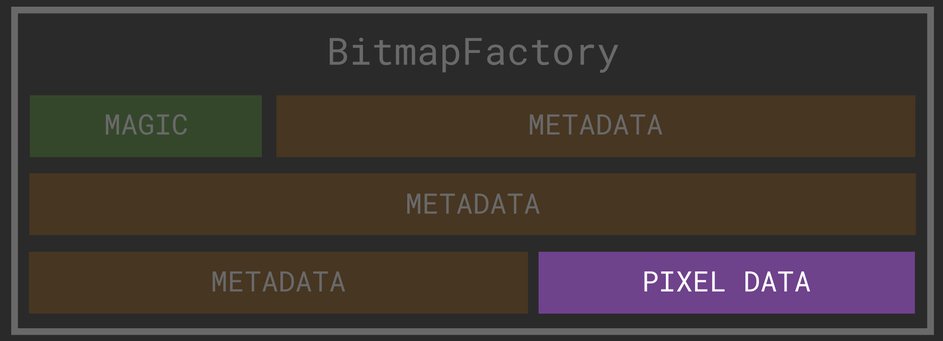
1920x1080


BitmapFactory
InputStream
PIXEL DATA

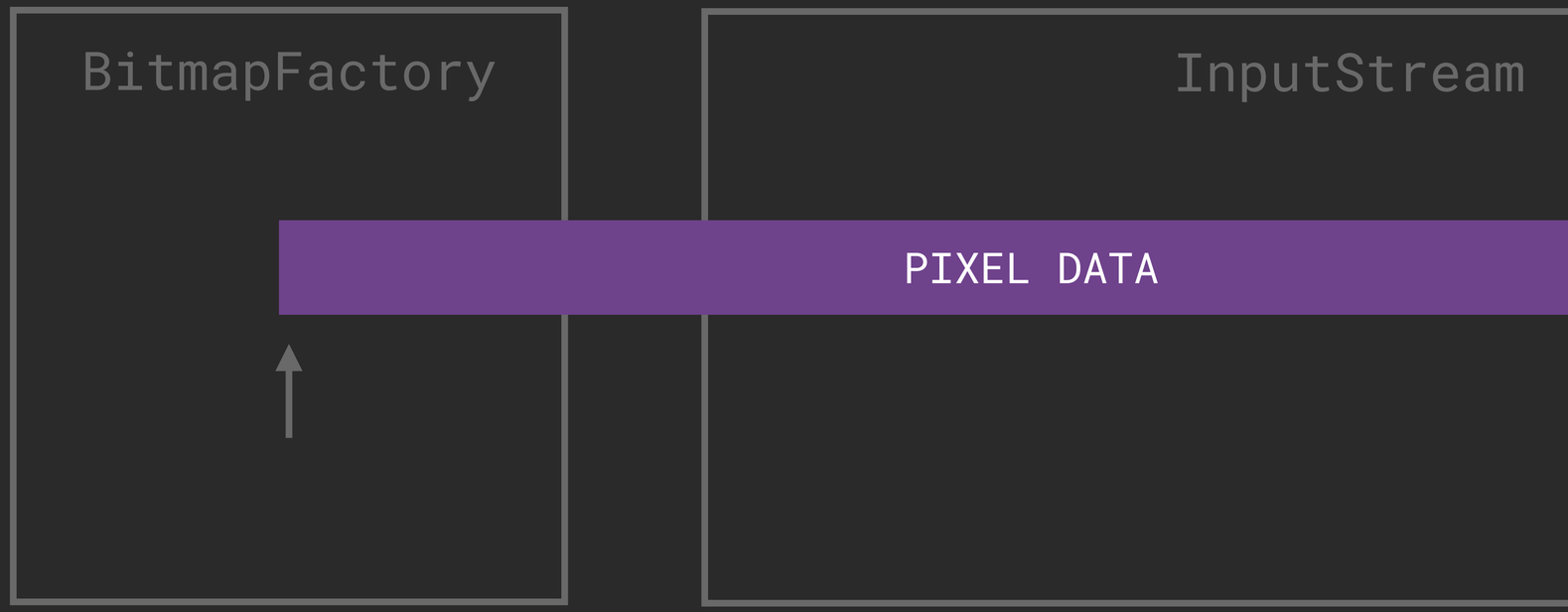

BitmapFactory
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

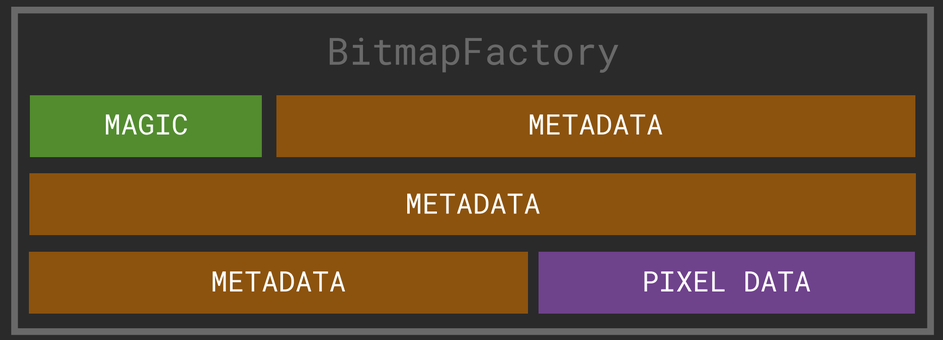
1920x1080

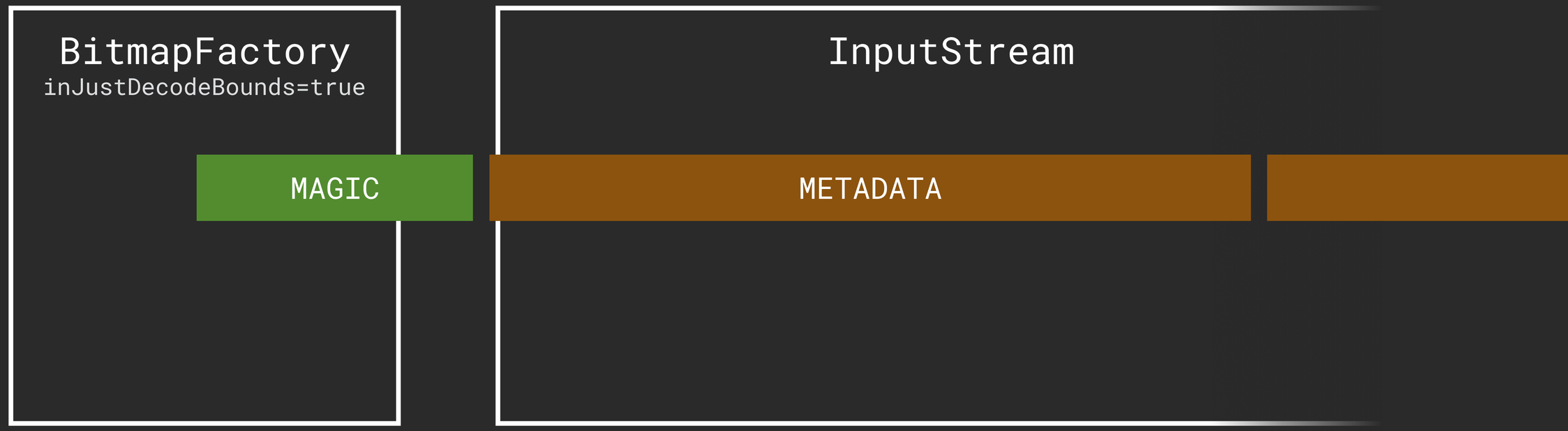

BitmapFactory
InputStream
PIXEL DATA

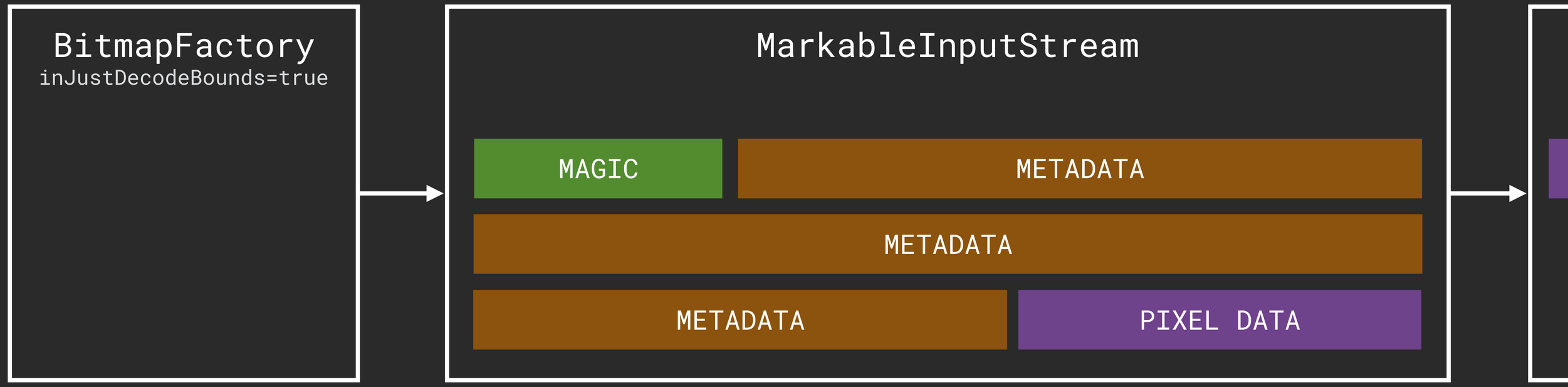

BitmapFactory
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

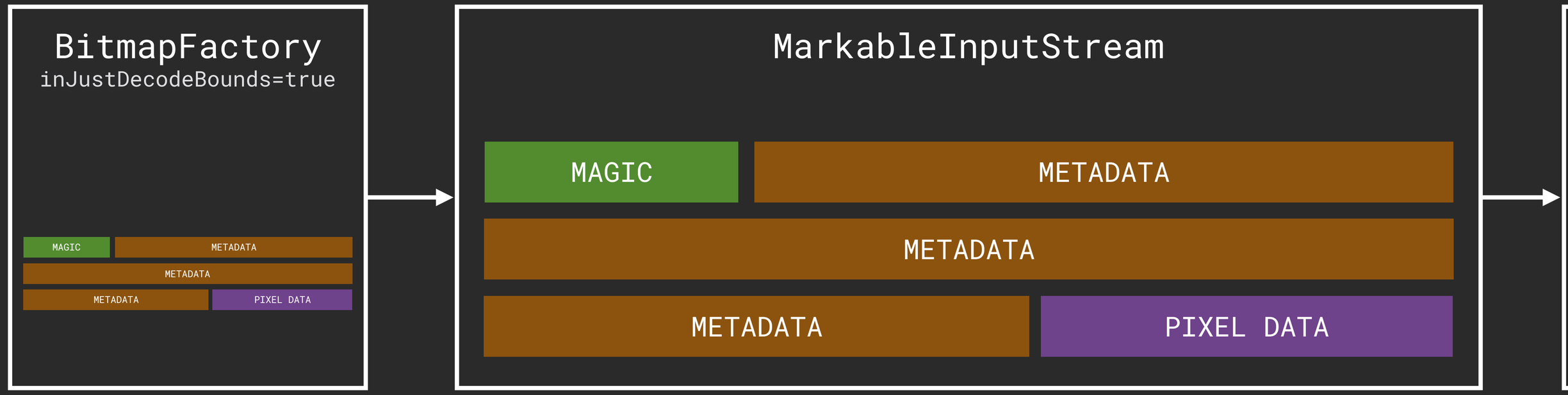
1920x1080

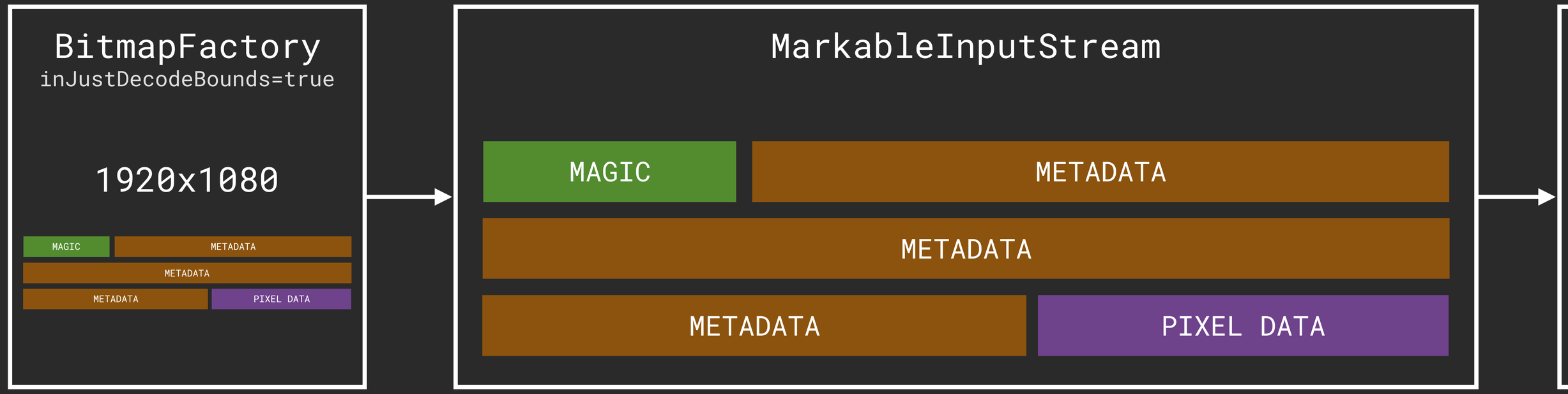

BitmapFactory
InputStream
PIXEL DATA


BitmapFactory
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

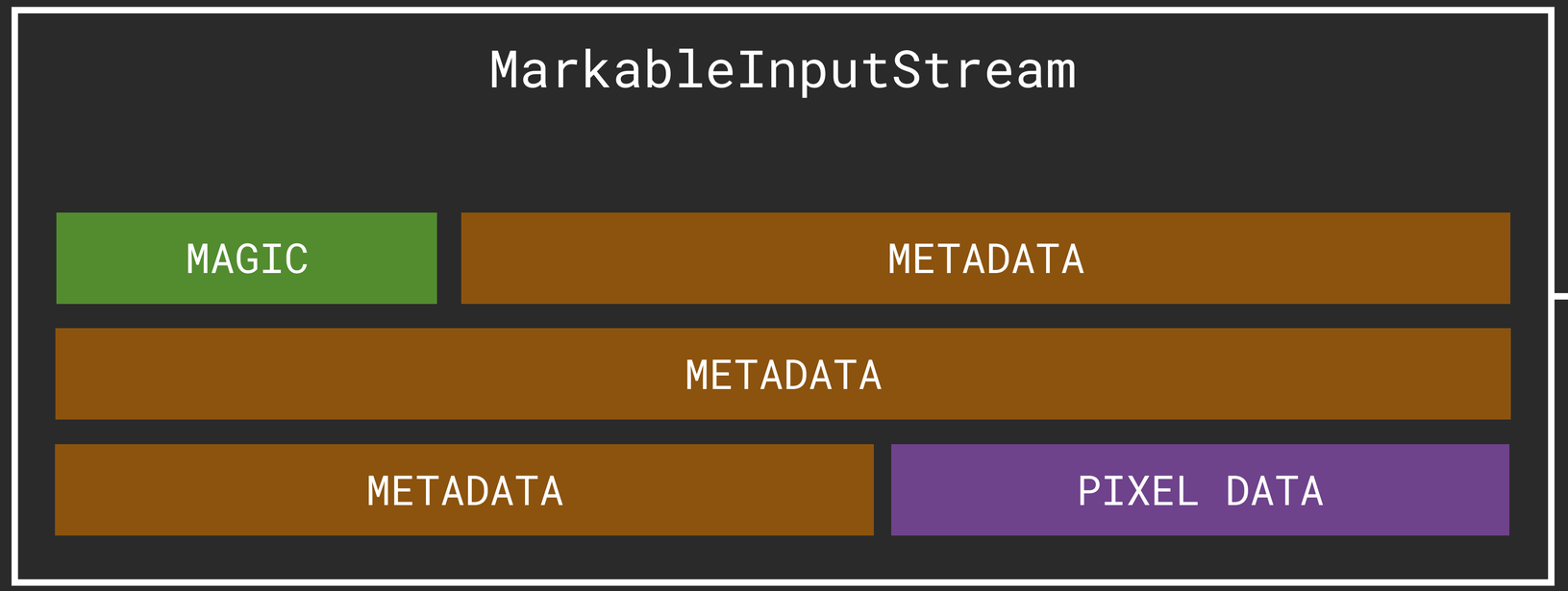

BitmapFactory
inJustDecodeBounds=true
InputStream
MAGIC
METADATA

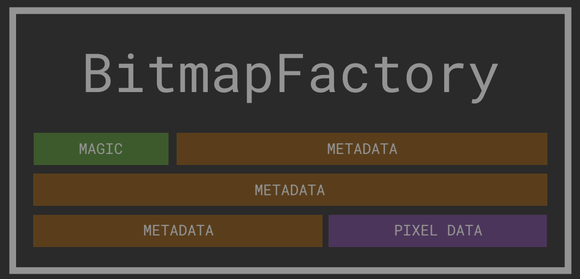

BitmapFactory
inJustDecodeBounds=true
MarkableInputStream
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA


BitmapFactory
inJustDecodeBounds=true
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
MarkableInputStream
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA


BitmapFactory
inJustDecodeBounds=true
1920x1080
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
MarkableInputStream
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

1920x1080


MarkableInputStream
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA


BitmapFactory
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

1920x1080

BitmapFactory

MarkableInputStream

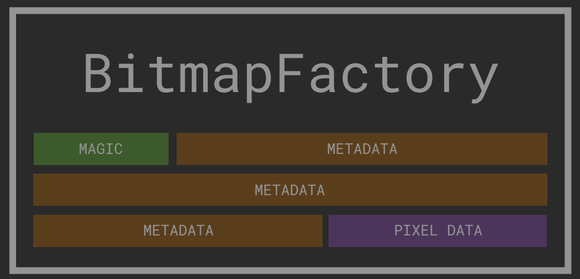
MAGIC

METADATA

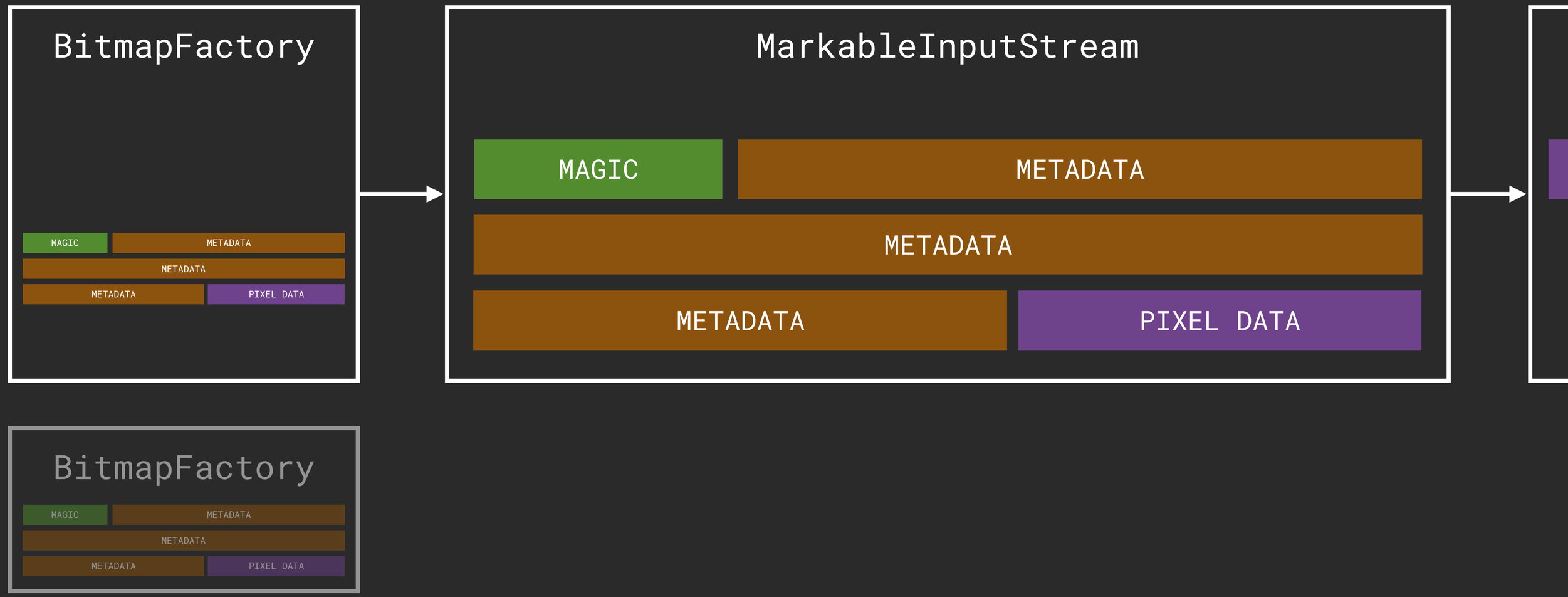
METADATA

METADATA

PIXEL DATA



1920x1080

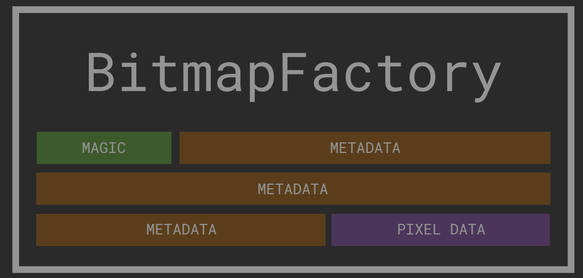

BitmapFactory
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
MarkableInputStream
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

BitmapFactory
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

1920x1080

BitmapFactory

MarkableInputStream



1920x1080

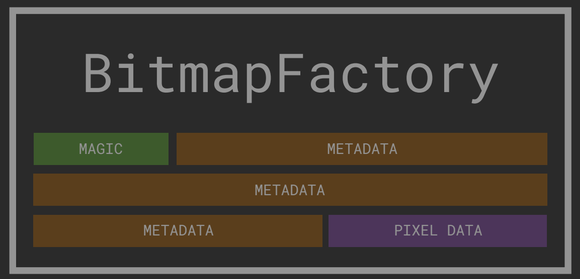
BitmapFactory

MarkableInputStream

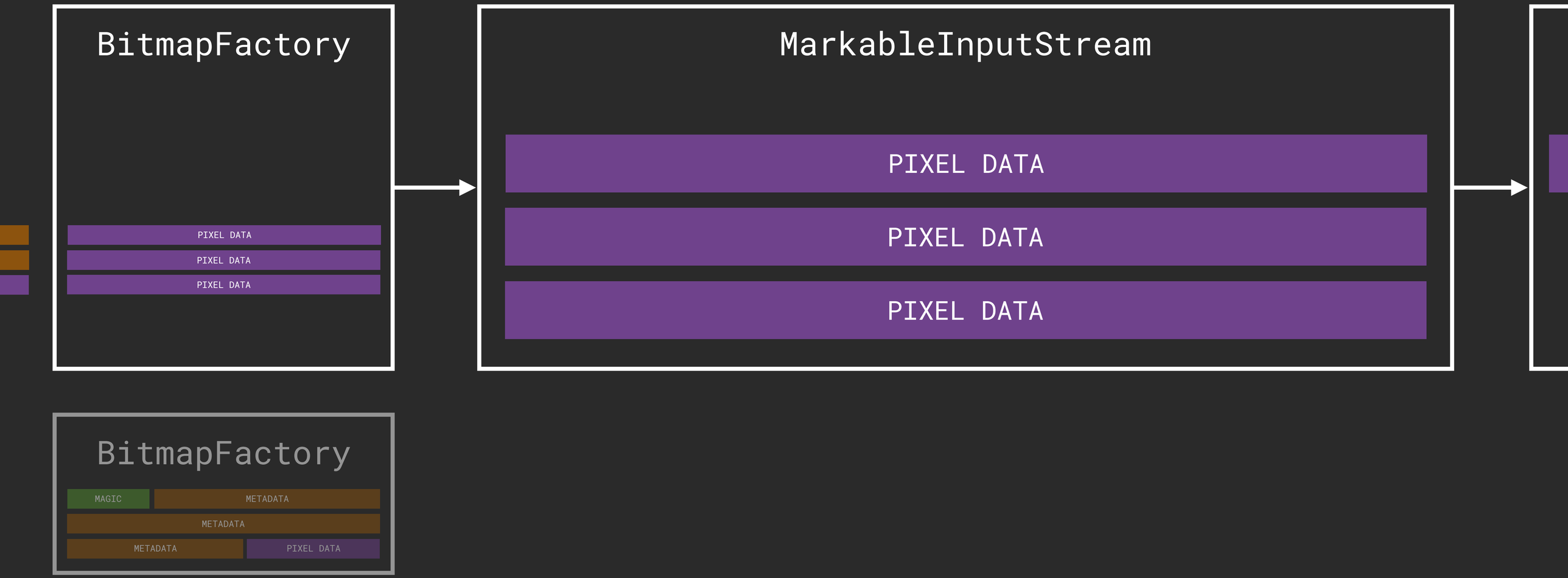
PIXEL DATA

PIXEL DATA

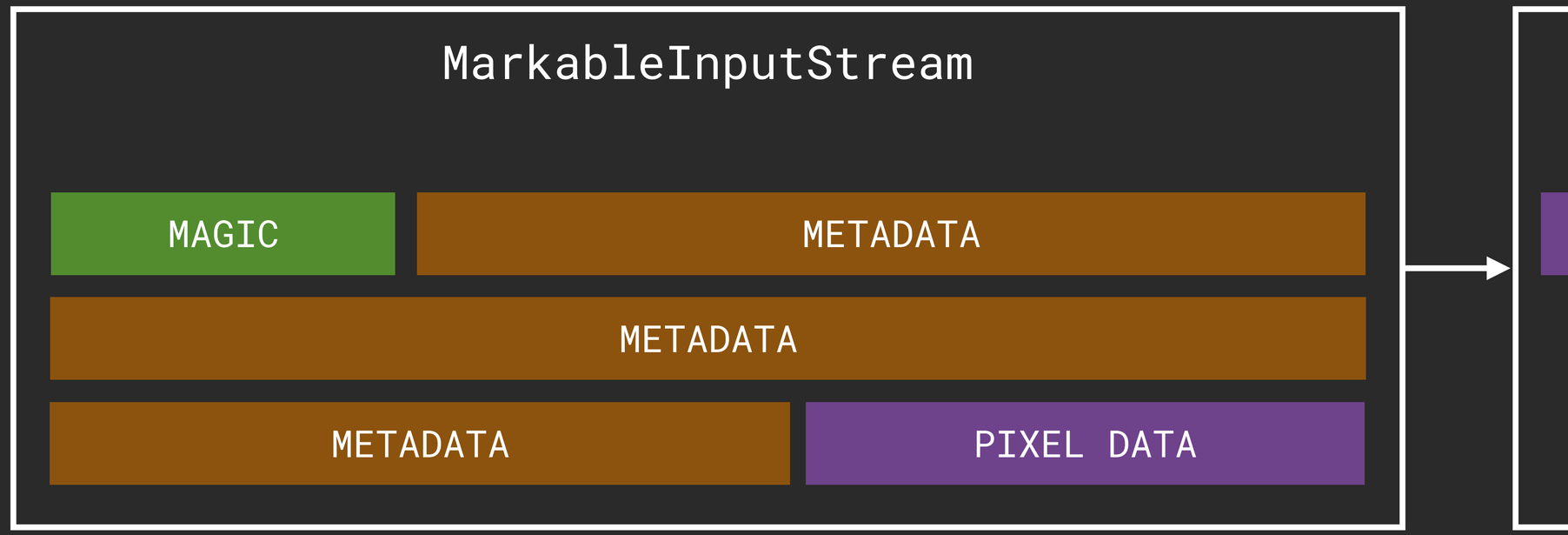
PIXEL DATA

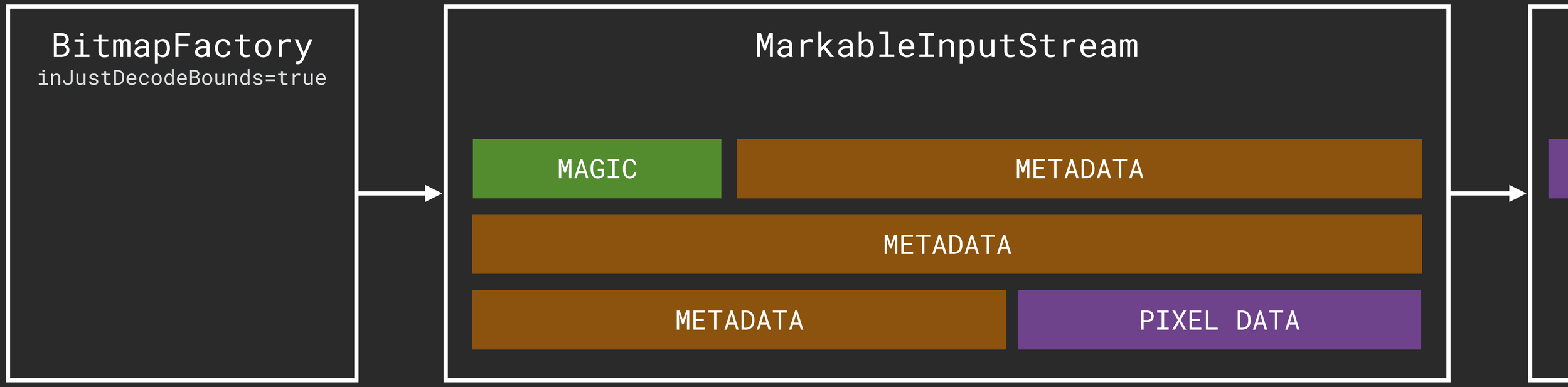


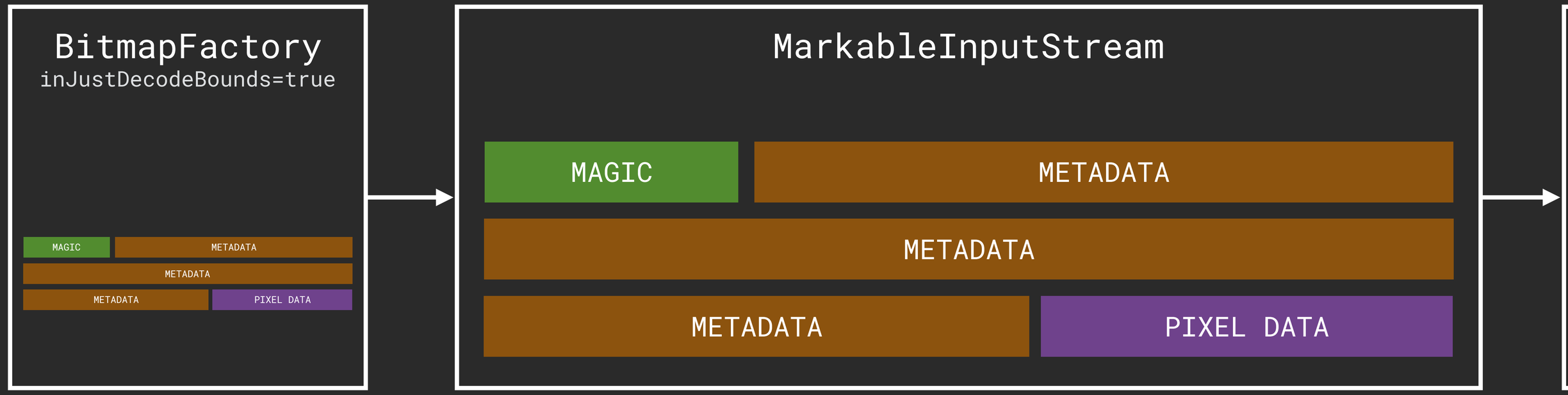
1920x1080

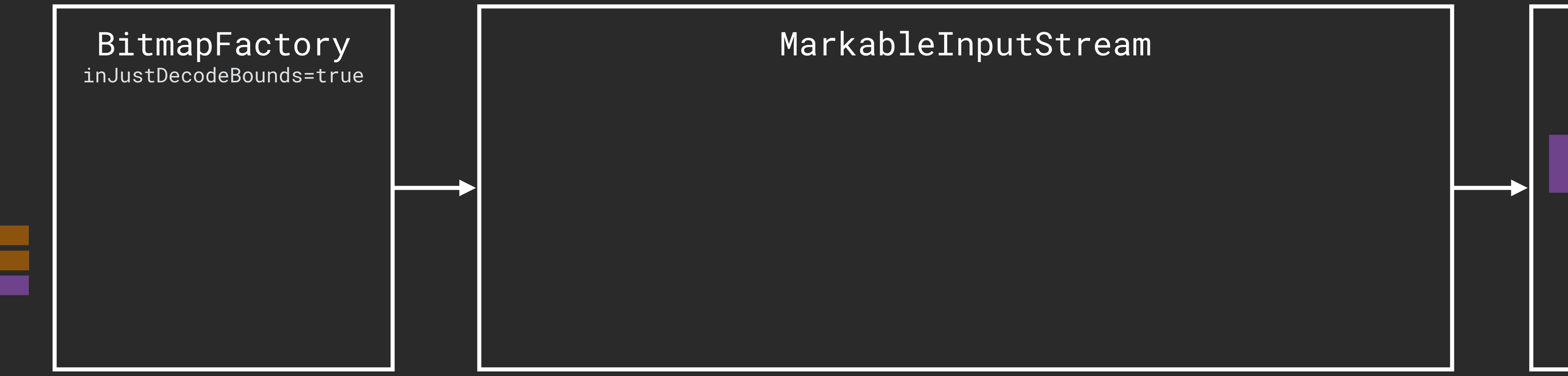

BitmapFactory
PIXEL DATA
PIXEL DATA
PIXEL DATA
MarkableInputStream
PIXEL DATA
PIXEL DATA
PIXEL DATA

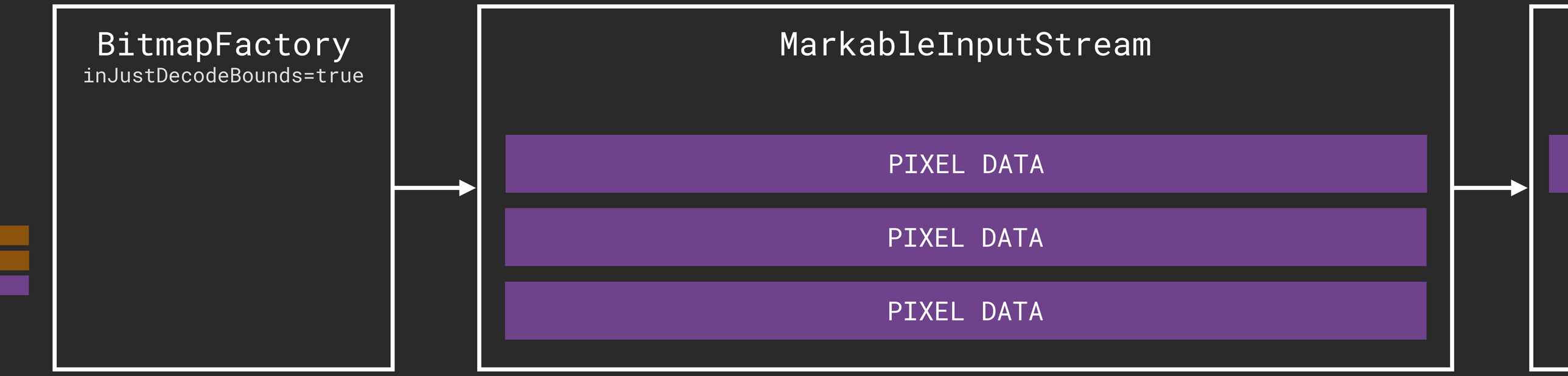
BitmapFactory
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

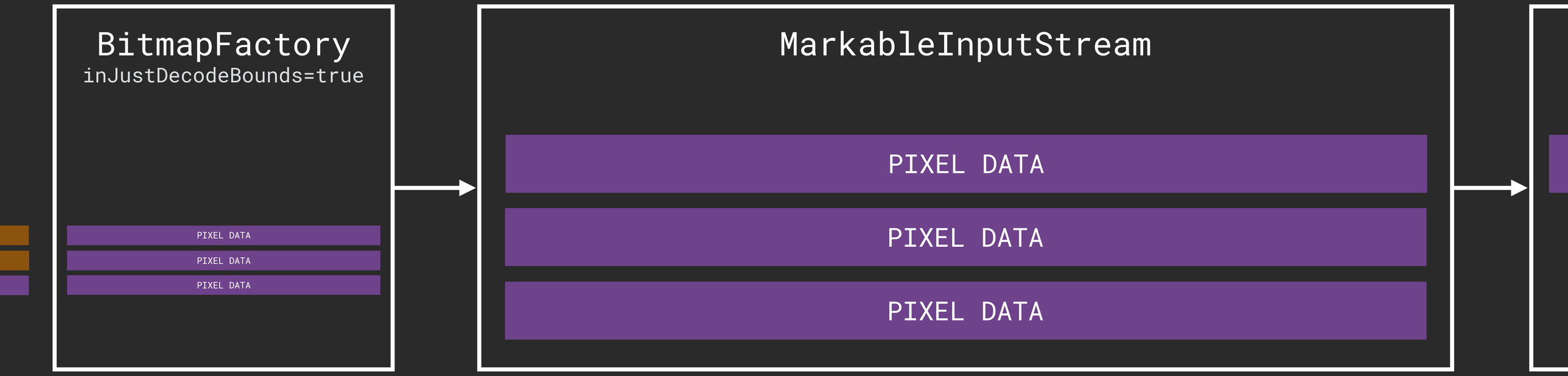

MarkableInputStream
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

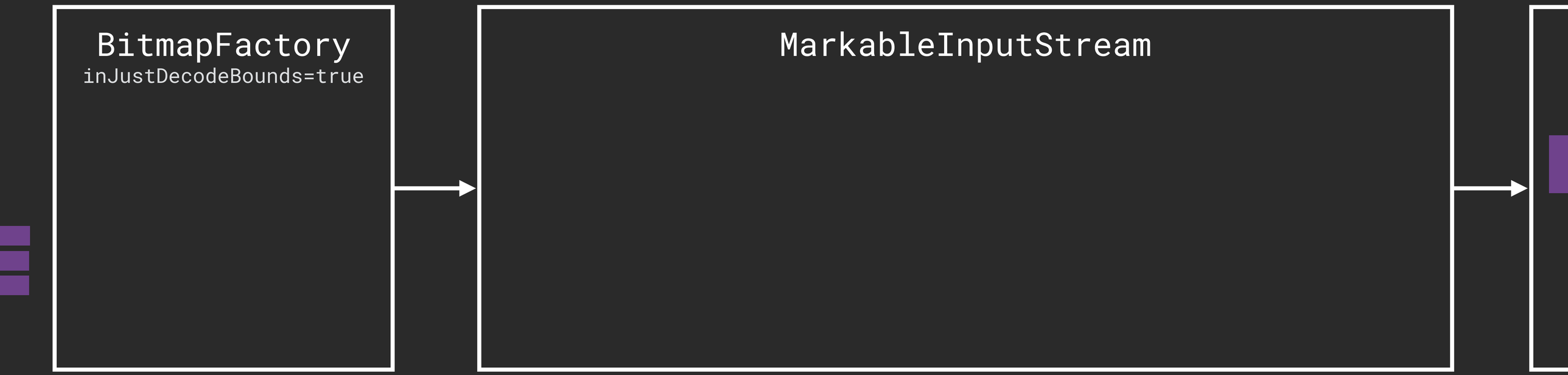

BitmapFactory
inJustDecodeBounds=true
MarkableInputStream
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

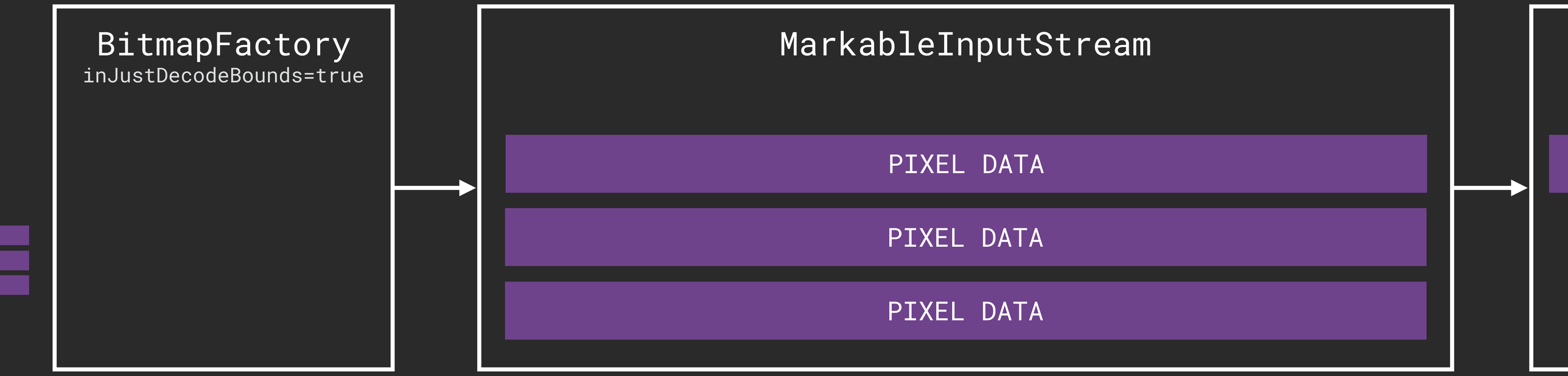

BitmapFactory
inJustDecodeBounds=true
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
MarkableInputStream
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

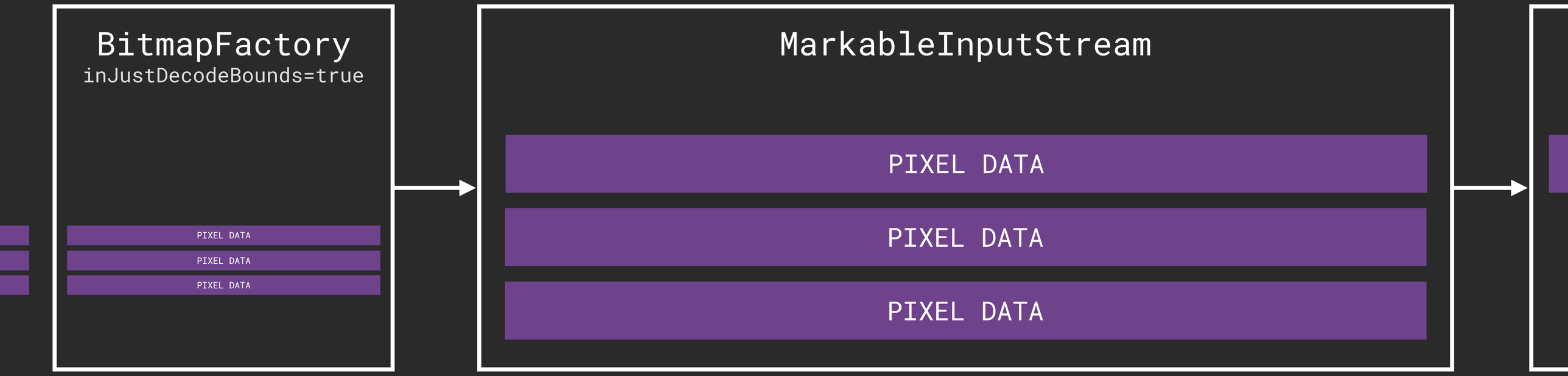

BitmapFactory
inJustDecodeBounds=true
MarkableInputStream

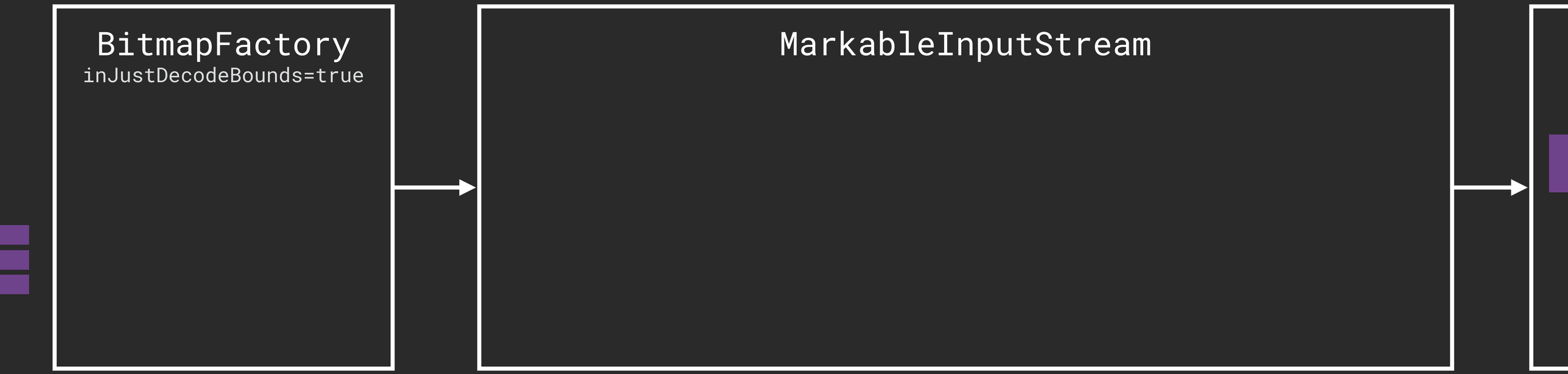

BitmapFactory
inJustDecodeBounds=true
MarkableInputStream
PIXEL DATA
PIXEL DATA
PIXEL DATA

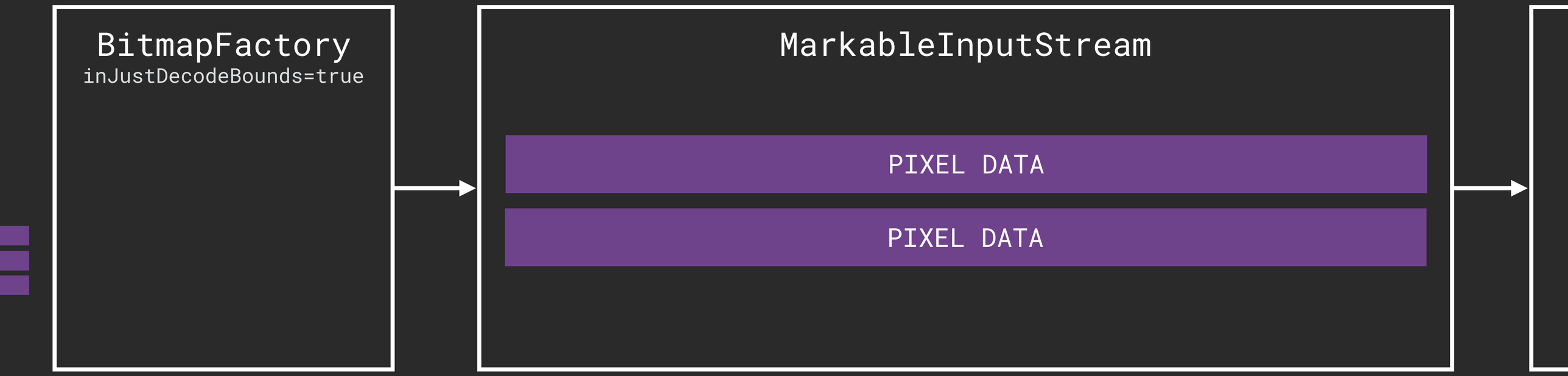

BitmapFactory
inJustDecodeBounds=true
PIXEL DATA
PIXEL DATA
PIXEL DATA
MarkableInputStream
PIXEL DATA
PIXEL DATA
PIXEL DATA

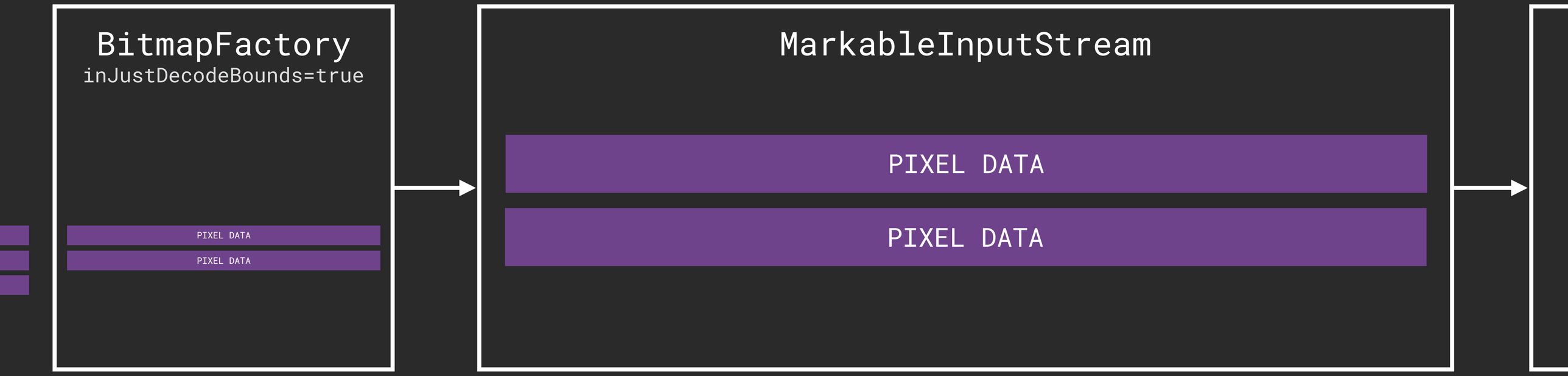

BitmapFactory
inJustDecodeBounds=true
MarkableInputStream

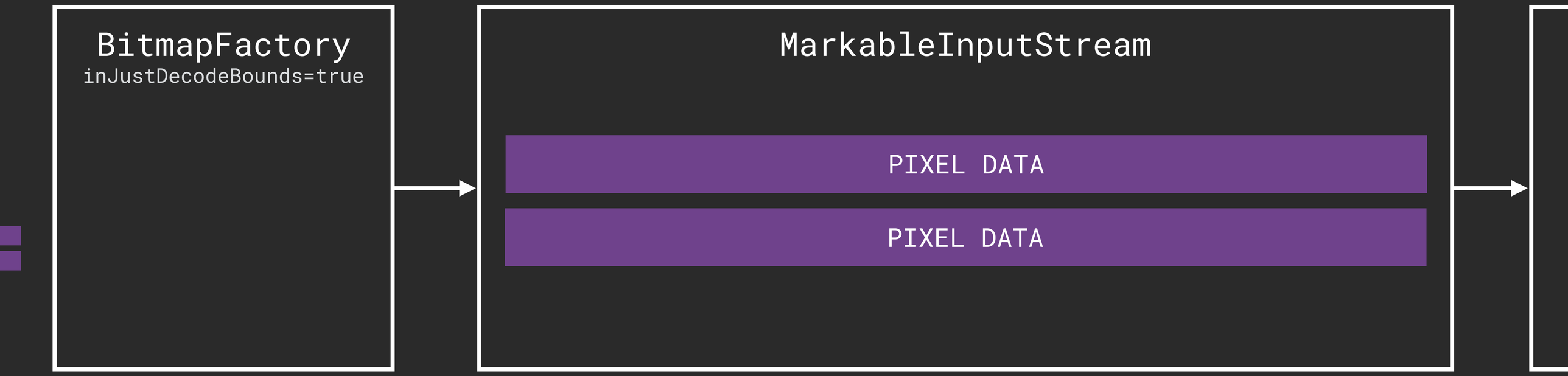

BitmapFactory
inJustDecodeBounds=true
MarkableInputStream
PIXEL DATA
PIXEL DATA
PIXEL DATA

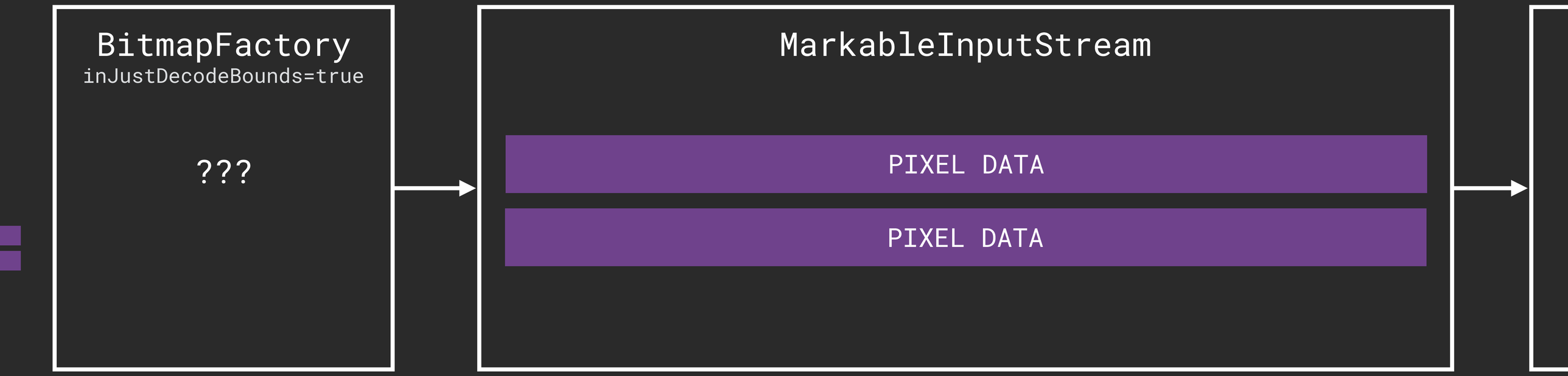

BitmapFactory
inJustDecodeBounds=true
PIXEL DATA
PIXEL DATA
PIXEL DATA
MarkableInputStream
PIXEL DATA
PIXEL DATA
PIXEL DATA


BitmapFactory
inJustDecodeBounds=true
MarkableInputStream

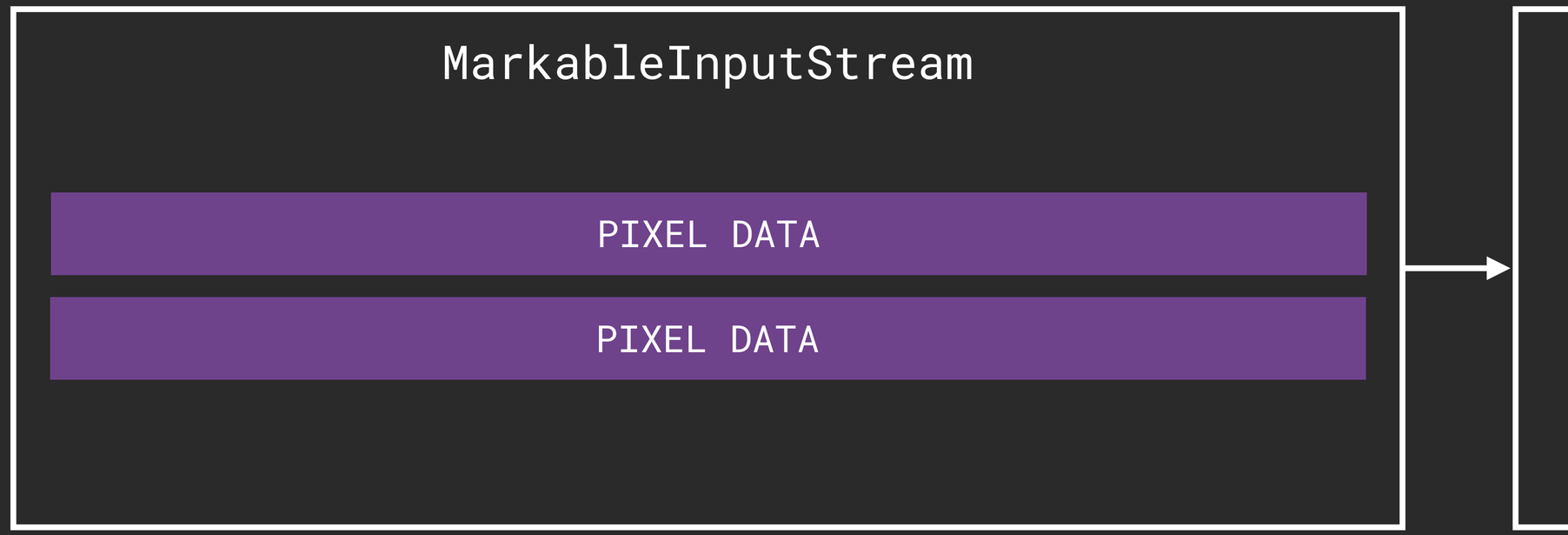

BitmapFactory
inJustDecodeBounds=true
MarkableInputStream
PIXEL DATA
PIXEL DATA


BitmapFactory
inJustDecodeBounds=true
PIXEL DATA
PIXEL DATA
MarkableInputStream
PIXEL DATA
PIXEL DATA

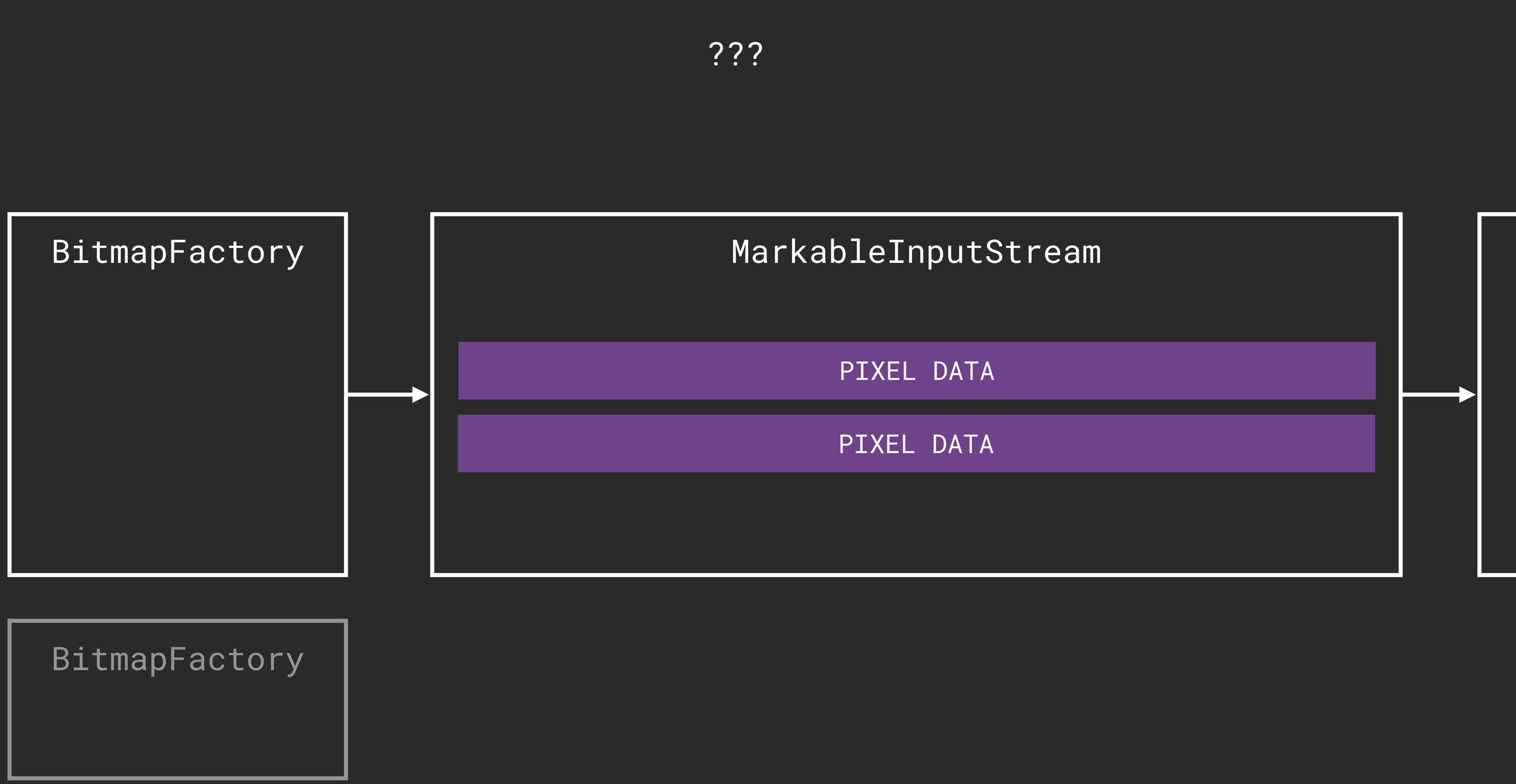

BitmapFactory
inJustDecodeBounds=true
MarkableInputStream
PIXEL DATA
PIXEL DATA


BitmapFactory
inJustDecodeBounds=true
???
MarkableInputStream
PIXEL DATA
PIXEL DATA

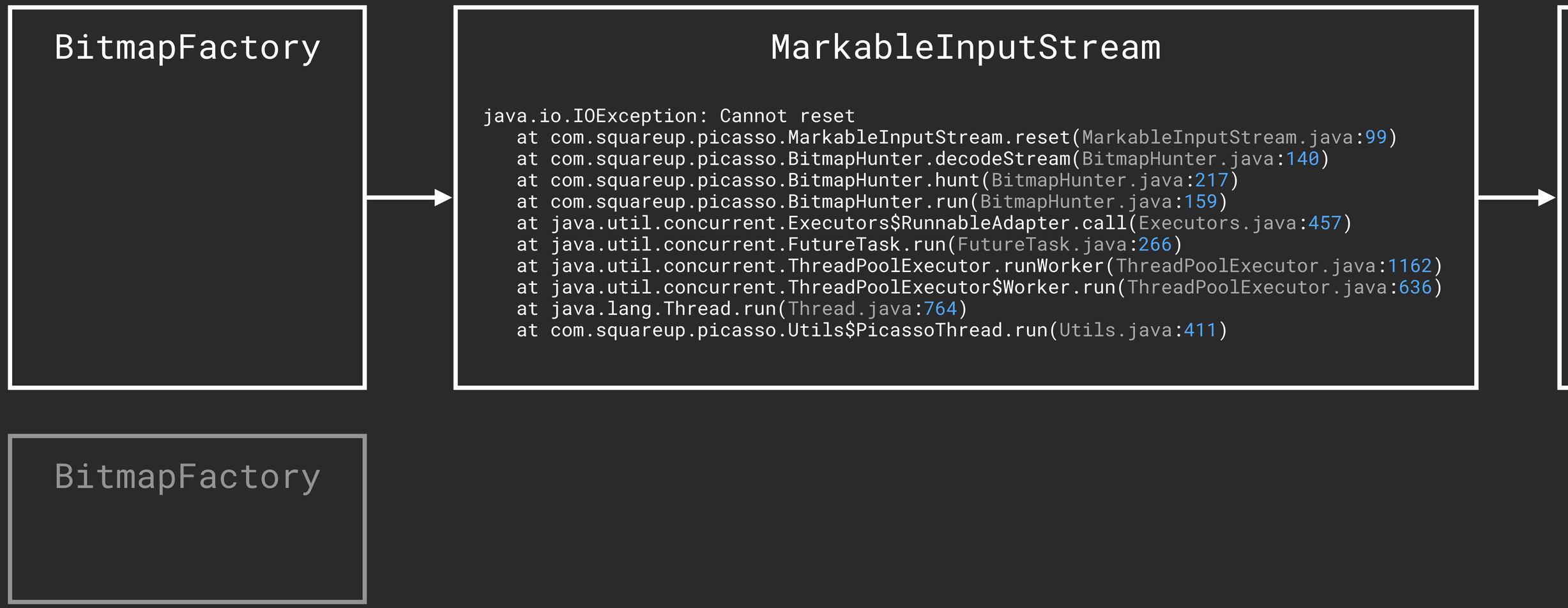
???



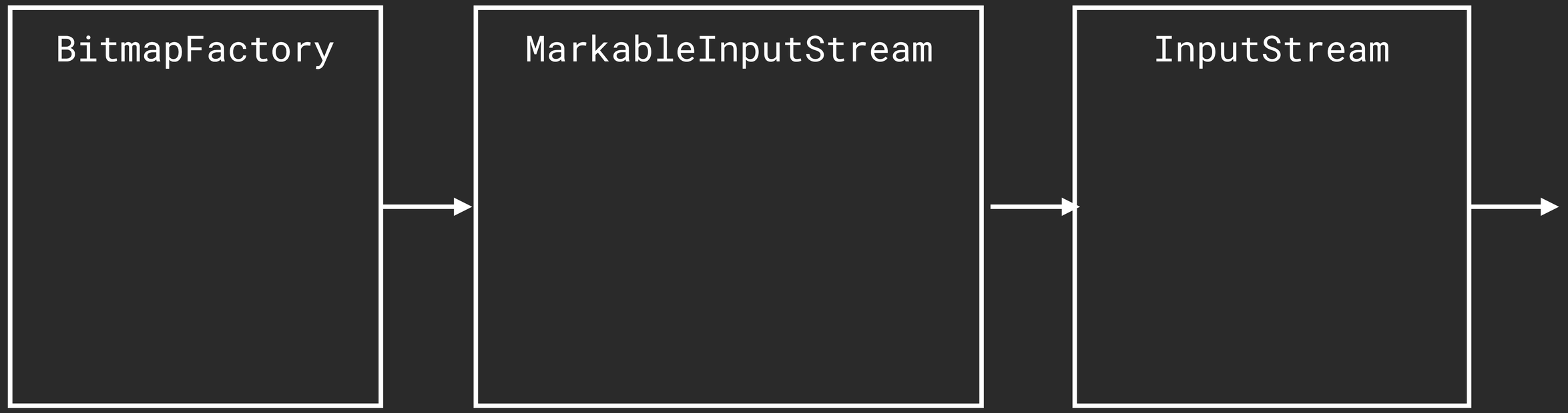
BitmapFactory


???
BitmapFactory
MarkableInputStream
PIXEL DATA
PIXEL DATA
BitmapFactory

???



BitmapFactory


BitmapFactory
MarkableInputStream
InputStream

Picasso 3.x

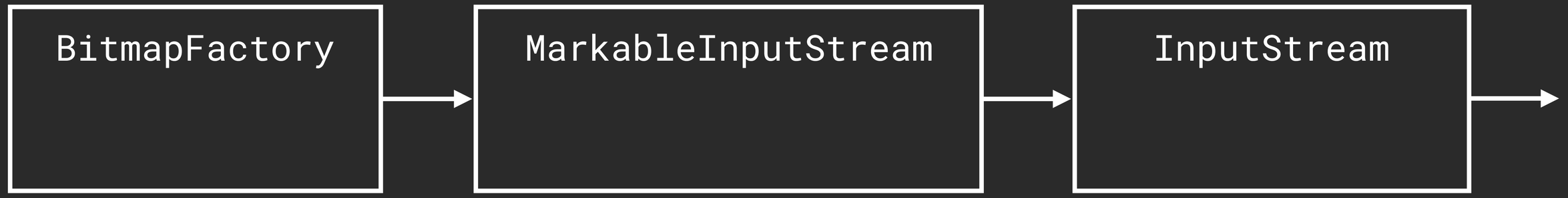
Picasso 2.x

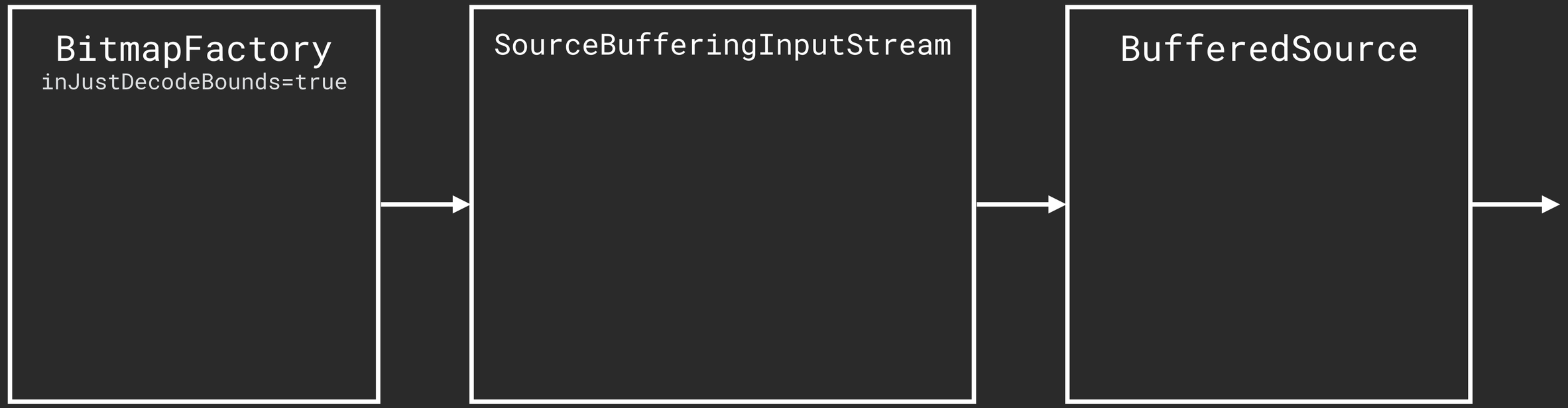
Picasso 3.x

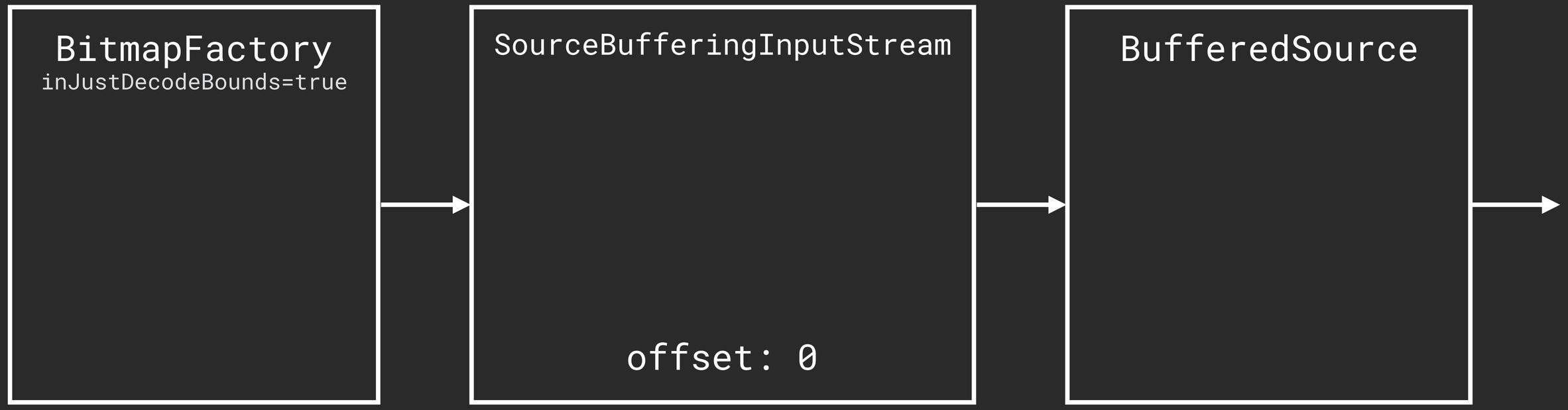
Picasso 2.x

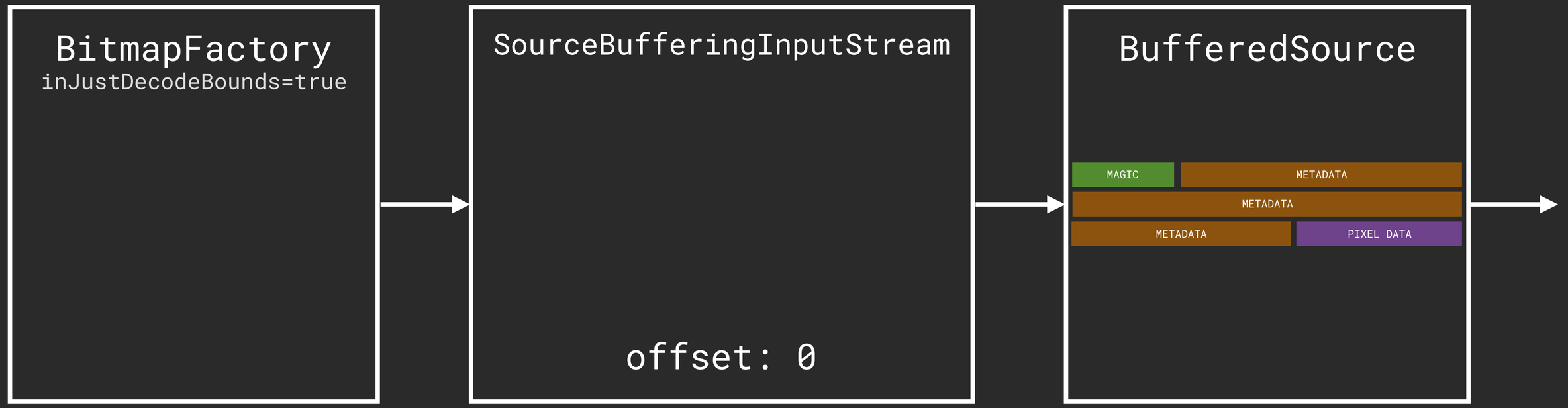
Picasso 3.x

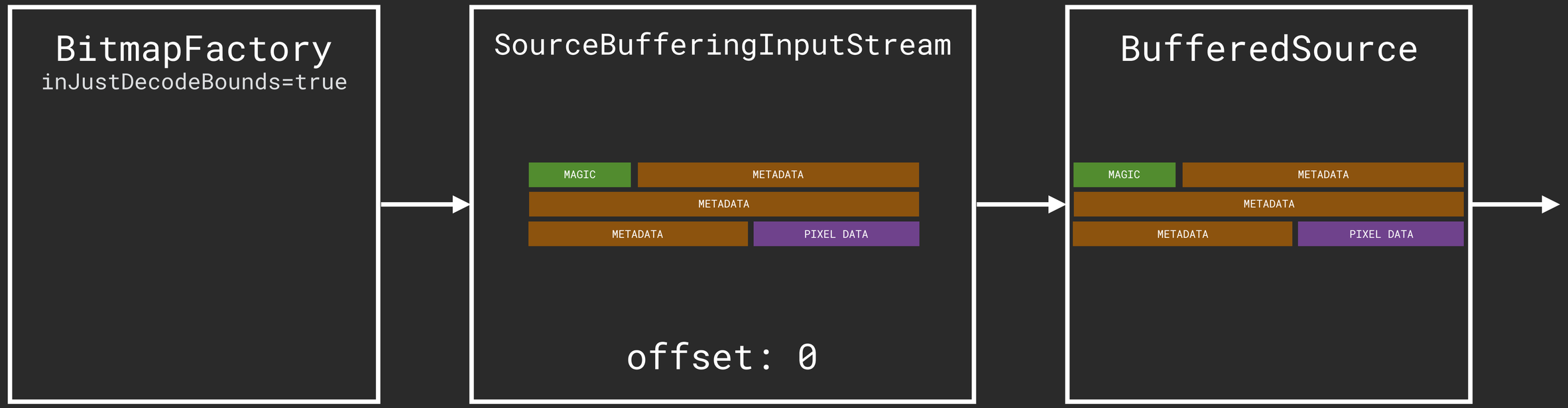
Picasso 2.x

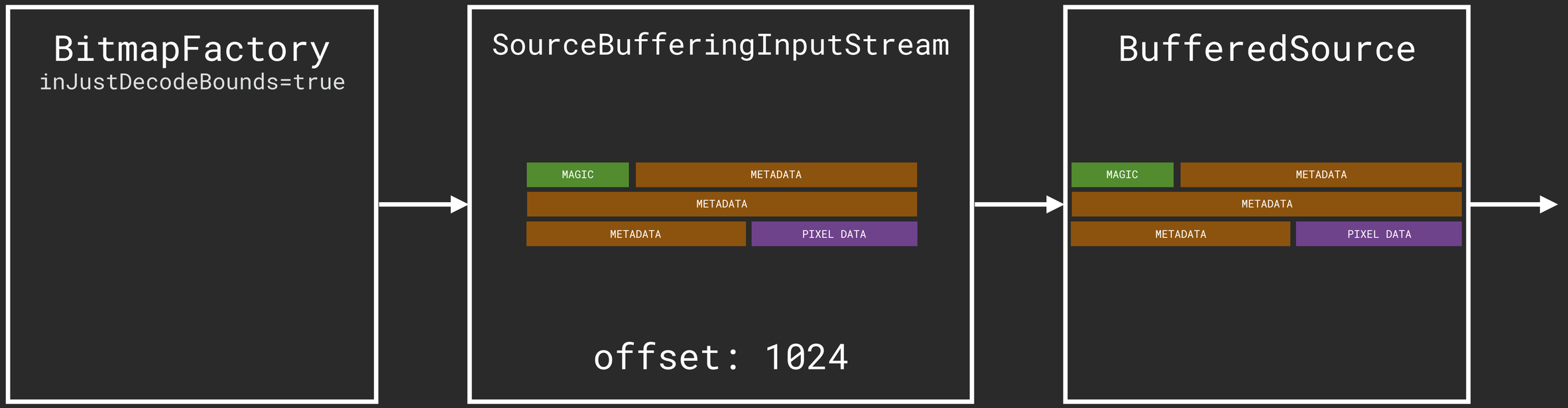


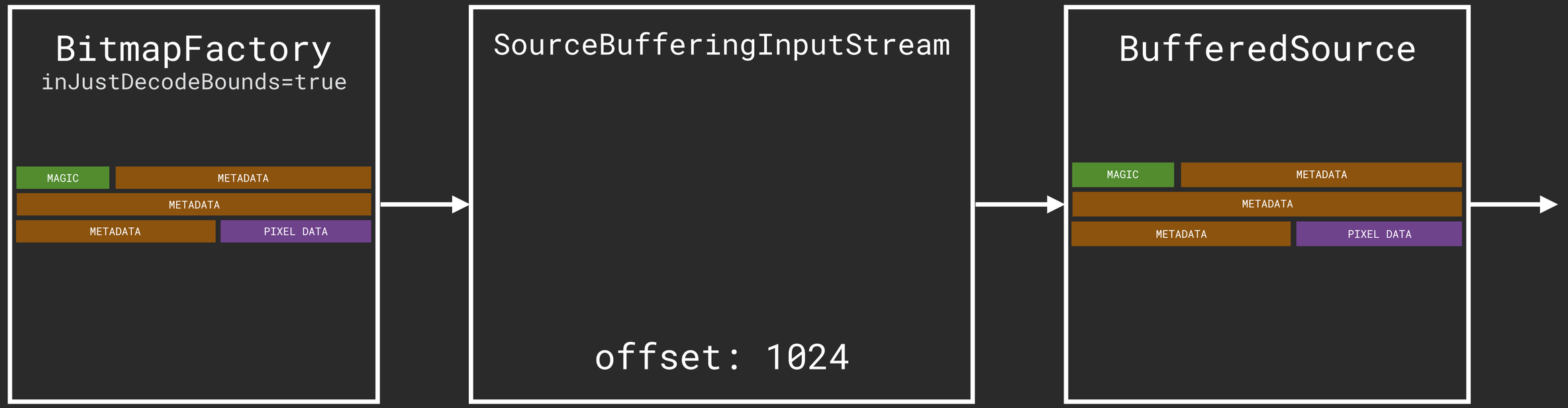

BitmapFactory
inJustDecodeBounds=true
SourceBufferingInputStream
BufferedSource

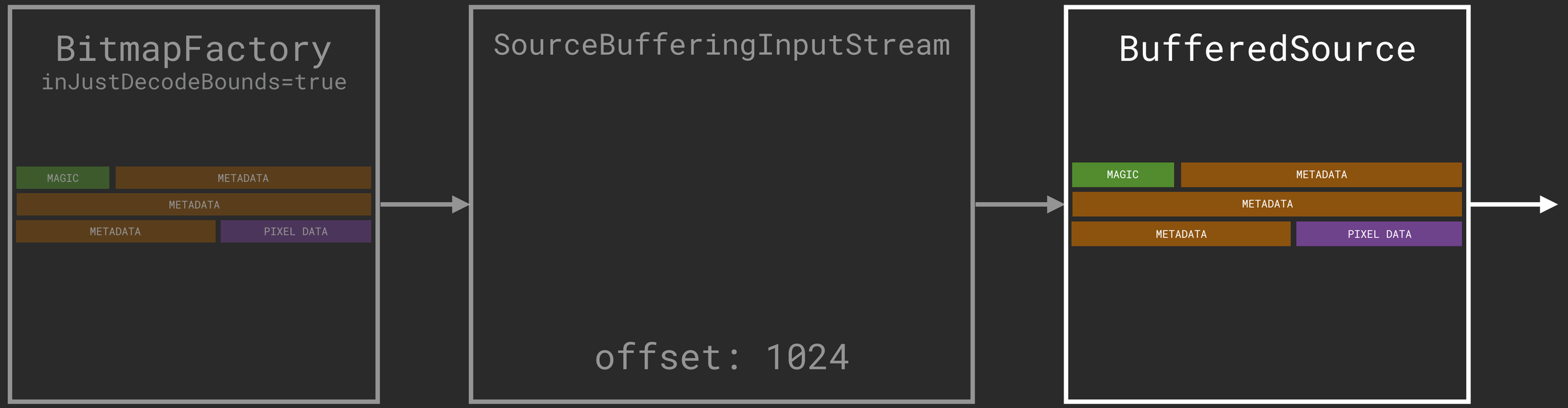

BitmapFactory
inJustDecodeBounds=true
SourceBufferingInputStream
offset: 0
BufferedSource

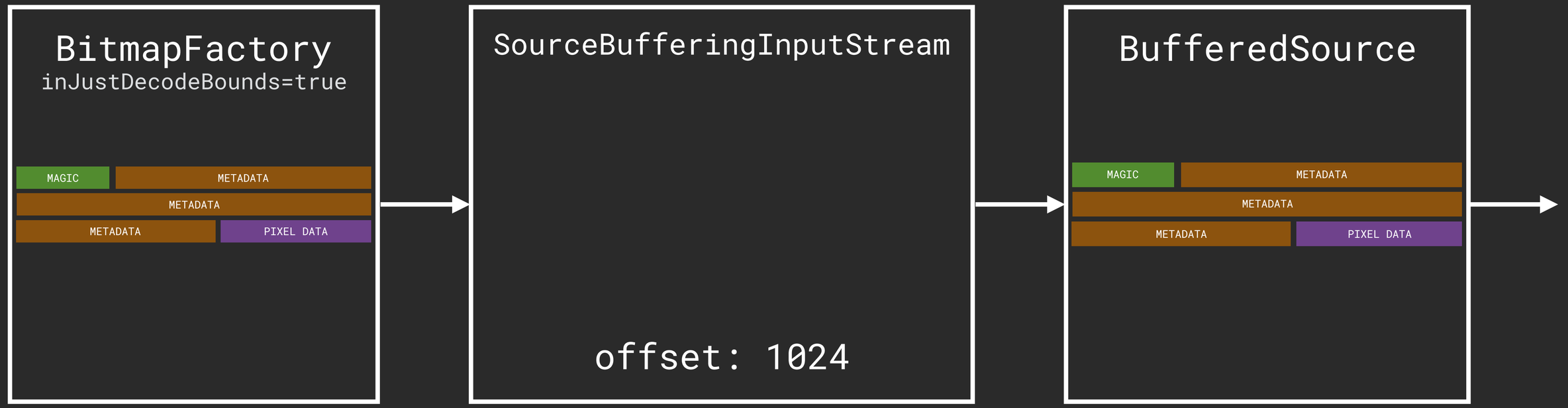

BitmapFactory
inJustDecodeBounds=true
SourceBufferingInputStream
offset: 0
BufferedSource
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

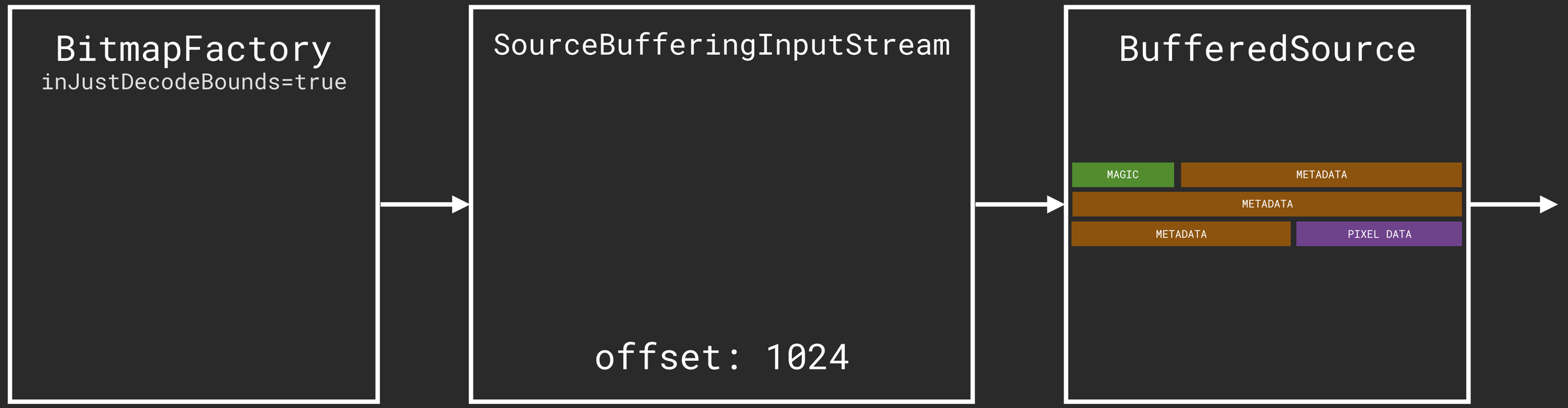

BitmapFactory
inJustDecodeBounds=true
SourceBufferingInputStream
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
offset: 0
BufferedSource
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

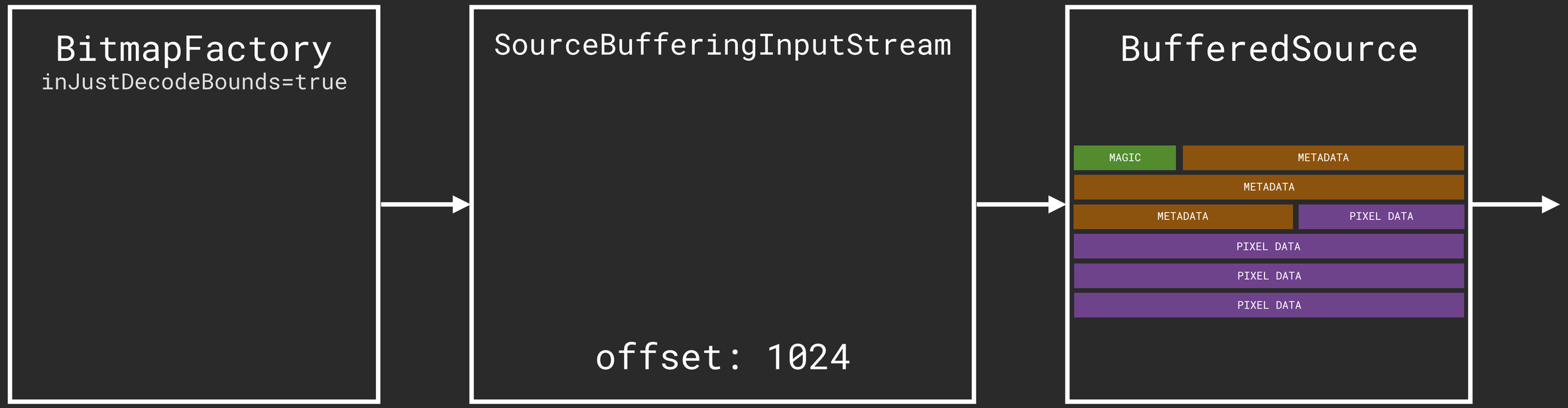

BitmapFactory
inJustDecodeBounds=true
SourceBufferingInputStream
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
offset: 1024
BufferedSource
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

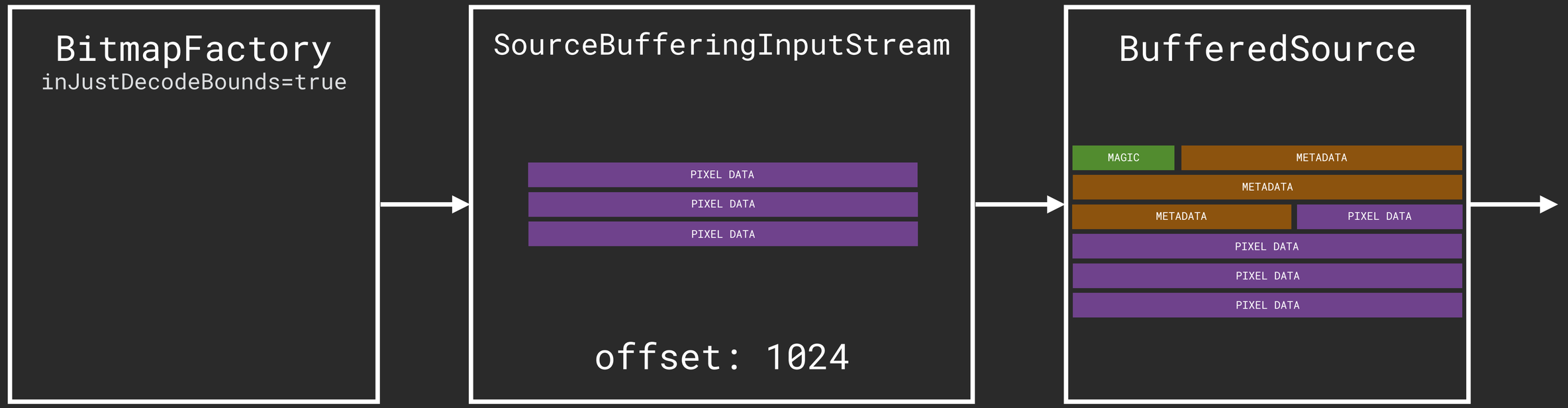

BitmapFactory
inJustDecodeBounds=true
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
SourceBufferingInputStream
offset: 1024
BufferedSource
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

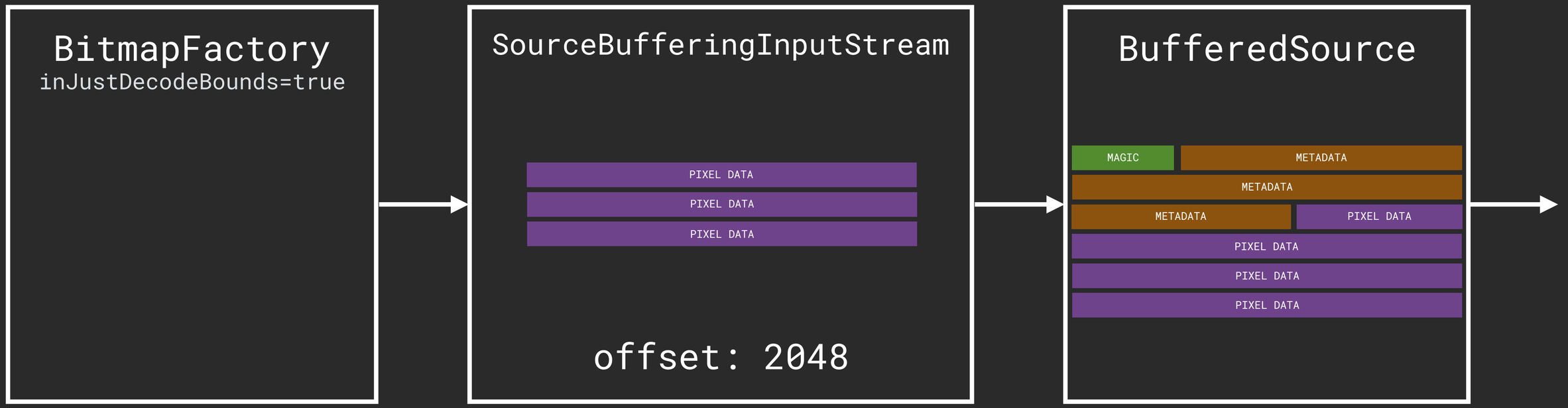

BitmapFactory
inJustDecodeBounds=true
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
SourceBufferingInputStream
offset: 1024
BufferedSource
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

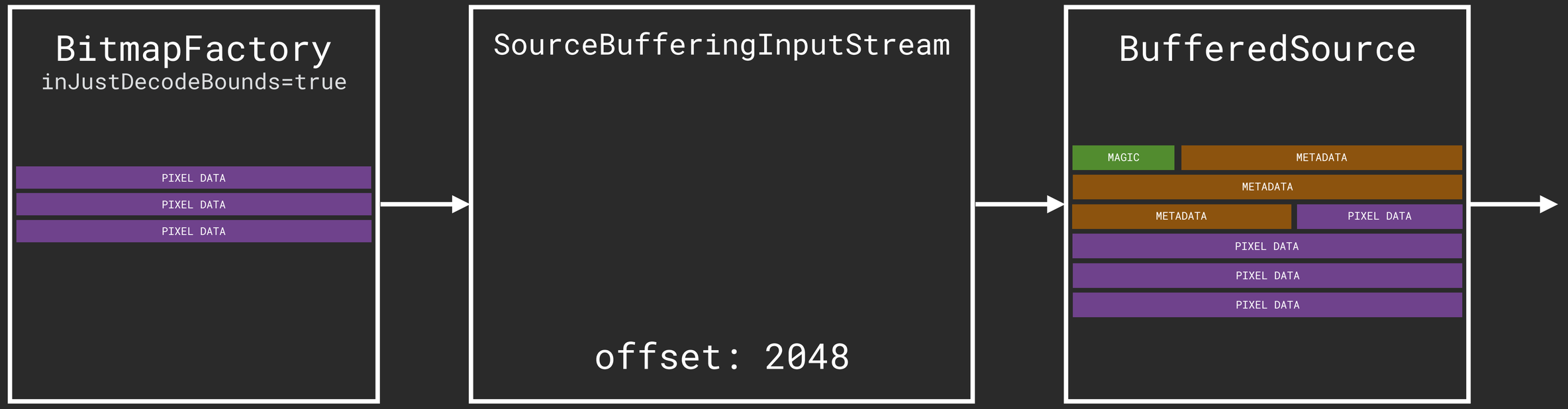

BitmapFactory
inJustDecodeBounds=true
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
SourceBufferingInputStream
offset: 1024
BufferedSource
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

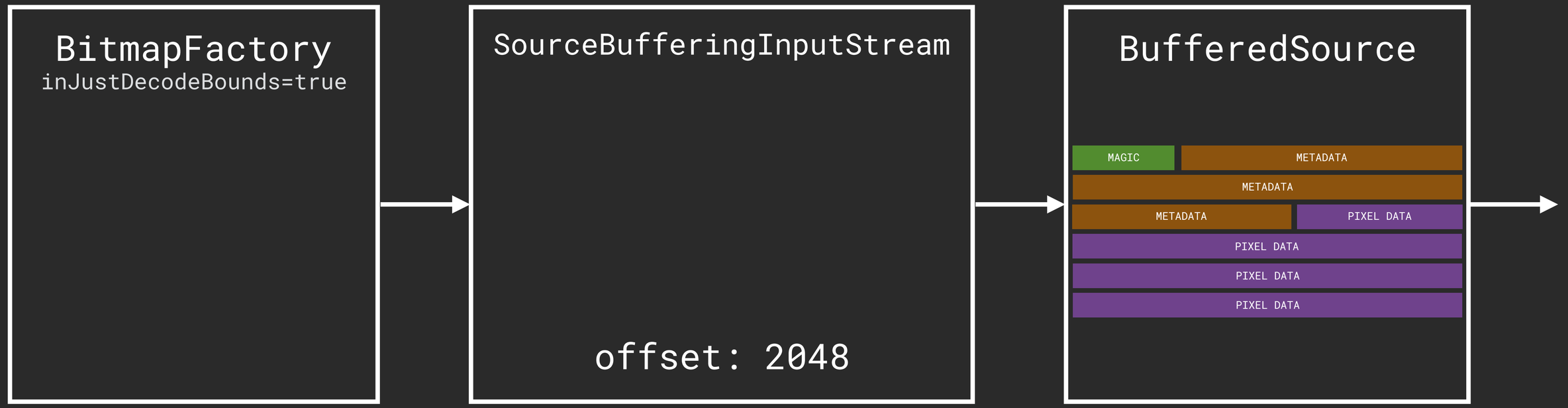

BitmapFactory
inJustDecodeBounds=true
SourceBufferingInputStream
offset: 1024
BufferedSource
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA

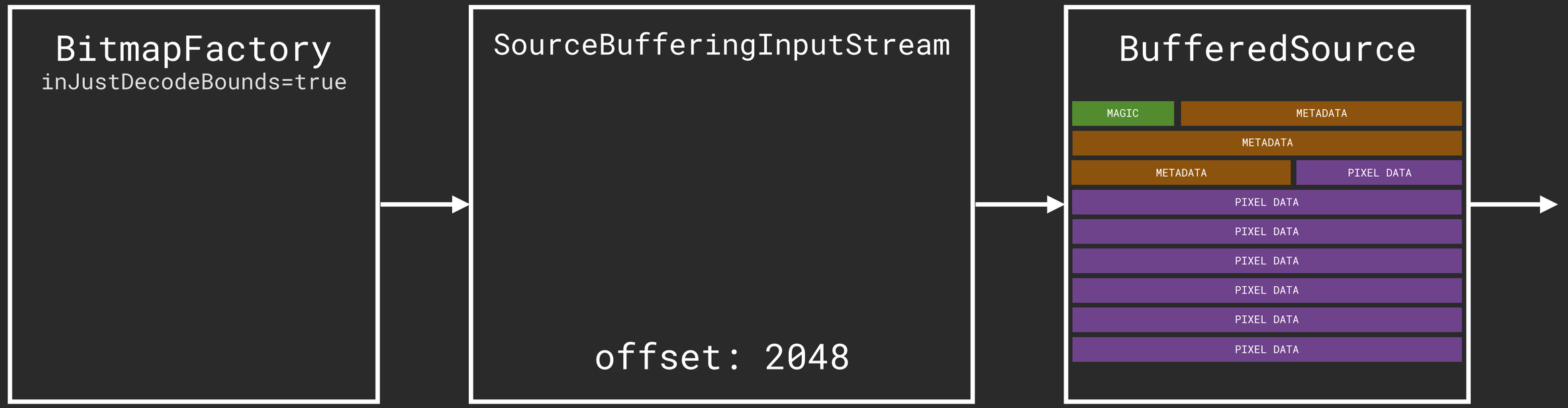

BitmapFactory
inJustDecodeBounds=true
SourceBufferingInputStream
offset: 1024
BufferedSource
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA

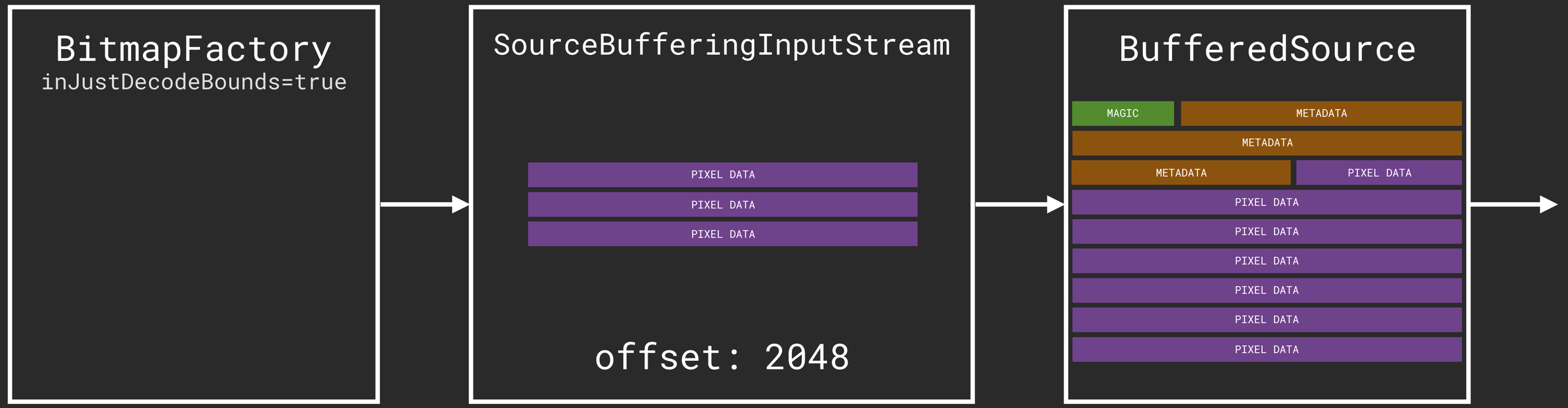

BitmapFactory
inJustDecodeBounds=true
SourceBufferingInputStream
PIXEL DATA
PIXEL DATA
PIXEL DATA
offset: 1024
BufferedSource
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA

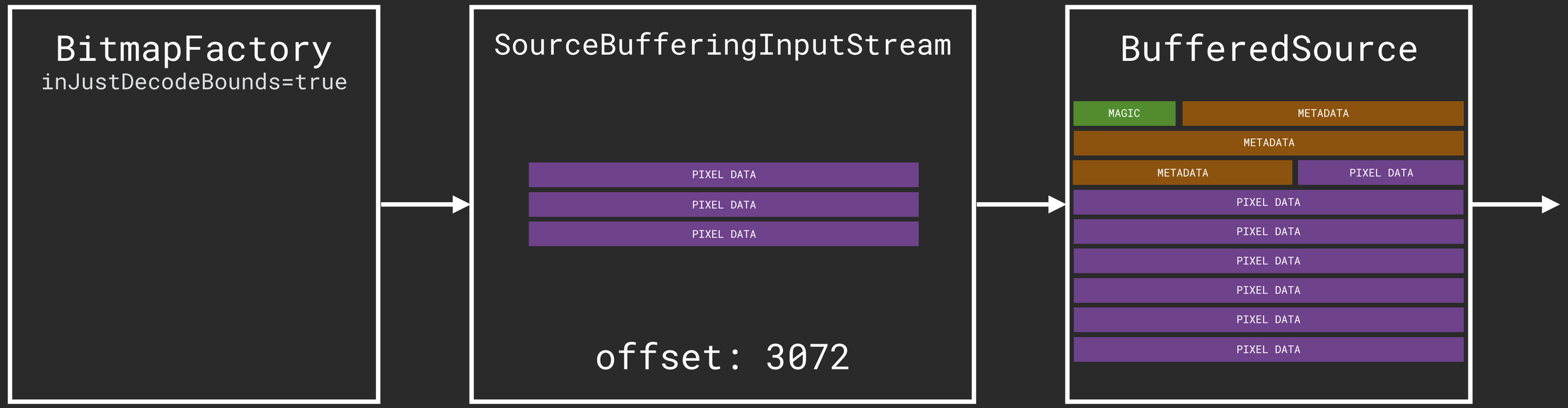

BitmapFactory
inJustDecodeBounds=true
SourceBufferingInputStream
PIXEL DATA
PIXEL DATA
PIXEL DATA
offset: 2048
BufferedSource
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA

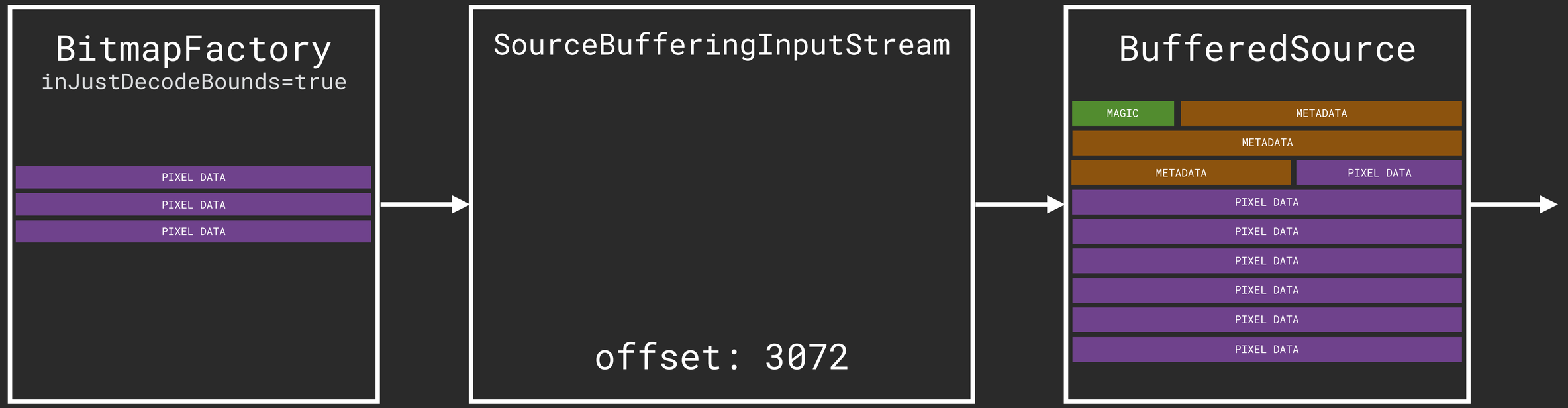

BitmapFactory
inJustDecodeBounds=true
PIXEL DATA
PIXEL DATA
PIXEL DATA
SourceBufferingInputStream
offset: 2048
BufferedSource
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA

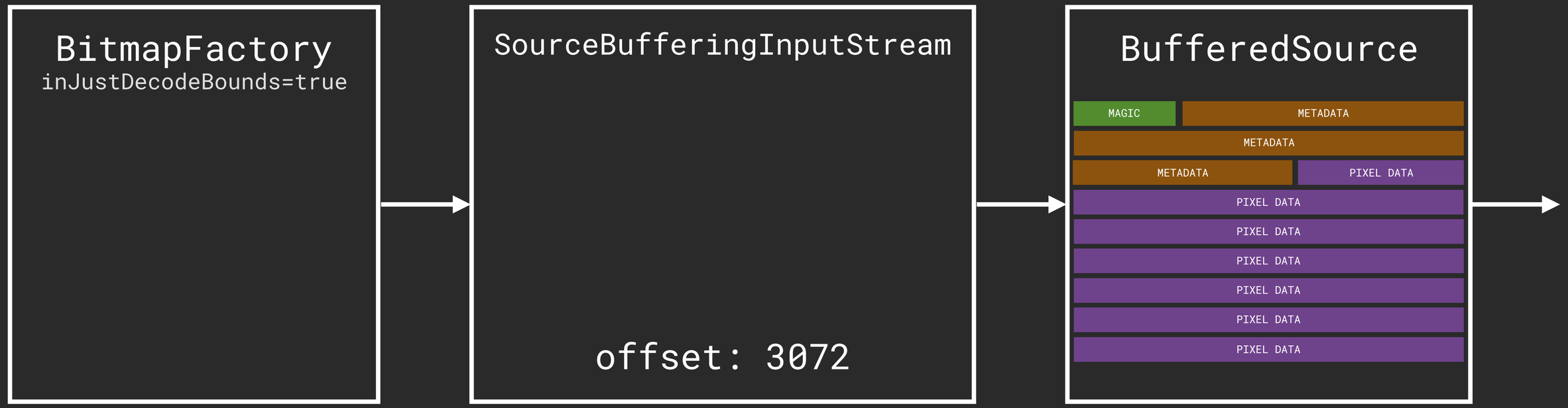

BitmapFactory
inJustDecodeBounds=true
SourceBufferingInputStream
offset: 2048
BufferedSource
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA

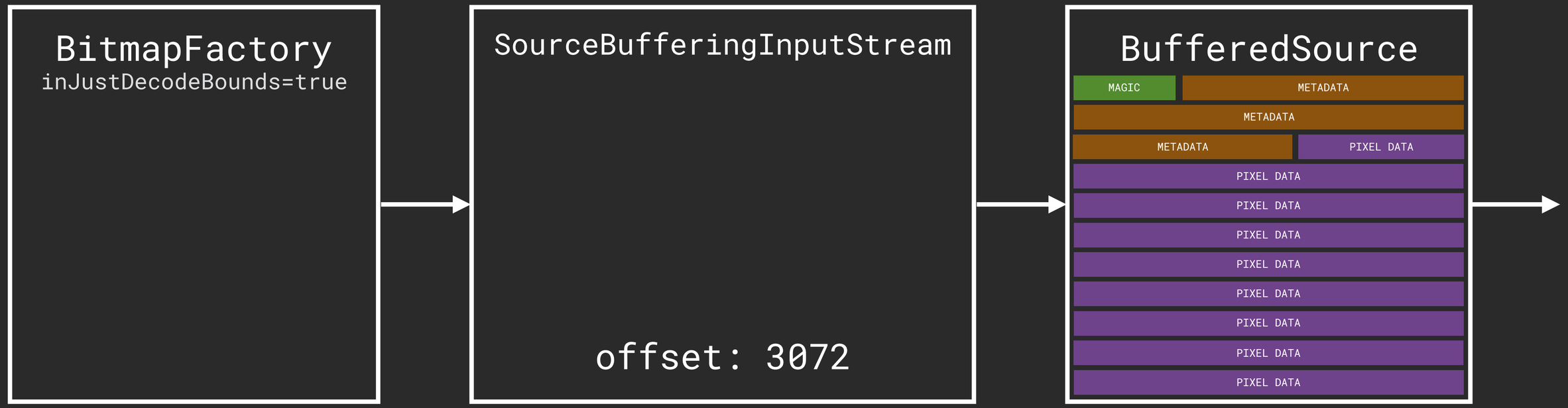

BitmapFactory
inJustDecodeBounds=true
SourceBufferingInputStream
offset: 2048
BufferedSource
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA

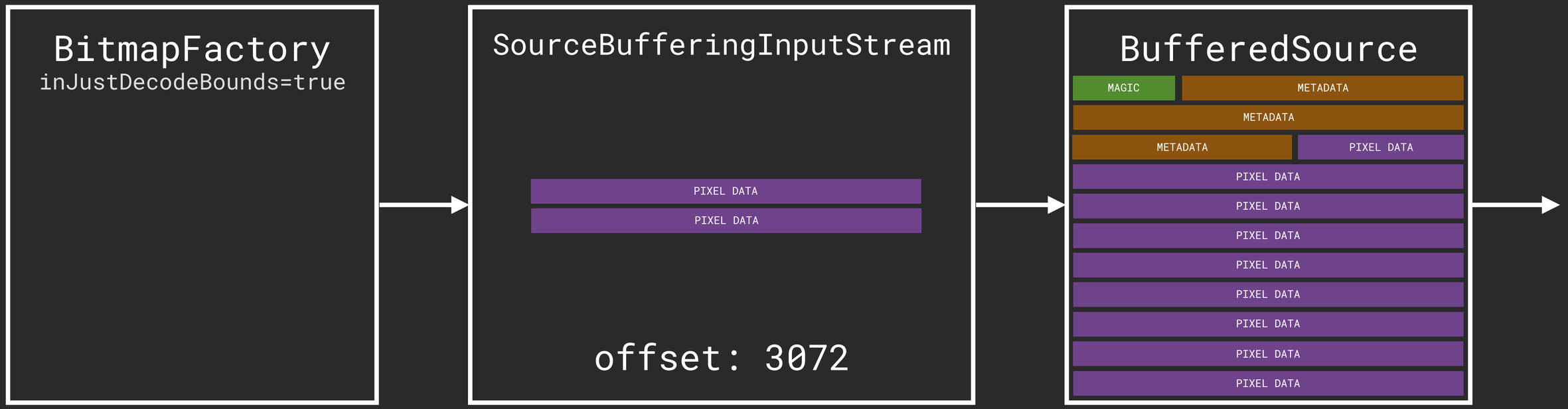

BitmapFactory
inJustDecodeBounds=true
SourceBufferingInputStream
PIXEL DATA
PIXEL DATA
PIXEL DATA
offset: 2048
BufferedSource
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA

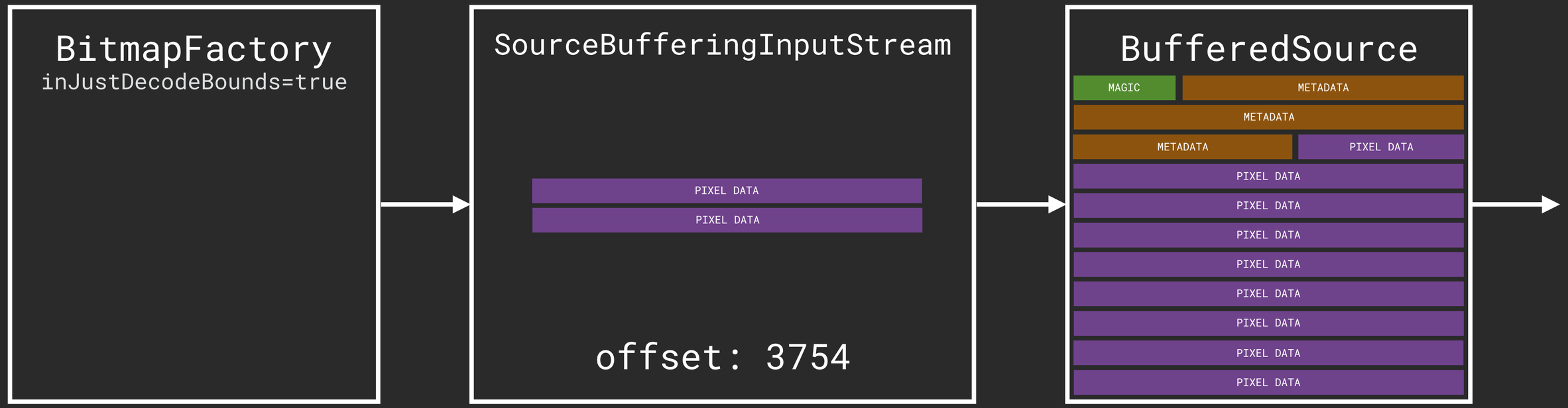

BitmapFactory
inJustDecodeBounds=true
SourceBufferingInputStream
PIXEL DATA
PIXEL DATA
PIXEL DATA
offset: 3072
BufferedSource
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA

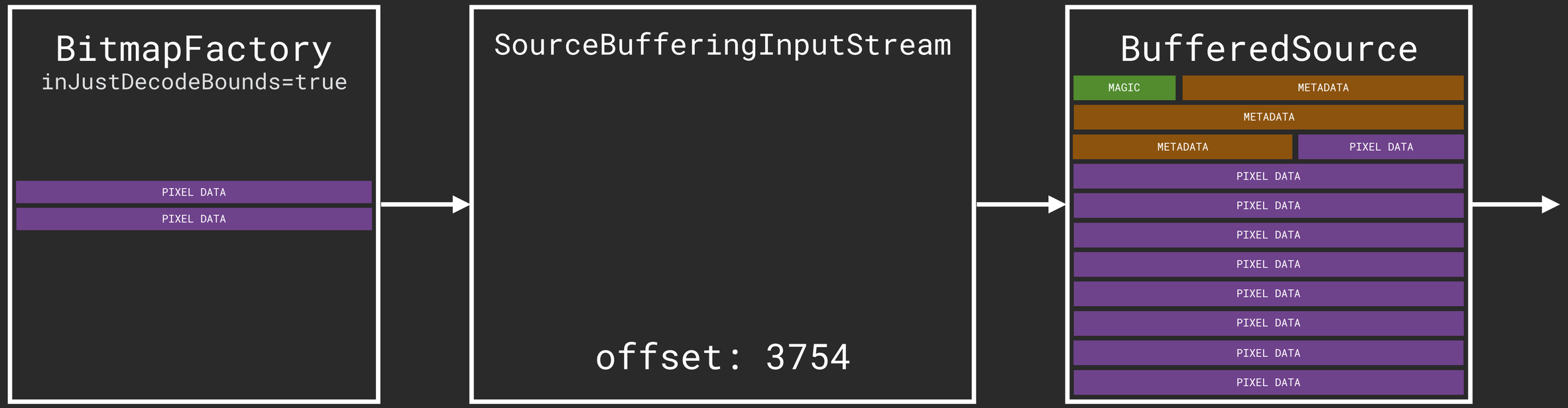

BitmapFactory
inJustDecodeBounds=true
PIXEL DATA
PIXEL DATA
PIXEL DATA
SourceBufferingInputStream
offset: 3072
BufferedSource
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA

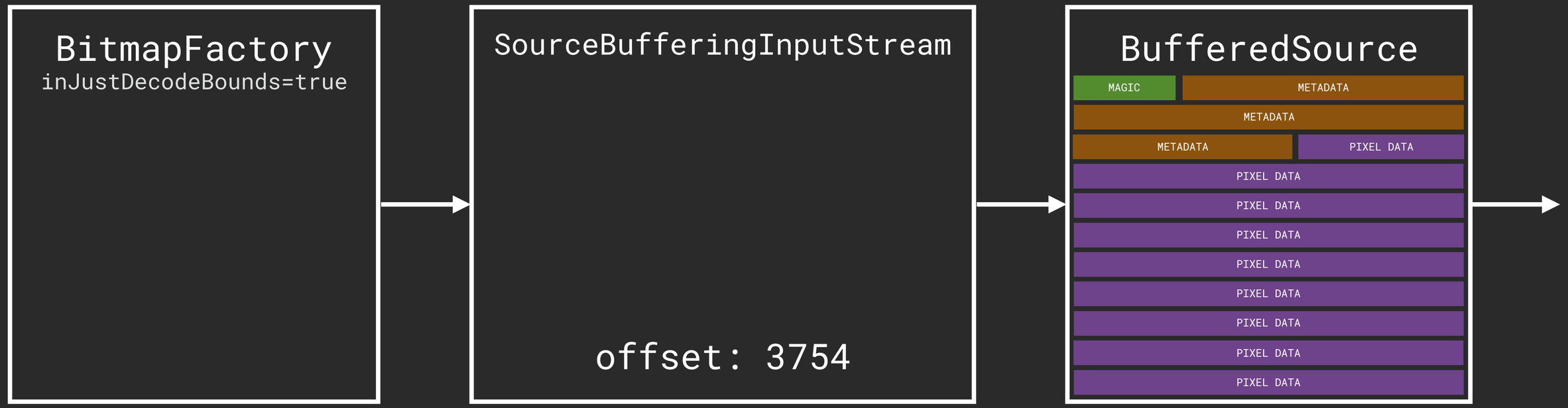

BitmapFactory
inJustDecodeBounds=true
SourceBufferingInputStream
offset: 3072
BufferedSource
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA

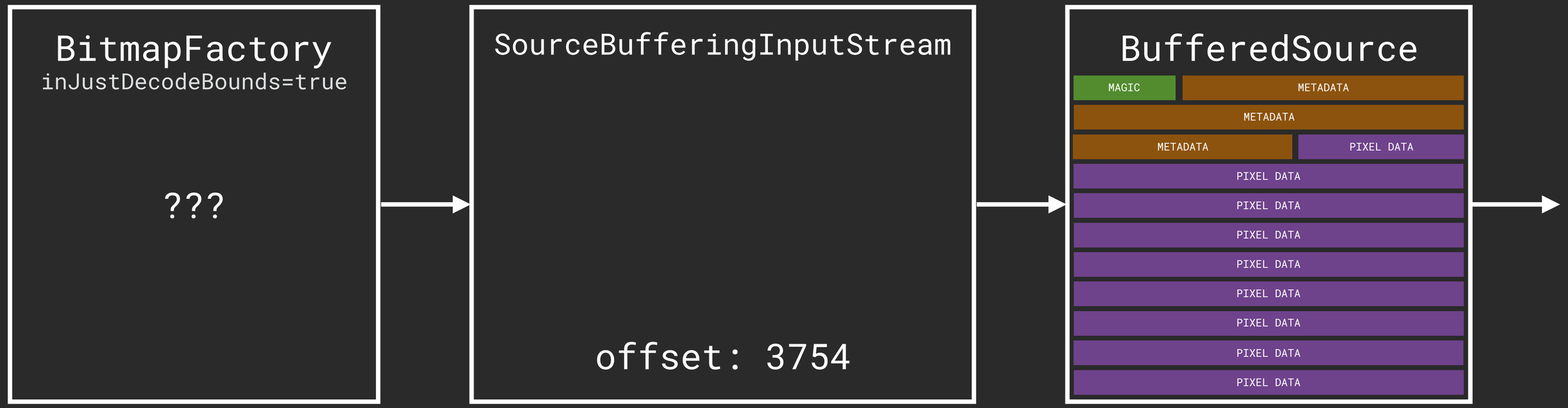

BitmapFactory
inJustDecodeBounds=true
SourceBufferingInputStream
offset: 3072
BufferedSource
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA


BitmapFactory
inJustDecodeBounds=true
SourceBufferingInputStream
PIXEL DATA
PIXEL DATA
offset: 3072
BufferedSource
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA

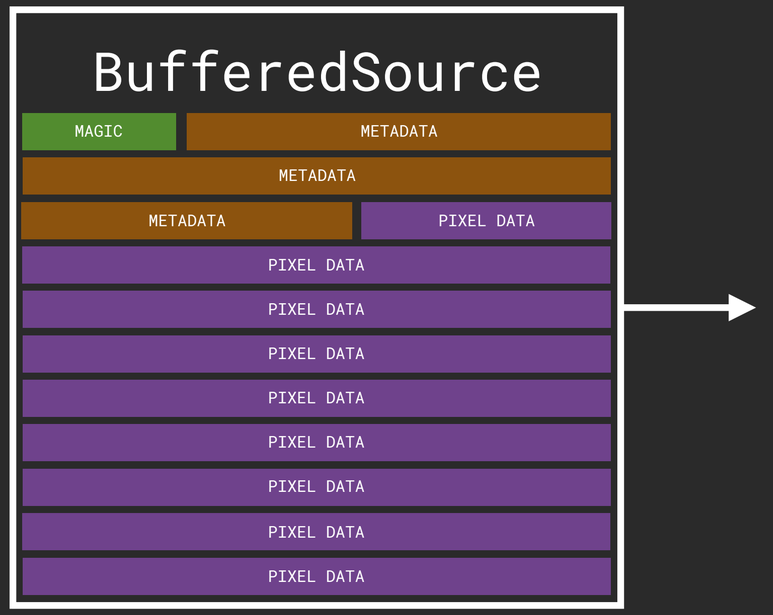

BitmapFactory
inJustDecodeBounds=true
SourceBufferingInputStream
PIXEL DATA
PIXEL DATA
offset: 3754
BufferedSource
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA

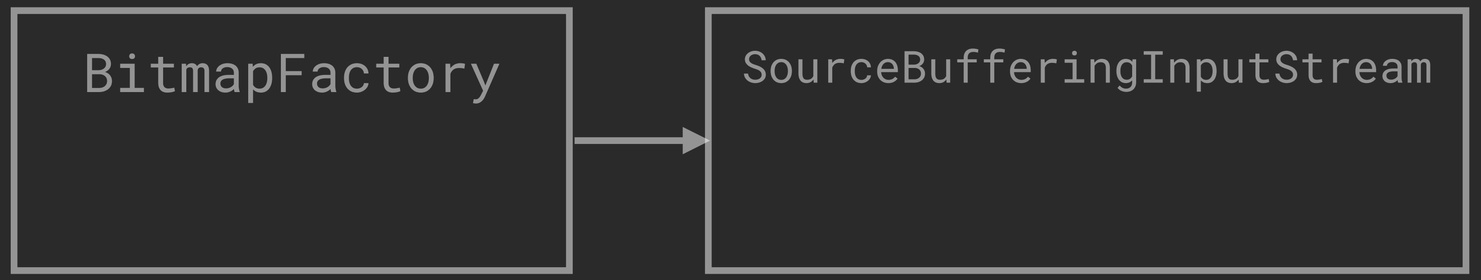

BitmapFactory
inJustDecodeBounds=true
PIXEL DATA
PIXEL DATA
SourceBufferingInputStream
offset: 3754
BufferedSource
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA

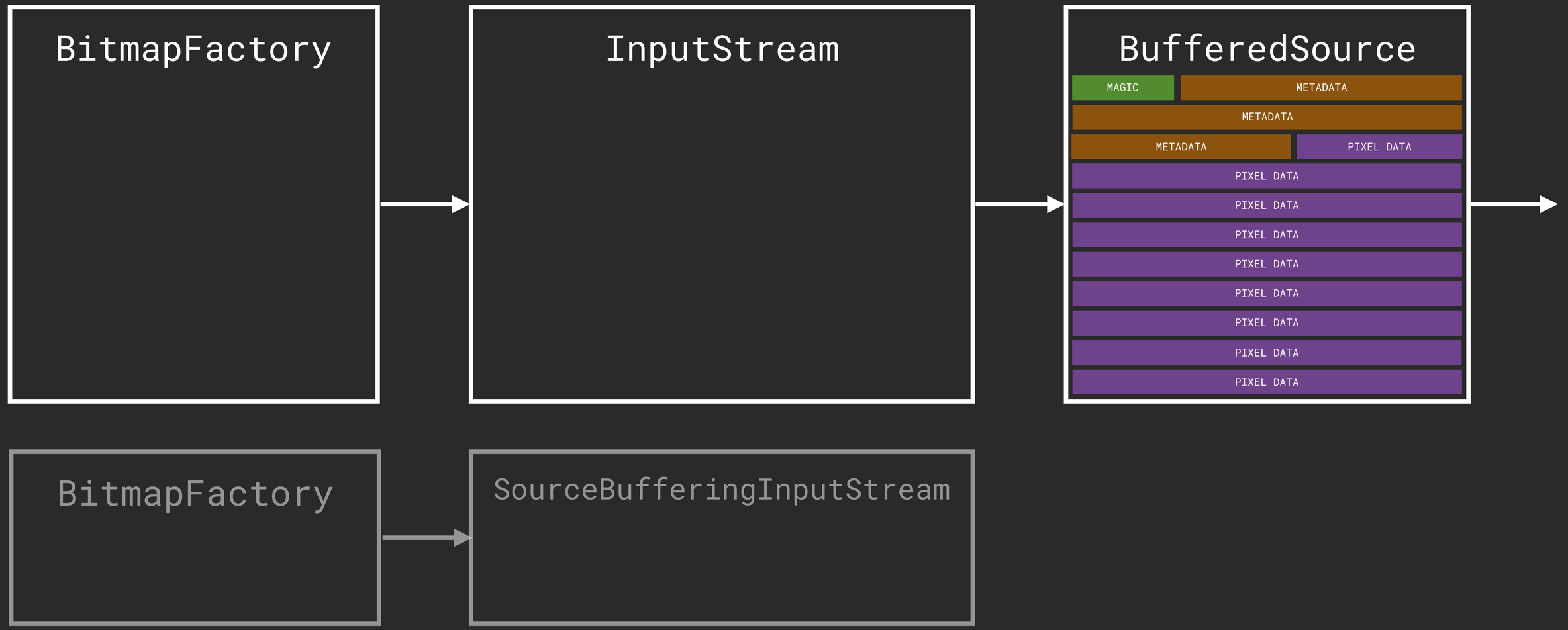

BitmapFactory
inJustDecodeBounds=true
SourceBufferingInputStream
offset: 3754
BufferedSource
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA

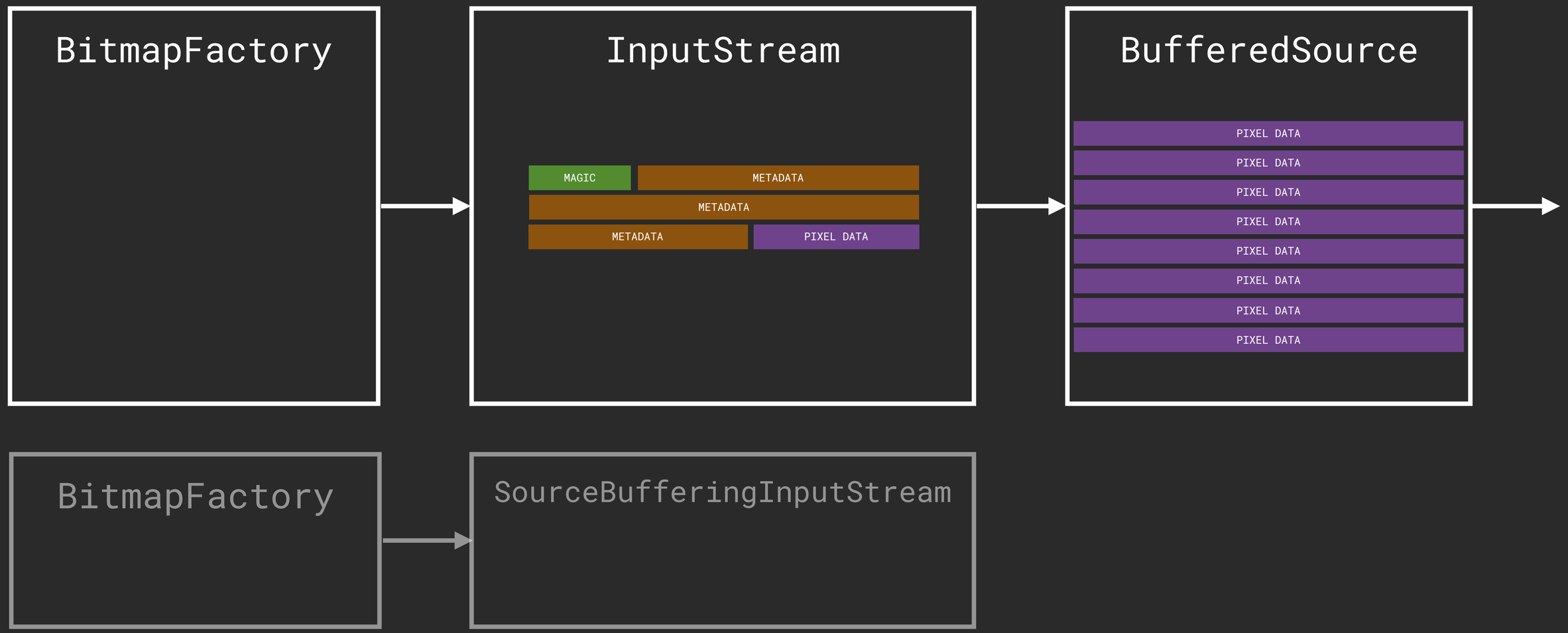

BitmapFactory
inJustDecodeBounds=true
???
SourceBufferingInputStream
offset: 3754
BufferedSource
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA

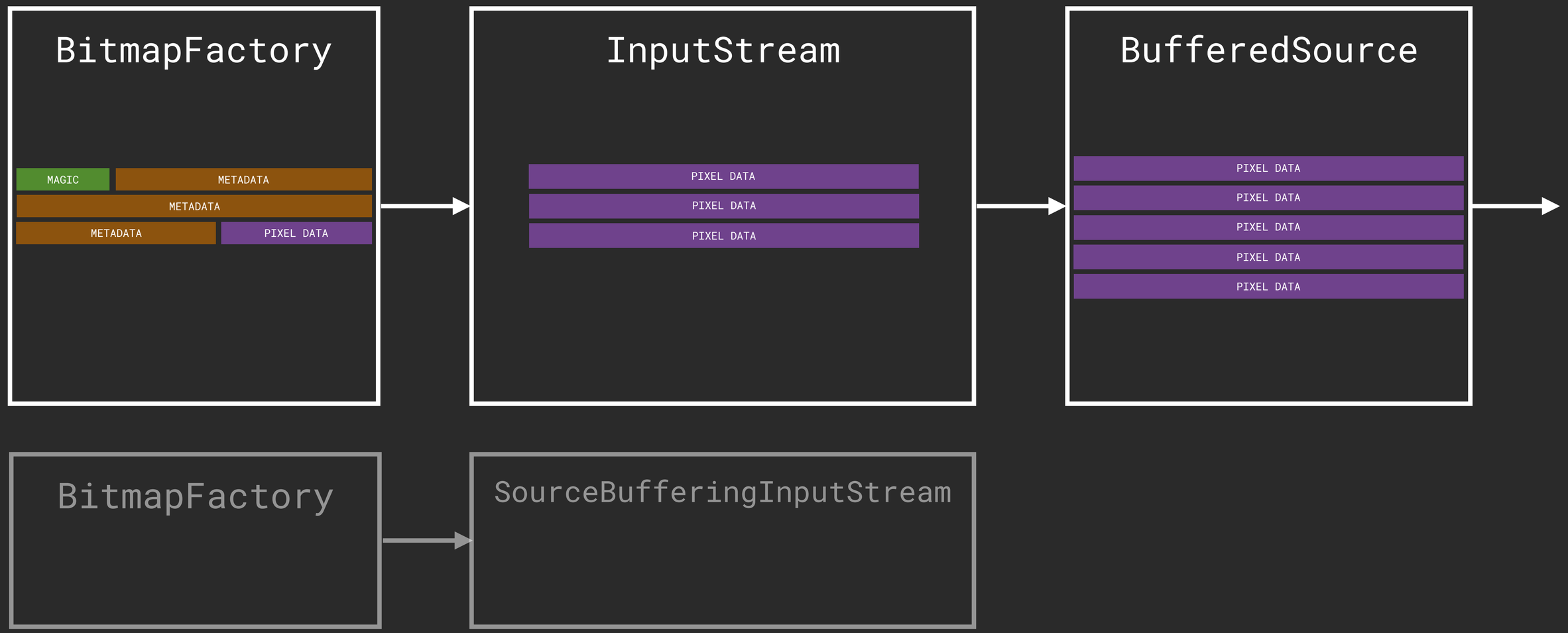
???

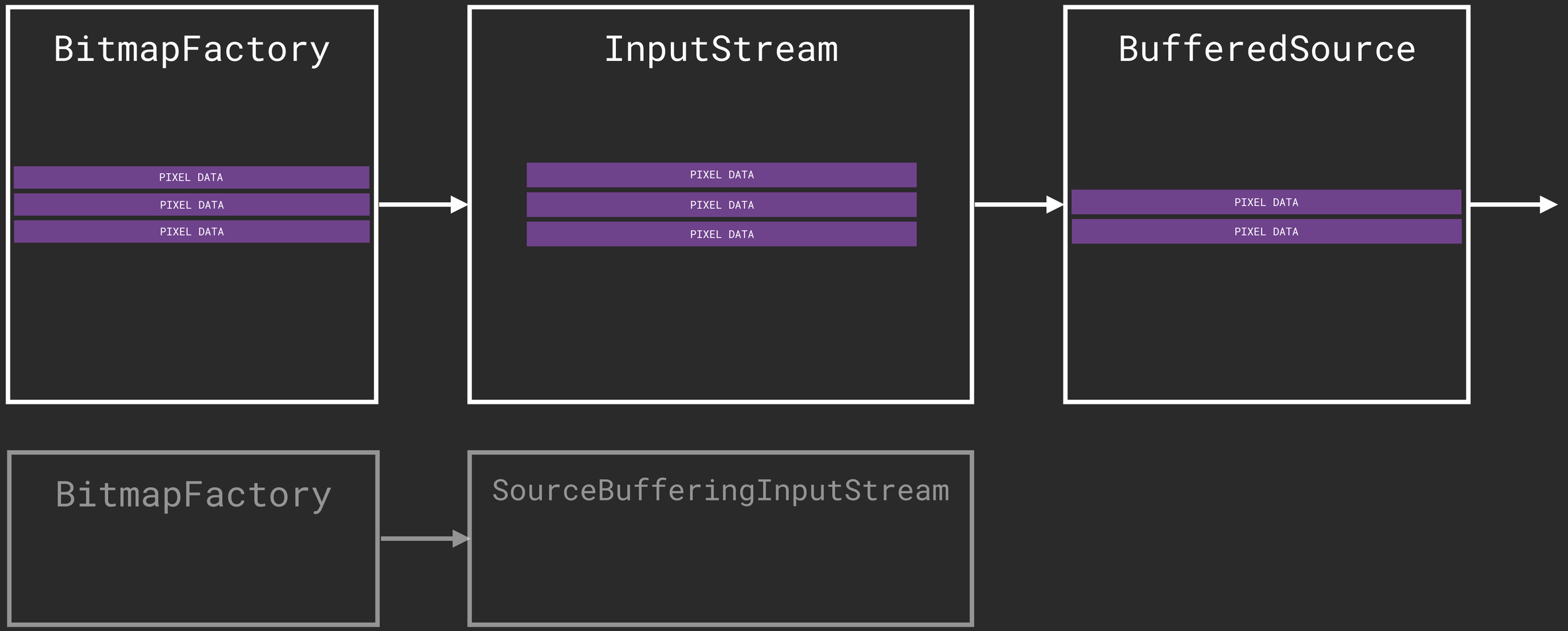

BufferedSource
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA

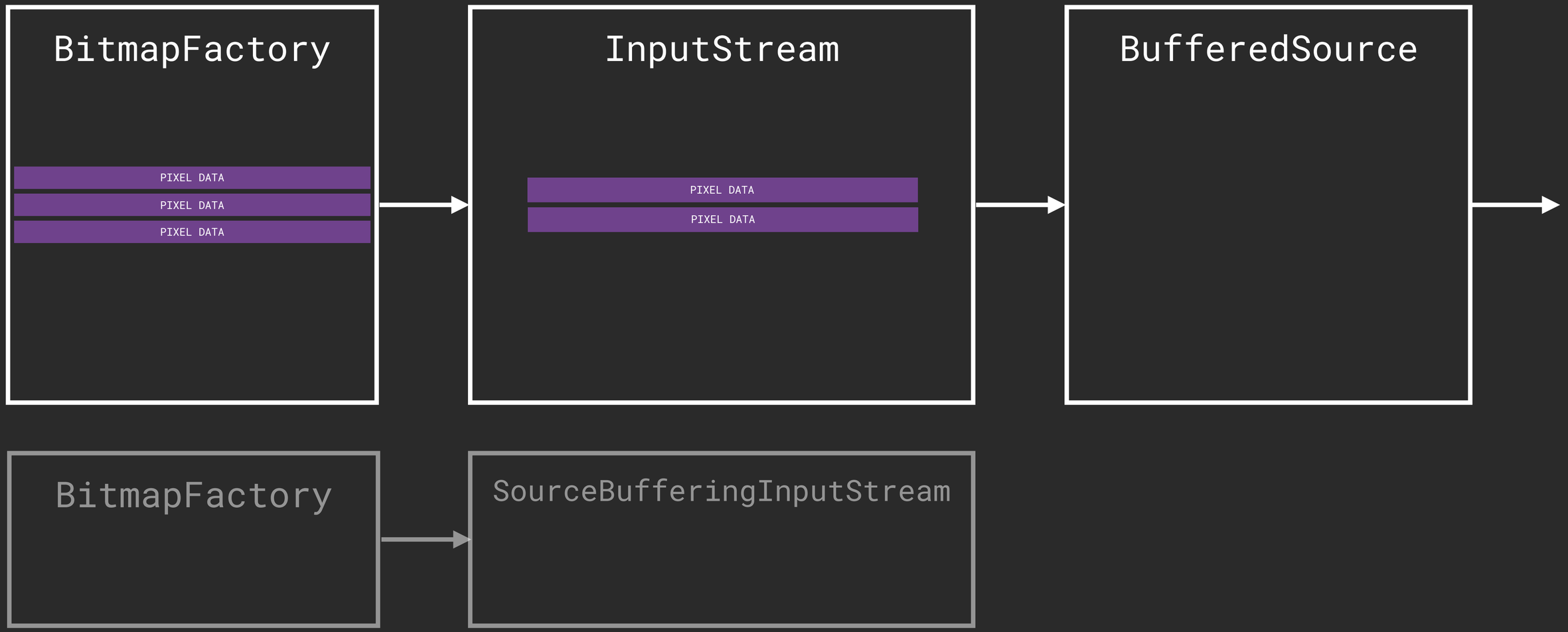

BitmapFactory
SourceBufferingInputStream

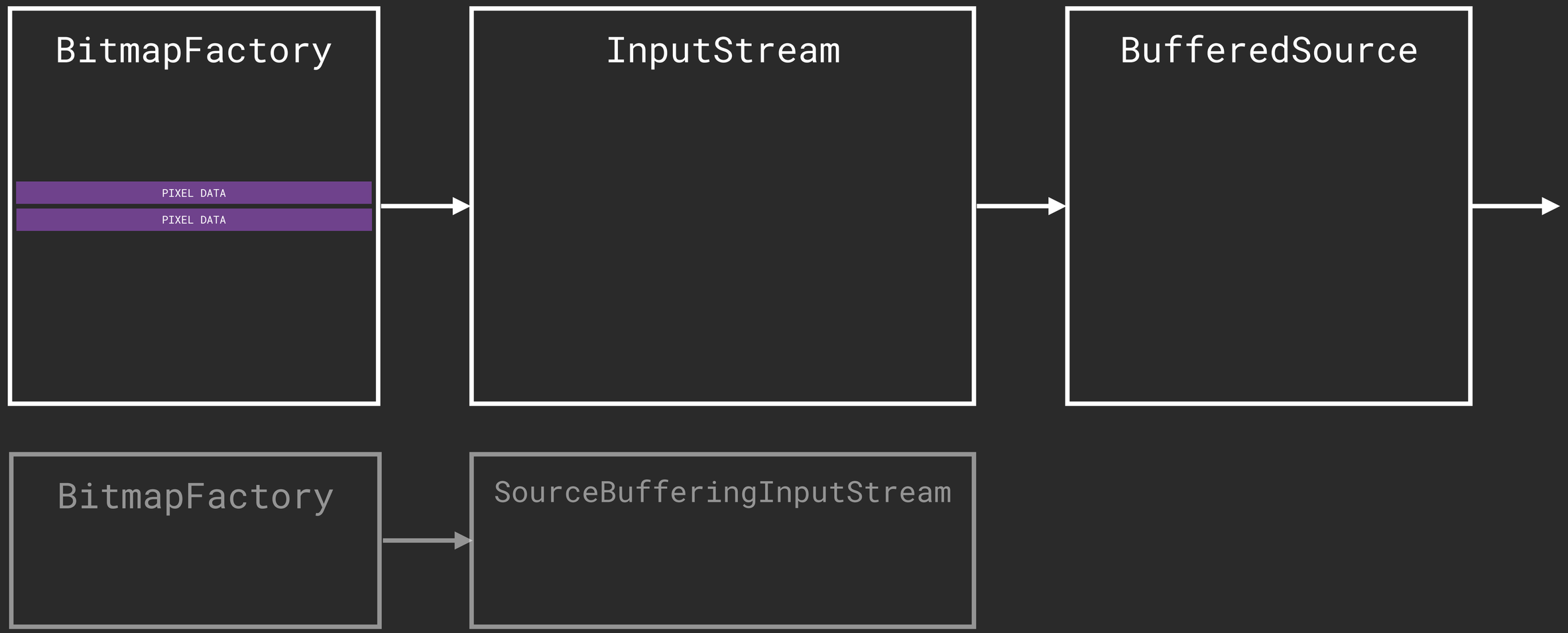

BitmapFactory
InputStream
BufferedSource
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
BitmapFactory
SourceBufferingInputStream

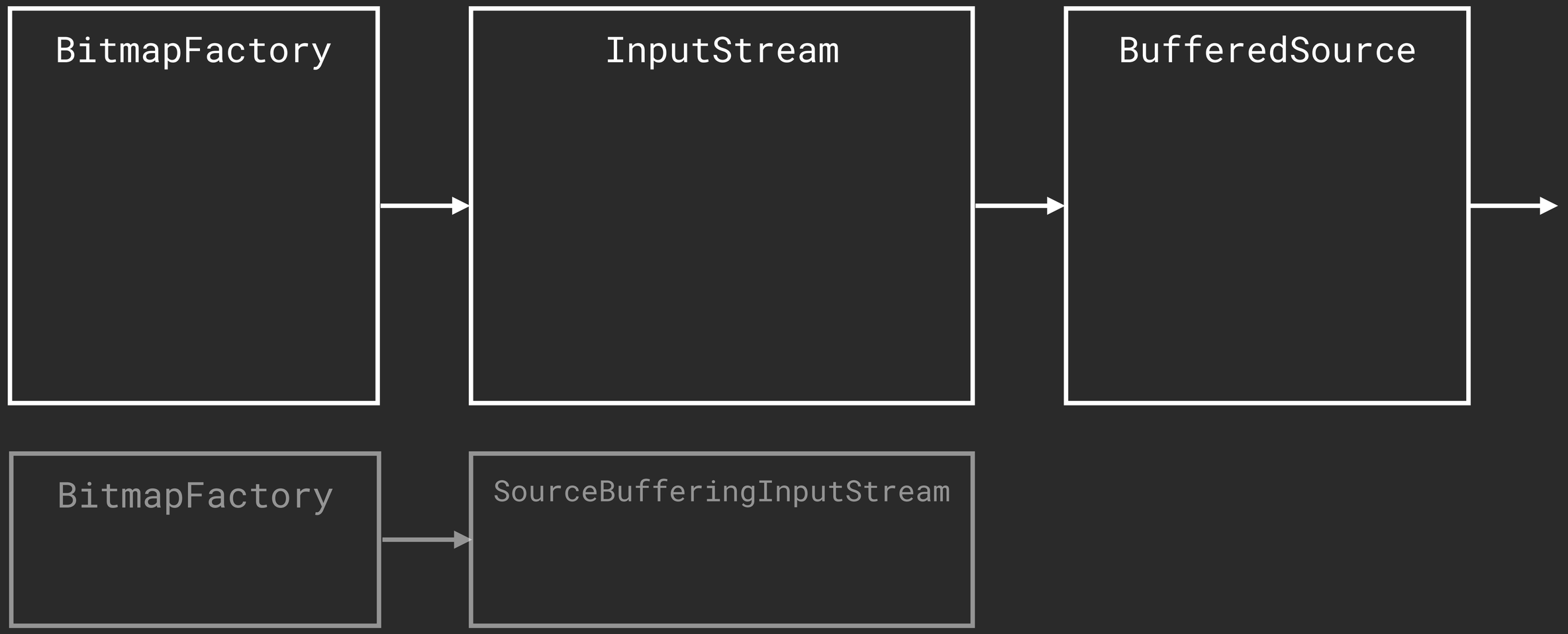

BitmapFactory
InputStream
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
BufferedSource
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
BitmapFactory
SourceBufferingInputStream

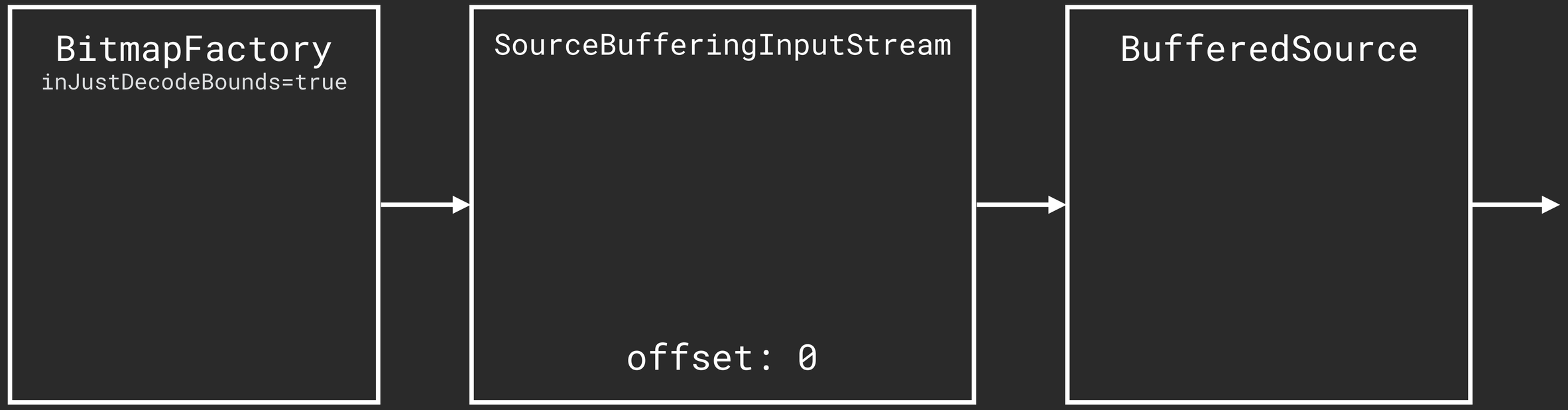

BitmapFactory
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
InputStream
PIXEL DATA
PIXEL DATA
PIXEL DATA
BufferedSource
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
PIXEL DATA
BitmapFactory
SourceBufferingInputStream

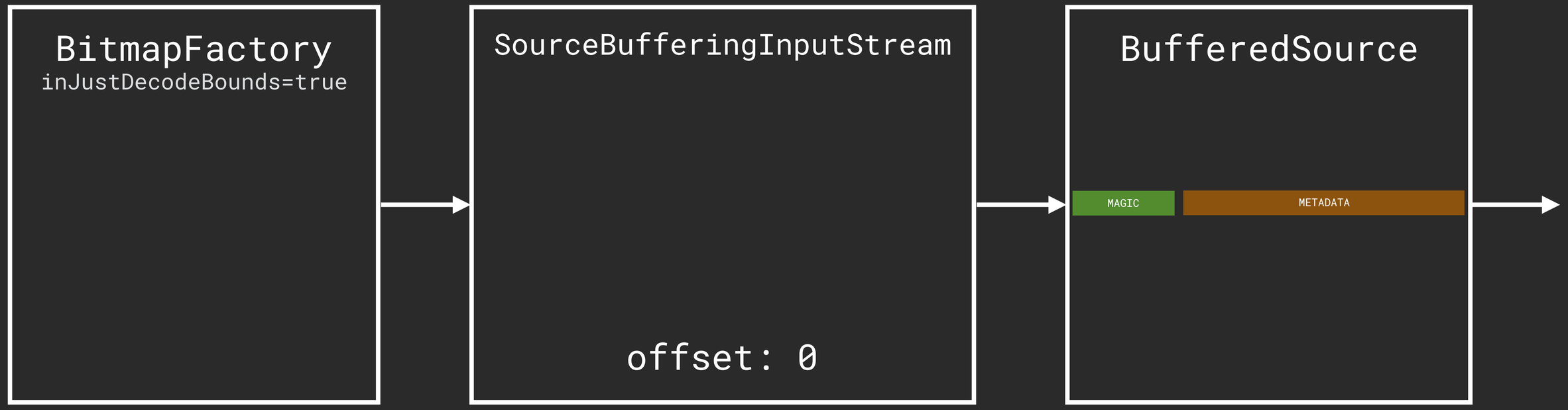

BitmapFactory
PIXEL DATA
PIXEL DATA
PIXEL DATA
InputStream
PIXEL DATA
PIXEL DATA
PIXEL DATA
BufferedSource
PIXEL DATA
PIXEL DATA
BitmapFactory
SourceBufferingInputStream

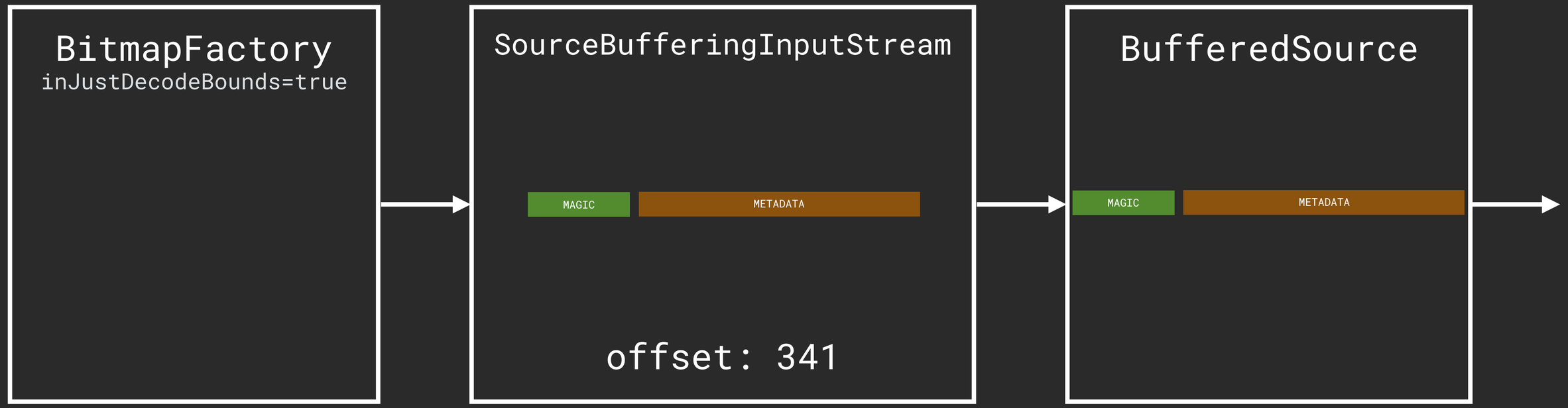

BitmapFactory
PIXEL DATA
PIXEL DATA
PIXEL DATA
InputStream
PIXEL DATA
PIXEL DATA
BufferedSource
BitmapFactory
SourceBufferingInputStream

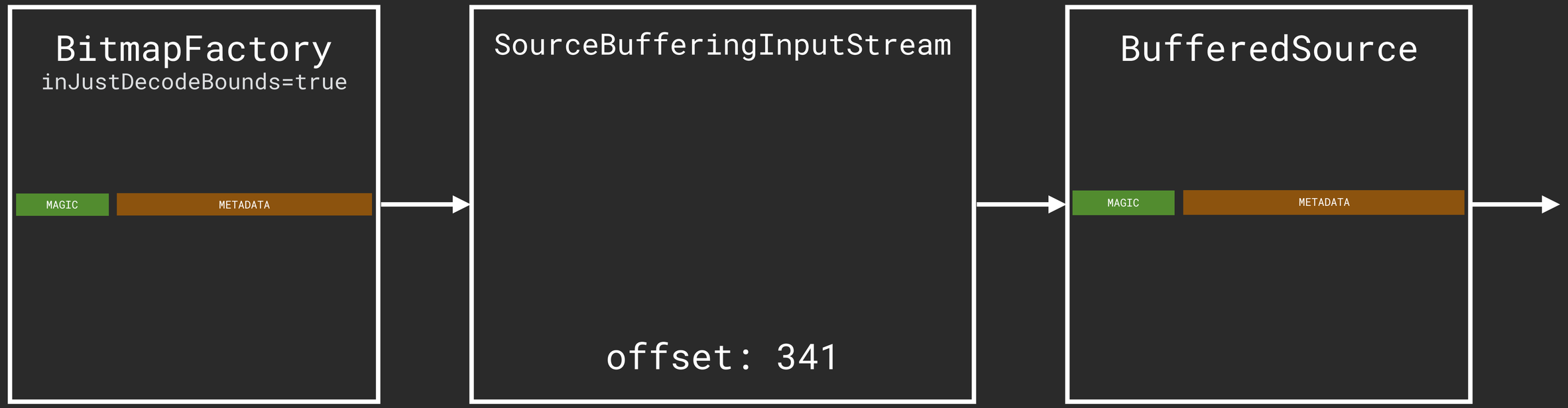

BitmapFactory
PIXEL DATA
PIXEL DATA
InputStream
BufferedSource
BitmapFactory
SourceBufferingInputStream

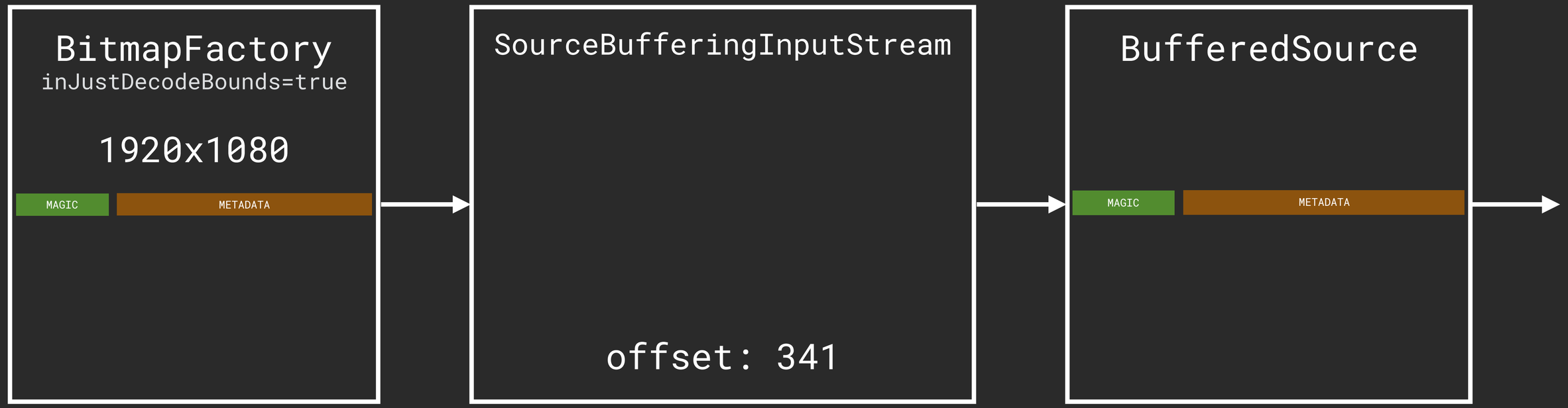

BitmapFactory
InputStream
BufferedSource
BitmapFactory
SourceBufferingInputStream


BitmapFactory
inJustDecodeBounds=true
SourceBufferingInputStream
offset: 0
BufferedSource

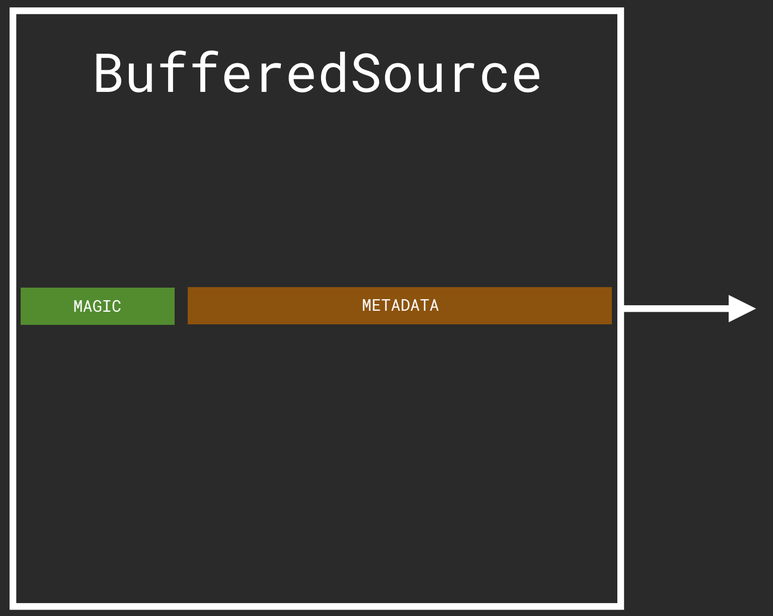

BitmapFactory
inJustDecodeBounds=true
SourceBufferingInputStream
offset: 0
BufferedSource
MAGIC
METADATA

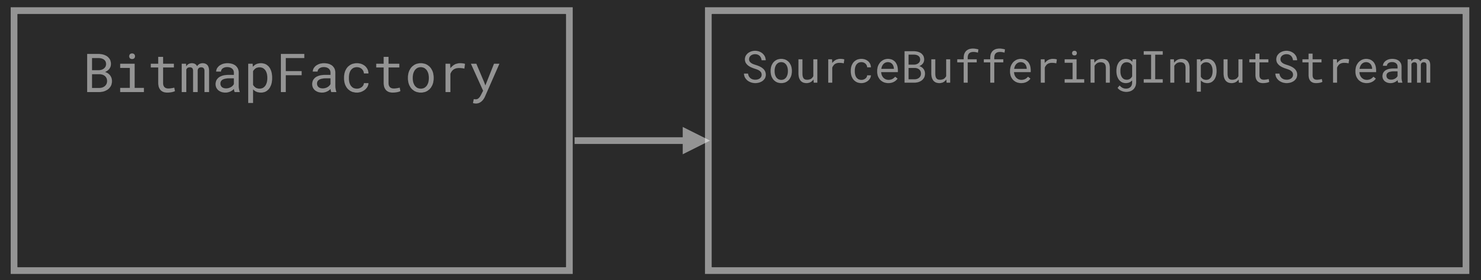

BitmapFactory
inJustDecodeBounds=true
SourceBufferingInputStream
MAGIC
METADATA
offset: 341
BufferedSource
MAGIC
METADATA


BitmapFactory
inJustDecodeBounds=true
MAGIC
METADATA
SourceBufferingInputStream
offset: 341
BufferedSource
MAGIC
METADATA

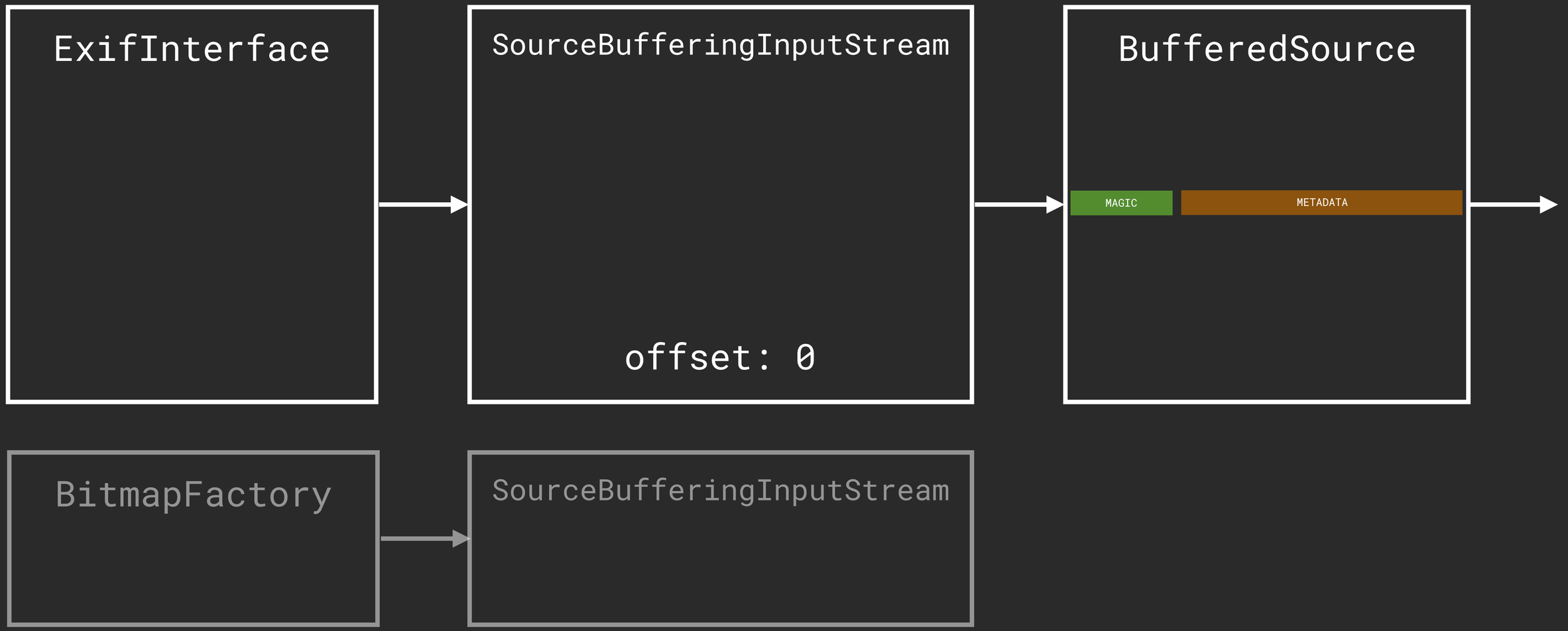

BitmapFactory
inJustDecodeBounds=true
1920x1080
MAGIC
METADATA
SourceBufferingInputStream
offset: 341
BufferedSource
MAGIC
METADATA

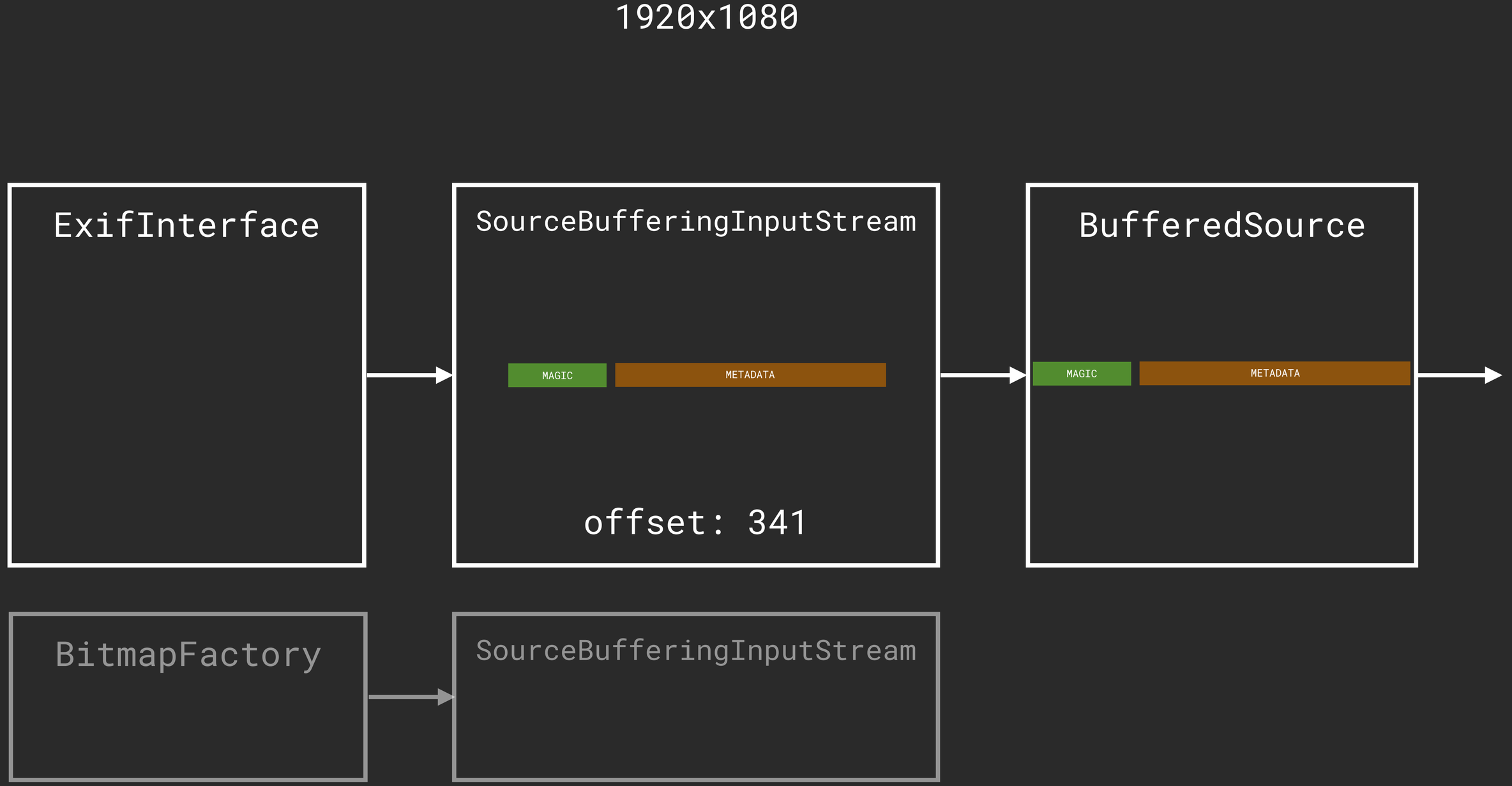
1920x1080

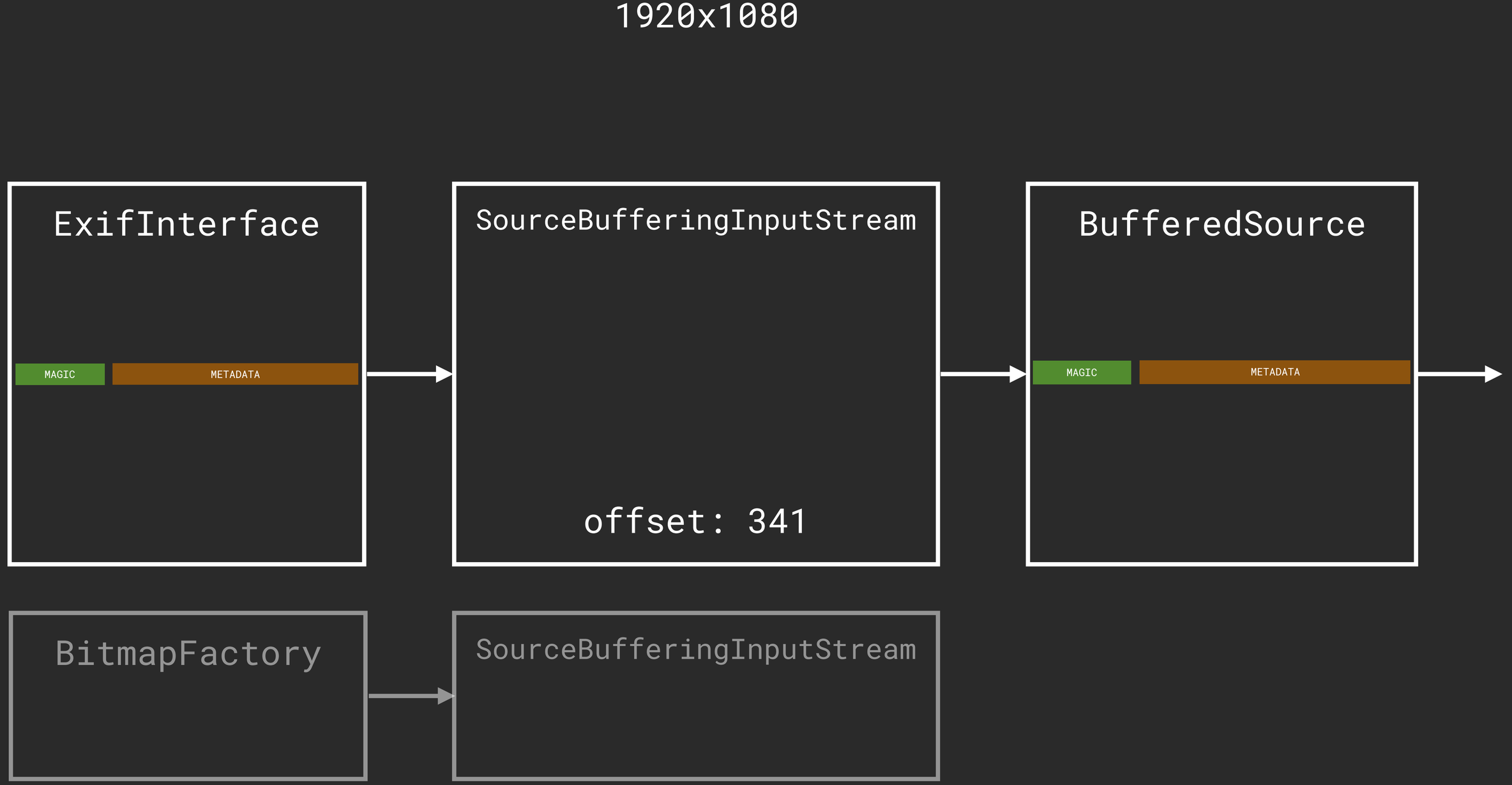

BufferedSource
MAGIC
METADATA


BitmapFactory
SourceBufferingInputStream

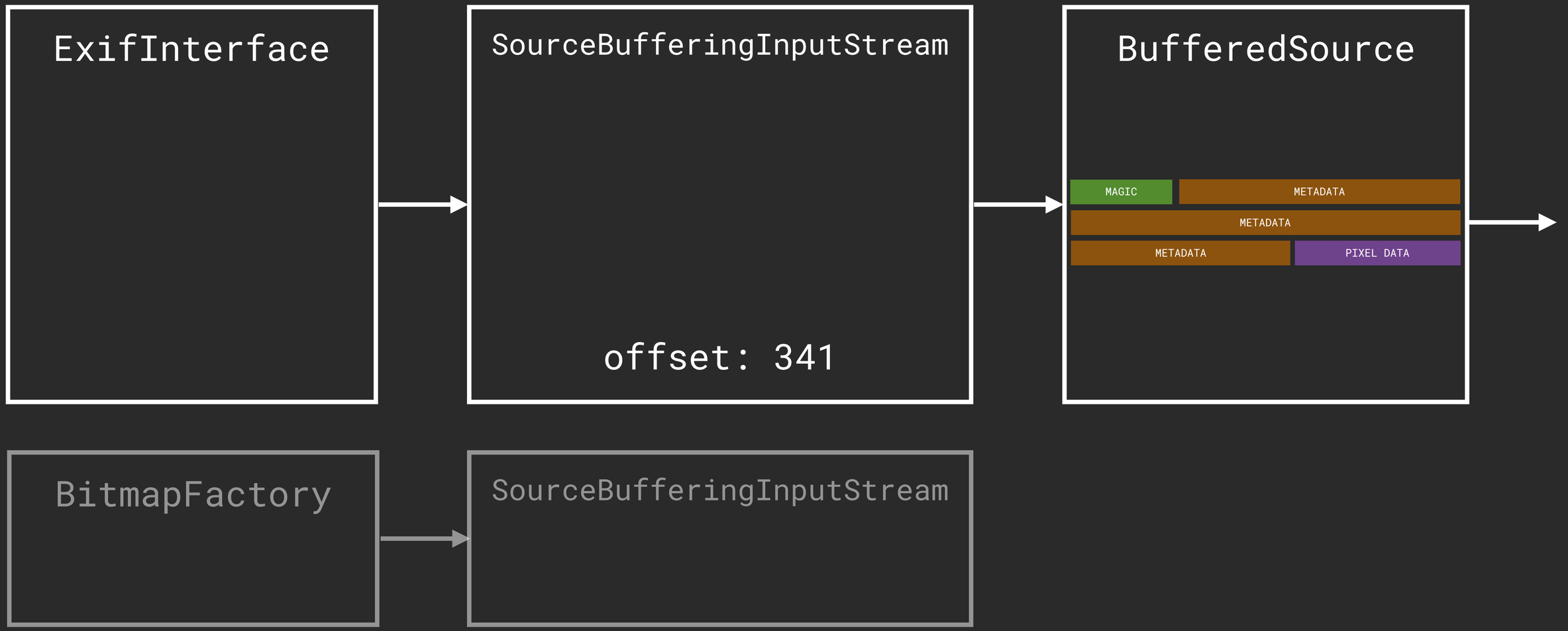
1920x1080

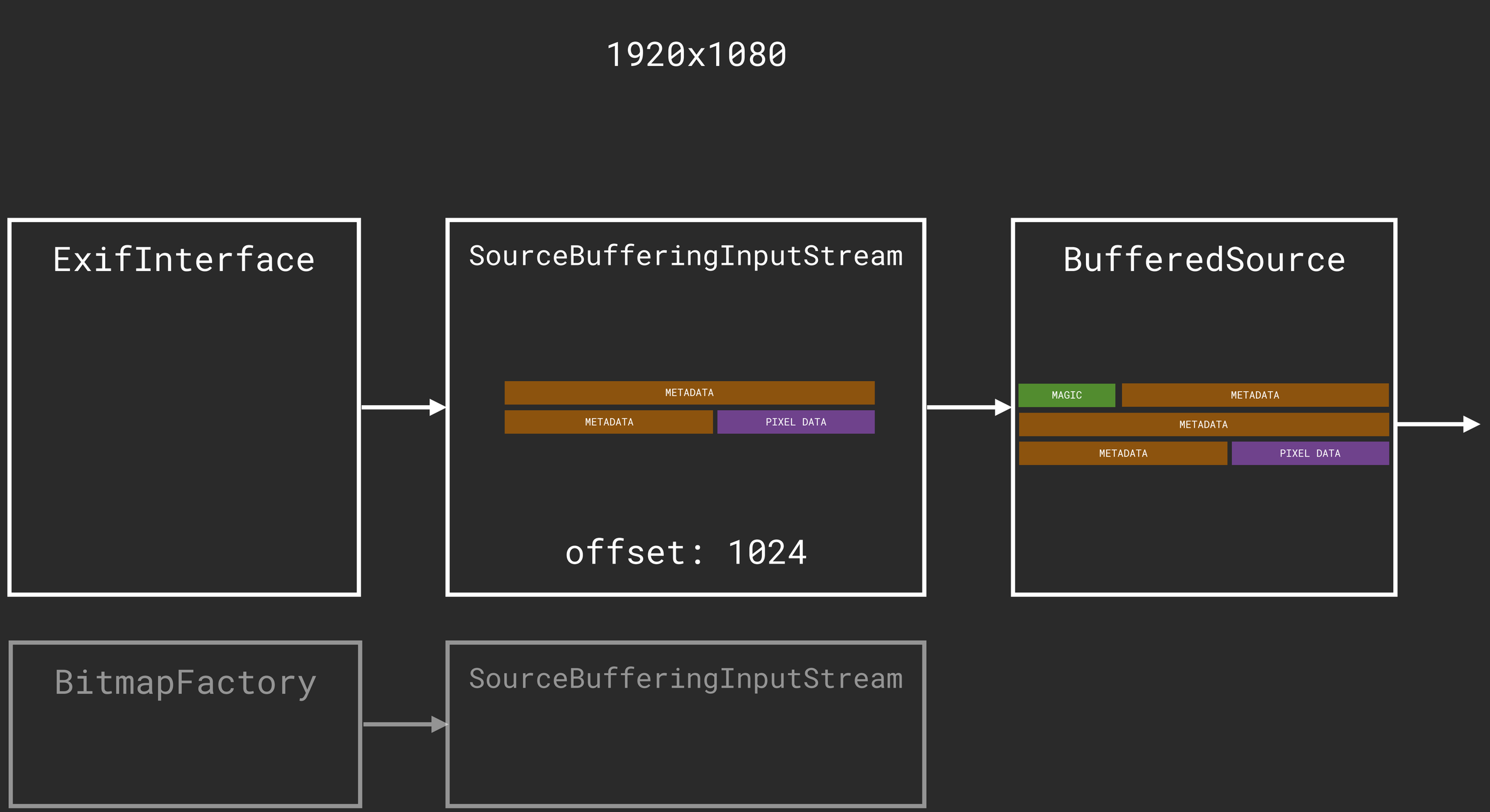

ExifInterface
SourceBufferingInputStream
offset: 0
BufferedSource
MAGIC
METADATA
BitmapFactory
SourceBufferingInputStream

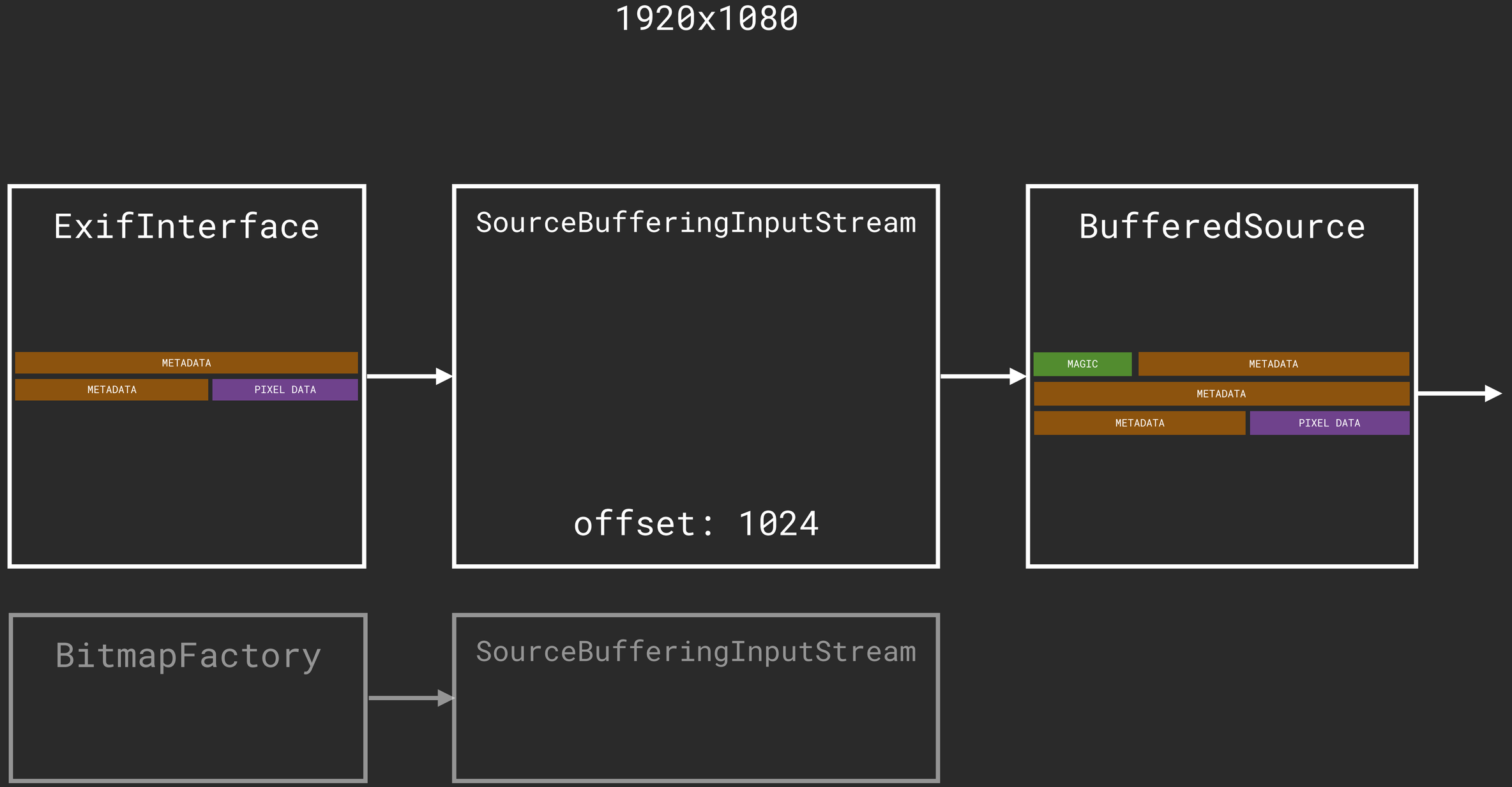

1920x1080
ExifInterface
SourceBufferingInputStream
MAGIC
METADATA
offset: 341
BufferedSource
MAGIC
METADATA
BitmapFactory
SourceBufferingInputStream

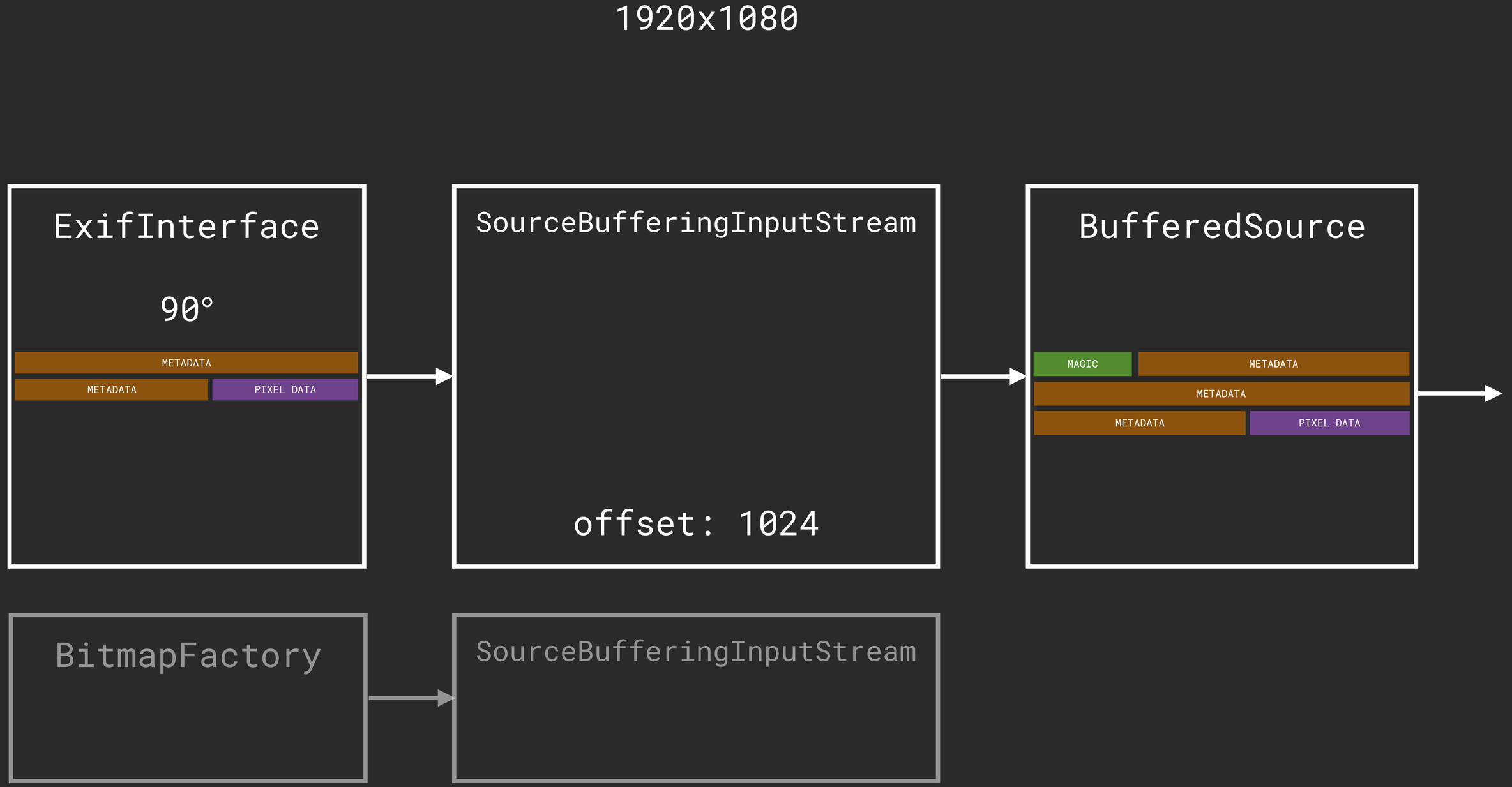

1920x1080
ExifInterface
MAGIC
METADATA
SourceBufferingInputStream
offset: 341
BufferedSource
MAGIC
METADATA
BitmapFactory
SourceBufferingInputStream

1920x1080


ExifInterface
SourceBufferingInputStream
offset: 341
BufferedSource
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
BitmapFactory
SourceBufferingInputStream

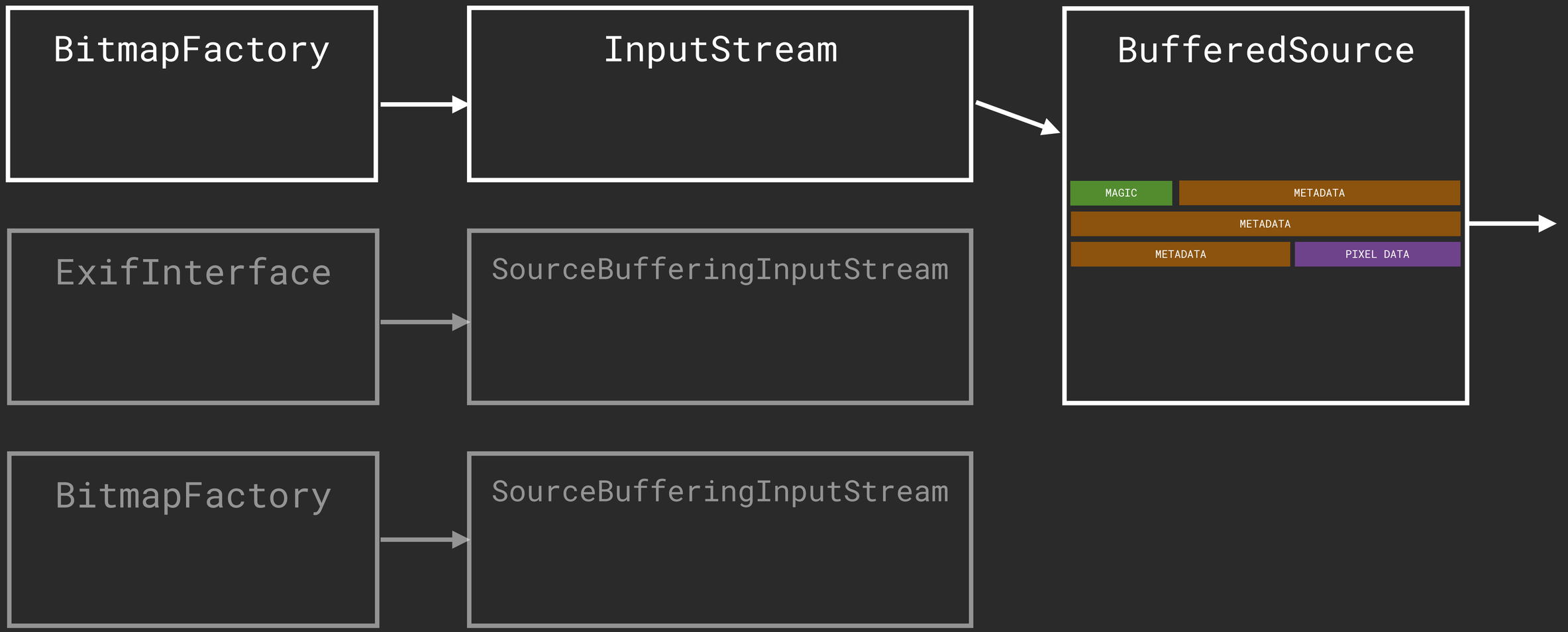

1920x1080
ExifInterface
SourceBufferingInputStream
METADATA
METADATA
PIXEL DATA
offset: 1024
BufferedSource
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
BitmapFactory
SourceBufferingInputStream

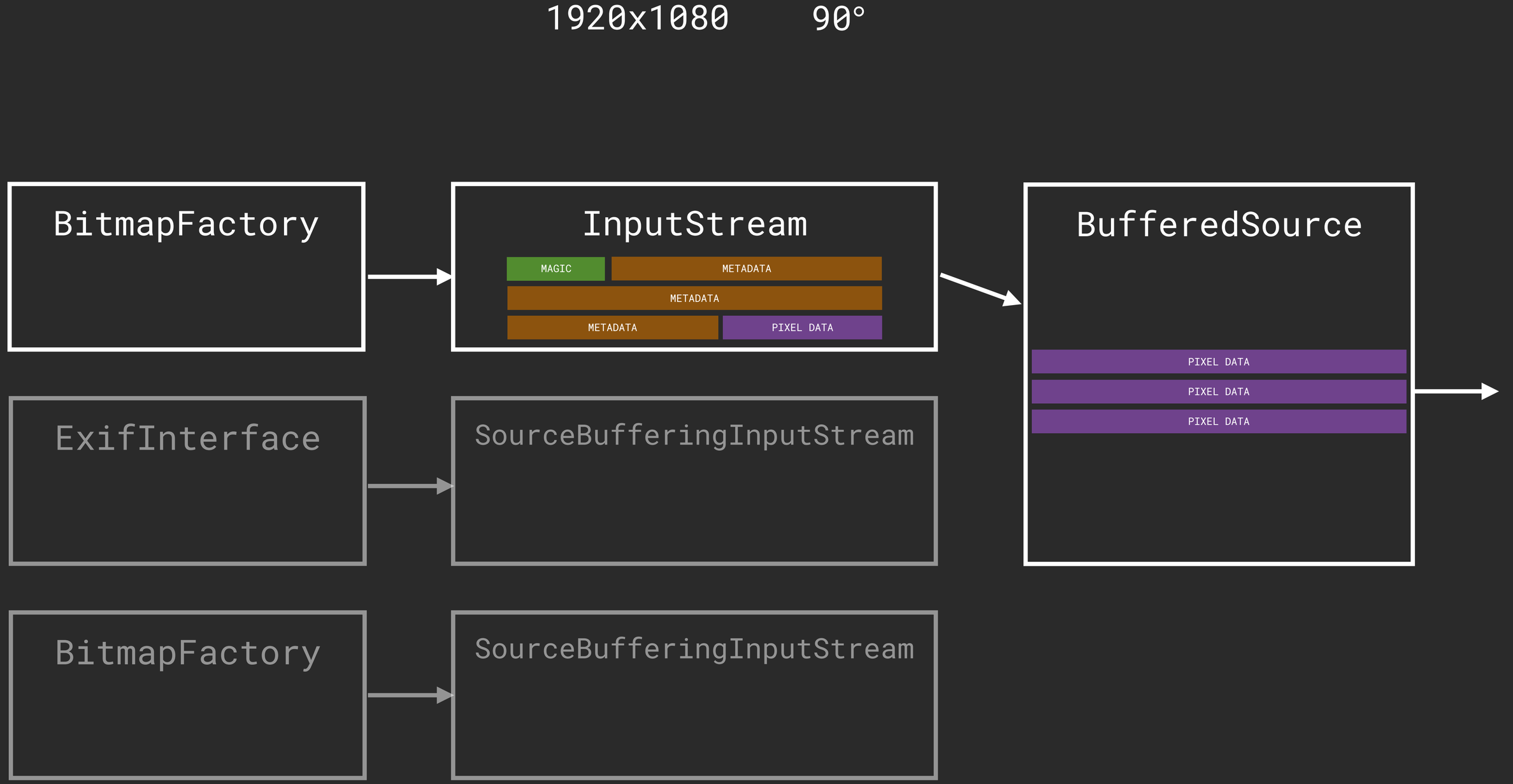

1920x1080
ExifInterface
METADATA
METADATA
PIXEL DATA
SourceBufferingInputStream
offset: 1024
BufferedSource
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
BitmapFactory
SourceBufferingInputStream

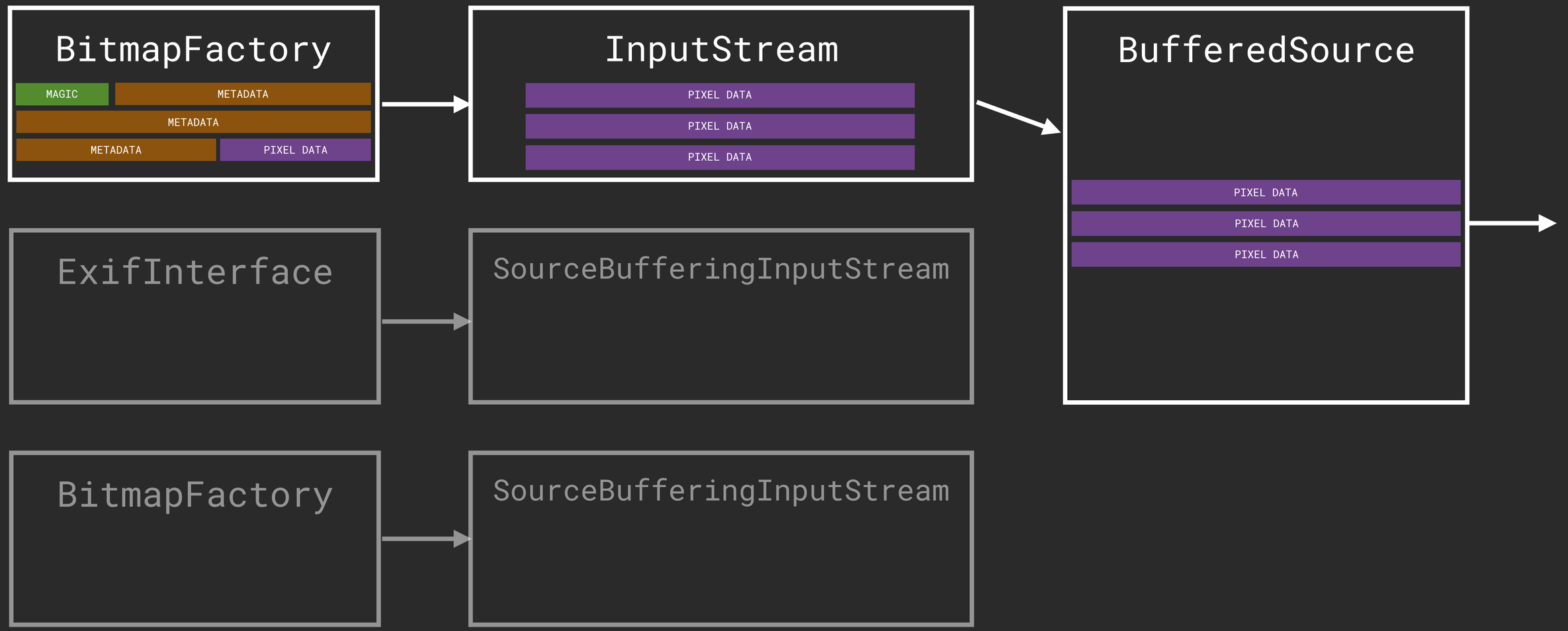

1920x1080
ExifInterface
90°
METADATA
METADATA
PIXEL DATA
SourceBufferingInputStream
offset: 1024
BufferedSource
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
BitmapFactory
SourceBufferingInputStream

1920x1080 90°

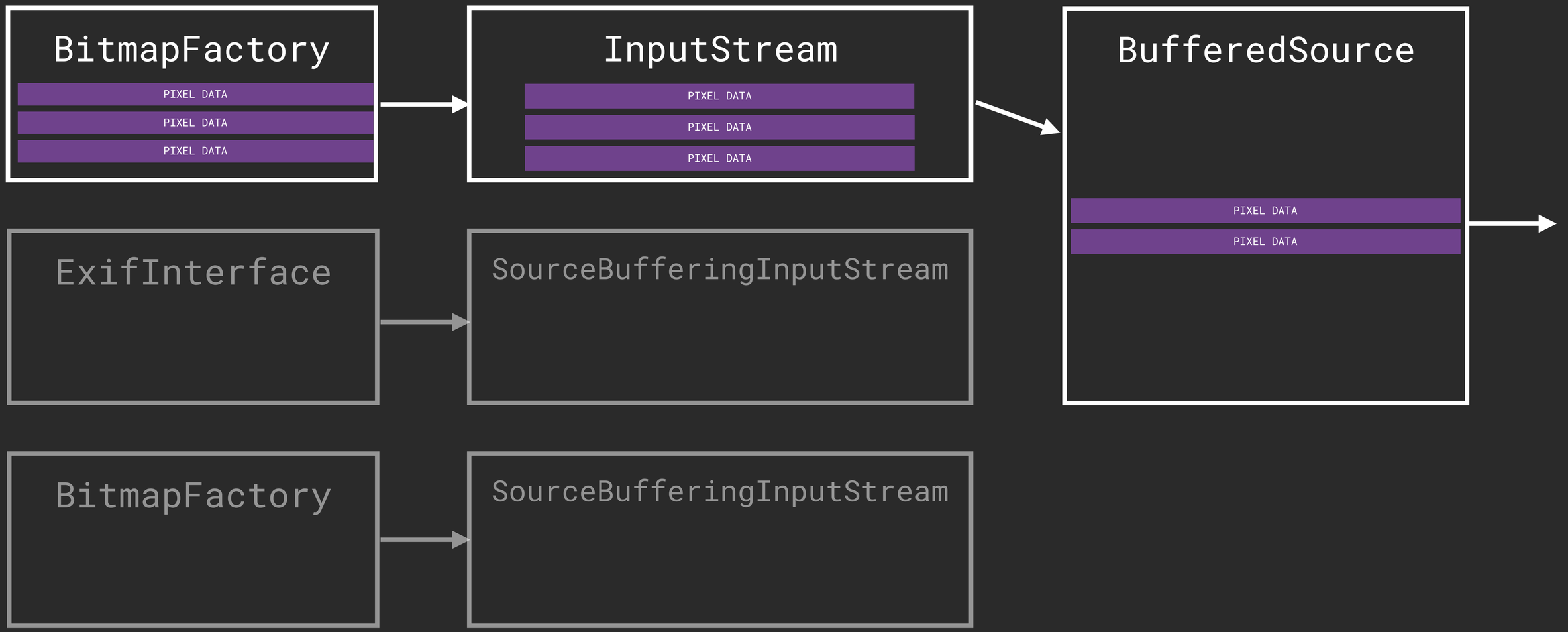
1920x1080 90°



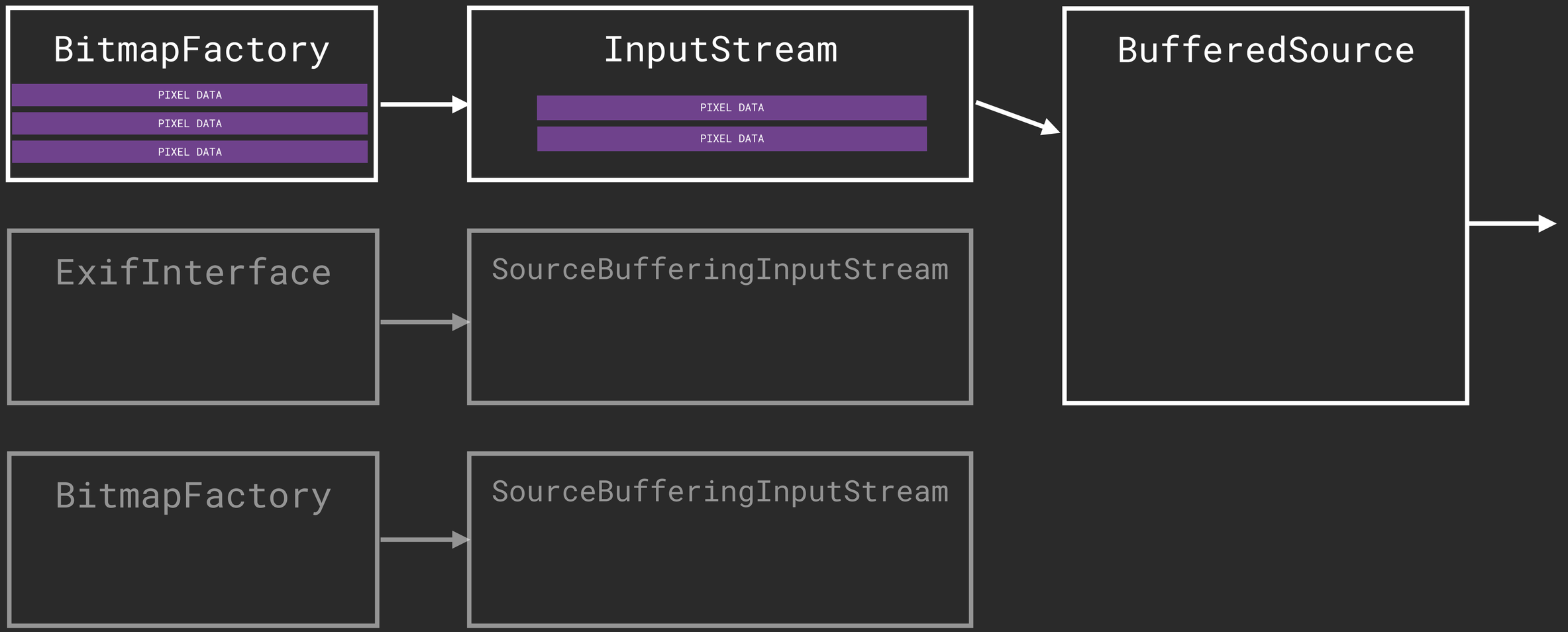

1920x1080
90°
BitmapFactory
InputStream
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
BufferedSource
PIXEL DATA
PIXEL DATA
PIXEL DATA
ExifInterface
SourceBufferingInputStream
BitmapFactory
SourceBufferingInputStream


1920x1080
90°
BitmapFactory
MAGIC
METADATA
METADATA
METADATA
PIXEL DATA
InputStream
PIXEL DATA
PIXEL DATA
PIXEL DATA
BufferedSource
PIXEL DATA
PIXEL DATA
PIXEL DATA
ExifInterface
SourceBufferingInputStream
BitmapFactory
SourceBufferingInputStream

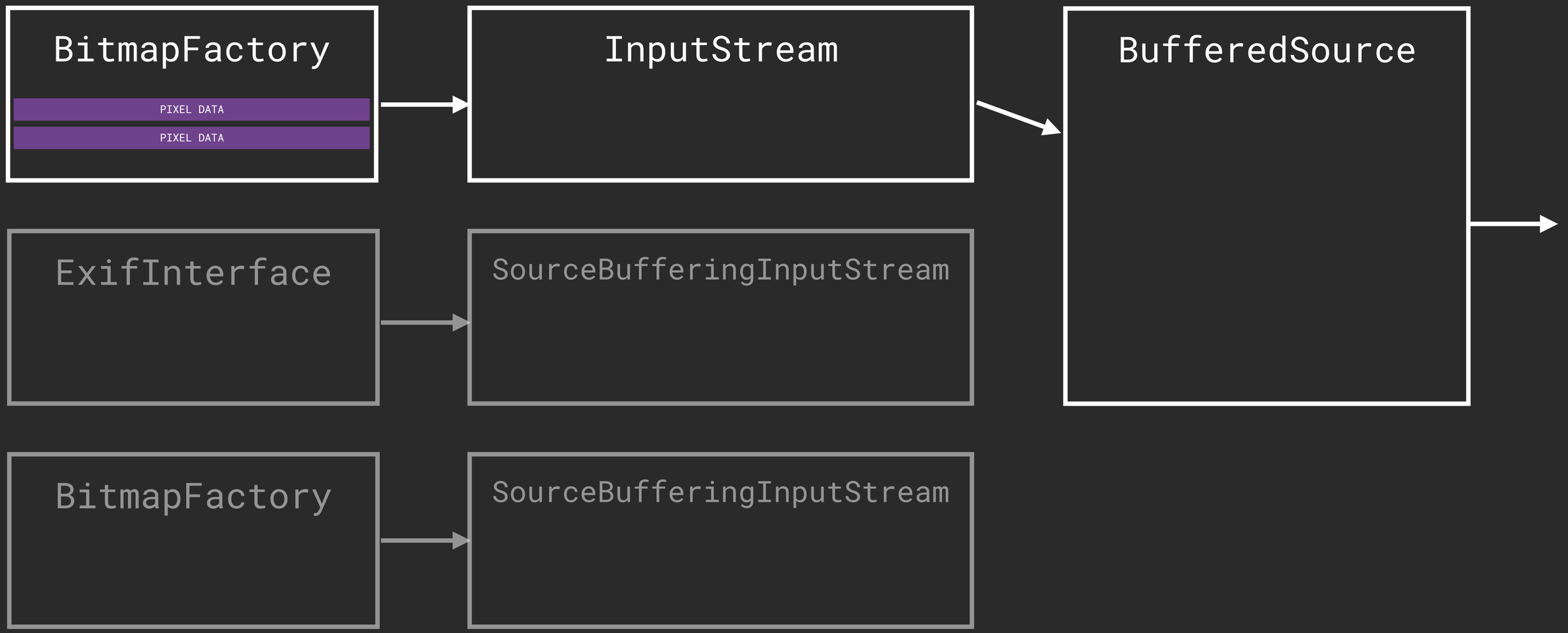
1920x1080 90°

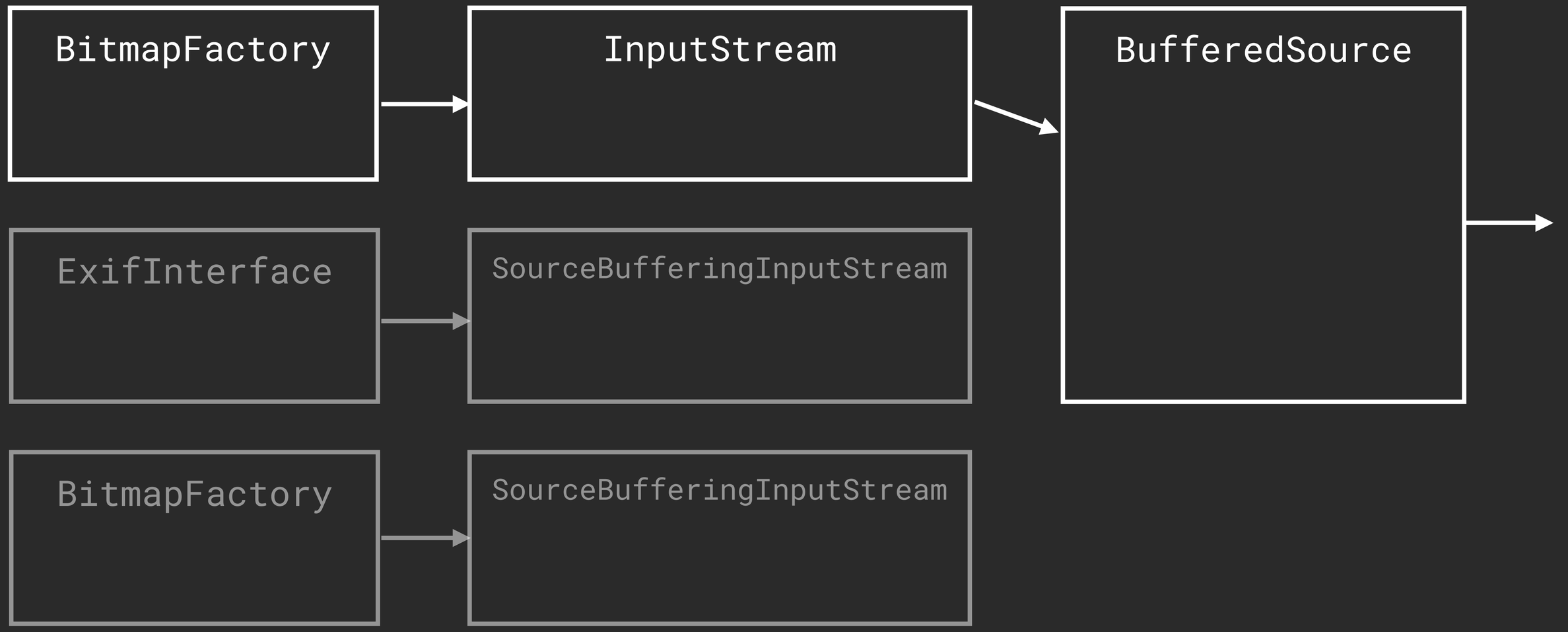


1920x1080 90°

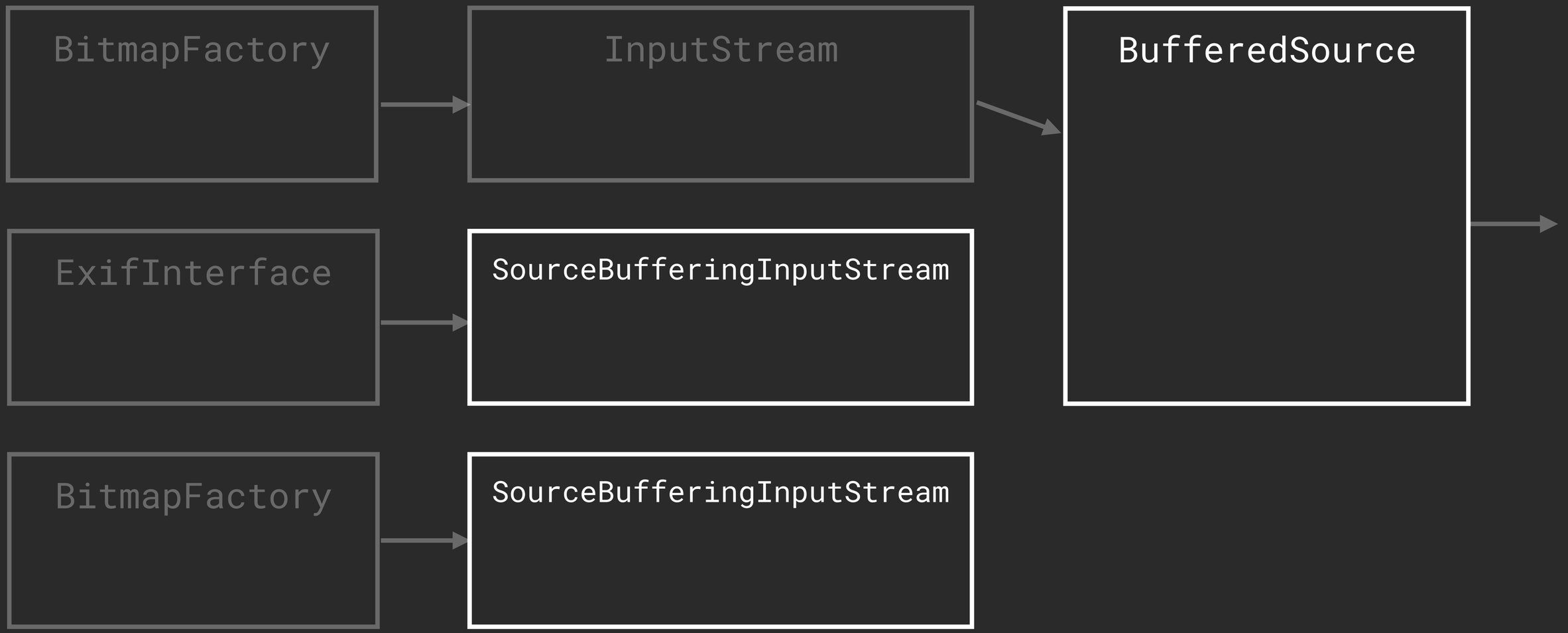


1920x1080 90°




1920x1080
90°
BitmapFactory
InputStream
BufferedSource
ExifInterface
SourceBufferingInputStream
BitmapFactory
SourceBufferingInputStream

1920x1080 90°



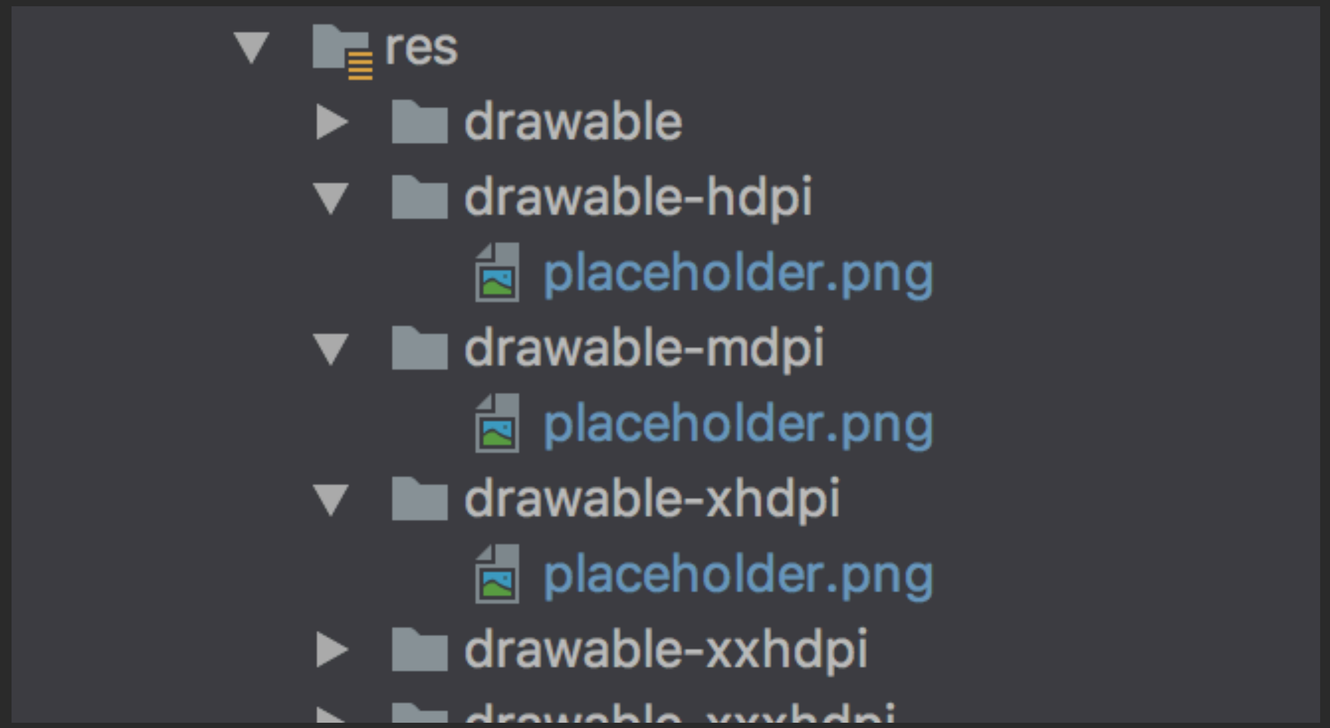
3.x Goals

- Android P
- OkHttp 2.x => 3.x
- Okio integration
- Improvements

Picasso

```
picasso.load(url)
    .placeholder(R.drawable.placeholder)
    .error(R.drawable.error)
    .fit()
    .into(view);
```

Picasso


res
drawable
drawable-hdpi
placeholder.png
drawable-mdpi
placeholder.png
drawable-xhdpi
placeholder.png
drawable-xxhdpi

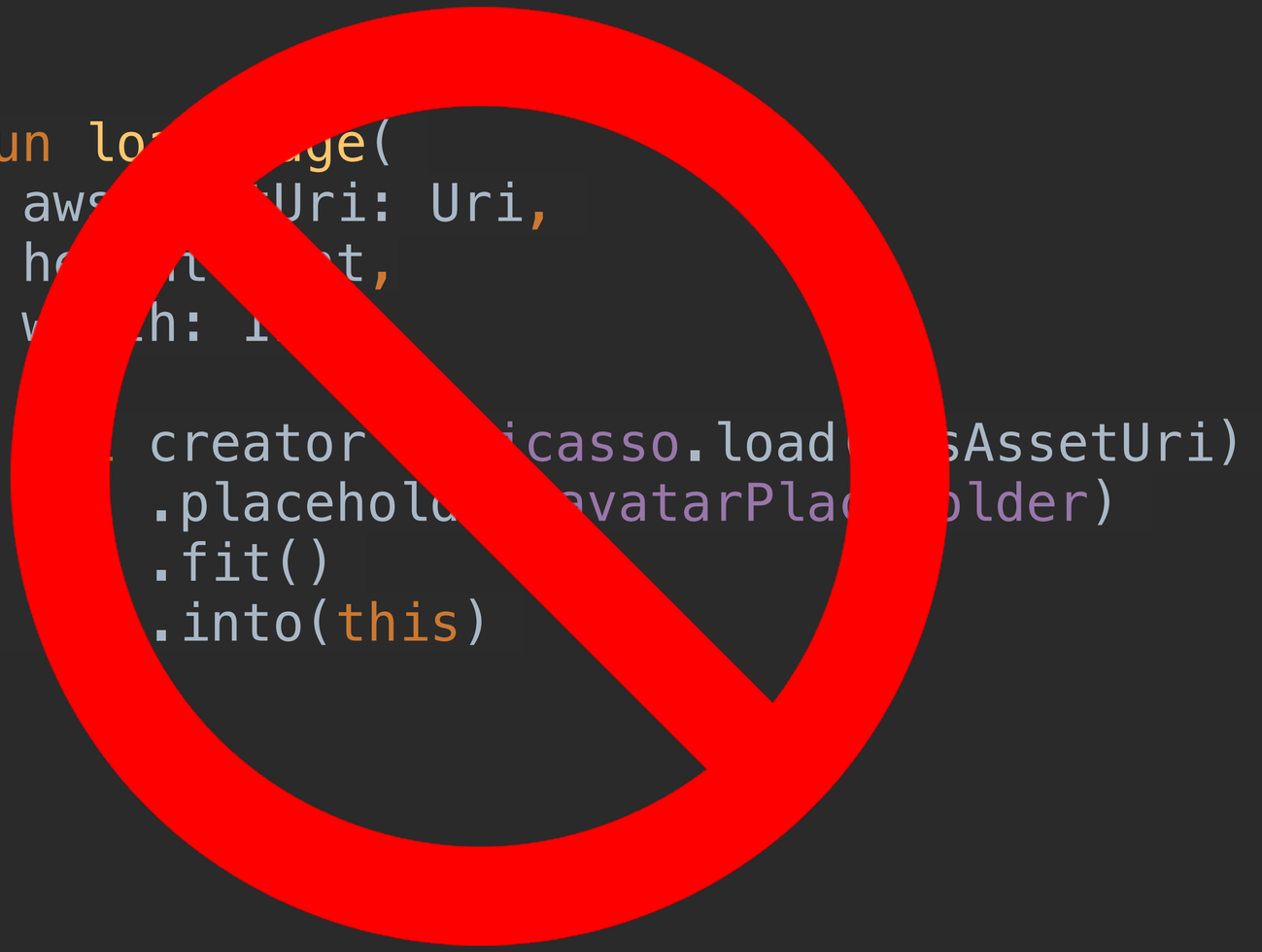
Picasso

```
fun loadImage(
  remoteUri: Uri,
  @ColorInt tintColor: Int? = null
) {
  val creator = picasso.load(remoteUri)
      .placeholder(avatarPlaceholder)
      .fit()

  if (transformTintColor != null) {
    creator.transform(TintTransformation(tintColor))
  }

  creator.into(this)
}
```

Picasso

```
fun loadImage(
  awsAssetUri: Uri,
  height: Int,
  width: Int
) {
  val creator = picasso.load(awsAssetUri)
      .placeholder(avatarPlaceholder)
      .fit()
      .into(this)
}
```

Picasso


fun lo ge(
aws Uri: Uri,
he t,
w h: I
)
creator icasso.load(sAssetUri)
.placehol avatarPla older)
.fit()
.into(this)
}

README.mkd

Survey

If you use thumbor, please take 1 minute and answer this survey? It's only 2 questions and one is multiple choice!!!

thumbor.

thumbor is a smart imaging service. It enables on-demand crop, resizing and flipping of images.

It also features a VERY smart detection of important points in the image for better cropping and resizing, using state-of-the-art face and feature detection algorithms (more on that in Detection Algorithms).

Using thumbor is very easy (after it is running). All you have to do is access it using an URL for an image, like this:

Picasso

```
picasso.load(“http://shrek.jpg”)
    .placeholder(R.drawable.placeholder)
    .error(R.drawable.error)
    .fit()
    .into(view);
```

Picasso

“http://shrek.jpg”

Request

Picasso

“http://shrek.jpg”

Request

Request Transformer

Picasso

“http://shrek.jpg”

Request

Request Transformer

“http://bar.com/20x20/shrek.jpg”

Request

http://shrek.jpg

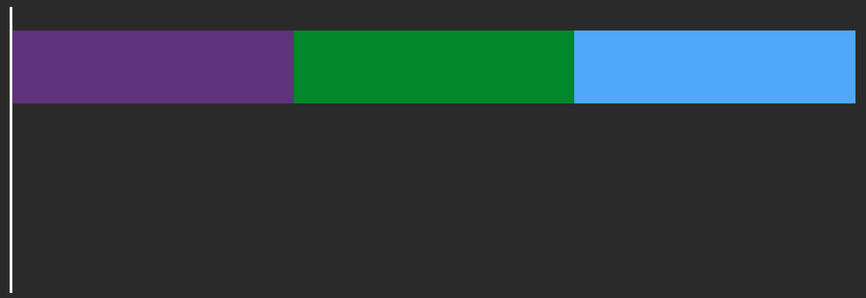
http://shrek.jpg

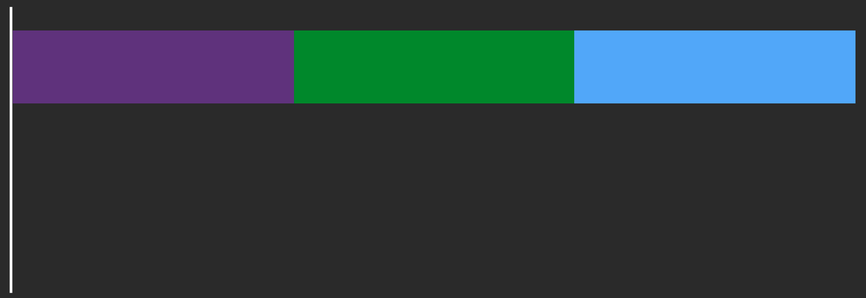
http://shrek.jpg

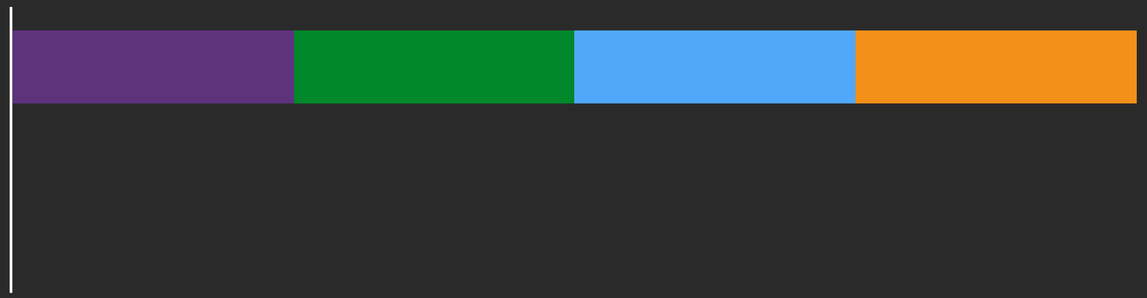
0101011010100110 1

0101011010100 1101

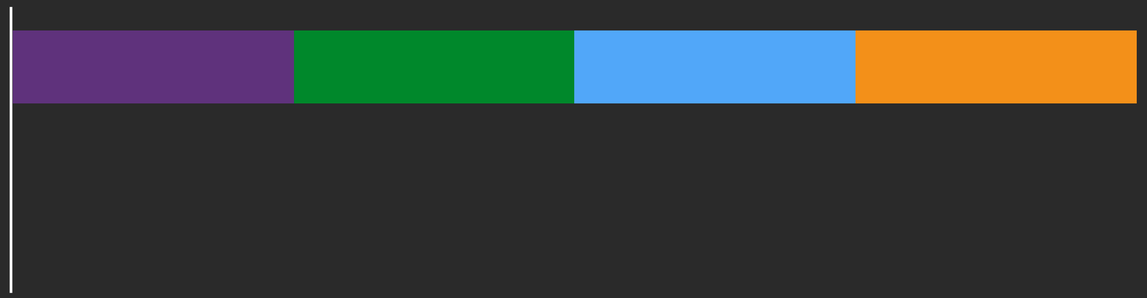
0101011010100 1101



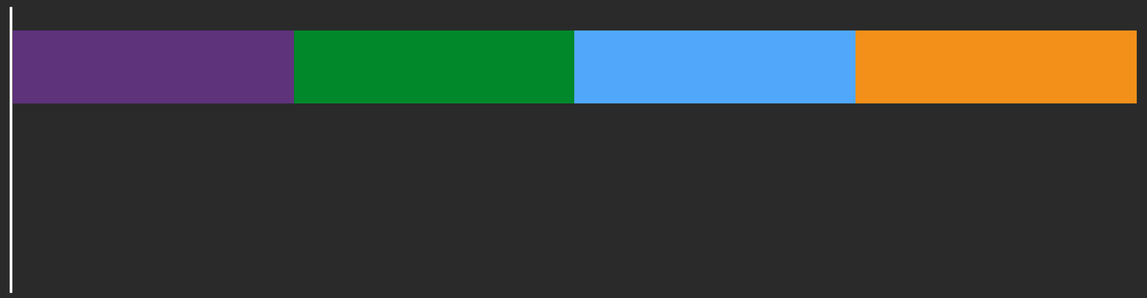




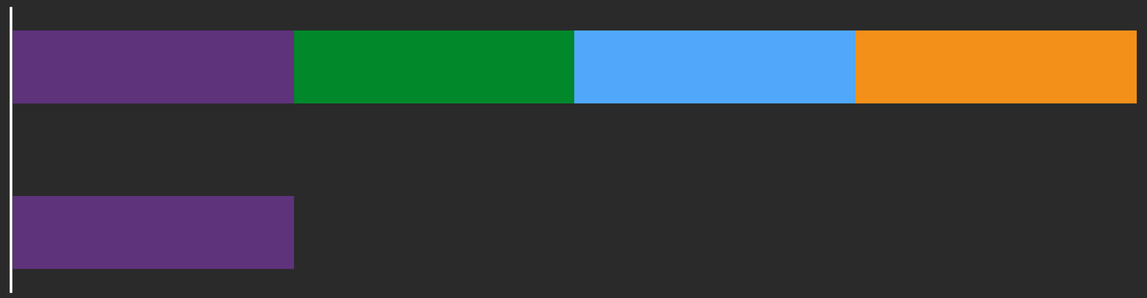
http://bar.com/20x20/shrek.jpg



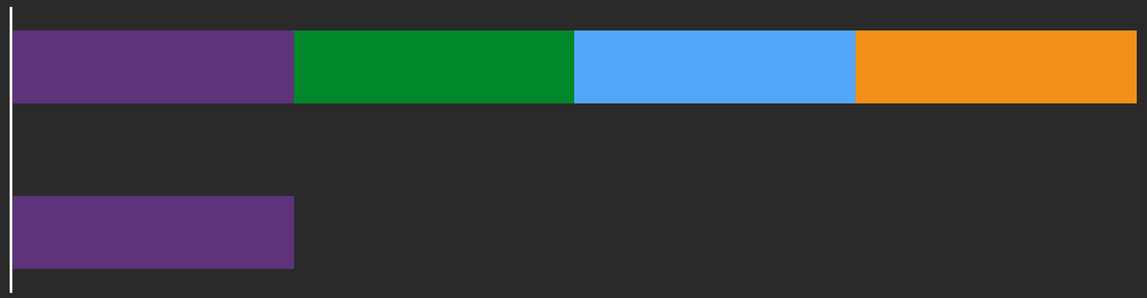
http://bar.com/20x20/shrek.jpg



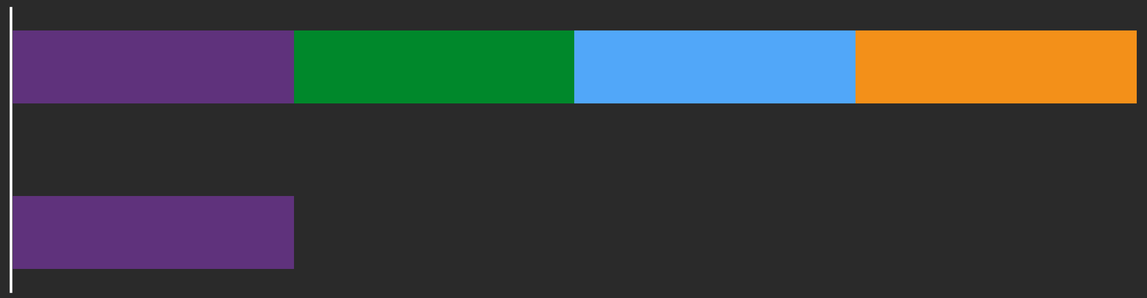
http://bar.com/20x20/shrek.jpg



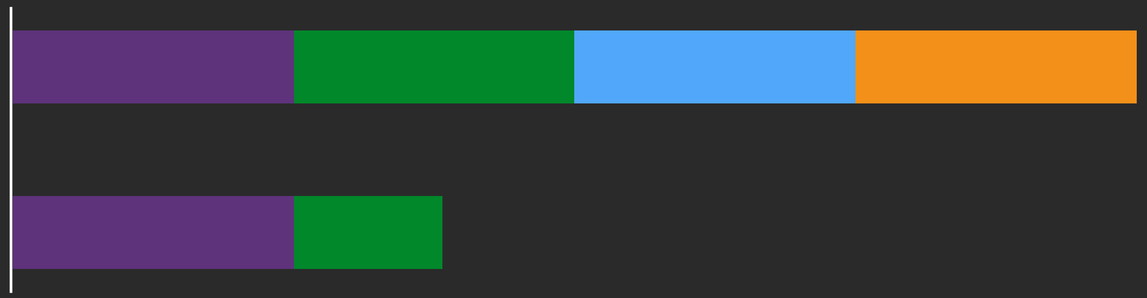
0101011



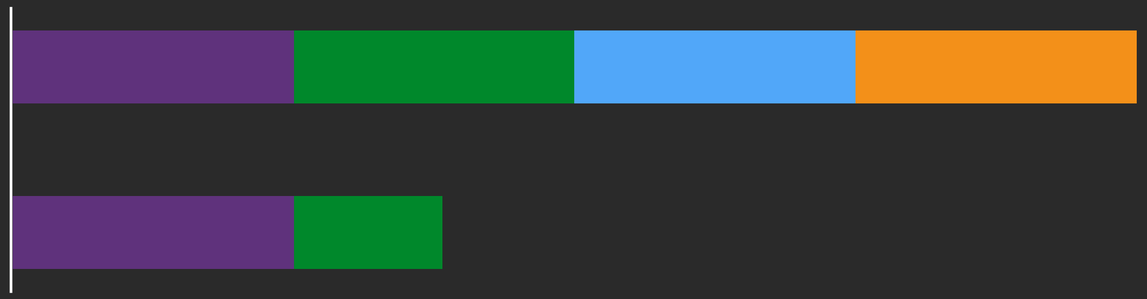
0101011



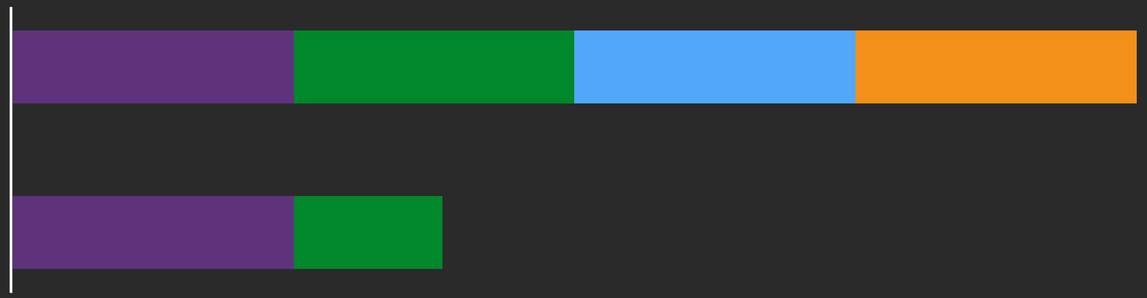
0101011

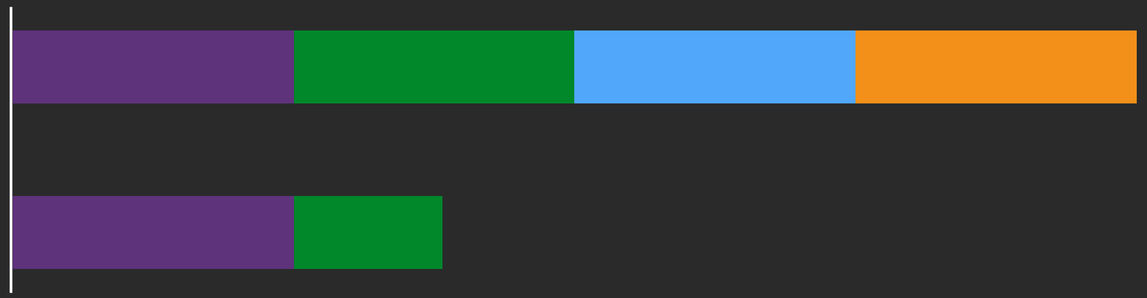


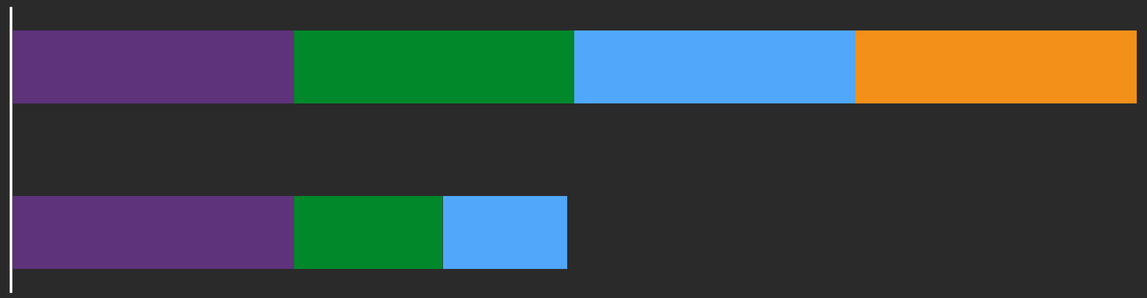
0101011

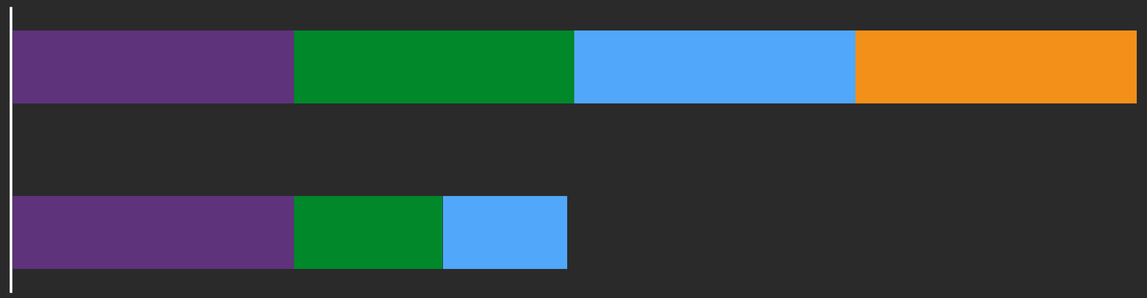


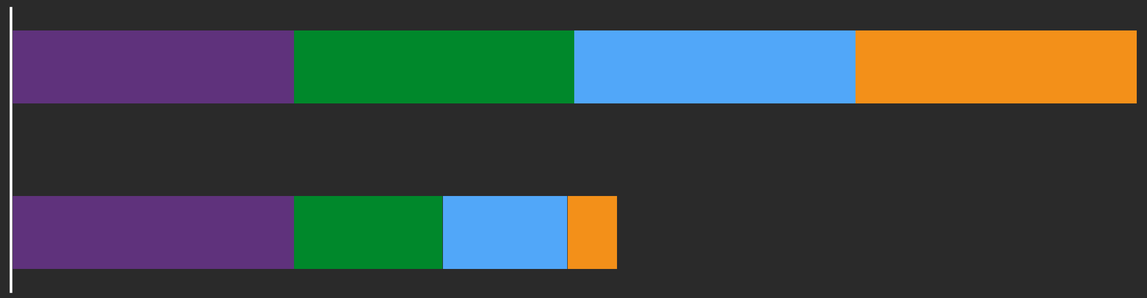
0101011

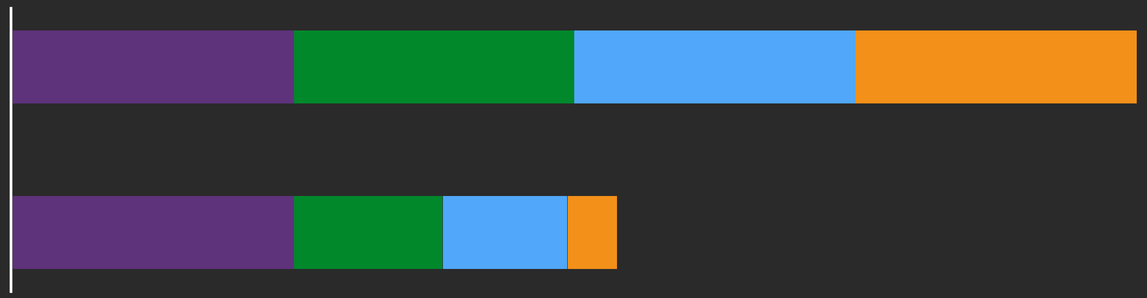


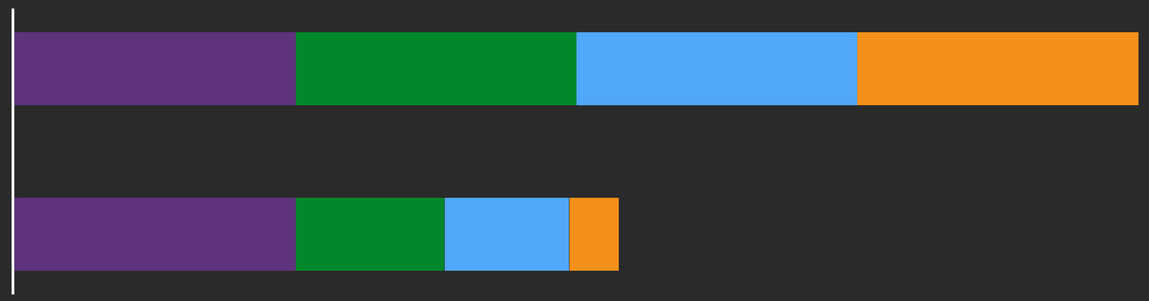


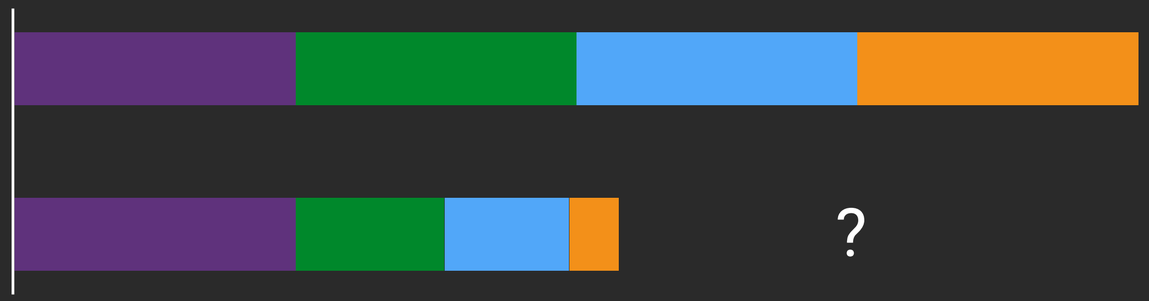












square / picasso

Watch 973 | Star 15,277 | Fork 3,812

Code | Issues 129 | Pull requests 17 | Insights

Branch: master | picasso / picasso-pollexor /

Create new file | Find file | History

runningcode and JakeWharton Add a flag to always transform remote images with thumbor when set. (#... | Latest commit f3c6695 28 days ago

..		
src	Add a flag to always transform remote images with thumbor when set. (#...	28 days ago
README.md	Change singleton getter to Picasso.get	4 months ago
build.gradle	More cleanup	a month ago
gradle.properties	Switch to gradle and AAR	2 years ago
lint.xml	Use OkHttp 3 as the sole shipping Downloader.	a year ago

README.md

Picasso Pollexor Request Transformer

A request transformer which uses a remote Thumbor install to perform image transformation on the server.

Picasso

“http://shrek.jpg”

Request

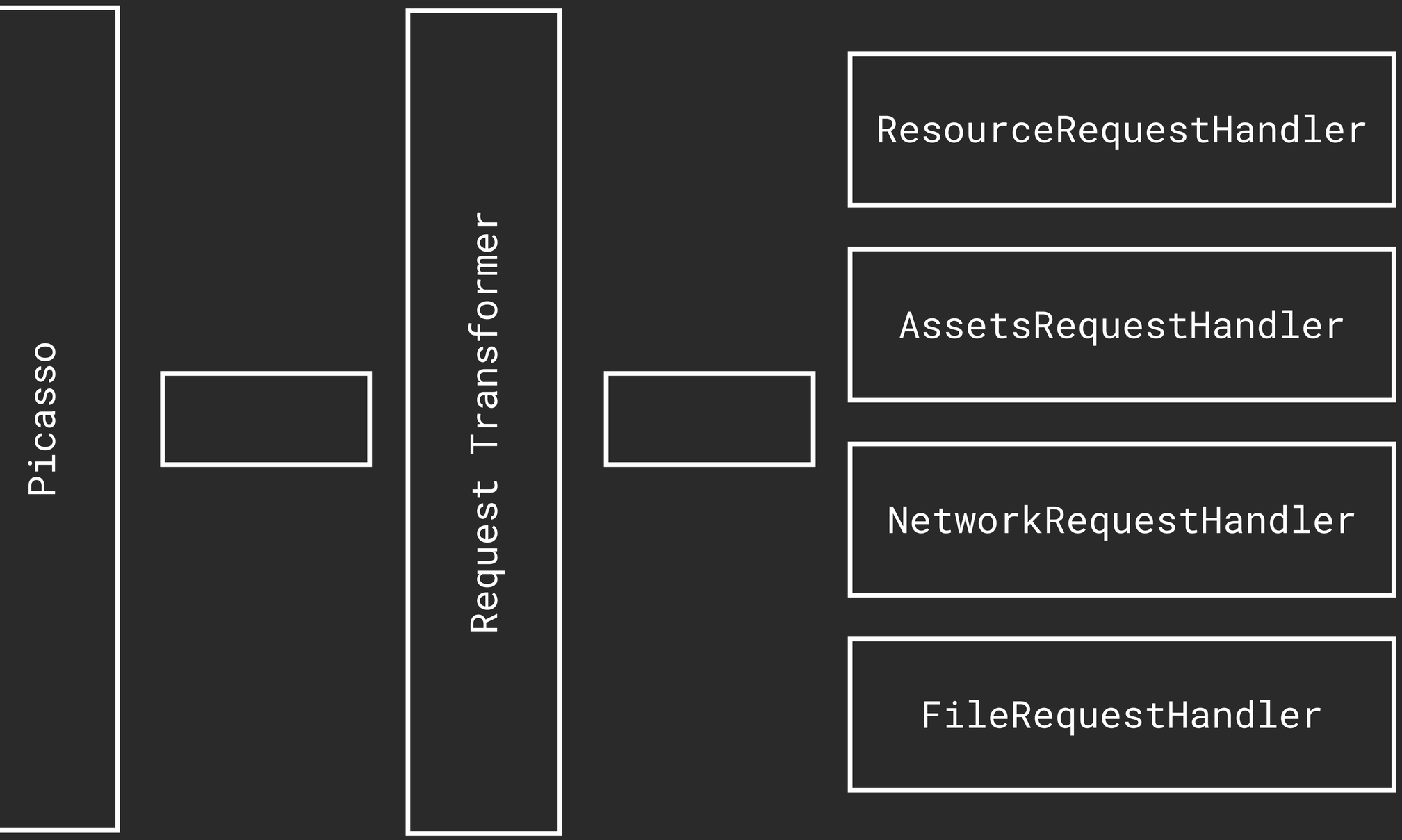
Request Transformer

“http://bar.com/20x20/shrek.jpg”

Request

Picasso

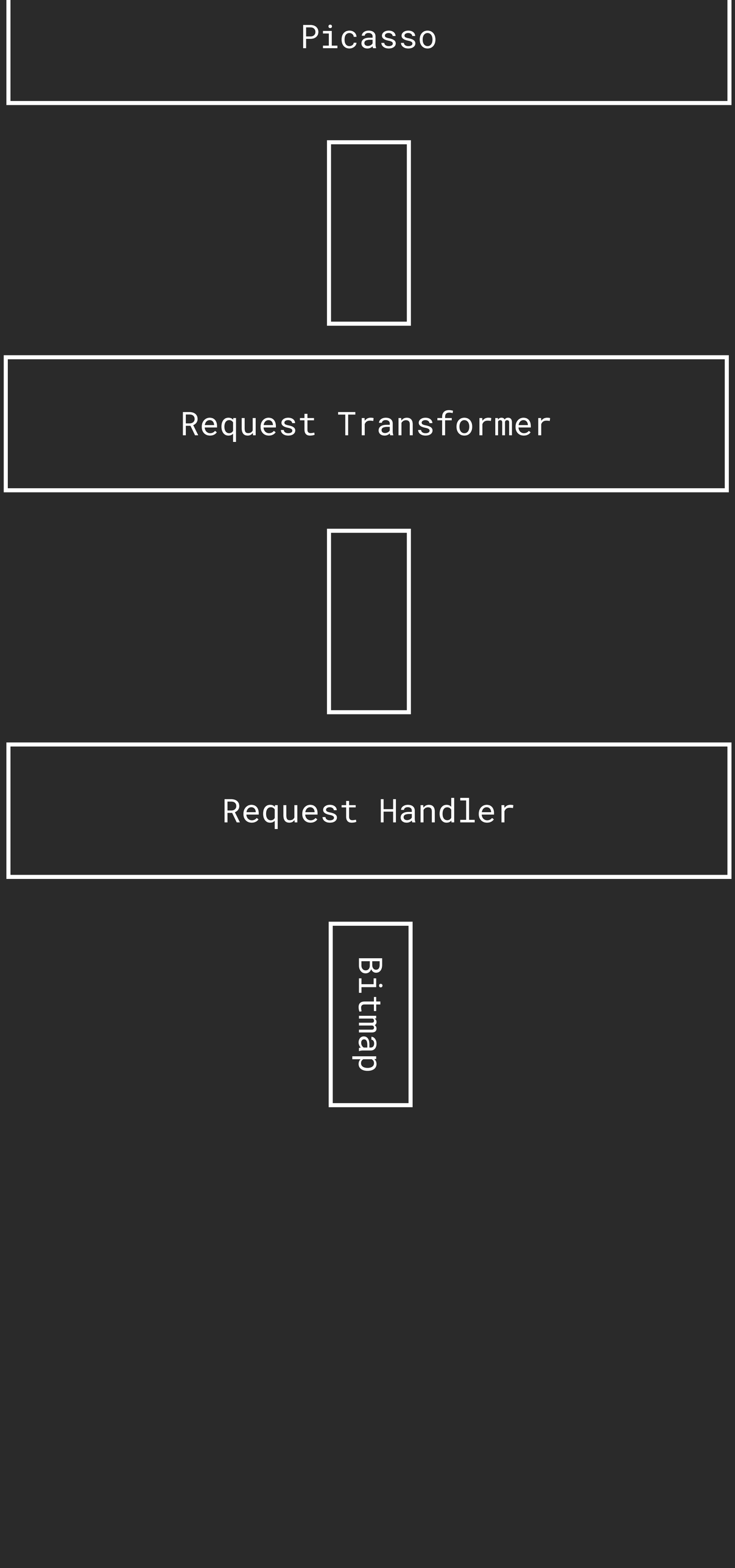
Request Transformer

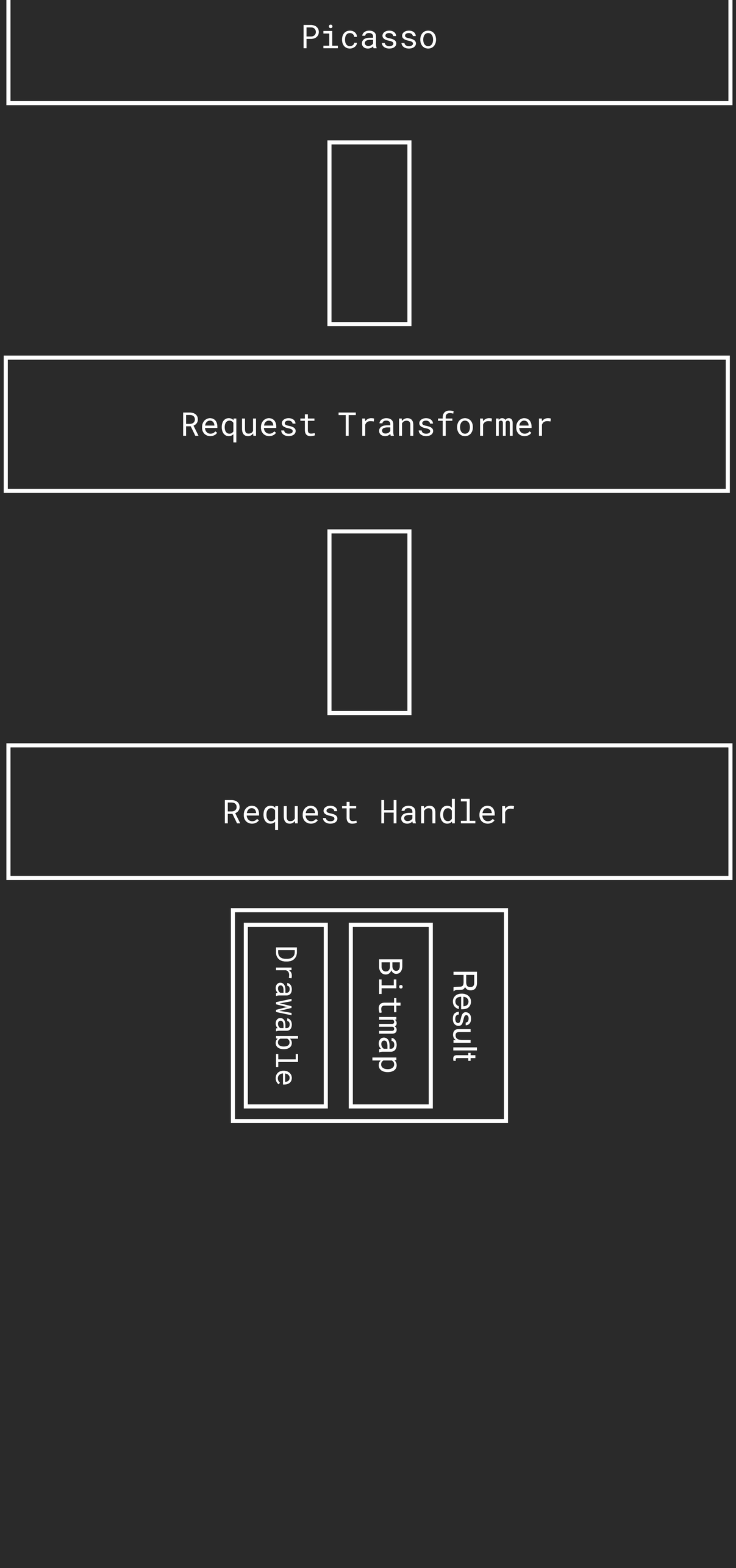

Picasso
Request Transformer
ResourceRequestHandler
AssetsRequestHandler
NetworkRequestHandler
FileRequestHandler

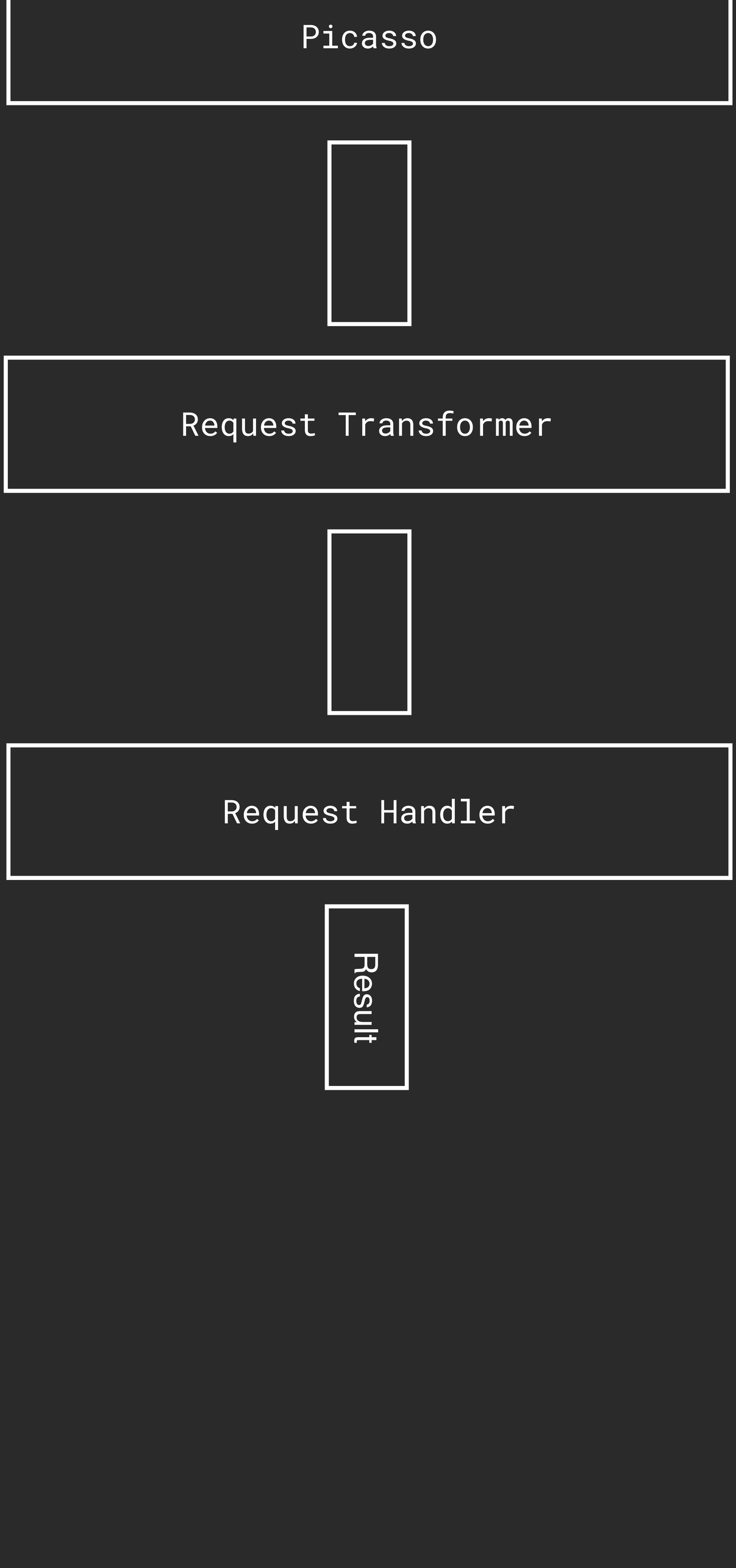
Picasso

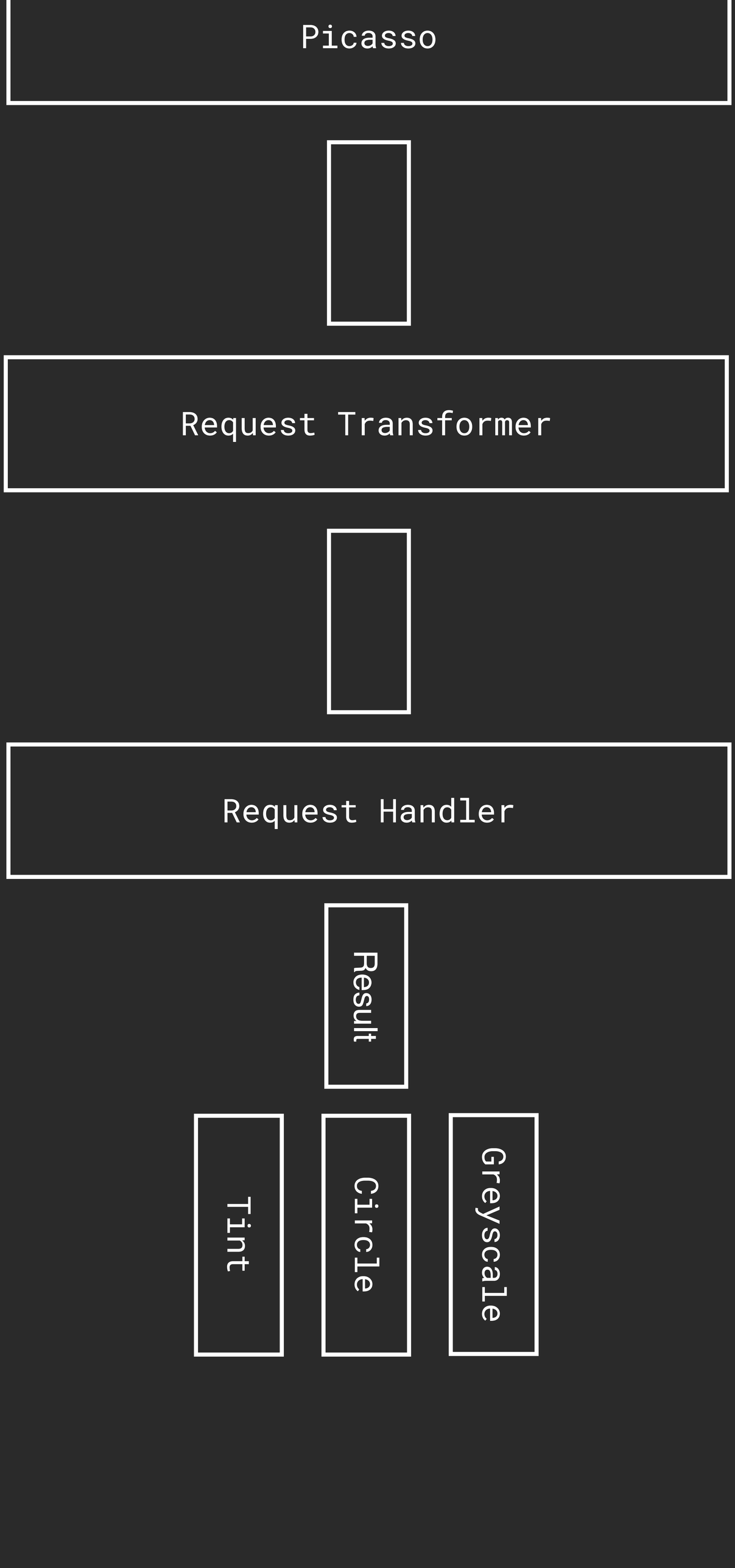
Request Transformer

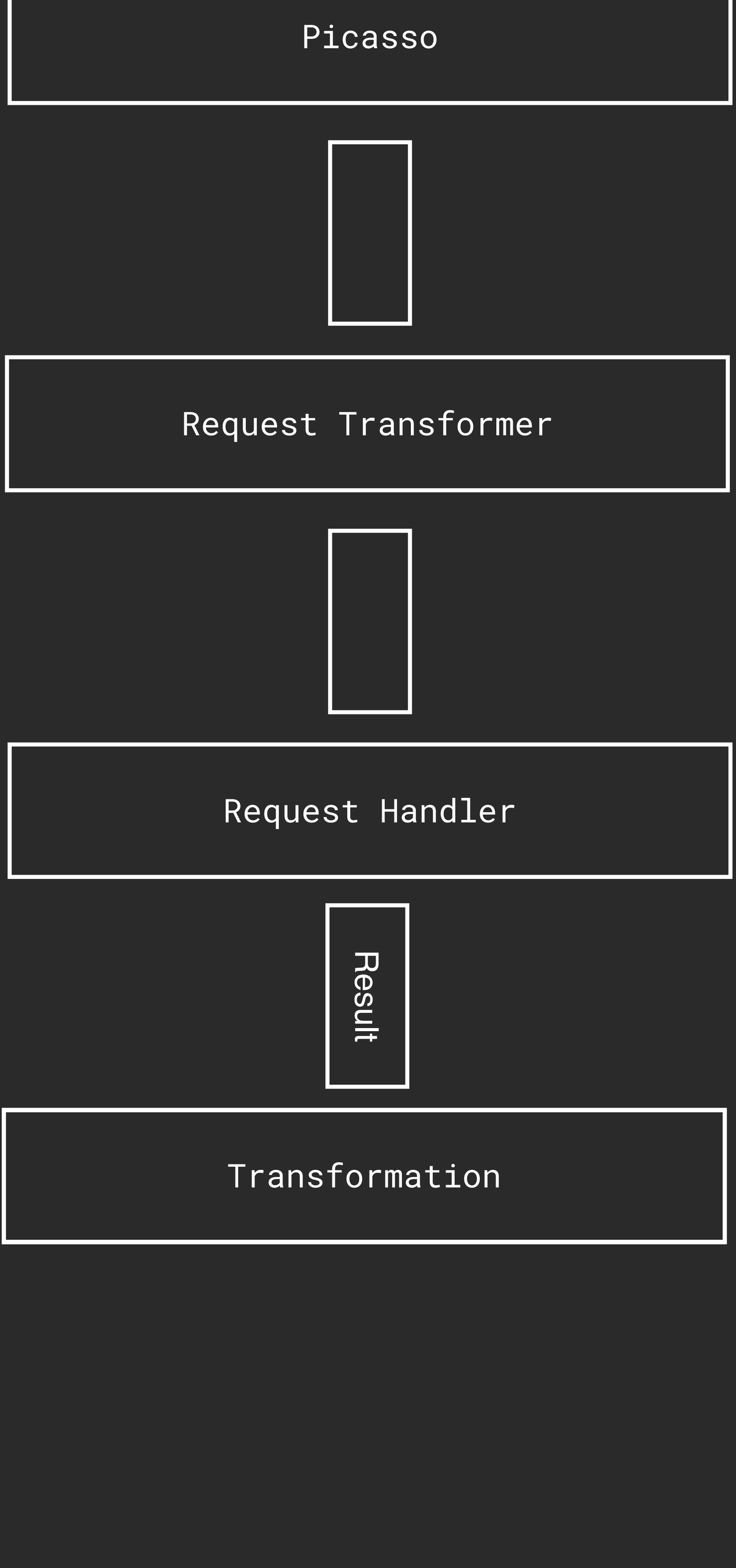
Request Handler

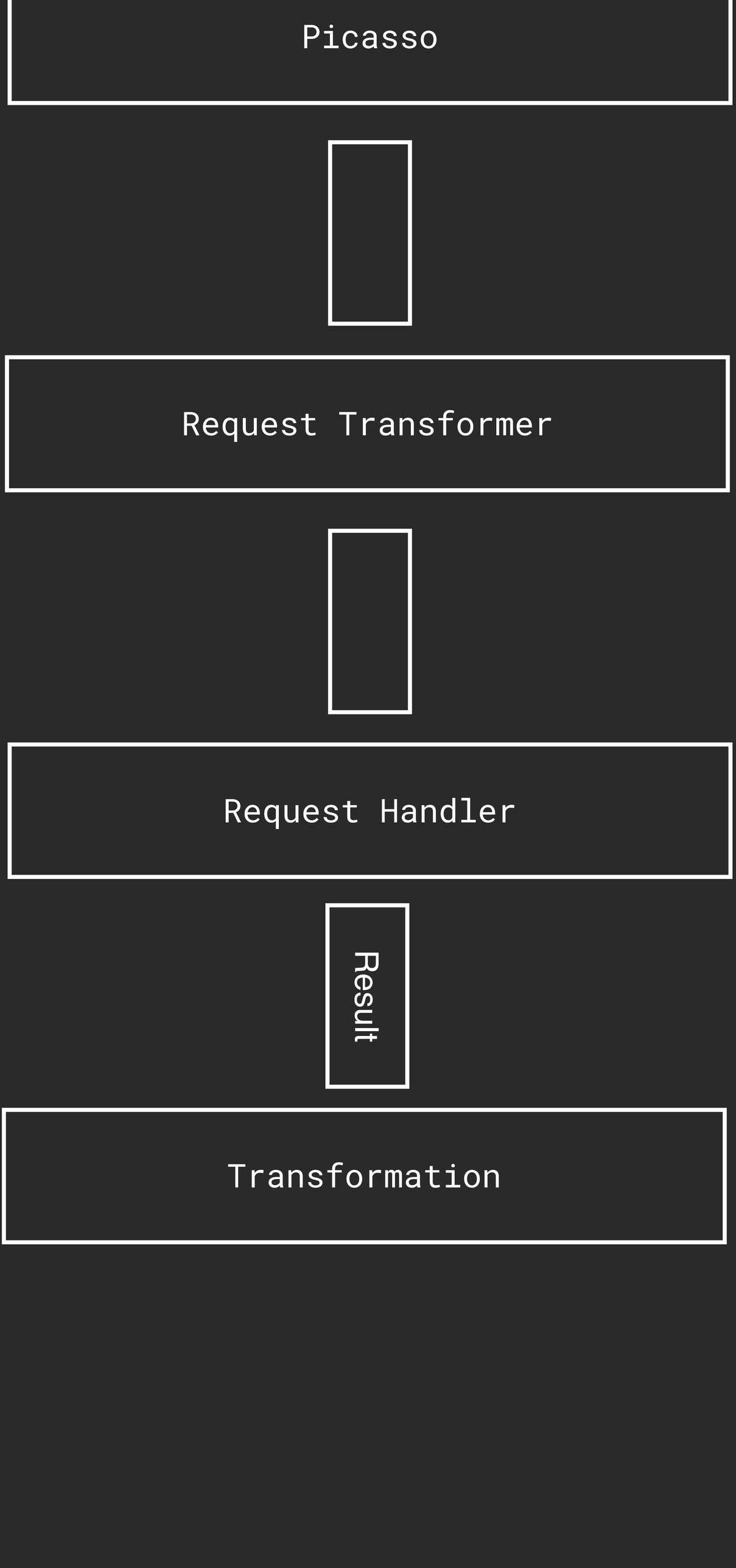

Picasso
Request Transformer
Request Handler
Bitmap

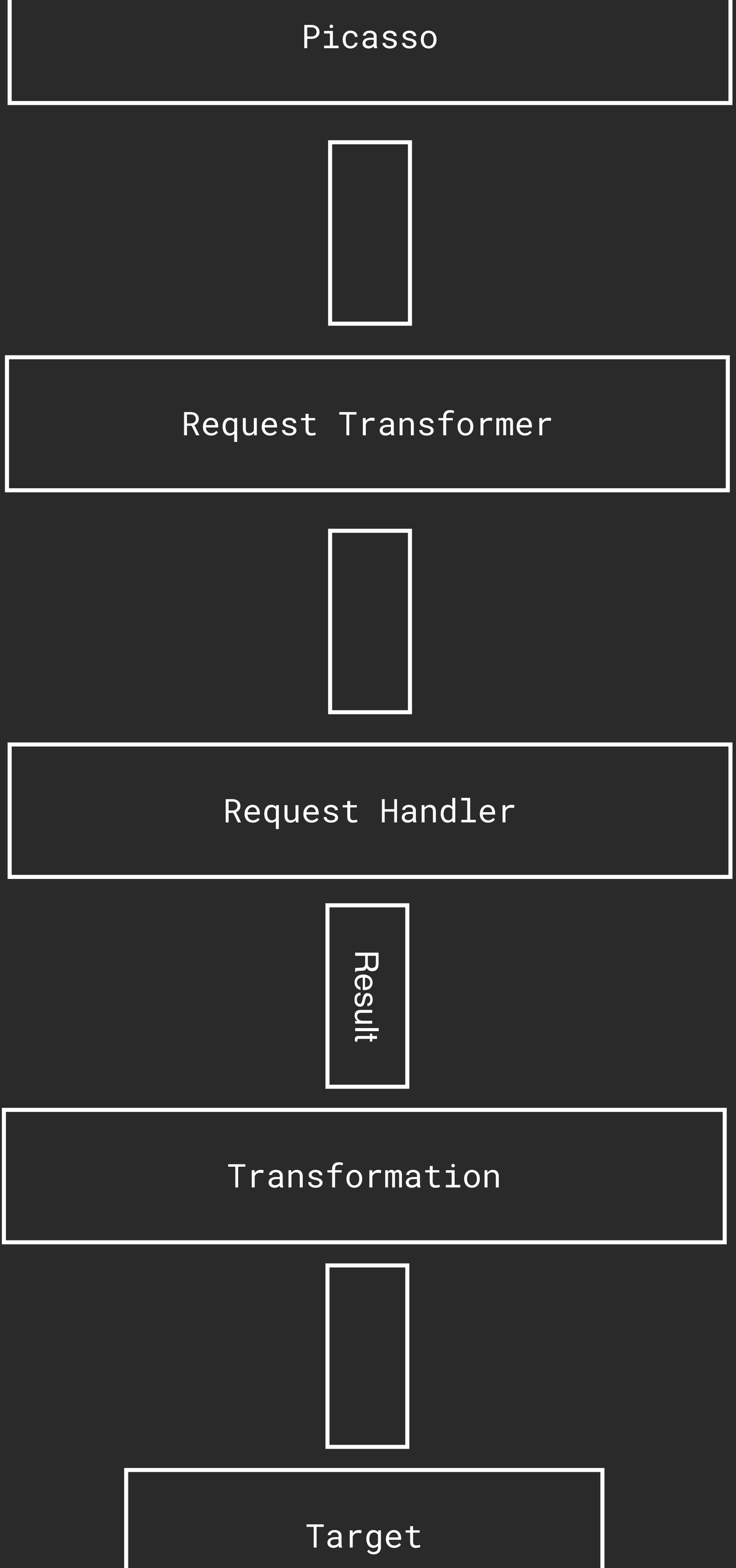

Picasso
Request Transformer
Request Handler
Result
Bitmap
Drawable

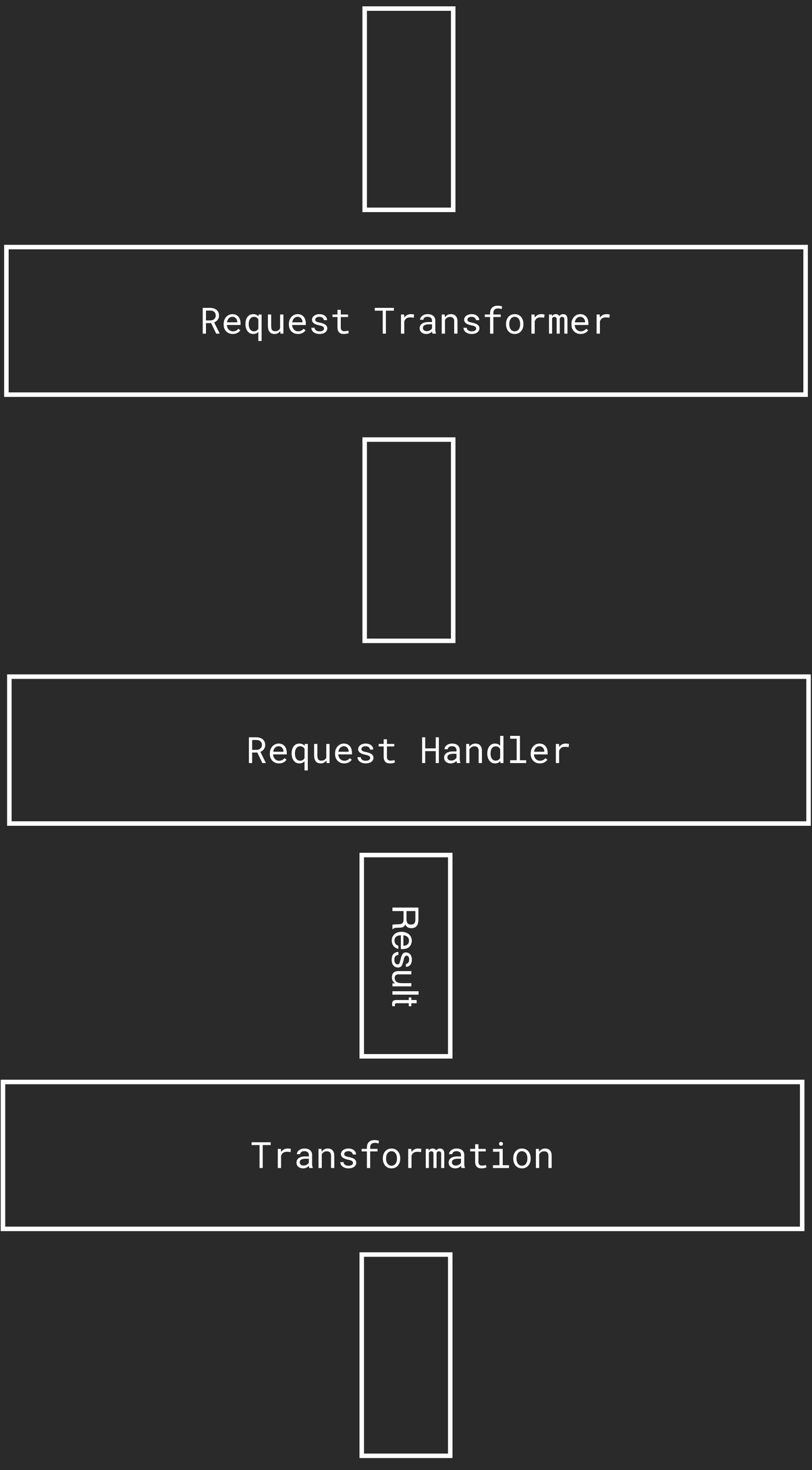

Picasso
Request Transformer
Request Handler
Result

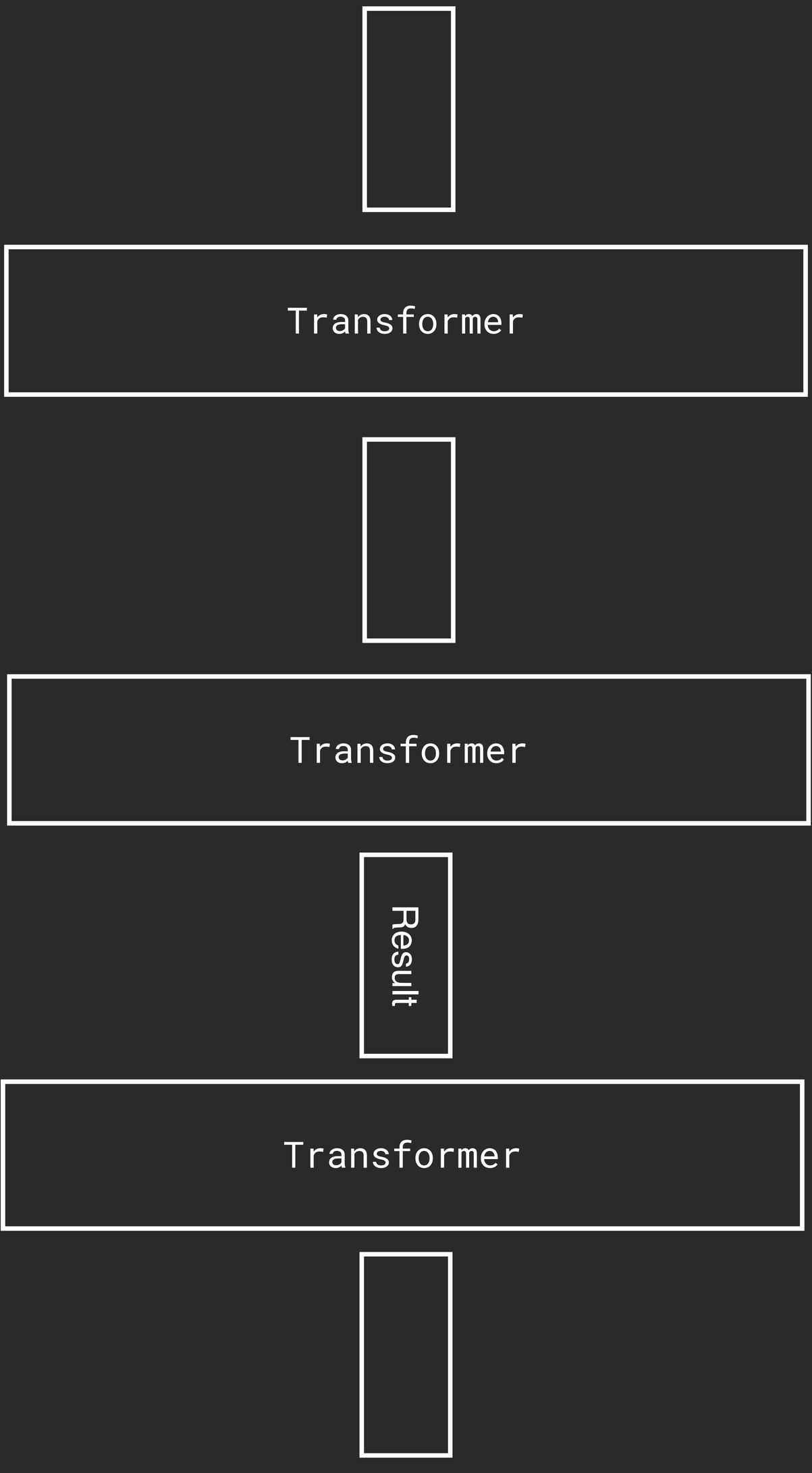

Picasso
Request Transformer
Request Handler
Result
Tint
Circle
Greyscale


Picasso
Request Transformer
Request Handler
Result
Transformation


Picasso
Request Transformer
Request Handler
Result
Transformation


Picasso
Request Transformer
Request Handler
Result
Transformation
Target

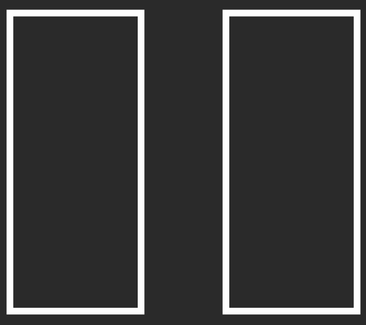

Request Transformer
Request Handler
Result
Transformation


Transformer
Transformer
Result
Transformer

Transformer

Transformer

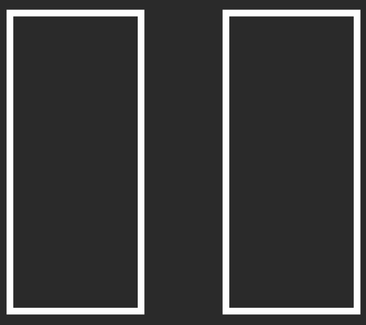
Transformer



Transformer

Transformer

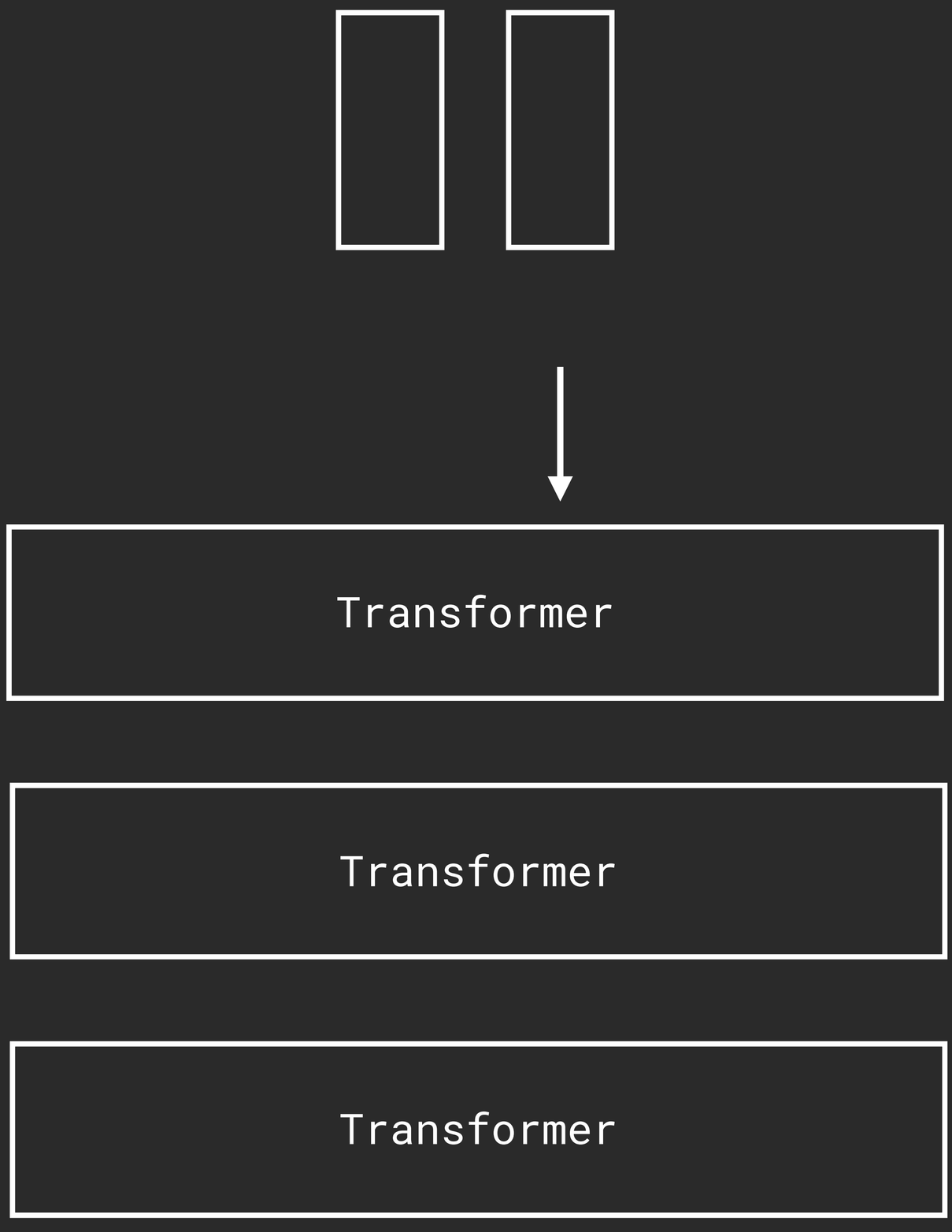
Transformer

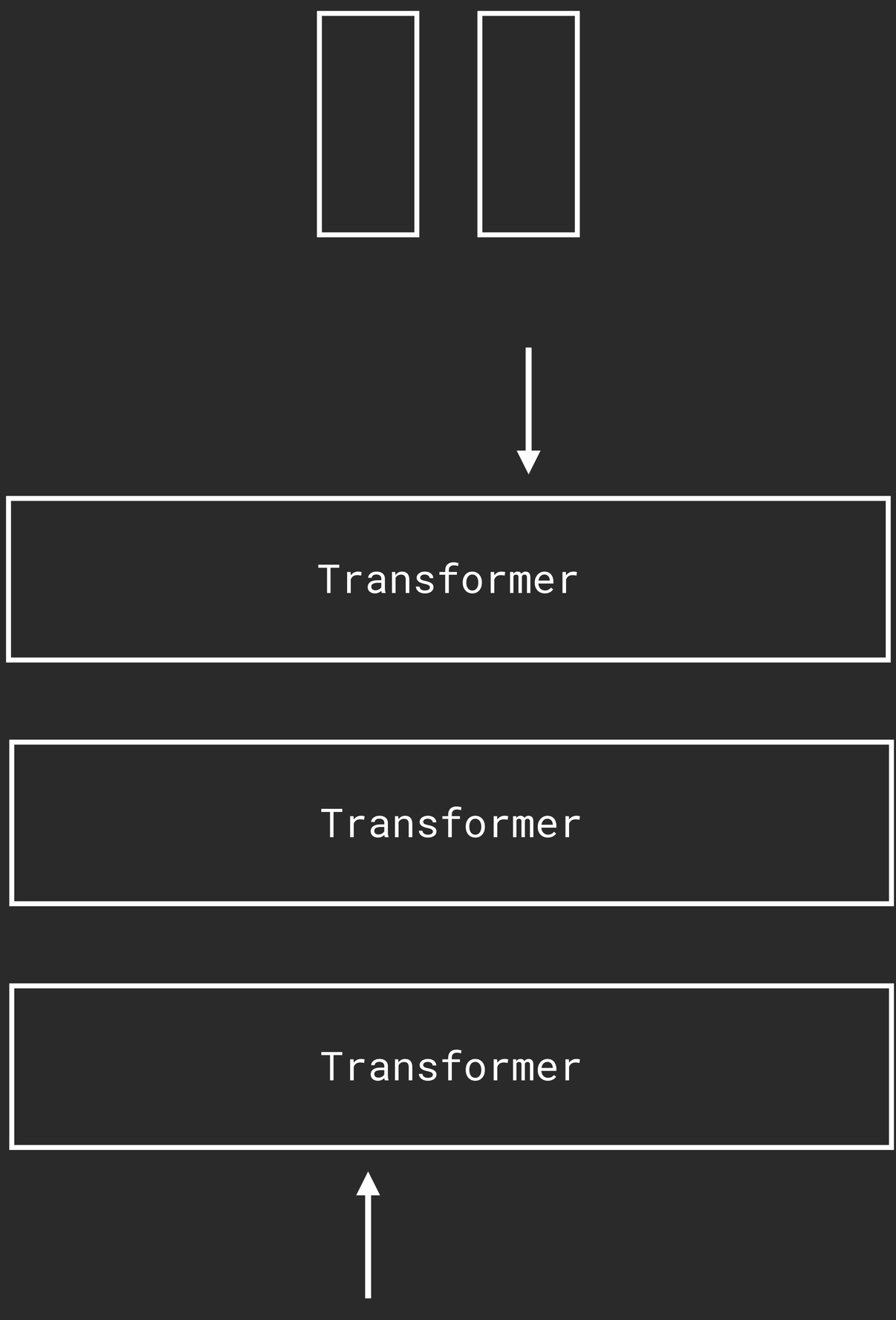


Transformer

Transformer

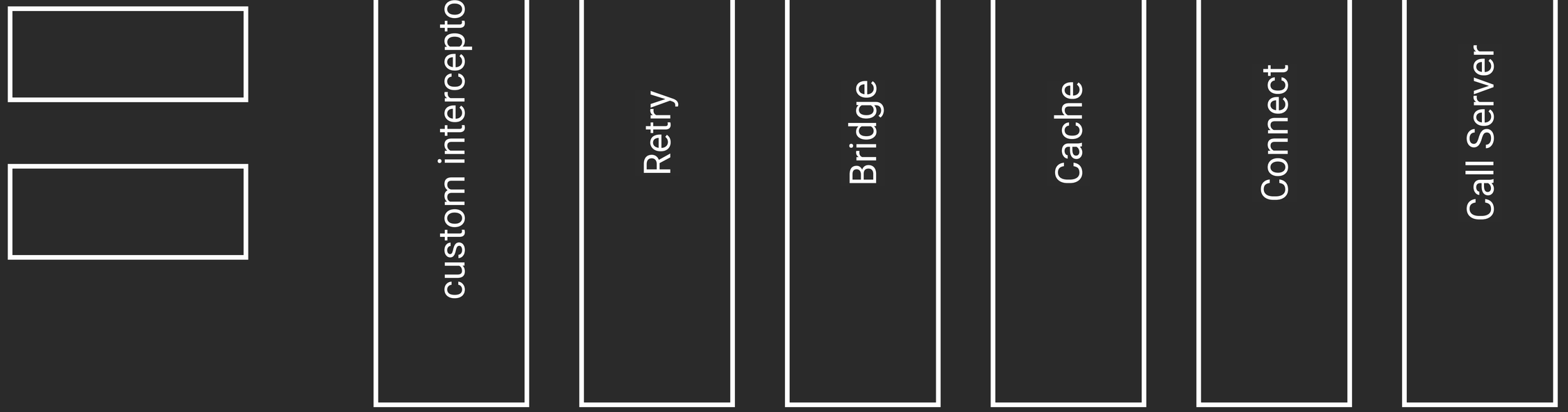
Transformer


Transformer
Transformer
Transformer

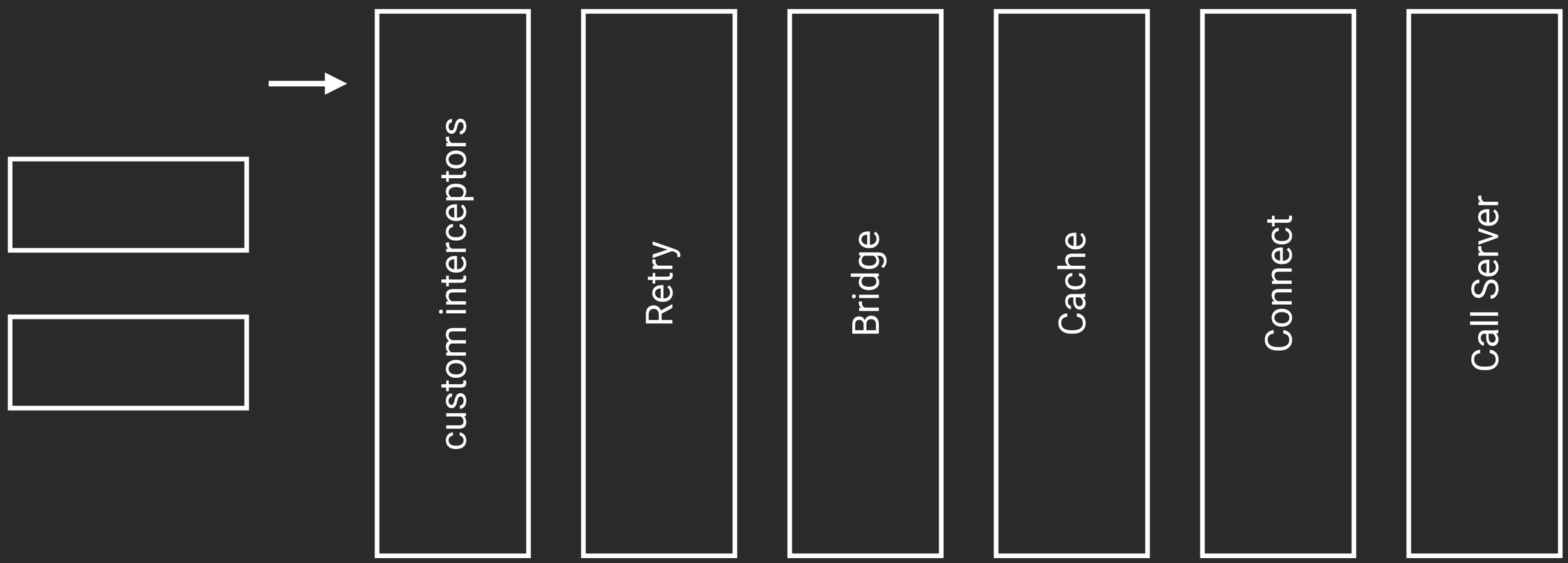

Transformer
Transformer
Transformer

square/okhttp

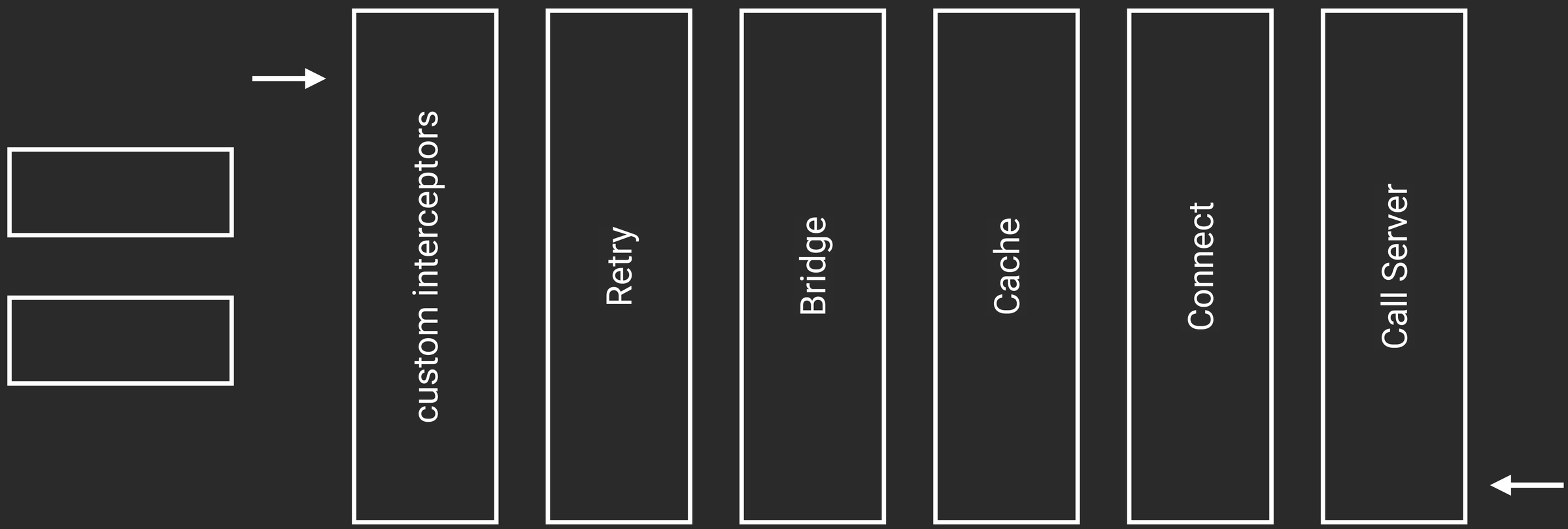
square/okhttp



square/okhttp



square/okhttp



Today, we would like to introduce and open source Picasso, our solution for image downloading and caching on Android.

Picasso aims to be fast and simple to use—often requiring only one line of code.

```
Picasso.with(context).load("http://example.com/logo.png").into(imageView);
```

And that's it! You can use this for downloading a single image or inside of an adapter's getView method to download many. Picasso automatically handles caching, recycling, and displaying the final bitmap into the target view.

Picasso also allows you to transform images.

```
Picasso.with(context)
    .load("http://example.com/logo.png")
    .resize(100, 100)
```

Today, we would like to introduce and open source Picasso, our solution for image downloading and caching on Android.

Picasso aims to be fast and simple to use—often requiring only one line of code.

```
Picasso.with(context) load("http://example.com/logo.png").into(imageView);
```

And that's it! You can use this for downloading a single image or inside of an adapter's getView method to download many. Picasso automatically handles caching, recycling, and displaying the final bitmap into the target view.

Picasso also allows you to transform images.

```
Picasso.with(context)
    .load("http://example.com/logo.png")
    .resize(100, 100)
```

```java
@Override
protected void onCreate(Bundle savedInstanceState) {
  super.onCreate(savedInstanceState);
  setContentView(R.layout.activity_main);

  Picasso.with(this)
      .load("http://example.com/logo.png")
      .into(imageView);
}
```

```java
@Override
protected void onCreate(Bundle savedInstanceState) {
  super.onCreate(savedInstanceState);
  setContentView(R.layout.activity_main);

  Picasso.with(this)
      .load("http://example.com/logo.png")
      .into(imageView);
}
```

```
public static Picasso with(Context context) {
  if (singleton == null) {
    synchronized (Picasso.class) {
      if (singleton == null) {
        singleton = new Builder(context).build();
      }
    }
  }
  return singleton;
}
```

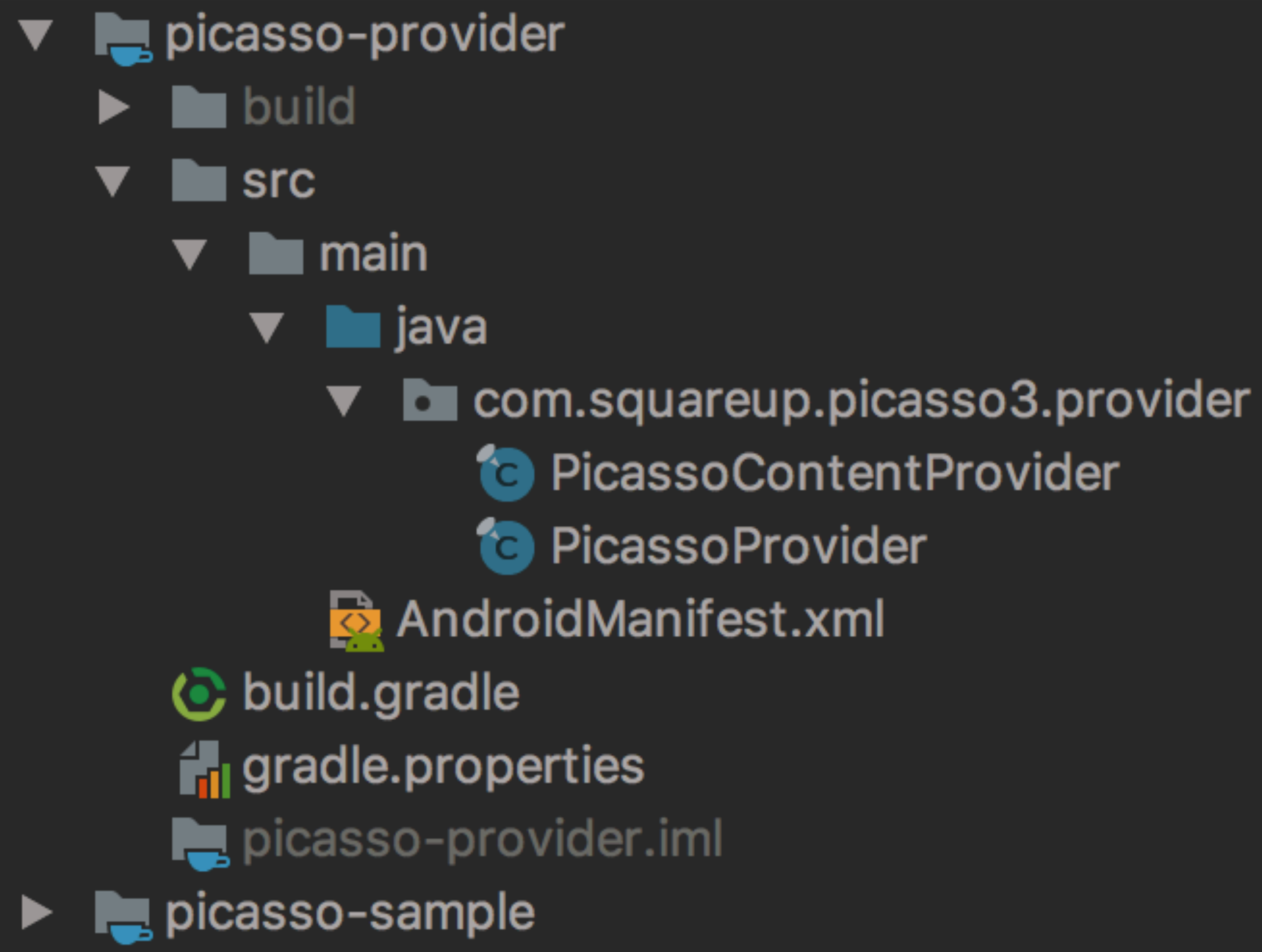
```
public static Picasso with(Context context) {
  if (singleton == null) {
    synchronized (Picasso.class) {
      if (singleton == null) {
        singleton = new Builder(context).build();
      }
    }
  }
  return singleton;
}
```

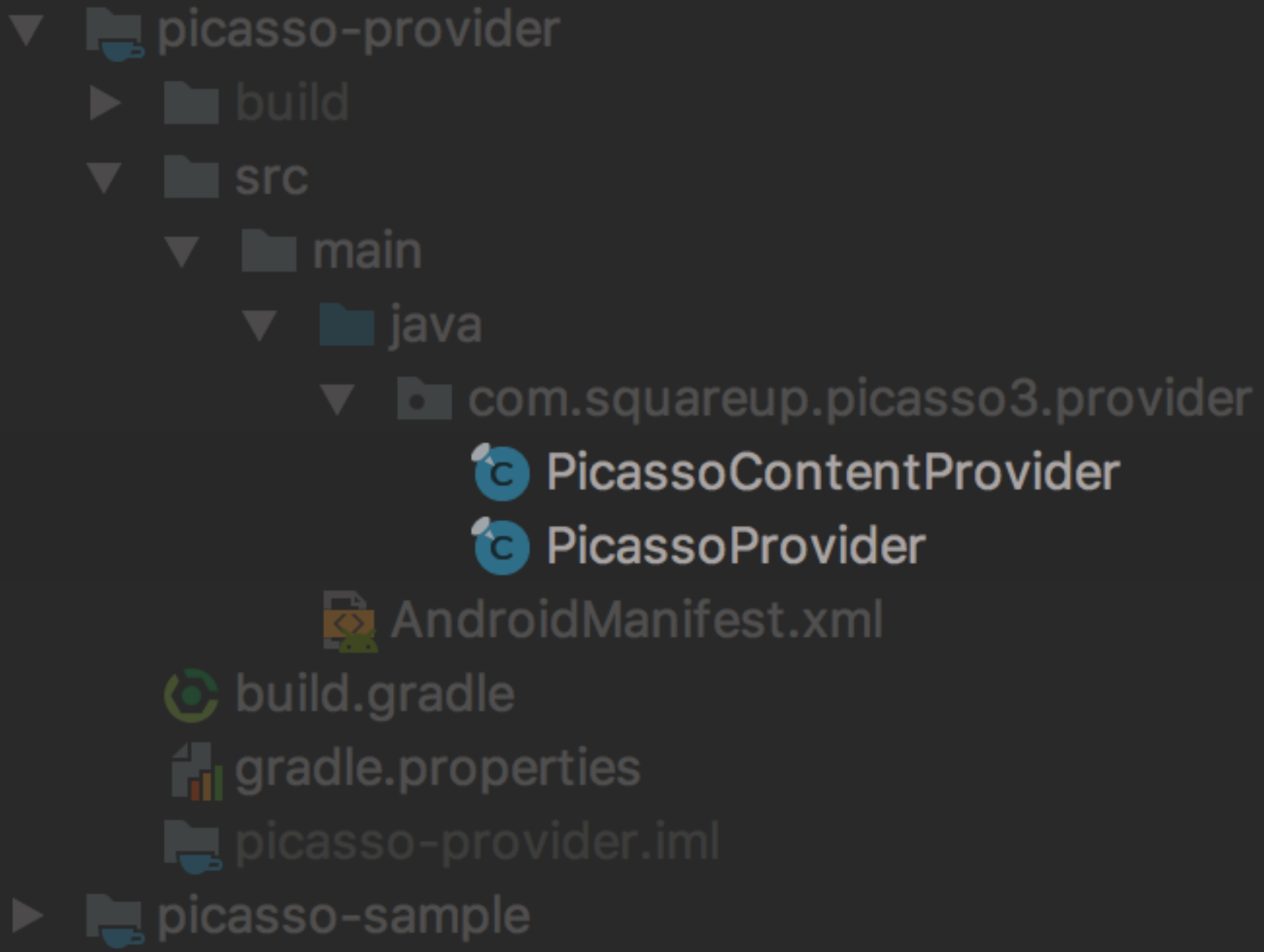
```
public static void setSingletonInstance(Picasso picasso) {
  synchronized (Picasso.class) {
    if (singleton != null) {
      throw new IllegalStateException(
                  "Singleton instance already exists.");
    }
    singleton = picasso;
  }
}
```

```
public static void setSingletonInstance(Picasso picasso) {
  synchronized (Picasso.class) {
    if (singleton != null) {
      throw new IllegalStateException(
                  "Singleton instance already exists.");
    }
    singleton = picasso;
  }
}
```

```
@Module(includes = ThumborModule.class)
final class ProductionPicassoModule {
  @Provides @Singleton static Picasso providePicasso(
      @App Context context,
      OkHttpClient client,
      Thumbor thumbor) {

    Picasso.RequestTransformer transformer =
        new PollexorRequestTransformer(thumbor);

    return new Picasso.Builder(context)
        .client(client)
        .requestTransformer(transformer)
        .listener((picasso, uri, e) -> Timber.d(e, uri.toString()));
  }
}
```

```
@Module(includes = ThumborModule.class)
final class ProductionPicassoModule {
  @Provides @Singleton static Picasso providePicasso(
      @App Context context,
      OkHttpClient client,
      Thumbor thumbor) {

    Picasso.RequestTransformer transformer =
        new PollexorRequestTransformer(thumbor);

    return new Picasso.Builder(context)
        .client(client)
        .requestTransformer(transformer)
        .listener((picasso, uri, e) -> Timber.d(e, uri.toString()));
  }
}
```


picasso-provider
build
src
main
java
com.squareup.picasso3.provider
PicassoContentProvider
PicassoProvider
AndroidManifest.xml
build.gradle
gradle.properties
picasso-provider.iml
picasso-sample


picasso-provider
build
src
main
java
com.squareup.picasso3.provider
PicassoContentProvider
PicassoProvider
AndroidManifest.xml
build.gradle
gradle.properties
picasso-provider.iml
picasso-sample

```java
public final class PicassoProvider {
  private static volatile Picasso instance;

  public static Picasso get() {
    if (instance == null) {
      synchronized (PicassoProvider.class) {
        if (instance == null) {
          if (PicassoContentProvider.context == null) {
            throw new IllegalStateException("context == null");
          }
          instance = new
                Picasso.Builder(PicassoContentProvider.context).build();
        }
      }
    }
    return instance;
  }
}
```

```
public final class PicassoProvider {
  private static volatile Picasso instance;

  public static Picasso get() {
    if (instance == null) {
      synchronized (PicassoProvider.class) {
        if (instance == null) {
          if (PicassoContentProvider.context == null) {
            throw new IllegalStateException("context == null");
          }
          instance = new
                Picasso.Builder(PicassoContentProvider.context).build();
        }
      }
    }
    return instance;
  }
}
```

```
public final class PicassoProvider {
  private static volatile Picasso instance;

  public static Picasso get() {
    if (instance == null) {
      synchronized (PicassoProvider.class) {
        if (instance == null) {
          if (PicassoContentProvider.context == null) {
            throw new IllegalStateException("context == null");
          }
          instance = new
                Picasso.Builder(PicassoContentProvider.context).build();
        }
      }
    }
    return instance;
  }
}
```

```
public final class PicassoContentProvider extends ContentProvider {
  static Context context;

  @Override public boolean onCreate() {
    context = getContext();
    return true;
  }


  …
}
```

The Firebase Blog

How does Firebase initialize on Android?

December 21, 2016

Doug Stevenson
Developer Advocate

If you've been working with Firebase on Android, you may have noticed that you don't normally have to write any lines of code to initialize a feature. You just grab the singleton object for that feature, and start using it right away. And, in the case of Firebase Crash Reporting, you don't even have to write any code at all for it to start capturing crashes! This question pops up from time to time, and I talked about it a bit at Google I/O 2016, but I'd also like to break it down in detail here.

The problem

About

Firebase gives you the tools and infrastructure to build better apps and grow successful businesses.
Learn more

Popular Posts

Introducing Cloud Firestore: Our New Document Database for Apps

Firebase expands to become a unified app platform

Welcoming Fabric to Google

The beginners guide to React Native and Firebase

Best Practices: Arrays in Firebase

Archive

3.x Goals

- Android P
- OkHttp 2.x => 3.x
- Okio integration
- Improvements

```
Picasso.with(context)
    .load(url)
    .placeholder(R.drawable.placeholder)
    .error(R.drawable.error)
    .fit()
    .into(view);
```

```
PicassoProvider.get()
    .load(url)
    .placeholder(R.drawable.placeholder)
    .error(R.drawable.error)
    .fit()
    .into(view);
```

```
picasso.load(url)
    .placeholder(R.drawable.placeholder)
    .error(R.drawable.error)
    .fit()
    .into(view);
```

```
abstract class RequestHandler {
  abstract boolean canHandleRequest(Request request);

  abstract Result load(Request request, int networkPolicy);
}
```

```
abstract class RequestHandler {
  abstract boolean canHandleRequest(Request request);

  abstract void load(Picasso picasso, Request request, Callback callback);
}
```

Coming Soon

Coming Soon

- Better separation of concerns: Source/Target/Load

Coming Soon

- Better separation of concerns: Source/Target/Load
- Propagating Drawables to Target (animated GIFs!)

Coming Soon

- Better separation of concerns: Source/Target/Load
- Propagating Drawables to Target (animated GIFs!)
- More? Let us know!

PICASSO WILL RETURN

Rinsing the Brush: Picasso 3.0

@jrodbx

@JakeWharton