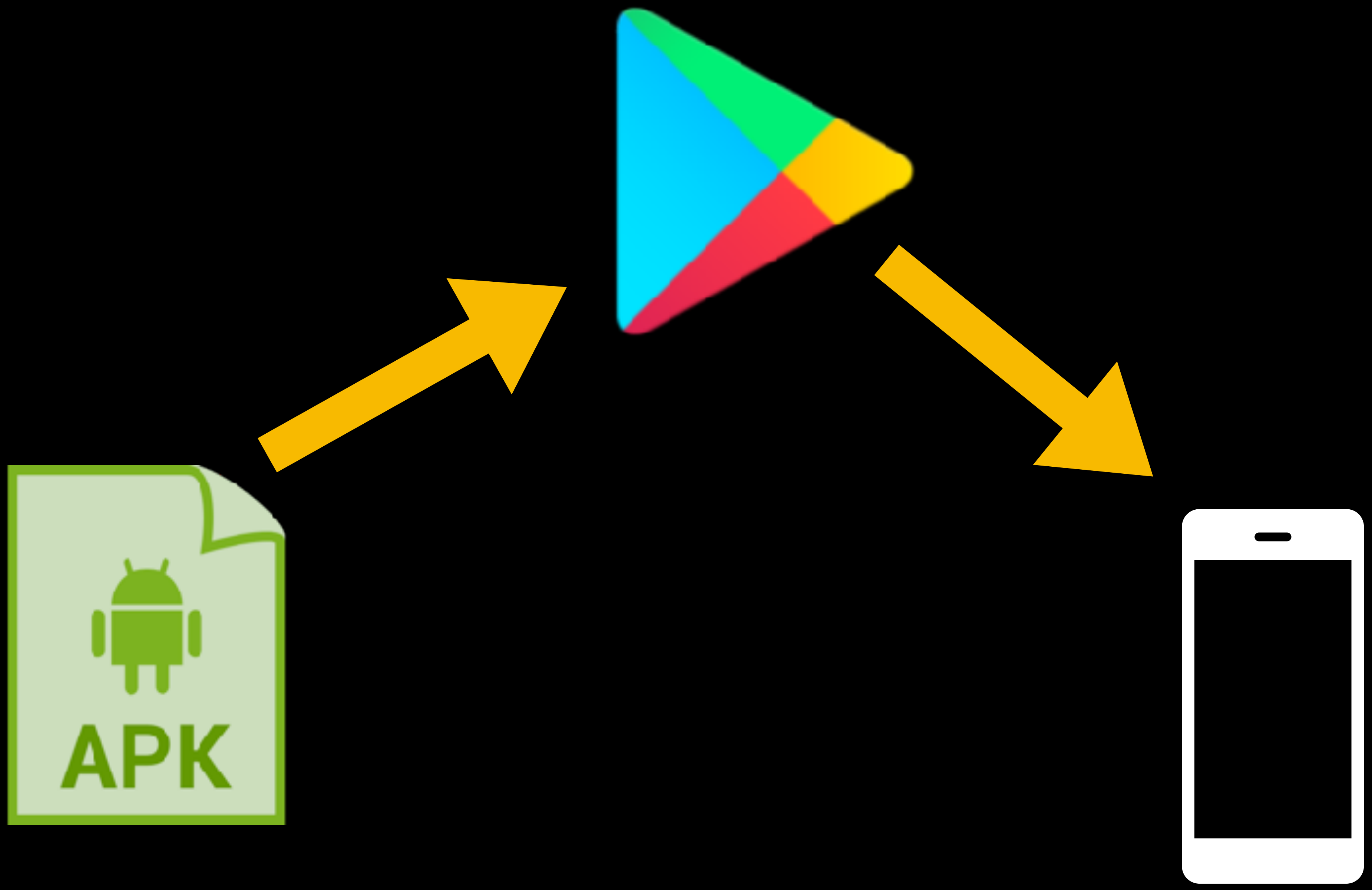
How ProGuard Works

Jeb Ware
American Express

How ProGuard Works

Jeb Ware
American Express


APK

```
➜  ~ adb shell pm path com.example.yourapp
➜  ~ adb shell pm path com.example.yourapp

package:/data/app/com.example.yourapp-1/base.apk
```

```
➜  ~ adb shell pm path com.example.yourapp

package:/data/app/com.example.yourapp-1/base.apk

➜  ~ adb pull /data/app/com.example.yourapp-1/base.apk
➜  ~ adb shell pm path com.example.yourapp

package:/data/app/com.example.yourapp-1/base.apk

➜  ~ adb pull /data/app/com.example.yourapp-1/base.apk

➜  ~ unzip base.apk
```

classes.dex + dex2jar ⟹ classes.jar

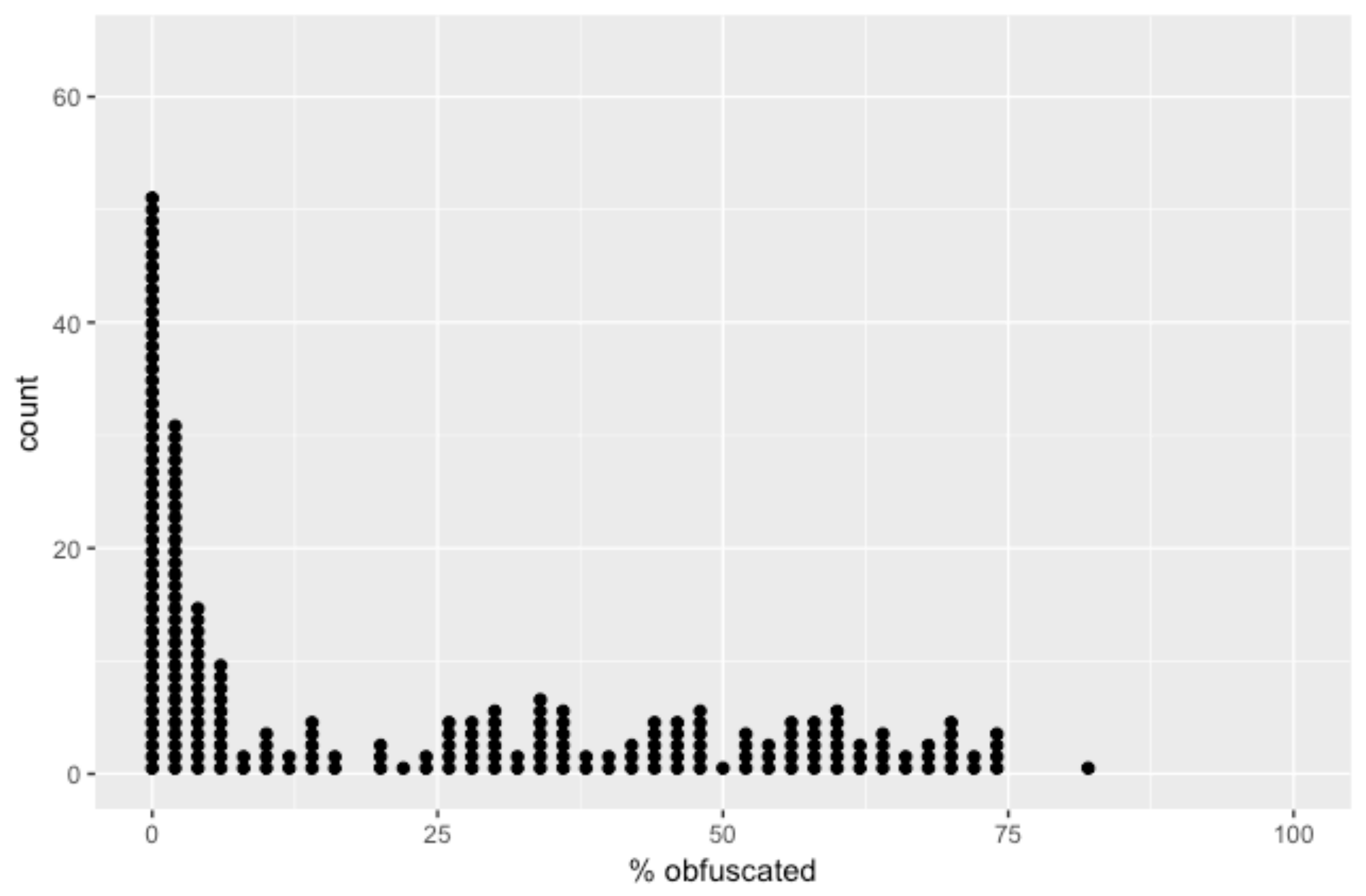
classes.jar + jd-cli ⟹ JavaSourceFiles.java

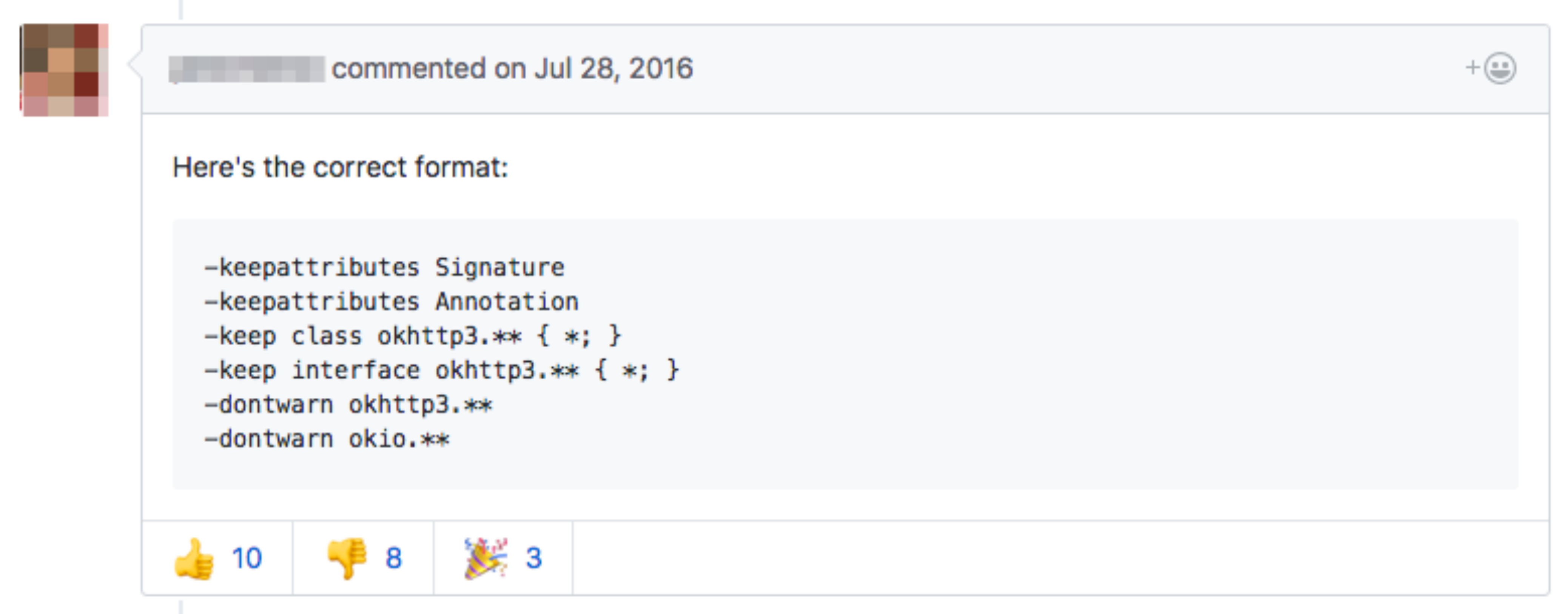
```java
public class MainActivity_ViewBinding implements Unbinder {
  private MainActivity target;
  private View view16908313;

  @UiThread
  public MainActivity_ViewBinding(final MainActivity paramMainActivity, View
paramView) {
    target = paramMainActivity;
    resultView = ((TextView)Utils.findRequiredViewAsType(paramView, 16908308,
"field 'resultView'", TextView.class));
    paramView = Utils.findRequiredView(paramView, 16908313, "method 'doIt'");
    view16908313 = paramView;
    paramView.setOnClickListener(new DebouncingOnClickListener() {
      public void doClick(View paramAnonymousView) {
        paramMainActivity.doIt((Button)Utils.castParam(paramAnonymousView,
"doClick", 0, "doIt", 0, Button.class));
      }
    });
  }
}

/* Location:
 * Qualified Name:     com.jebware.demo.MainActivity_ViewBinding
 * Java Class Version: 6 (50.0)
 * JD-Core Version:    0.7.1
 */
```

```
public class MainActivity_ViewBinding implements Unbinder {
  private MainActivity target;
  private View view16908313;

  @UiThread
  public MainActivity_ViewBinding(final MainActivity paramMainActivity, View
paramView) {
    target = paramMainActivity;
    resultView = ((TextView)Utils.findRequiredViewAsType(paramView, 16908308,
"field 'resultView'", TextView.class));
    paramView = Utils.findRequiredView(paramView, 16908313, "method 'doIt'");
    view16908313 = paramView;
    paramView.setOnClickListener(new DebouncingOnClickListener() {
      public void doClick(View paramAnonymousView) {
        paramMainActivity.doIt((Button)Utils.castParam(paramAnonymousView,
"doClick", 0, "doIt", 0, Button.class));
      }
    });
  }
}

/* Location:
 * Qualified Name:     com.jebware.demo.MainActivity_ViewBinding
 * Java Class Version: 6 (50.0)
 * JD-Core Version:    0.7.1
 */
```

```
final class c<T> implements c.e<ad, T> {
  private final com.a.a.e a;
  private final t<T> b;

  c(com.a.a.e parame, t<T> paramt) {
    a = parame;
    b = paramt;
  }

  public T a(ad paramad) {
    Object localObject1 = a.a(paramad.d());
    try {
      localObject1 = b.b((a)localObject1);
      return (T)localObject1;
    } finally {
      paramad.close();
    }
  }
}
```

Too many apps don’t obfuscate

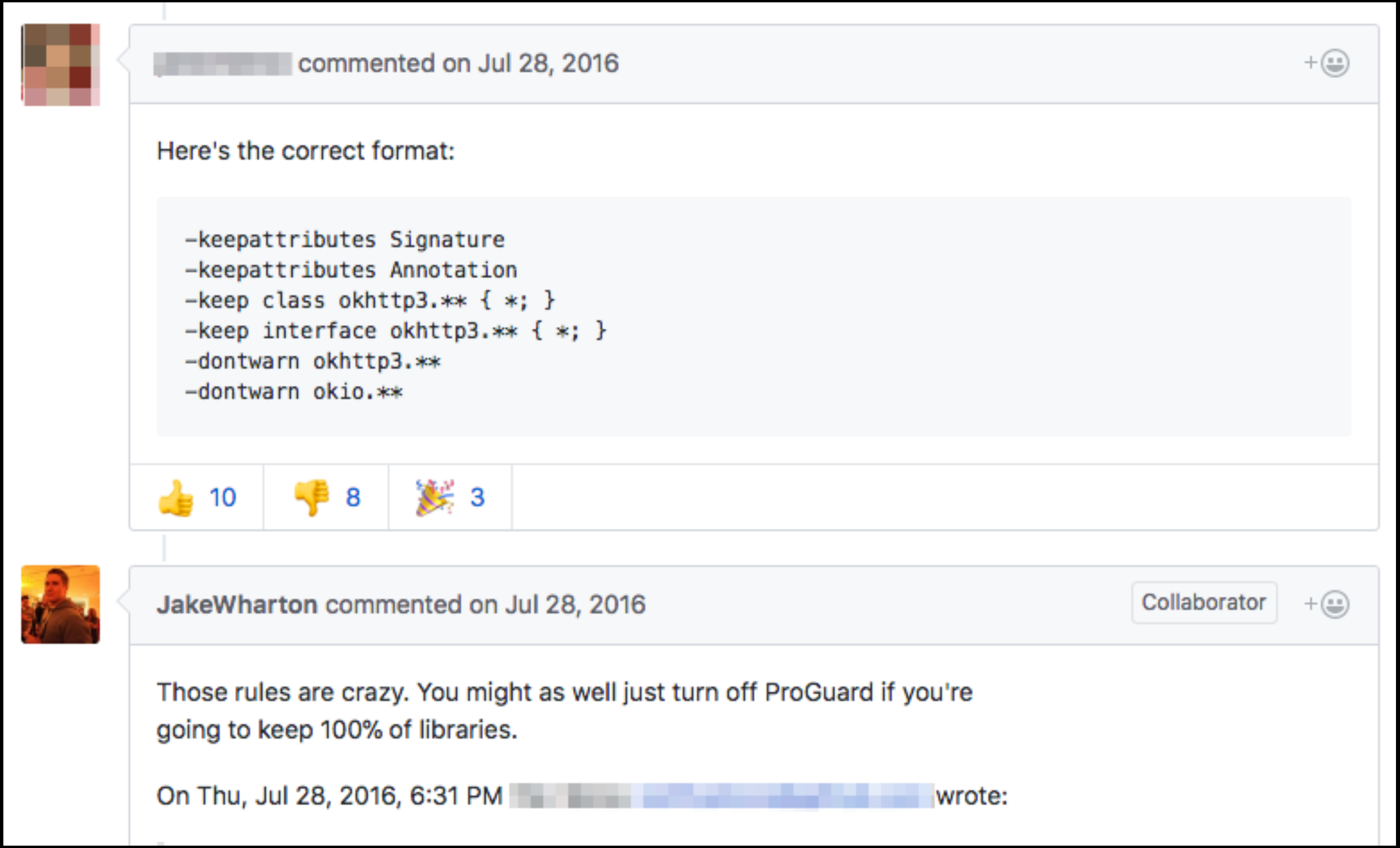




[redacted] commented on Jul 28, 2016

Here's the correct format:

```
-keepattributes Signature
-keepattributes Annotation
-keep class okhttp3.** { *; }
-keep interface okhttp3.** { *; }
-dontwarn okhttp3.**
-dontwarn okio.**
```



[redacted] commented on Jul 28, 2016

Here's the correct format:

```
-keepattributes Signature
-keepattributes Annotation
-keep class okhttp3.** { *; }
-keep interface okhttp3.** { *; }
-dontwarn okhttp3.**
-dontwarn okio.**
```

👍 10 👎 8 🎉 3

JakeWharton commented on Jul 28, 2016 Collaborator

Those rules are crazy. You might as well just turn off ProGuard if you're going to keep 100% of libraries.

On Thu, Jul 28, 2016, 6:31 PM [redacted] wrote:

ProGuard

- shrinking
- obfuscation
- optimization

ProGuard

- shrinking
- obfuscation
- ~~optimization~~

configuration file(s)

```
-dontoptimize

-dontusemixedcaseclassnames
-dontskipnonpubliclibraryclasses
-verbose

# Preserve some attributes that may be required for reflection.
-keepattributes *Annotation*,Signature,InnerClasses,EnclosingMethod

-keep public class com.google.vending.licensing.ILicensingService
-keep public class com.android.vending.licensing.ILicensingService
-keep public class com.google.android.vending.licensing.ILicensingService
-dontnote com.android.vending.licensing.ILicensingService
-dontnote com.google.vending.licensing.ILicensingService
-dontnote com.google.android.vending.licensing.ILicensingService

# For native methods, see http://proguard.sourceforge.net/manual/examples.html#native
-keepclasseswithmembernames class * {
    native <methods>;
}

# Keep setters in Views so that animations can still work.
-keepclassmembers public class * extends android.view.View {
    void set*(***);
    *** get*();
}
```

class pool(s)

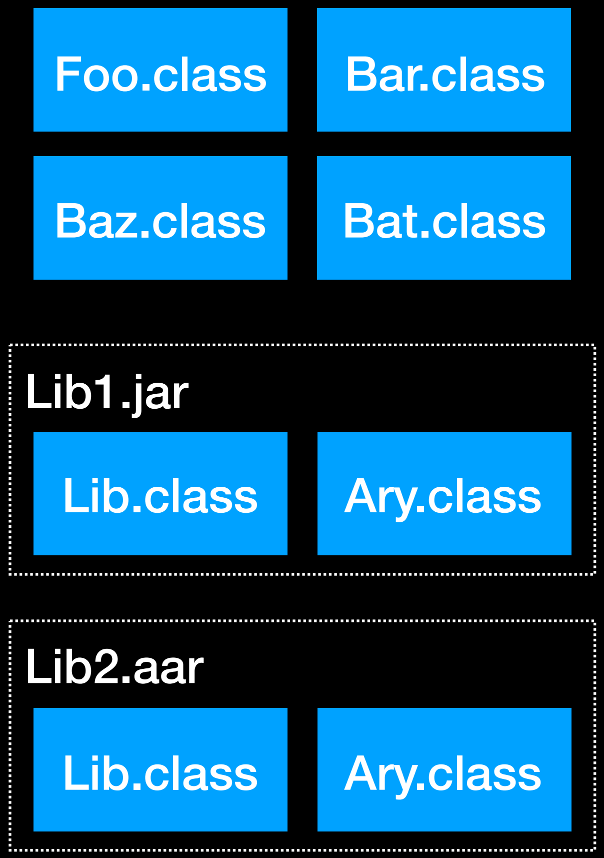
Foo.class

Bar.class

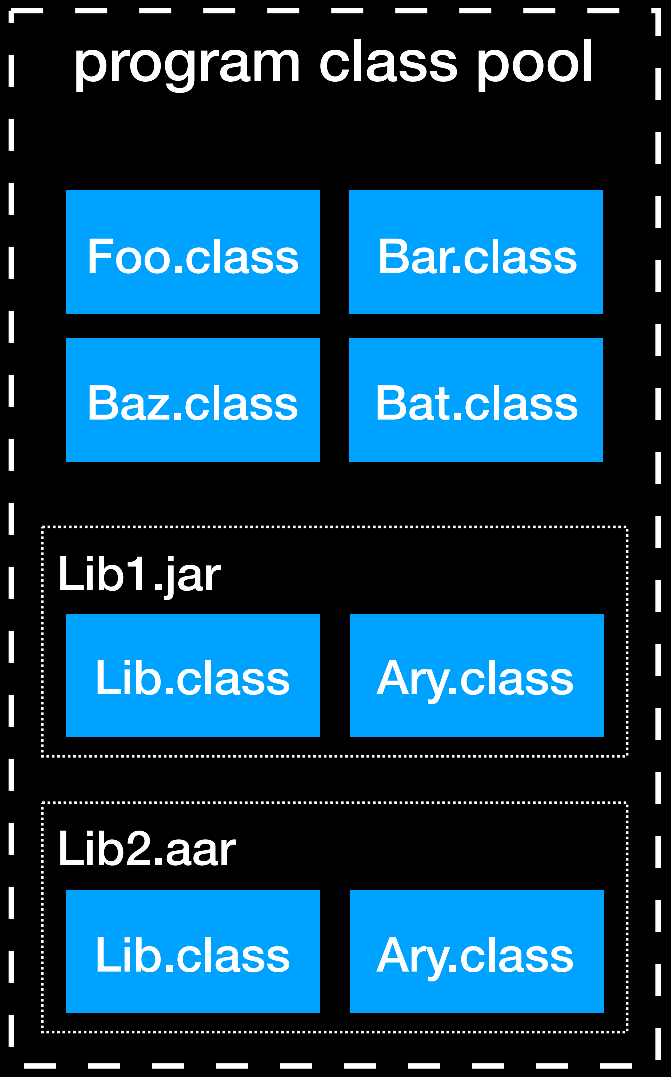
Baz.class

Bat.class

class pool(s)


Foo.class
Bar.class
Baz.class
Bat.class
Lib1.jar
Lib.class
Ary.class
Lib2.aar
Lib.class
Ary.class

class pool(s)



program class pool

Foo.class Bar.class

Baz.class Bat.class

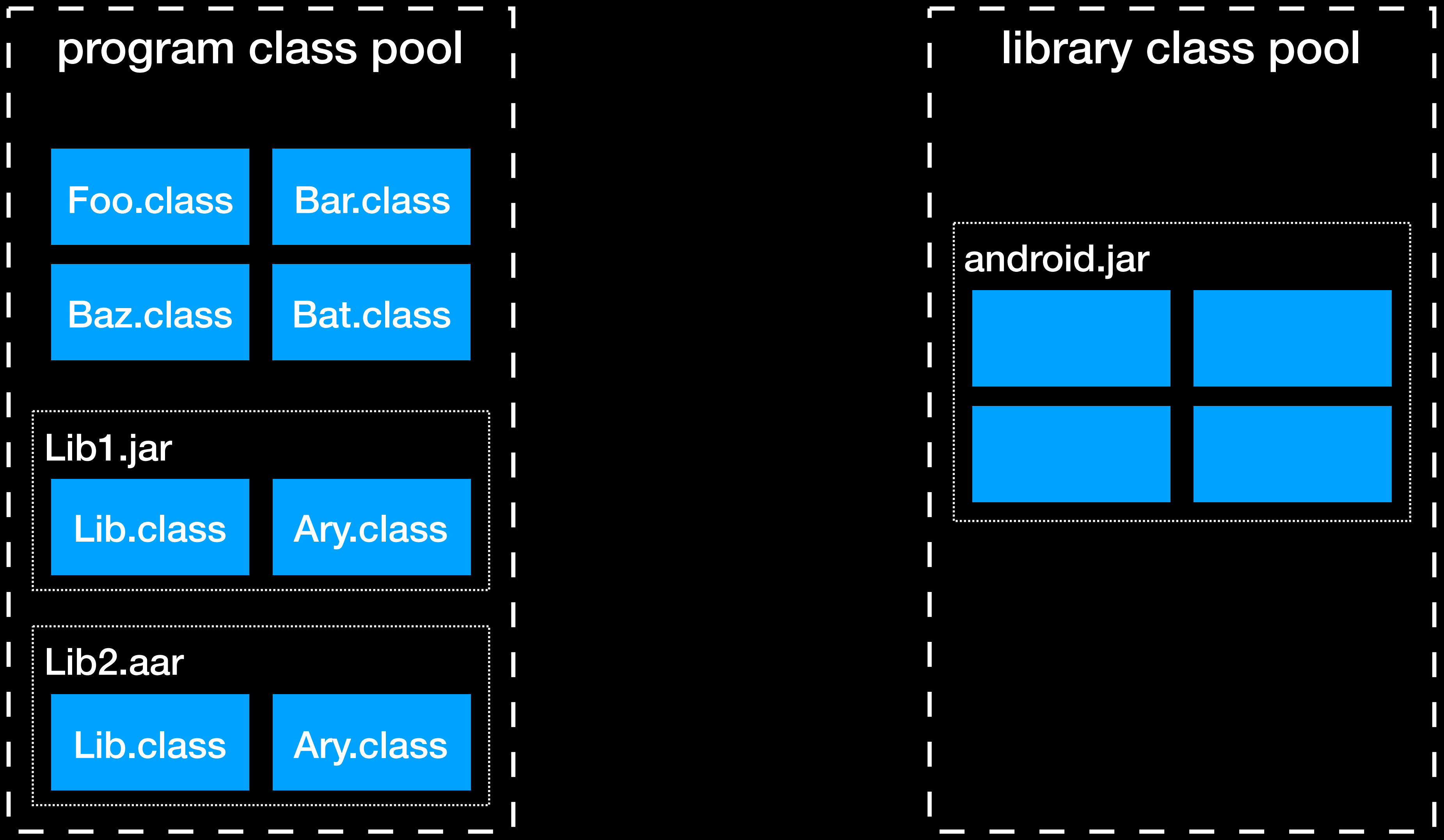
Lib1.jar

Lib.class Ary.class

Lib2.aar

Lib.class Ary.class

class pool(s)


program class pool
Foo.class
Bar.class
Baz.class
Bat.class
Lib1.jar
Lib.class
Ary.class
Lib2.aar
Lib.class
Ary.class
library class pool
android.jar

find seeds

```
✓ public class MainActivity extends AppCompatActivity {

      private String name = “name";

      @Override
✓     protected void onCreate(Bundle paramBundle) {
          String hello = StringUtil.getHello(name);
      }
  }

  public class StringUtil {

      static String helloPrefix = "Hello, “;
      static String goodbyePrefix = "Goodbye, “;

      public static String getHello(String paramString) {
          return helloPrefix + paramString;
      }
  }

  public class Foo {

      private String bar = "bar";

      public String getBar() {
          return bar;
      }
  }
```

```
✓ public class MainActivity extends AppCompatActivity {

      private String name = “name";

      @Override
✓     protected void onCreate(Bundle paramBundle) {
          String hello = StringUtil.getHello(name);
      }
  }
```

seeds.txt

```
com.jebware.demo.MainActivity
com.jebware.demo.MainActivity: onCreate(android.os.Bundle)
```

```
          return helloPrefix + name;
      }
  }

  public class Foo {

      private String bar = "bar";

      public String getBar() {
          return bar;
      }
  }
```

shrinking

```
public class MainActivity extends AppCompatActivity {

    private String name = “name";

    @Override
    protected void onCreate(Bundle paramBundle) {
        String hello = StringUtil.getHello(name);
    }
}

public class StringUtil {

    static String helloPrefix = "Hello, “;
    static String goodbyePrefix = "Goodbye, “;

    public static String getHello(String paramString) {
        return helloPrefix + paramString;
    }
}

public class Foo {

    private String bar = "bar";

    public String getBar() {
        return bar;
    }
}
```

shrinking

```
✓ public class MainActivity extends AppCompatActivity {

      private String name = “name";

      @Override
      protected void onCreate(Bundle paramBundle) {
          String hello = StringUtil.getHello(name);
      }
  }

  public class StringUtil {

      static String helloPrefix = "Hello, “;
      static String goodbyePrefix = "Goodbye, “;

      public static String getHello(String paramString) {
          return helloPrefix + paramString;
      }
  }

  public class Foo {

      private String bar = "bar";

      public String getBar() {
          return bar;
      }
  }
```

shrinking

```
✓ public class MainActivity extends AppCompatActivity {

      private String name = “name";

      @Override
✓     protected void onCreate(Bundle paramBundle) {
          String hello = StringUtil.getHello(name);
      }
  }

  public class StringUtil {

      static String helloPrefix = "Hello, “;
      static String goodbyePrefix = "Goodbye, “;

      public static String getHello(String paramString) {
          return helloPrefix + paramString;
      }
  }

  public class Foo {

      private String bar = "bar";

      public String getBar() {
          return bar;
      }
  }
```

shrinking

```
✓ public class MainActivity extends AppCompatActivity {

✓     private String name = “name";

      @Override
✓     protected void onCreate(Bundle paramBundle) {
          String hello = StringUtil.getHello(name);
      }
  }

  public class StringUtil {

      static String helloPrefix = "Hello, “;
      static String goodbyePrefix = "Goodbye, “;

      public static String getHello(String paramString) {
          return helloPrefix + paramString;
      }
  }

  public class Foo {

      private String bar = "bar";

      public String getBar() {
          return bar;
      }
  }
```

shrinking

```
✓ public class MainActivity extends AppCompatActivity {

✓     private String name = “name";

      @Override
✓     protected void onCreate(Bundle paramBundle) {
          String hello = StringUtil.getHello(name);
      }
  }

✓ public class StringUtil {

      static String helloPrefix = "Hello, “;
      static String goodbyePrefix = "Goodbye, “;

      public static String getHello(String paramString) {
          return helloPrefix + paramString;
      }
  }

  public class Foo {

      private String bar = "bar";

      public String getBar() {
          return bar;
      }
  }
```

shrinking

```
✓ public class MainActivity extends AppCompatActivity {

✓     private String name = “name";

      @Override
✓     protected void onCreate(Bundle paramBundle) {
          String hello = StringUtil.getHello(name);
      }
  }

✓ public class StringUtil {

      static String helloPrefix = "Hello, “;
      static String goodbyePrefix = "Goodbye, “;

✓     public static String getHello(String paramString) {
          return helloPrefix + paramString;
      }
  }

  public class Foo {

      private String bar = "bar";

      public String getBar() {
          return bar;
      }
  }
```

shrinking

```
✓ public class MainActivity extends AppCompatActivity {

✓     private String name = “name";

      @Override
✓     protected void onCreate(Bundle paramBundle) {
          String hello = StringUtil.getHello(name);
      }
  }

✓ public class StringUtil {

✓     static String helloPrefix = "Hello, “;
      static String goodbyePrefix = "Goodbye, “;

✓     public static String getHello(String paramString) {
          return helloPrefix + paramString;
      }
  }

  public class Foo {

      private String bar = "bar";

      public String getBar() {
          return bar;
      }
  }
```

shrinking

```
✓ public class MainActivity extends AppCompatActivity {

✓     private String name = “name";

      @Override
✓     protected void onCreate(Bundle paramBundle) {
          String hello = StringUtil.getHello(name);
      }
  }

✓ public class StringUtil {

✓     static String helloPrefix = "Hello, “;
✗     static String goodbyePrefix = "Goodbye, “;

✓     public static String getHello(String paramString) {
          return helloPrefix + paramString;
      }
  }

✗ public class Foo {

✗     private String bar = "bar";

✗     public String getBar() {
          return bar;
      }
  }
```

shrinking

```
✓ public class MainActivity extends AppCompatActivity {

✓     private String name = “name";

      @Override
✓     protected void onCreate(Bundle paramBundle) {
          String hello = StringUtil.getHello(name);
      }
  }

✓ public class StringUtil {

✓     static String helloPrefix = "Hello, “;
✗     static String goodbyePrefix = "Goodbye, “;

✓     public static String getHello(String paramString) {
          return helloPrefix + paramString;
      }
  }

✗ public class Foo {

✗     private String bar = "bar";

✗     public String getBar() {
          return bar;
      }
  }
```

shrinking

```
✓ public class MainActivity extends AppCompatActivity {

✓     private String name = “name";

      @Override
✓
```

usage.txt

```
com.jebware.demo.StringUtil:
    static String goodbyePrefix
com.jebware.demo.Foo
```

```
✓
✓     static String helloPrefix = "Hello, “;
✗     static String goodbyePrefix = "Goodbye, “;

✓     public static String getHello(String name) {
          return helloPrefix + name;
      }
  }

✗ public class Foo {

✗     private String bar = "bar";

✗     public String getBar() {
          return bar;
      }
  }
```

obfuscation

```
public class MainActivity extends AppCompatActivity {

    private String name = “name";

    @Override
    protected void onCreate(Bundle paramBundle) {
        String hello = StringUtil.getHello(name);
    }
}

public class StringUtil {

    static String helloPrefix = "Hello, “;

    public static String getHello(String paramString) {
        return helloPrefix + paramString;
    }
}
```

obfuscation

```
✓ public class MainActivity extends AppCompatActivity {

      private String name = “name";

      @Override
✓     protected void onCreate(Bundle paramBundle) {
          String hello = StringUtil.getHello(name);
      }
  }

  public class StringUtil {

      static String helloPrefix = "Hello, “;

      public static String getHello(String paramString) {
          return helloPrefix + paramString;
      }
  }
```

obfuscation

```
✓ public class MainActivity extends AppCompatActivity {
```

mapping.txt

```
com.jebware.demo.MainActivity -> com.jebware.demo.MainActivity:
    String name -> a
    void onCreate(android.os.Bundle) -> onCreate
com.jebware.demo.StringUtil -> com.jebware.demo.a
    String helloPrefix -> a
    String getHello(String) -> b
```

```
    public static String getHello(String name) {
        return helloPrefix + name;
    }
}
```

obfuscation

```
public class MainActivity extends AppCompatActivity {

    private String name = “name";

    @Override
    protected void onCreate(Bundle paramBundle) {
        String hello = StringUtil.getHello(name);
    }
}

public class StringUtil {

    static String helloPrefix = "Hello, “;

    public static String getHello(String paramString) {
        return helloPrefix + paramString;
    }
}
```

obfuscation

```
public class MainActivity extends AppCompatActivity {

    private String a = “name";

    @Override
    protected void onCreate(Bundle paramBundle) {
        String hello = StringUtil.getHello(name);
    }
}

public class a {

    static String a = "Hello, “;

    public static String b(String paramString) {
        return helloPrefix + paramString;
    }
}
```

obfuscation

```
public class MainActivity extends AppCompatActivity {

    private String a = “name";

    @Override
    protected void onCreate(Bundle paramBundle) {
        String hello = StringUtil.getHello(name);
    }
}

public class a {

    static String a = "Hello, “;

    public static String b(String paramString) {
        return helloPrefix + paramString;
    }
}
```

obfuscation

```
public class MainActivity extends AppCompatActivity {

    private String a = “name";

    @Override
    protected void onCreate(Bundle paramBundle) {
        String hello = StringUtil.getHello(name);
    }
}

public class a {

    static String a = "Hello, “;

    public static String b(String paramString) {
        return helloPrefix + paramString;
    }
}
```

obfuscation

```
public class MainActivity extends AppCompatActivity {

    private String a = “name";

    @Override
    protected void onCreate(Bundle paramBundle) {
        String hello = StringUtil.getHello(name);
    }
}

public class a {

    static String a = "Hello, “;

    public static String b(String paramString) {
        return helloPrefix + paramString;
    }
}
```

obfuscation

```
public class MainActivity extends AppCompatActivity {

    private String a = “name";

    @Override
    protected void onCreate(Bundle paramBundle) {
        String hello = StringUtil.getHello(name);
    }
}

public class a {

    static String a = "Hello, “;

    public static String b(String paramString) {
        return helloPrefix + paramString;
    }
}
```

obfuscation

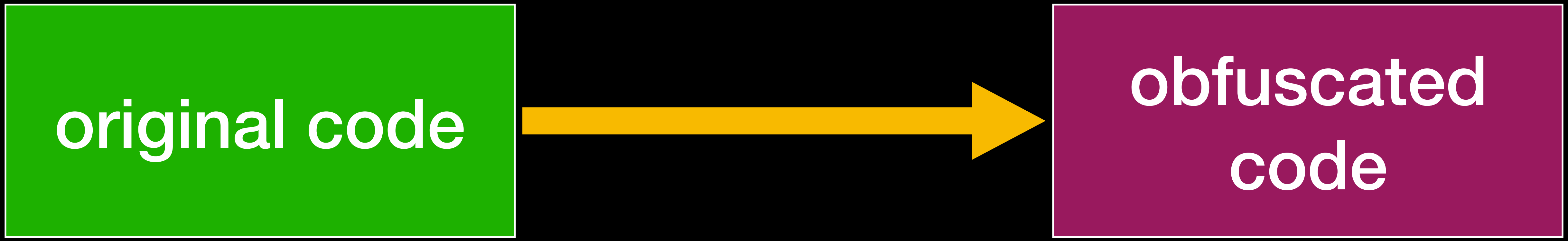
```java
public class MainActivity extends AppCompatActivity {

    private String a = “name";

    @Override
    protected void onCreate(Bundle paramBundle) {
        String hello = a.b(a);
    }
}

public class a {

    static String a = "Hello, “;

    public static String b(String paramString) {
        return a + paramString;
    }
}
```

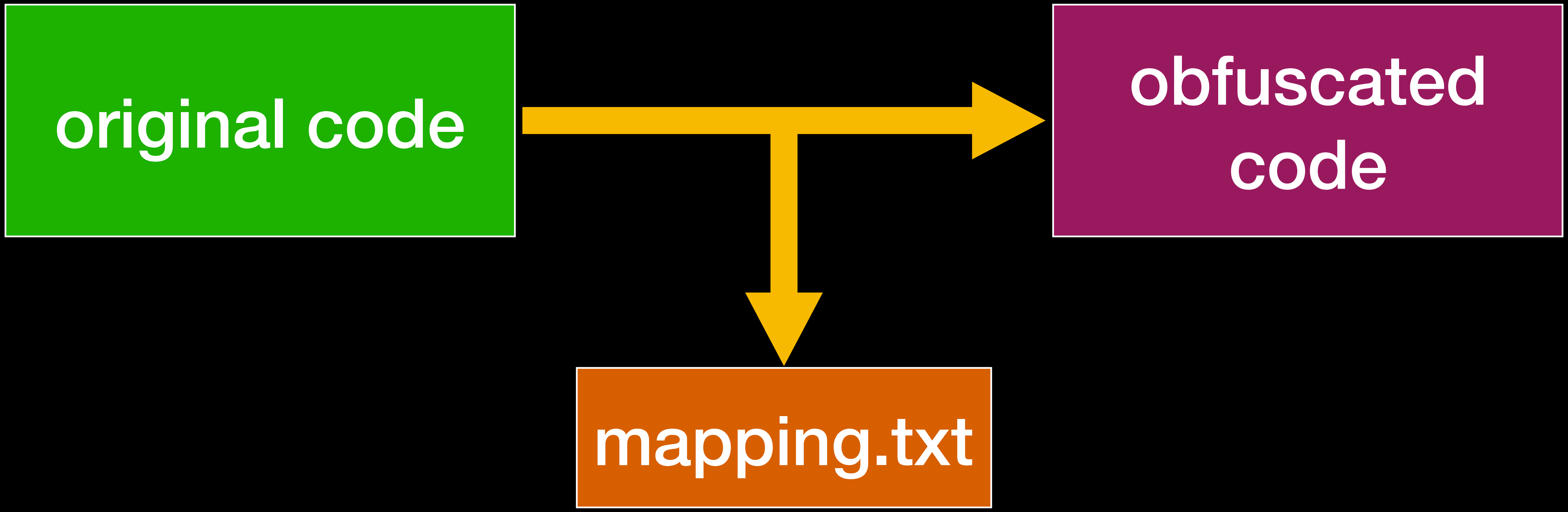
obfuscated stacktrace

```
java.io.IOException: Can't read [dummy.jar] (No such file or directory)
   at proguard.y.a(MyApplication:188)
   at proguard.y.a(MyApplication:158)
   at proguard.y.a(MyApplication:136)
   at proguard.y.a(MyApplication:66)
   at proguard.ProGuard.c(MyApplication:218)
   at proguard.ProGuard.a(MyApplication:82)
   at proguard.ProGuard.main(MyApplication:538)
Caused by: java.io.IOException: No such file or directory
   at proguard.d.q.a(MyApplication:50)
   at proguard.y.a(MyApplication:184)
   ... 6 more
```

de-obfuscated stacktrace

```
java.io.IOException: Can't read [dummy.jar] (No such file or directory)
   at proguard.InputReader.readInput(InputReader.java:188)
   at proguard.InputReader.readInput(InputReader.java:158)
   at proguard.InputReader.readInput(InputReader.java:136)
   at proguard.InputReader.execute(InputReader.java:66)
   at proguard.ProGuard.readInput(ProGuard.java:218)
   at proguard.ProGuard.execute(ProGuard.java:82)
   at proguard.ProGuard.main(ProGuard.java:538)
Caused by: java.io.IOException: No such file or directory
   at proguard.io.DirectoryPump.pumpDataEntries(DirectoryPump.java:50)
   at proguard.InputReader.readInput(InputReader.java:184)
   ... 6 more
```

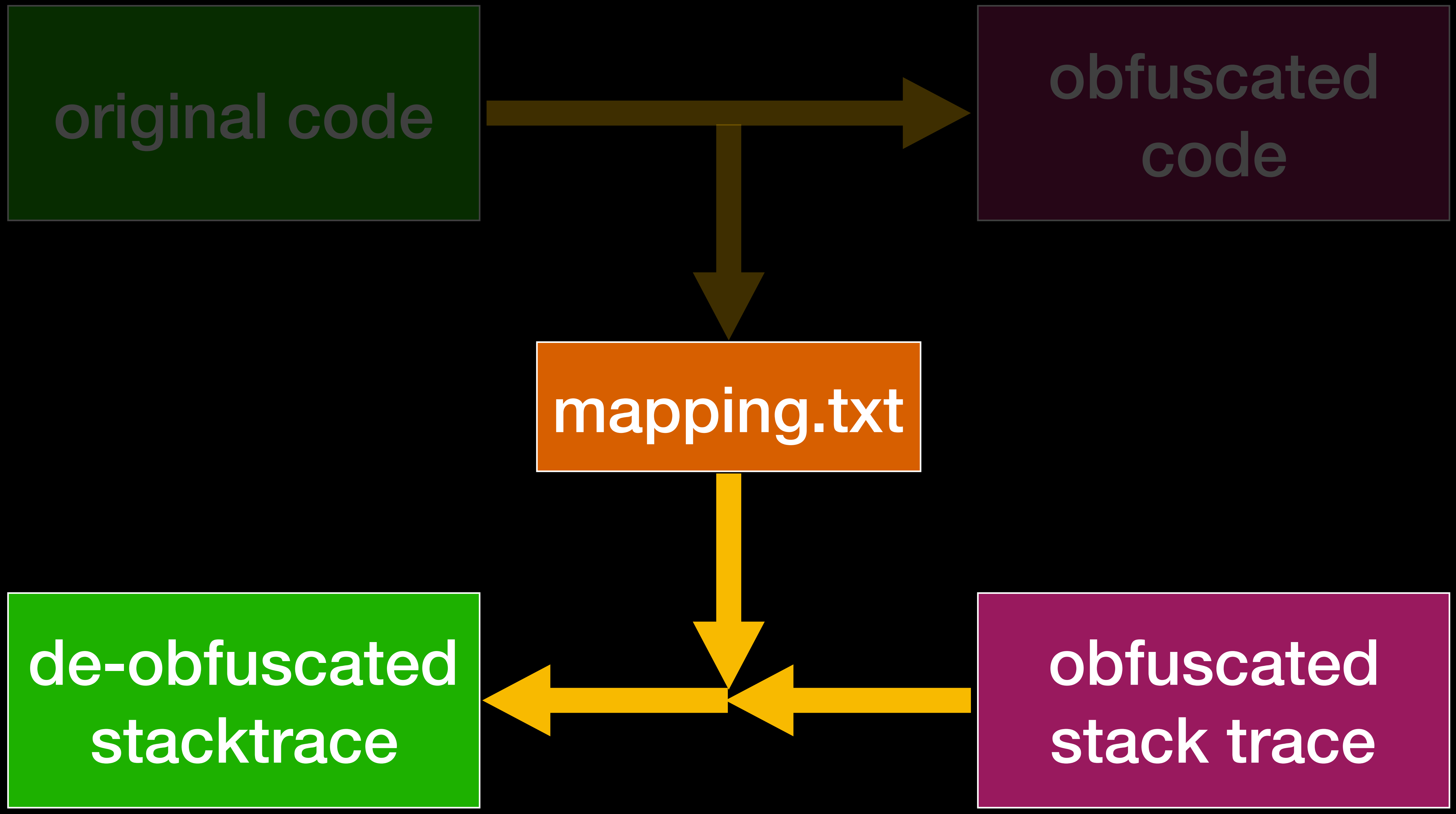

original code
obfuscated code



original code

obfuscated code

mapping.txt



original code

obfuscated code

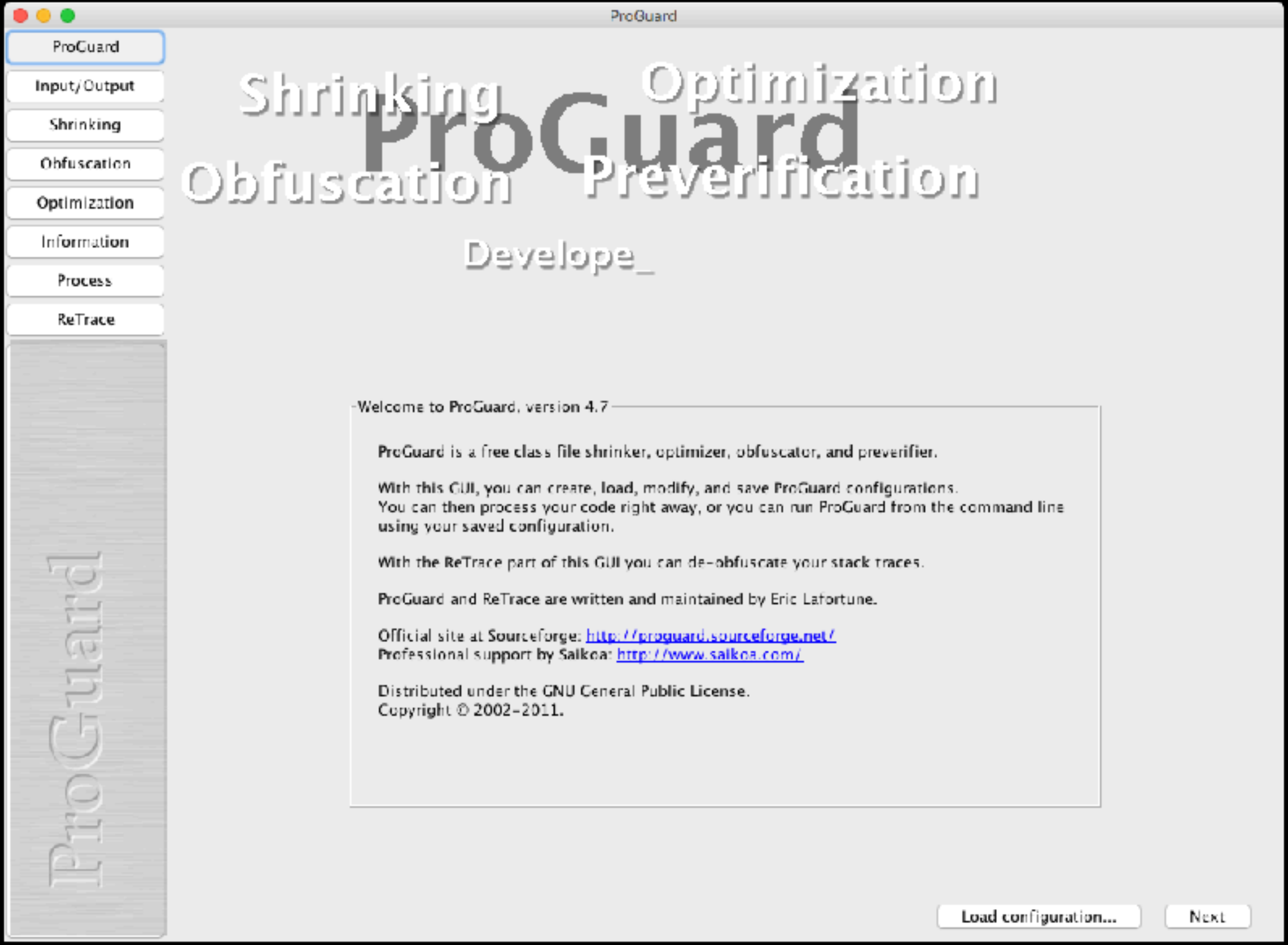
mapping.txt

de-obfuscated stacktrace

obfuscated stack trace

retrace.sh

```
$ANDROID_SDK/tools/proguard/bin/

retrace.sh mapping.txt stacktrace.txt
```



Welcome to ProGuard, version 4.7

ProGuard is a free class file shrinker, optimizer, obfuscator, and preverifier.

With this GUI, you can create, load, modify, and save ProGuard configurations.
You can then process your code right away, or you can run ProGuard from the command line using your saved configuration.

With the ReTrace part of this GUI you can de-obfuscate your stack traces.

ProGuard and ReTrace are written and maintained by Eric Lafortune.

Official site at Sourceforge: http://proguard.sourceforge.net/
Professional support by Saikoa: http://www.saikoa.com/

Distributed under the GNU General Public License.
Copyright © 2002-2011.

```
$ANDROID_SDK/tools/proguard/bin/proguardgui.sh
```

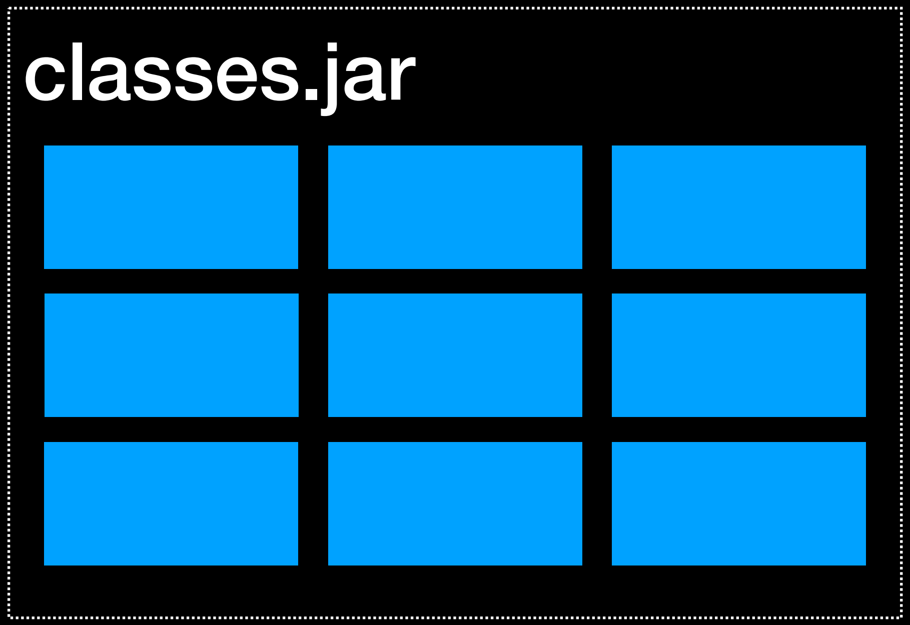
obfuscation

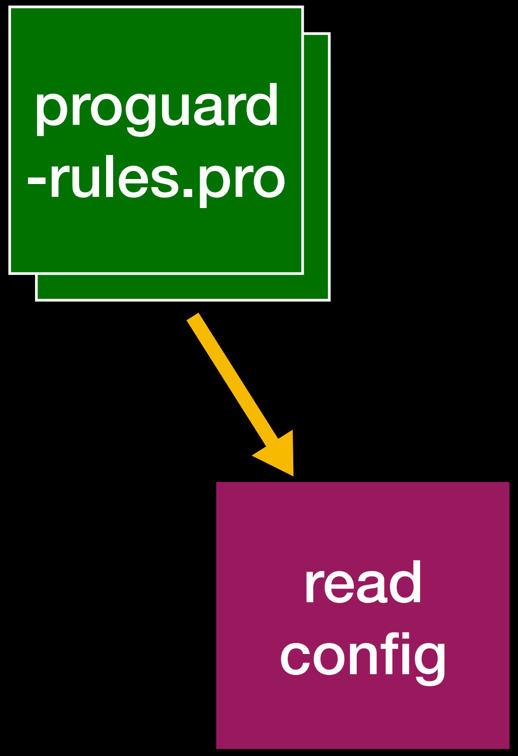
```
public class MainActivity extends AppCompatActivity {

    private String a = “name";

    @Override
    protected void onCreate(Bundle paramBundle) {
        String hello = a.b(a);
    }
}

public class a {

    static String a = "Hello, “;

    public static String b(String paramString) {
        return a + paramString;
    }
}
```

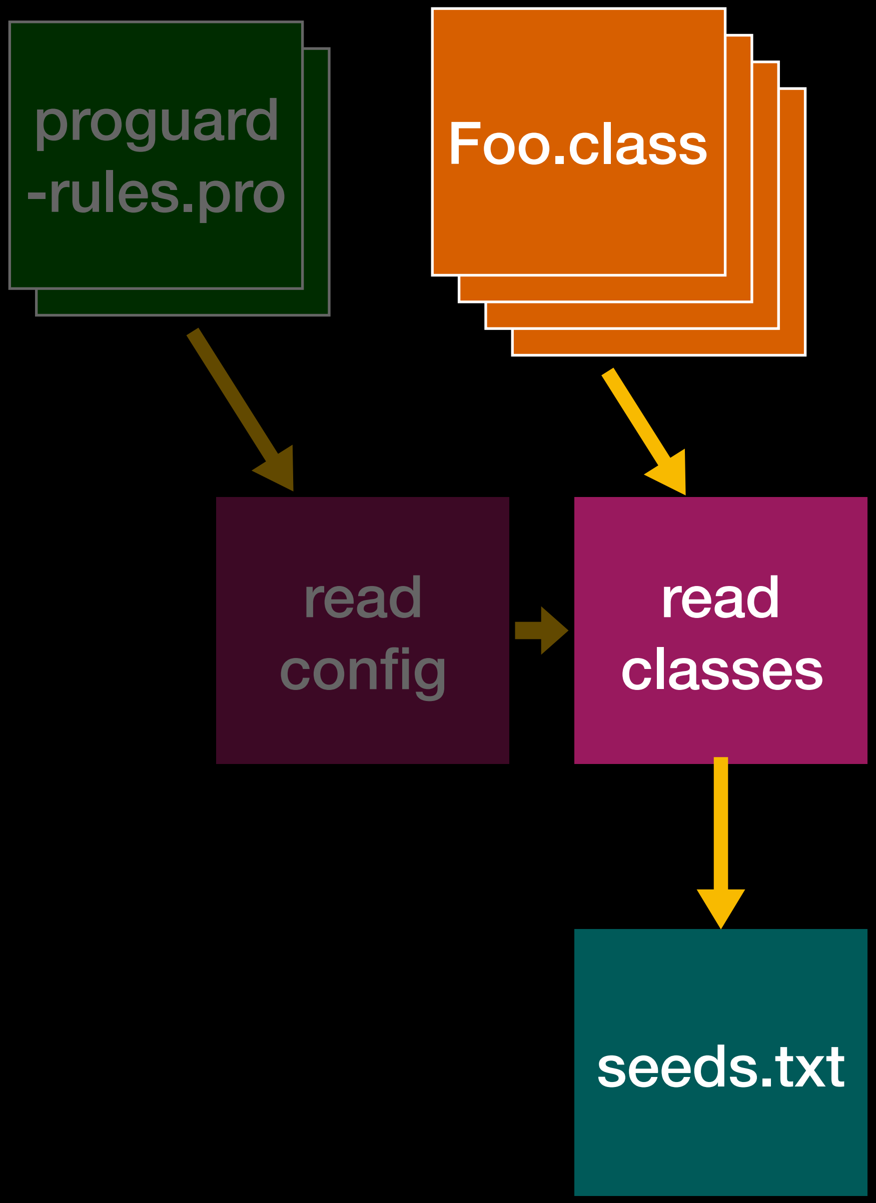
dump.txt

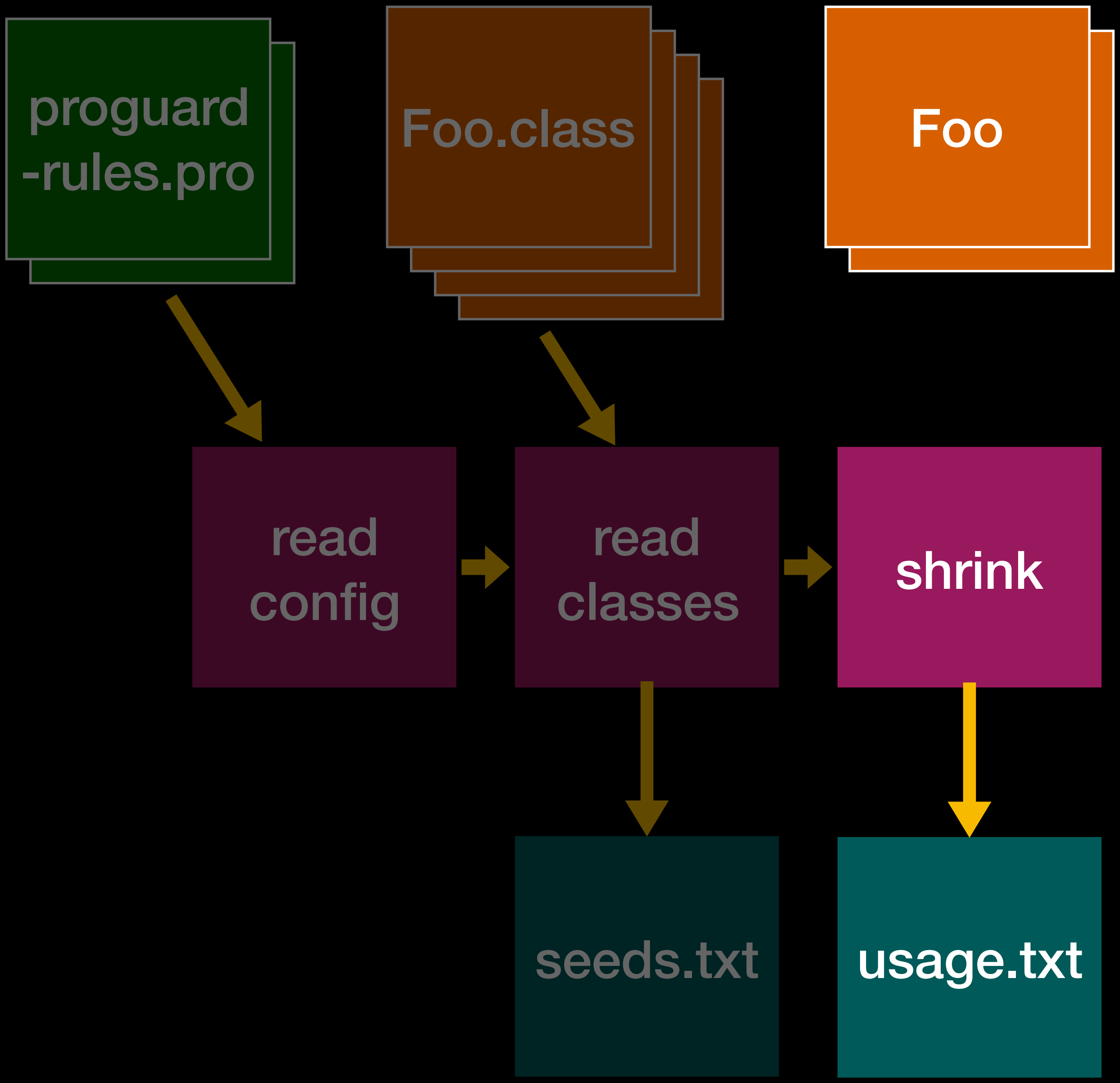
```
_____________________________________________________________________
- Program class: com/jebware/demo/a/a/a
  Superclass:    java/lang/Object
  Major version: 0x33
  Minor version: 0x0
    = target 1.7
  Access flags:  0x601
    = public interface com.jebware.demo.a.a.a extends java.lang.Object

Interfaces (count = 0):

Constant Pool (count = 12):
  - Class [com/jebware/demo/a/a/a]
  - Class [java/lang/Object]
  - Utf8 [()Lc/b;]
  - Utf8 [()Lc/b<Lcom/jebware/demo/a/b/a;>;]
  - Utf8 [/demo/hello.json]
  - Utf8 [Lc/b/f;]
  - Utf8 [RuntimeVisibleAnnotations]
  - Utf8 [Signature]
  - Utf8 [a]
  - Utf8 [com/jebware/demo/a/a/a]
  - Utf8 [java/lang/Object]

Fields (count = 0):

Methods (count = 1):
  - Method:       a()Lc/b;
    Access flags: 0x401
      = public abstract c.b a()
    Class member attributes (count = 2):
    - Runtime visible annotations attribute:
      - Annotation [Lc/b/f;]:
        - Constant element value [a 's']
          - Utf8 [/demo/hello.json]
    - Signature attribute:
      - Utf8 [()Lc/b<Lcom/jebware/demo/a/b/a;>;]

Class file attributes (count = 0):
```

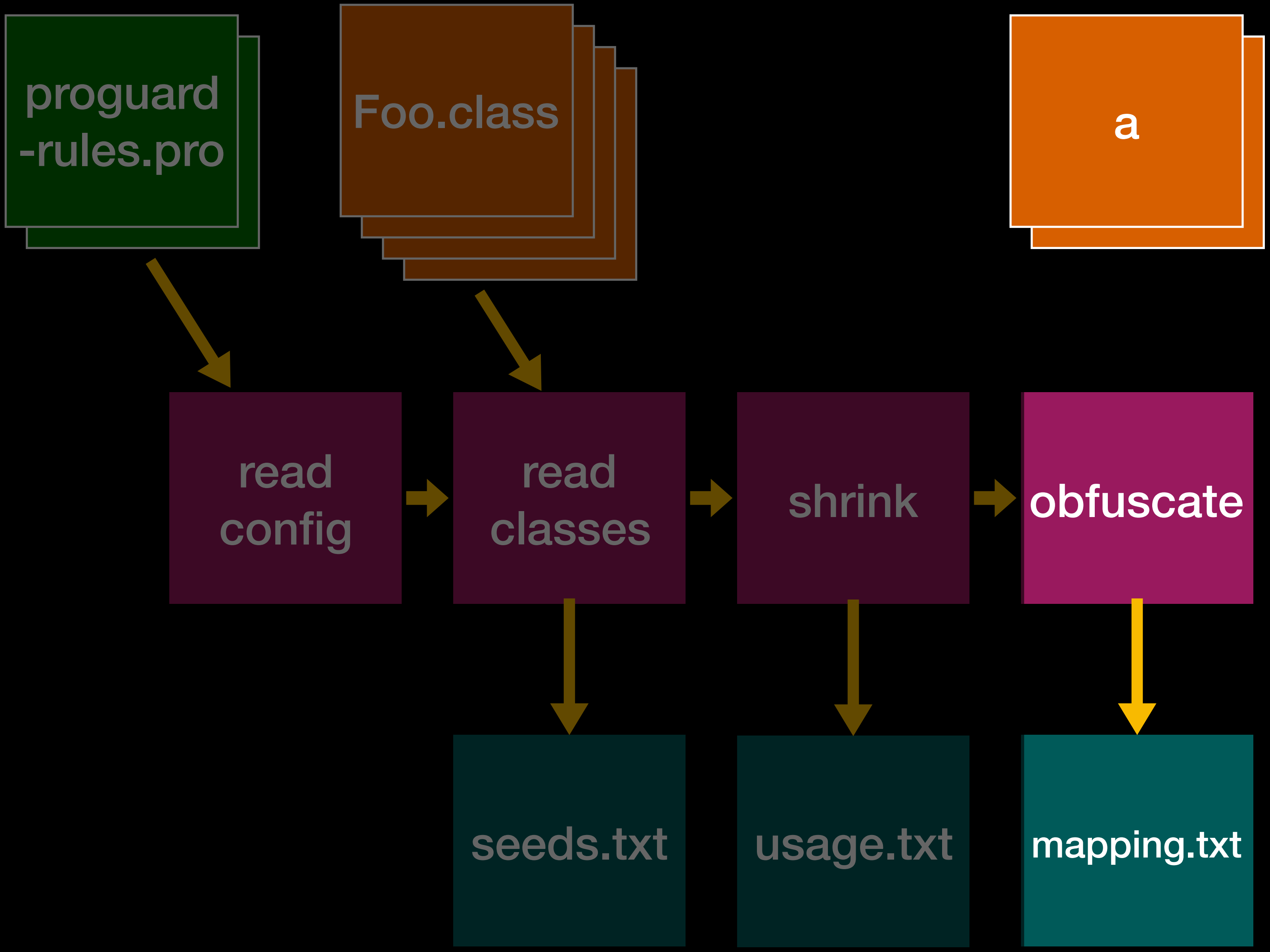
write out to .jar

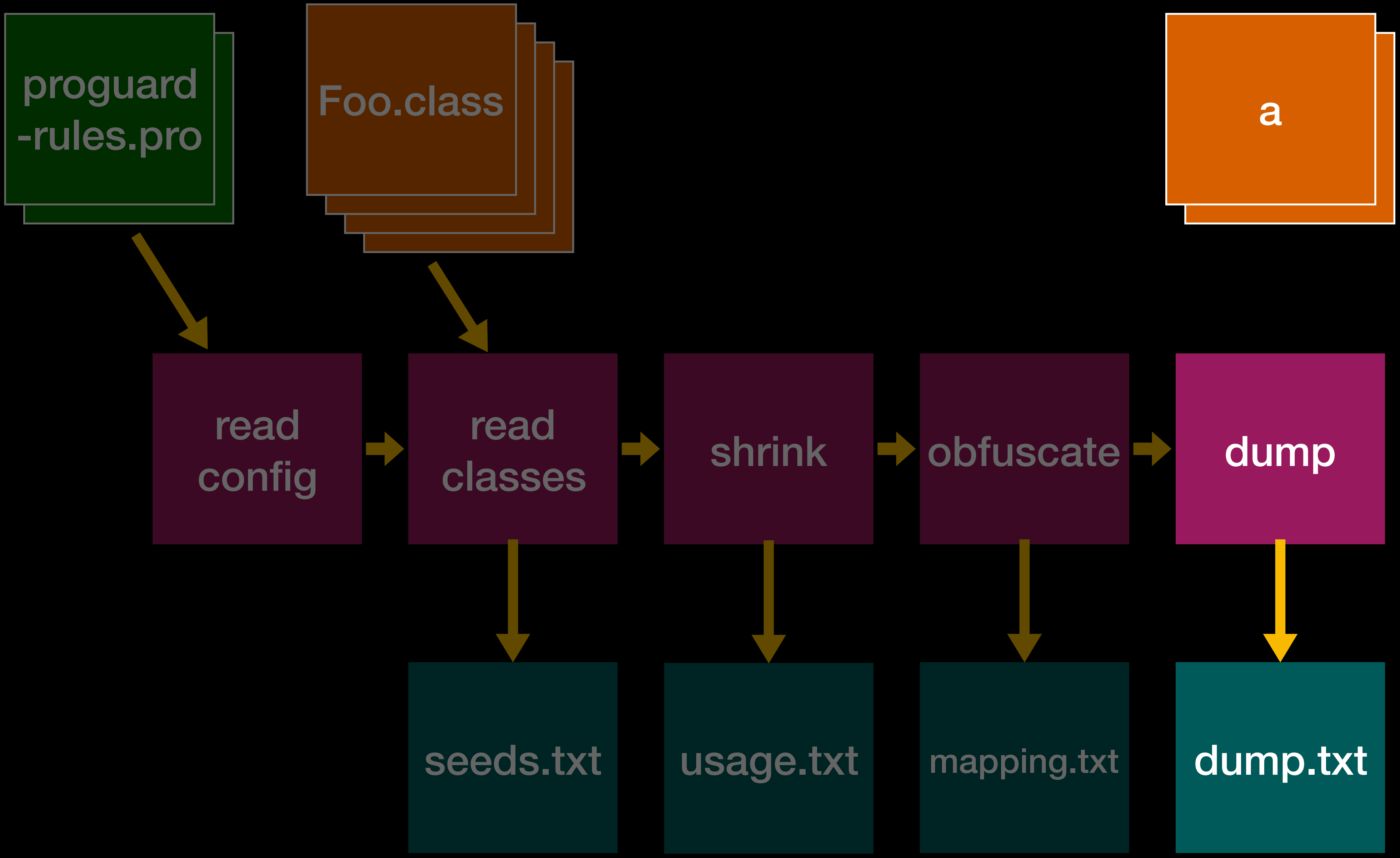


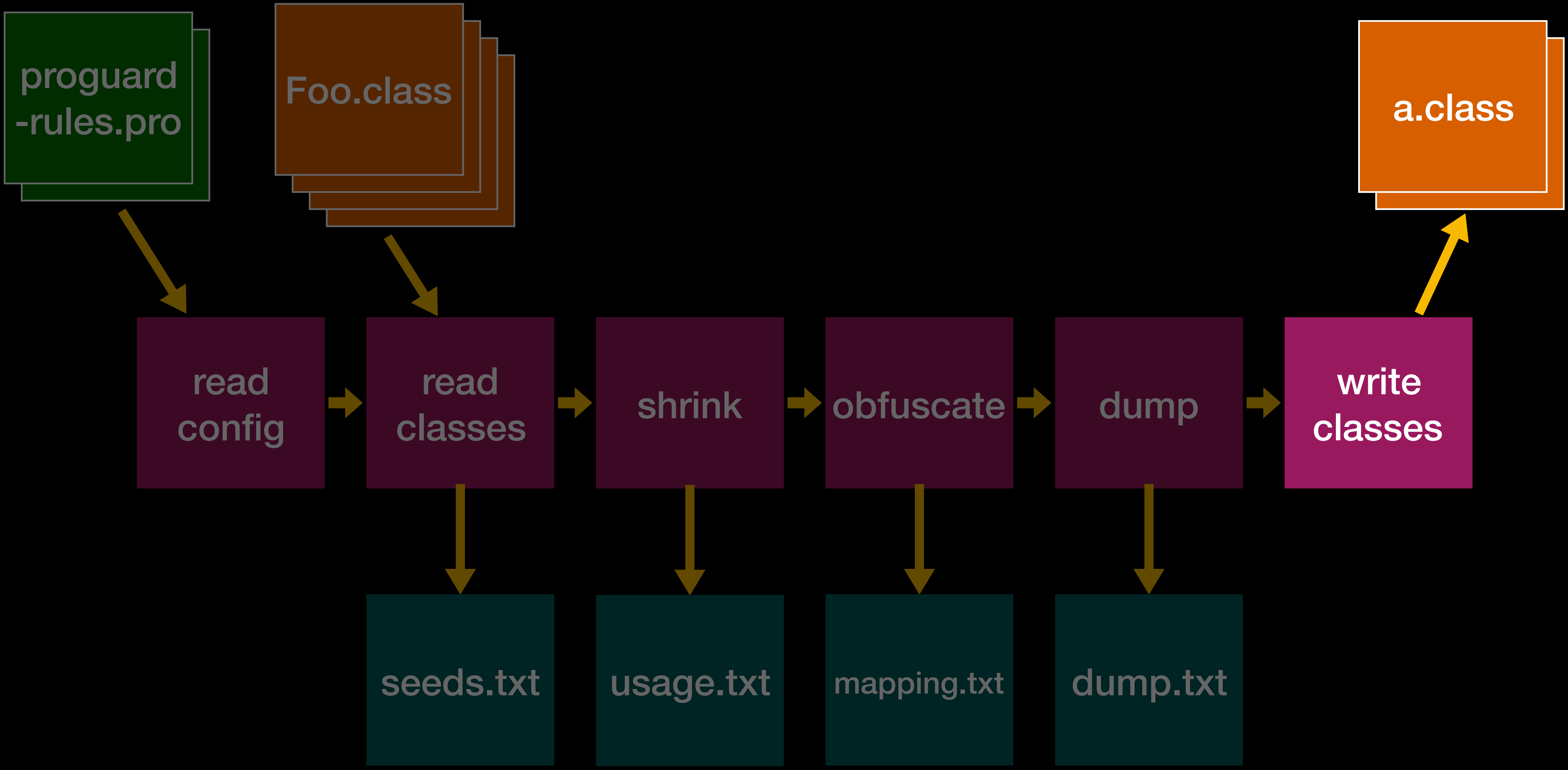

proguard
-rules.pro
read
config

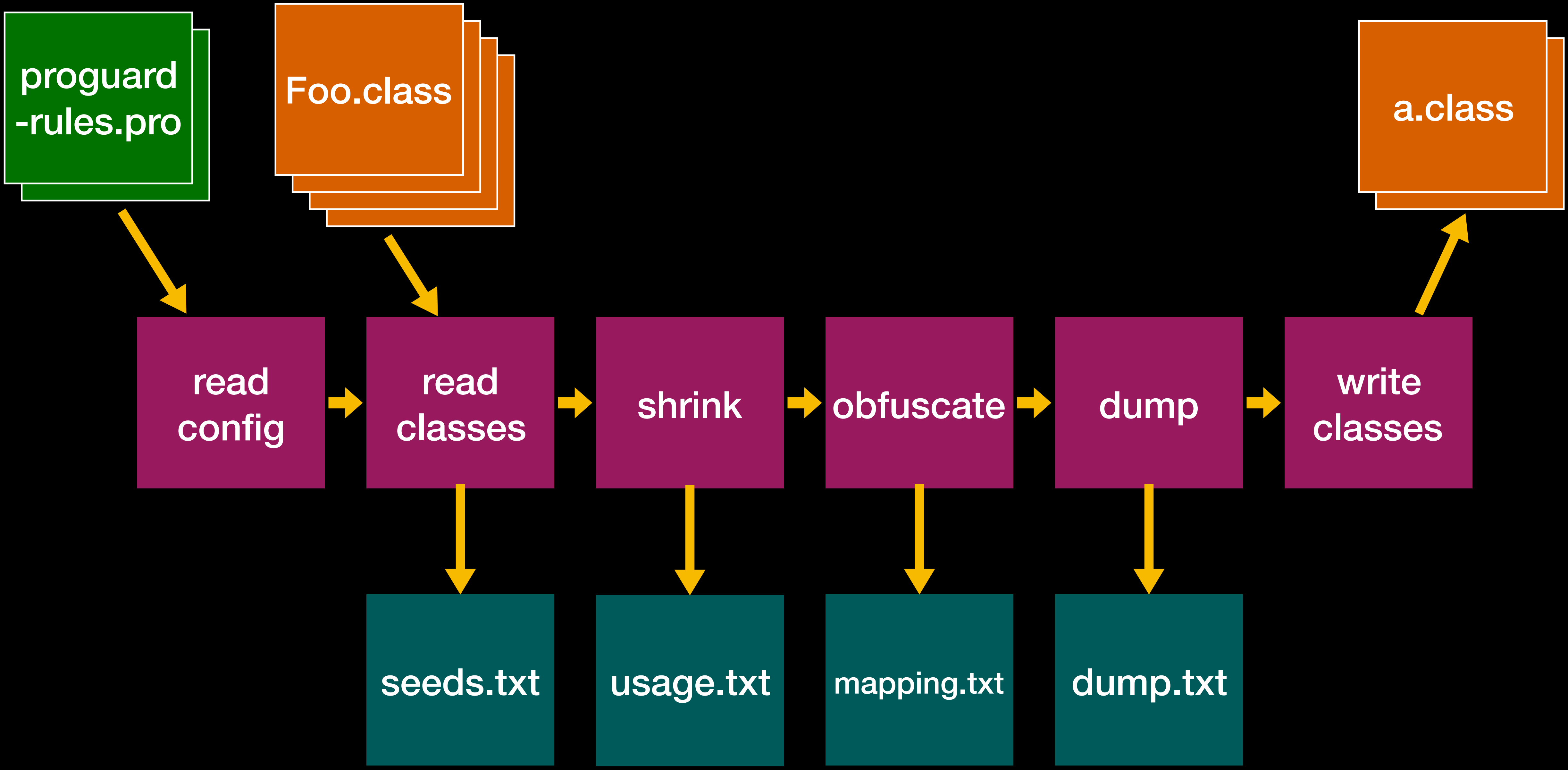

proguard
-rules.pro
Foo.class
read
config
read
classes
seeds.txt


proguard
-rules.pro
Foo.class
Foo
read config
read classes
shrink
seeds.txt
usage.txt


proguard-rules.pro
Foo.class
a
read config
read classes
shrink
obfuscate
seeds.txt
usage.txt
mapping.txt


proguard
-rules.pro
Foo.class
a
read
config
read
classes
shrink
obfuscate
dump
seeds.txt
usage.txt
mapping.txt
dump.txt


proguard -rules.pro
Foo.class
a.class
read config
read classes
shrink
obfuscate
dump
write classes
seeds.txt
usage.txt
mapping.txt
dump.txt


proguard -rules.pro
Foo.class
a.class
read config
read classes
shrink
obfuscate
dump
write classes
seeds.txt
usage.txt
mapping.txt
dump.txt

keep rules

```
-keepclassmembers public class * extends android.view.View {
    void set*(***);
    *** get*();
}

-keepclasseswithmembernames class * {
    native <methods>;
}

-keep @android.support.annotation.Keep class * {*;}
```

https://www.guardsquare.com/en/proguard/manual/introduction

keep rules

```
-keep
-keepclassmembers
-keepnames
-keepclassmembernames
-keepclasseswithmembers
-keepclasseswithmembernames
```

(no rule)

	classes	members
shrink	✓	✓
obfuscate	✓	✓

-keep

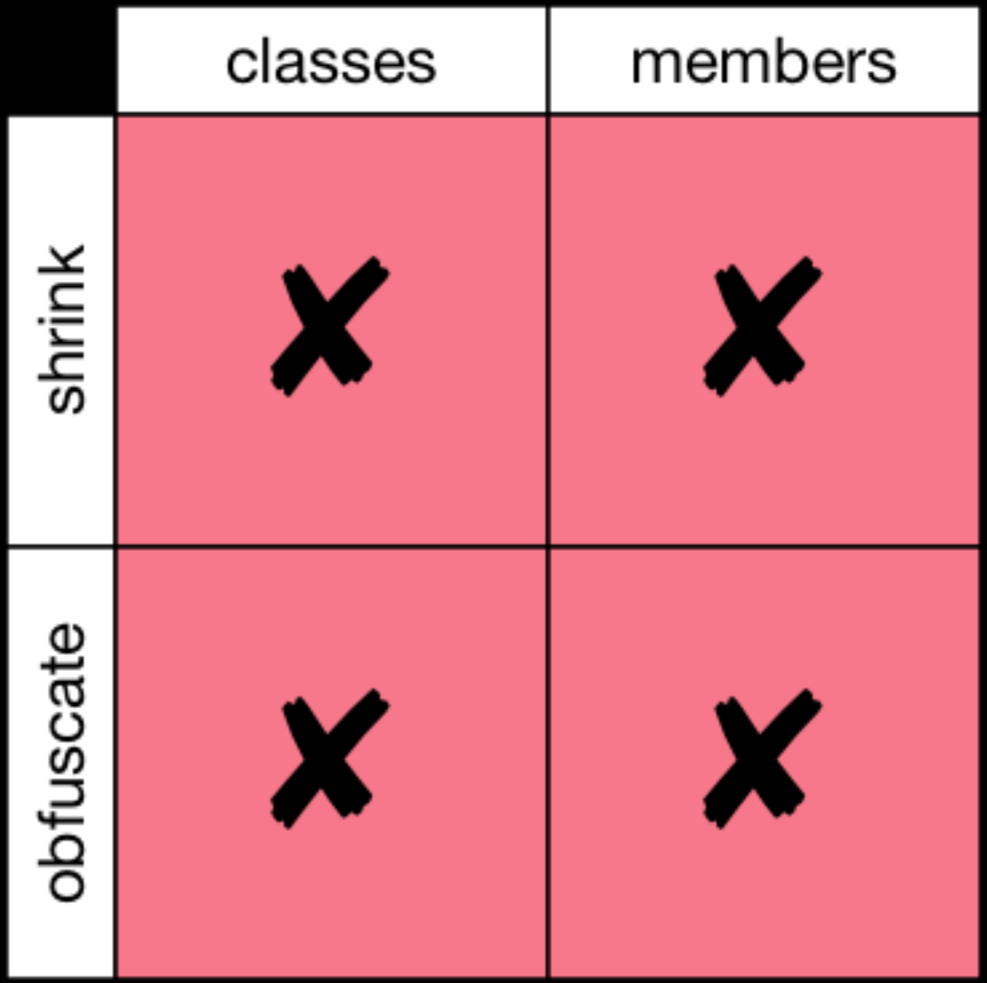
	classes	members
shrink	✗	✗
obfuscate	✗	✗

-keep

	classes	members
shrink	✗	✗
obfuscate	✗	✗

```
-keep @android.support.annotation.Keep class * {
    *;
}
```

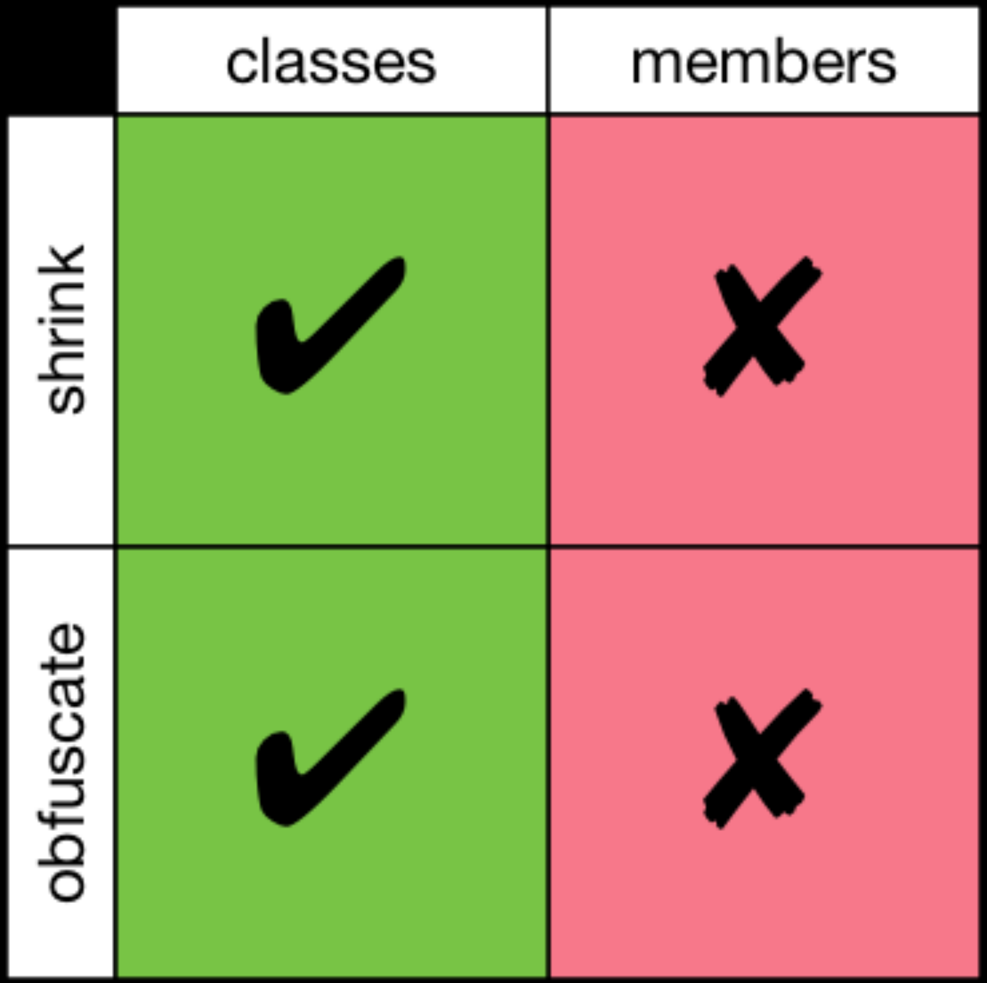
-keepclasseswithmembers



	classes	members
shrink	✗	✗
obfuscate	✗	✗

```
-keepclasseswithmembers class * {
    @android.support.annotation.Keep <methods>;
}
```

-keepclassmembers



	classes	members
shrink	✓	✗
obfuscate	✓	✗

```
-keepclassmembers class * implements android.os.Parcelable {
    public static final ** CREATOR;
}
```

-keepnames

	classes	members
shrink	✔	✔
obfuscate	✘	✘

also, -keepclasseswithmembernames

-keepclassmembernames

	classes	members
shrink	✓	✓
obfuscate	✓	✗

```
-keepclassmembernames class com.example.models.** {
    !static !transient <fields>;
}
```

-addconfigurationdebugging

-addconfigurationdebugging

```
ProGuard: The class 'com.google.gson.internal.bind
.ReflectiveTypeAdapterFactory'
is calling Class.getDeclaredFields on class
'com.jebware.demo.service.model.HelloResult' to
retrieve its fields.

You might consider preserving all fields with their
original names, with a setting like:

-keepclassmembers class
    com.jebware.demo.service.model.HelloResult {

    <fields>;
}
```

-addconfigurationdebugging

ProGuard v6

AGP v3.2.0

-addconfigurationdebugging

ProGuard v6

AGP v3.2.0

!!ONLY IN DEBUG BUILDS!!

config

config

- `proguardFiles ‘proguard-rules.pro’`

config

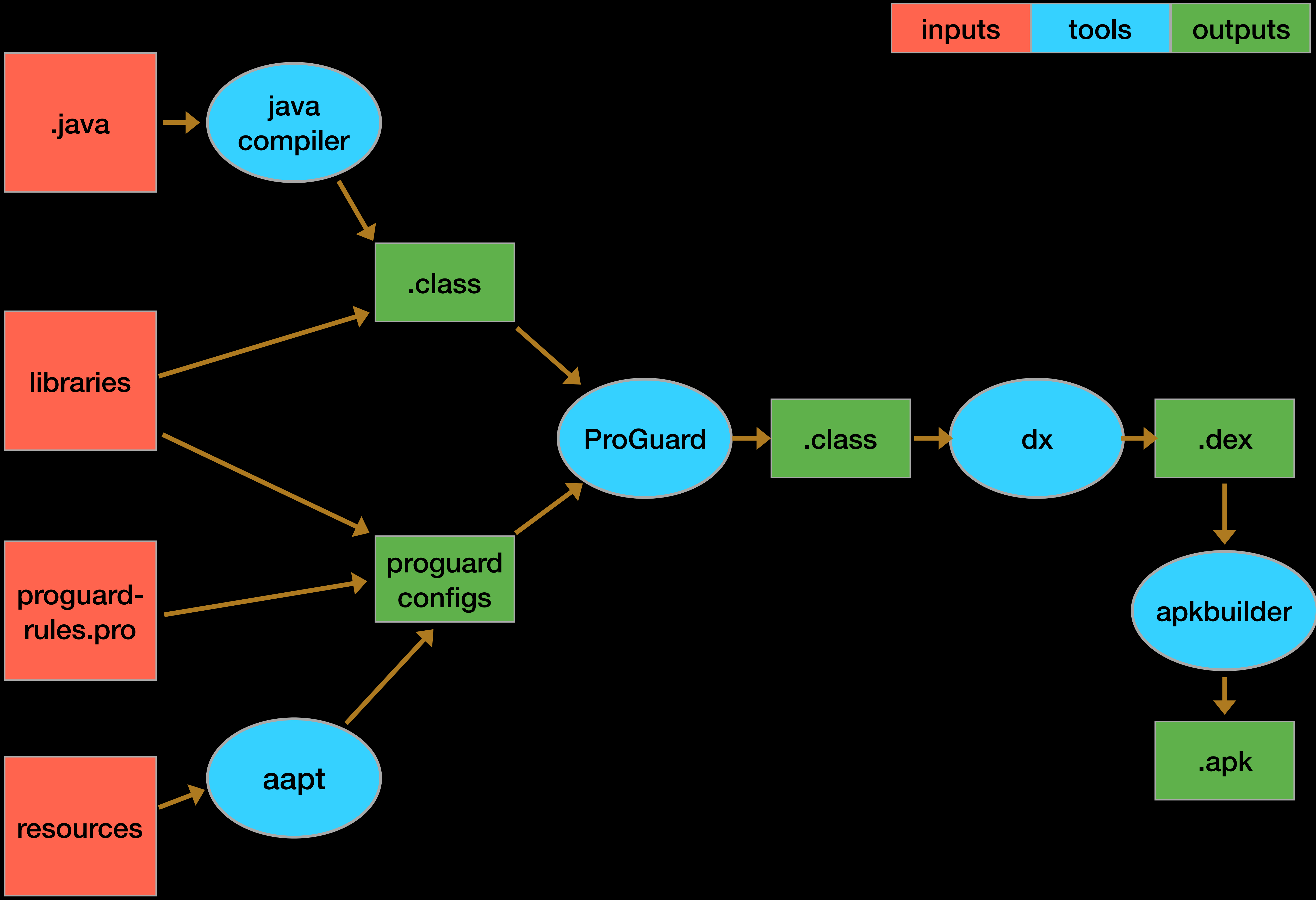
- `proguardFiles ‘proguard-rules.pro’`
- `proguardFiles getDefaultProguardFile(‘proguard-android.txt’)`

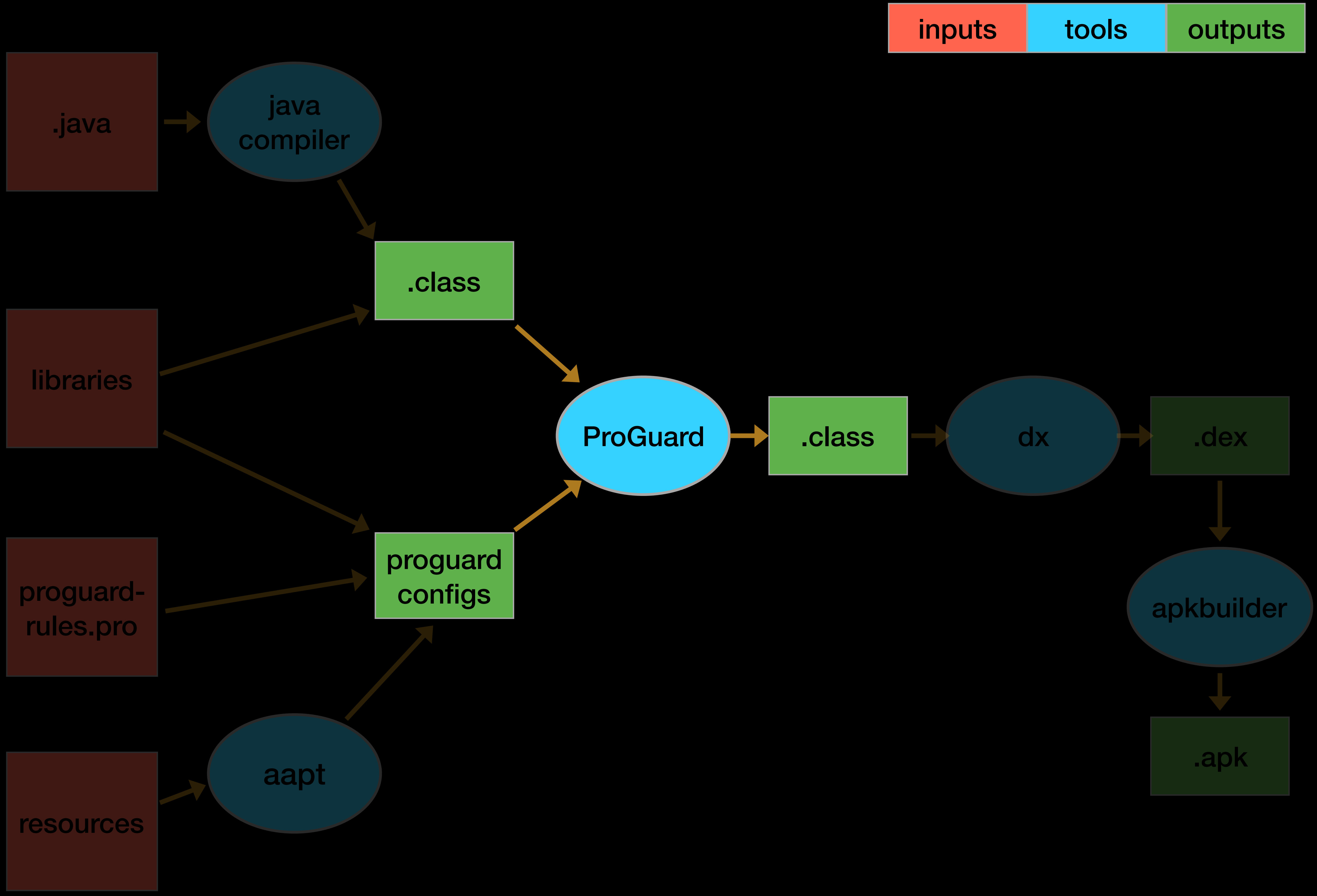
config

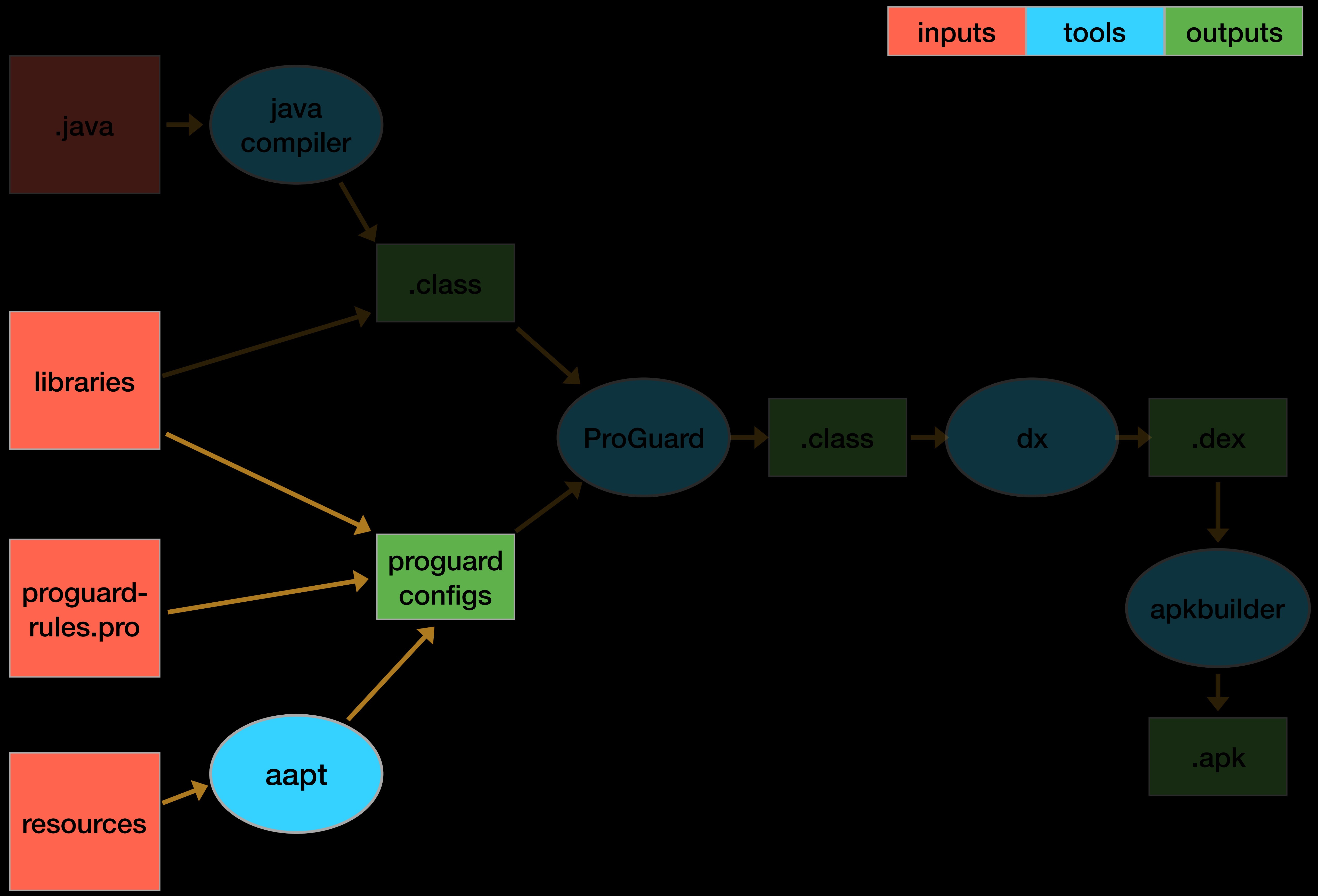
- `proguardFiles ‘proguard-rules.pro’`
- `proguardFiles getDefaultProguardFile(‘proguard-android.txt’)`
- Libraries `consumerProguardFiles ‘proguard-rules.pro’`

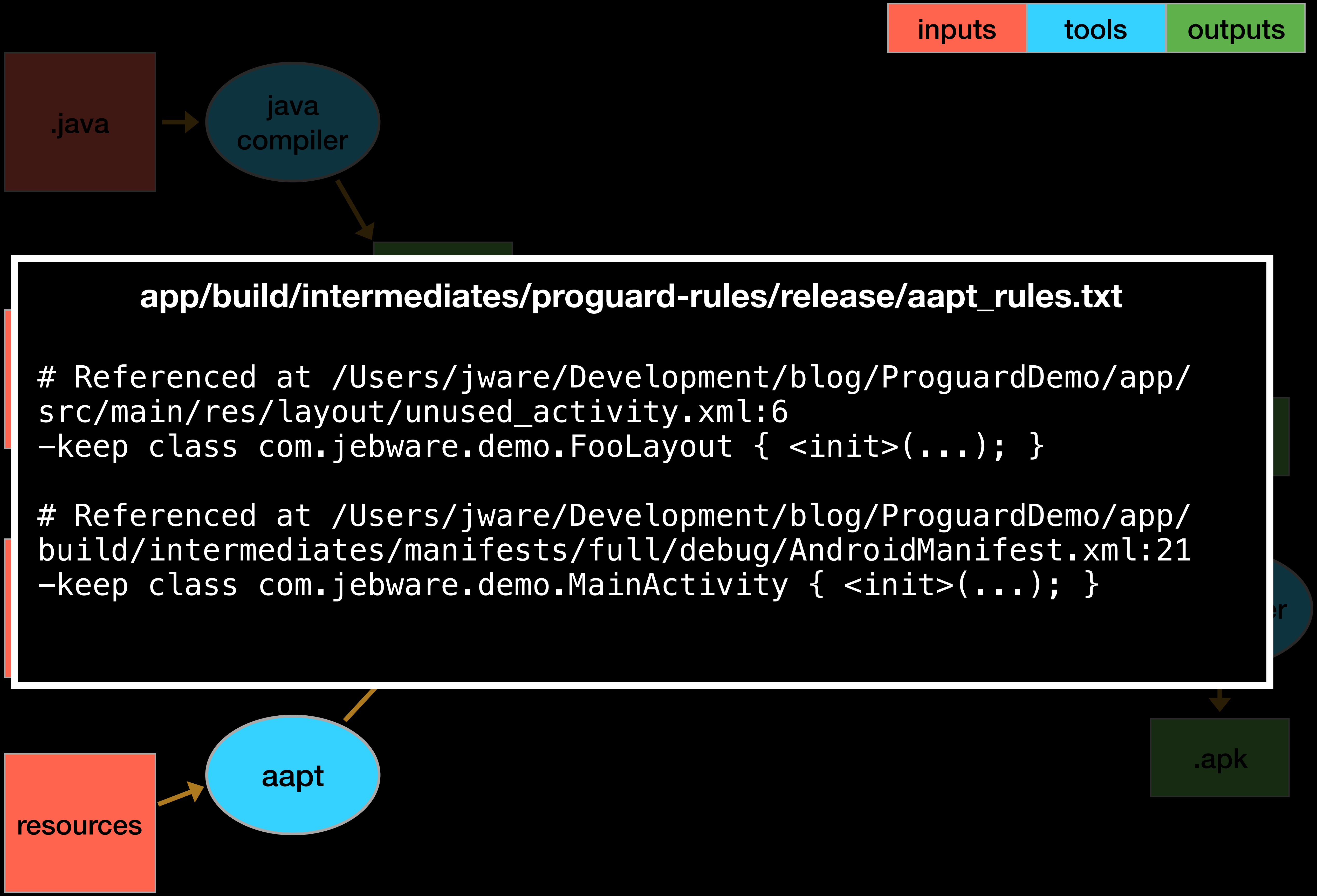
config

- `proguardFiles ‘proguard-rules.pro’`
- `proguardFiles getDefaultProguardFile(‘proguard-android.txt’)`
- Libraries
  `consumerProguardFiles ‘proguard-rules.pro’`
- AAPT


inputs
tools
outputs
.java
java compiler
.class
libraries
ProGuard
.class
dx
.dex
proguard-rules.pro
proguard configs
apkbuilder
resources
aapt
.apk

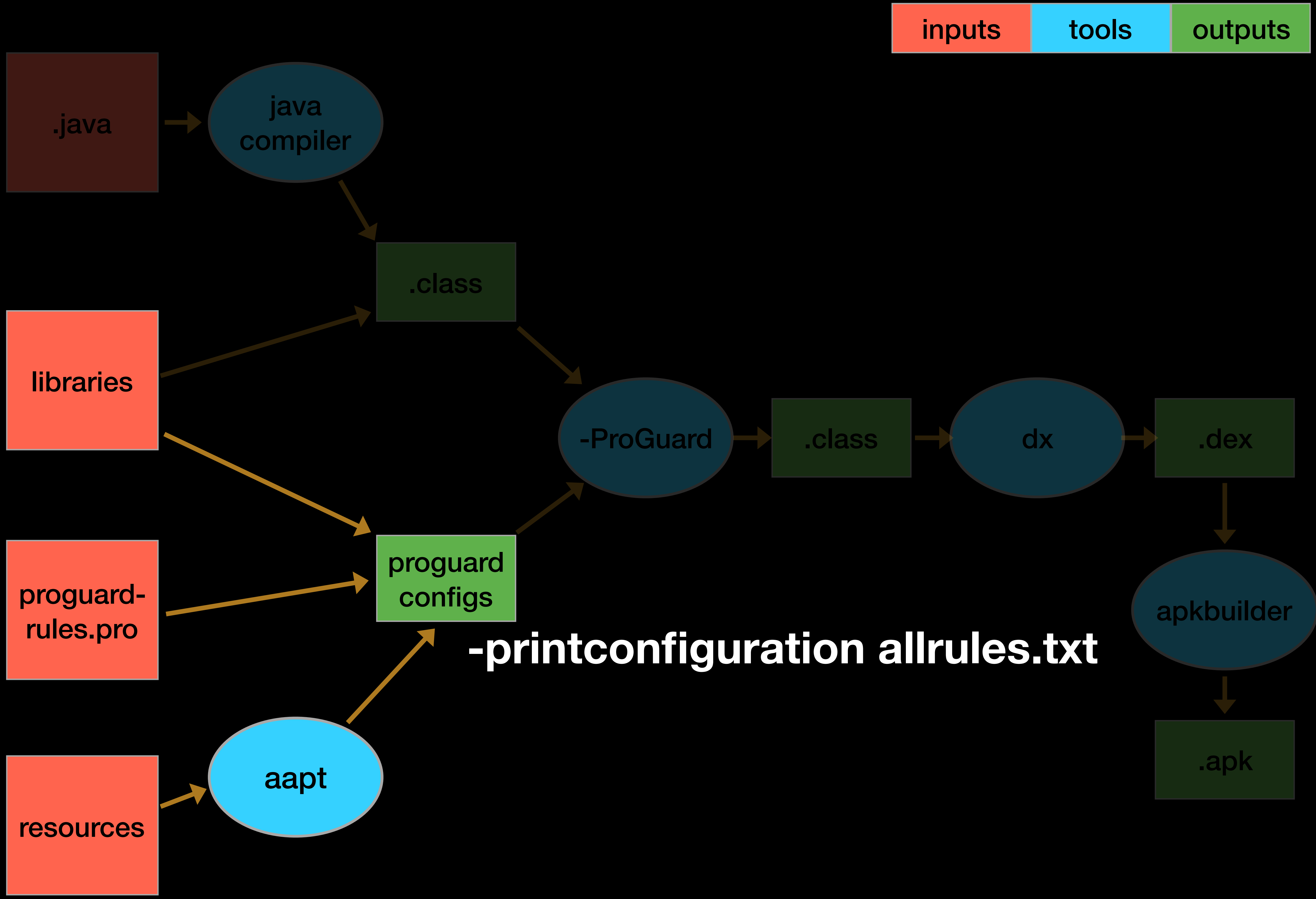

inputs
tools
outputs
.java
java compiler
.class
libraries
ProGuard
.class
dx
.dex
proguard-rules.pro
proguard configs
apkbuilder
resources
aapt
.apk


inputs
tools
outputs
.java
java compiler
.class
libraries
ProGuard
.class
dx
.dex
proguard configs
proguard-rules.pro
apkbuilder
aapt
resources
.apk

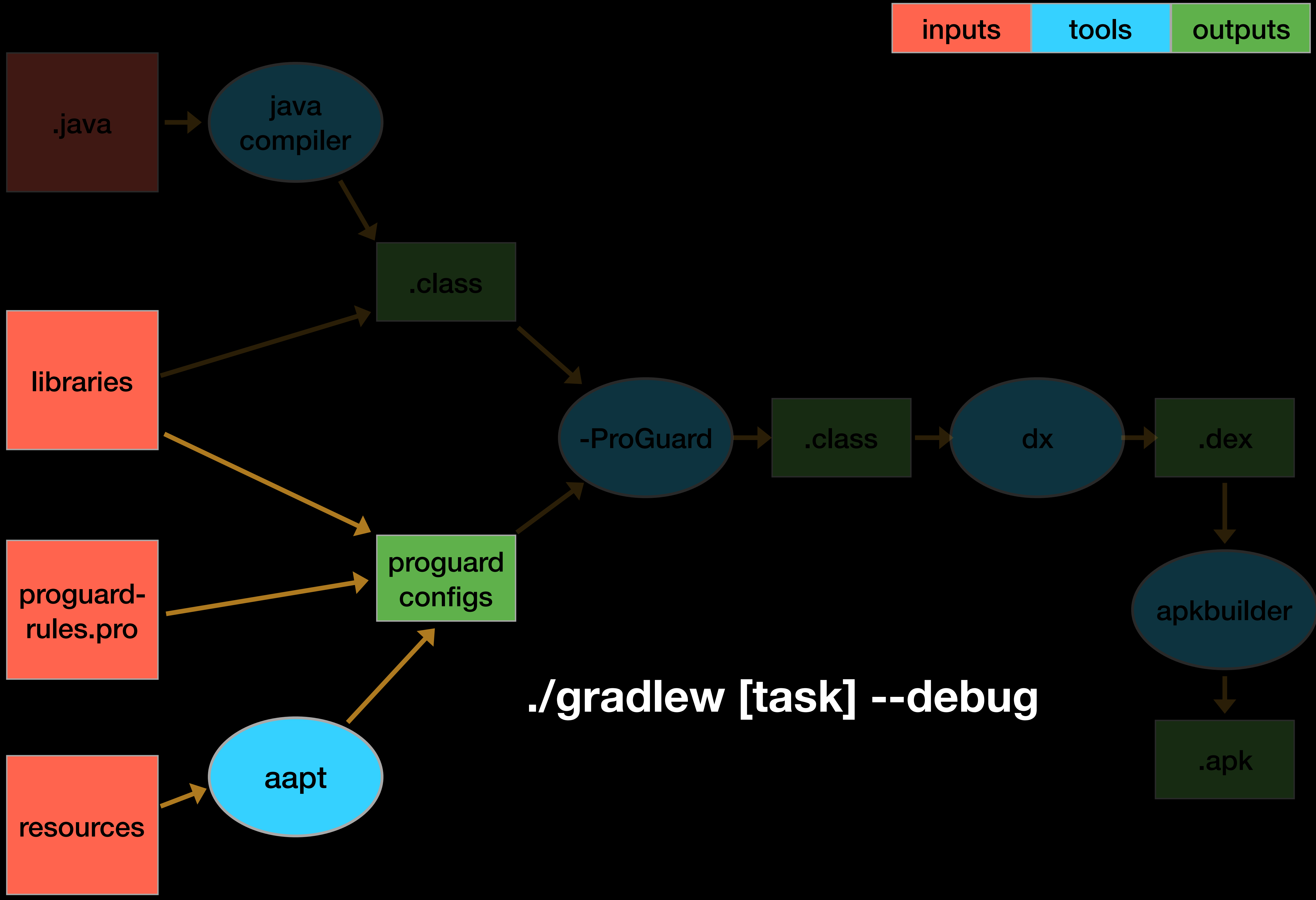


app/build/intermediates/proguard-rules/release/aapt_rules.txt

```
# Referenced at /Users/jware/Development/blog/ProguardDemo/app/
src/main/res/layout/unused_activity.xml:6
-keep class com.jebware.demo.FooLayout { <init>(...); }

# Referenced at /Users/jware/Development/blog/ProguardDemo/app/
build/intermediates/manifests/full/debug/AndroidManifest.xml:21
-keep class com.jebware.demo.MainActivity { <init>(...); }
```



-printconfiguration allrules.txt



inputs

tools

outputs

.java

java compiler

.class

libraries

-ProGuard

.class

dx

.dex

proguard-rules.pro

proguard configs

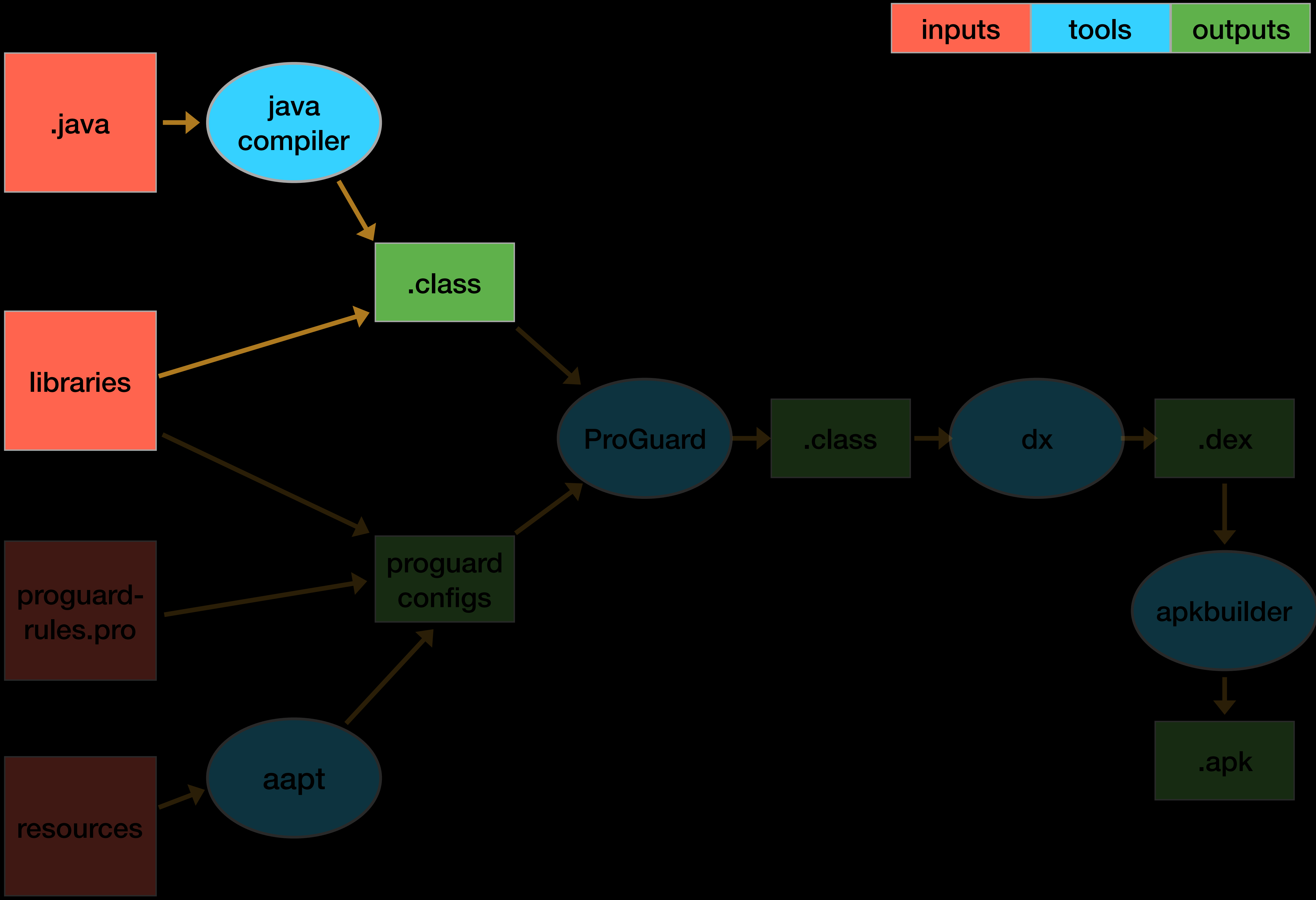
apkbuilder

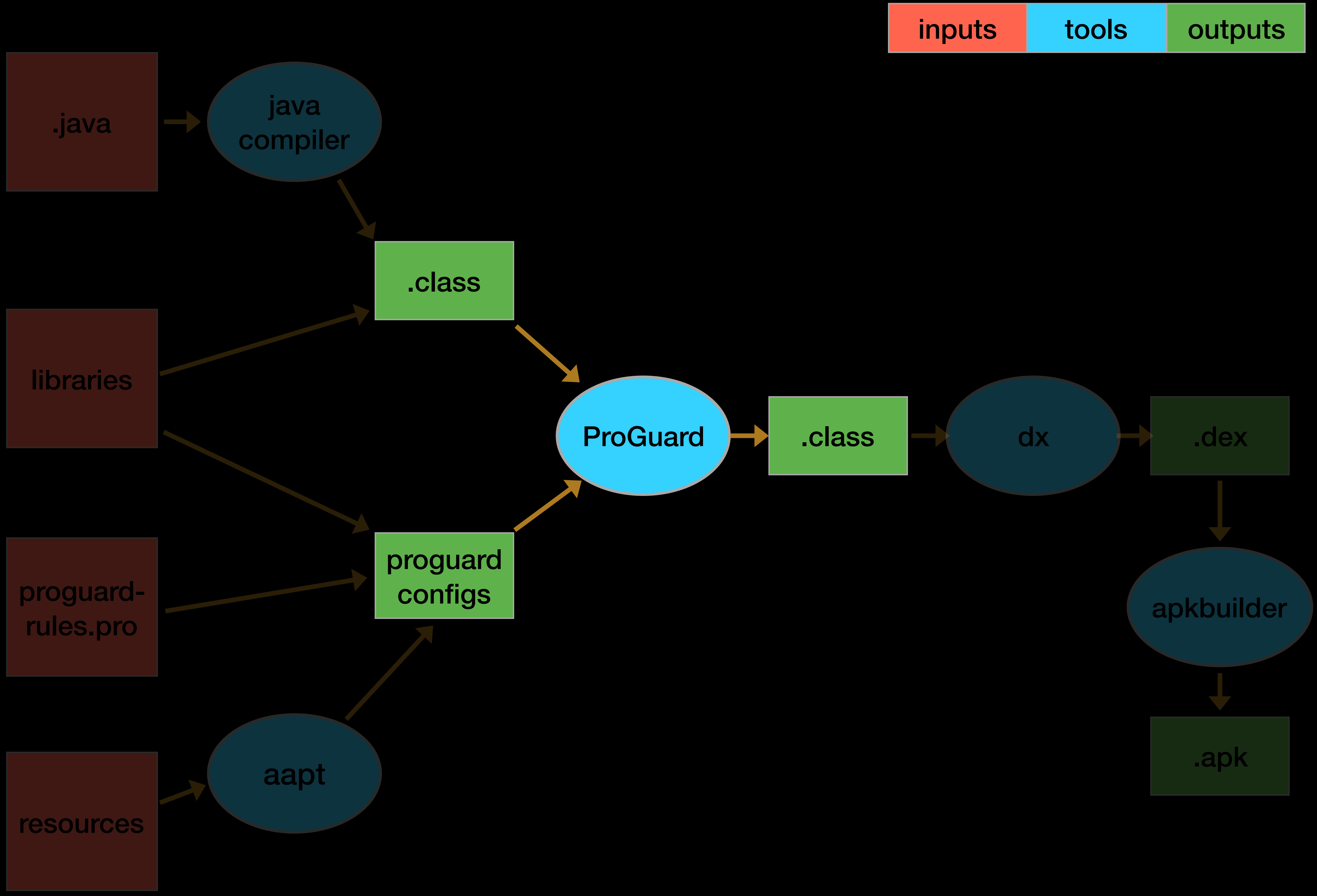
resources

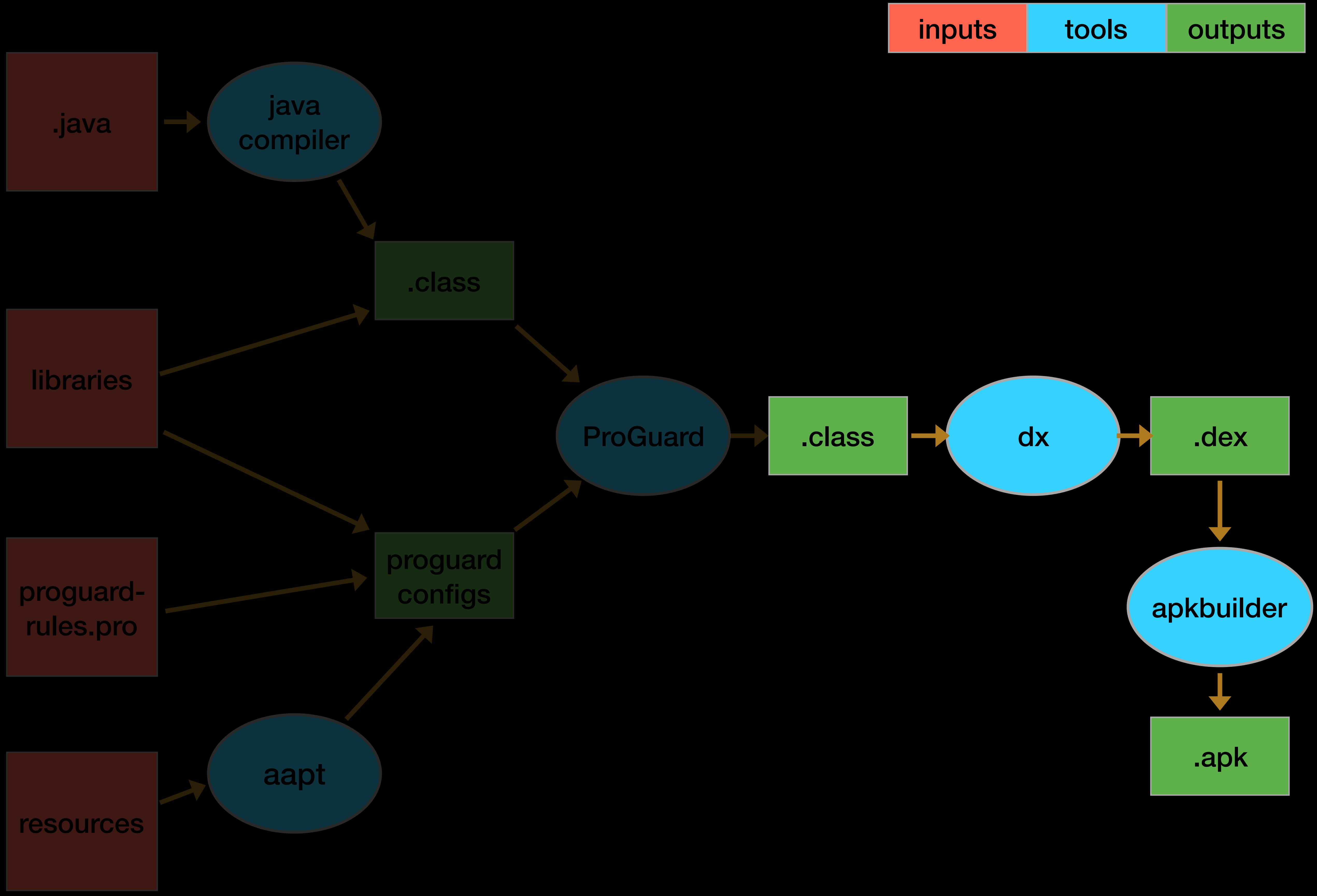
aapt

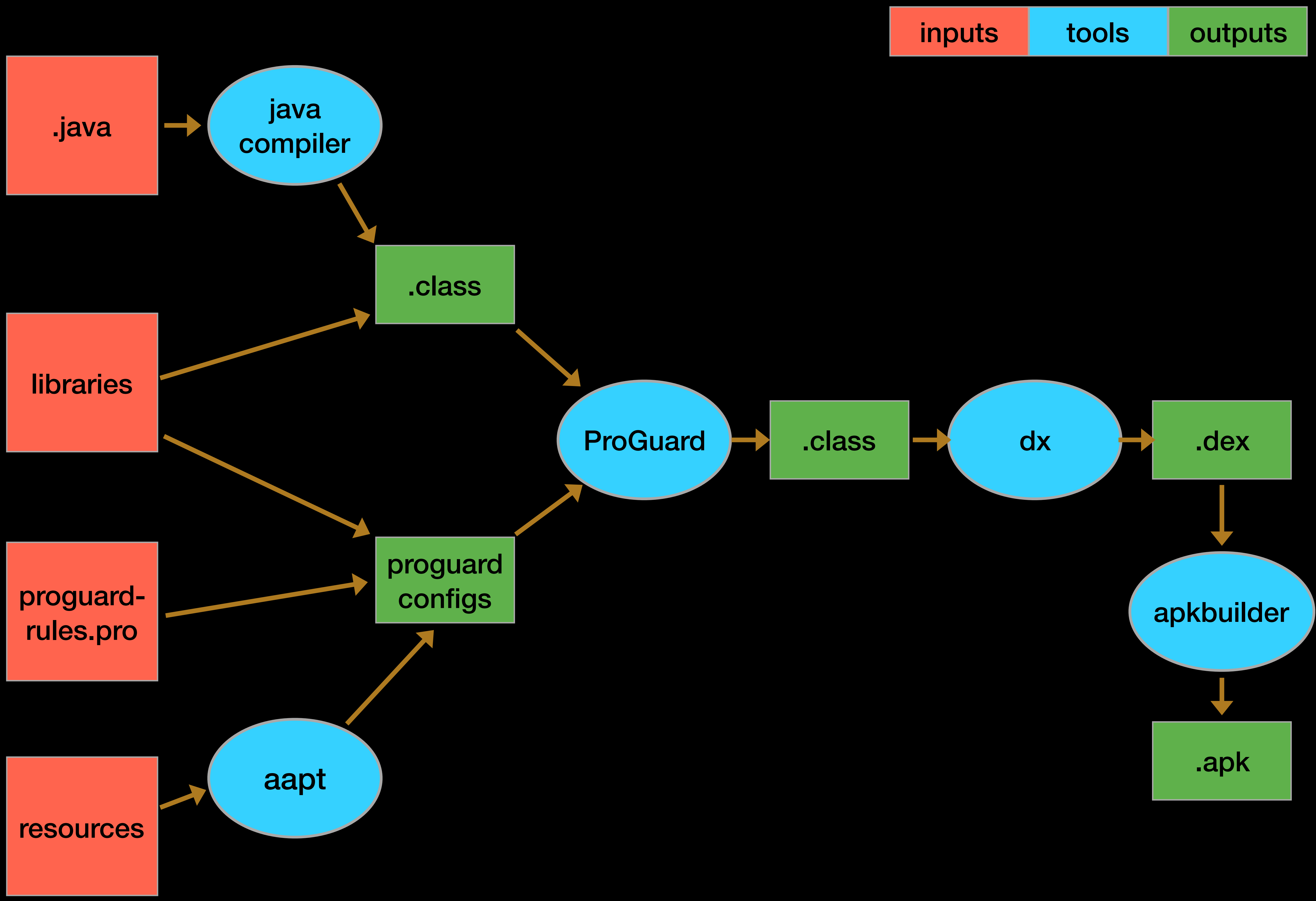
.apk

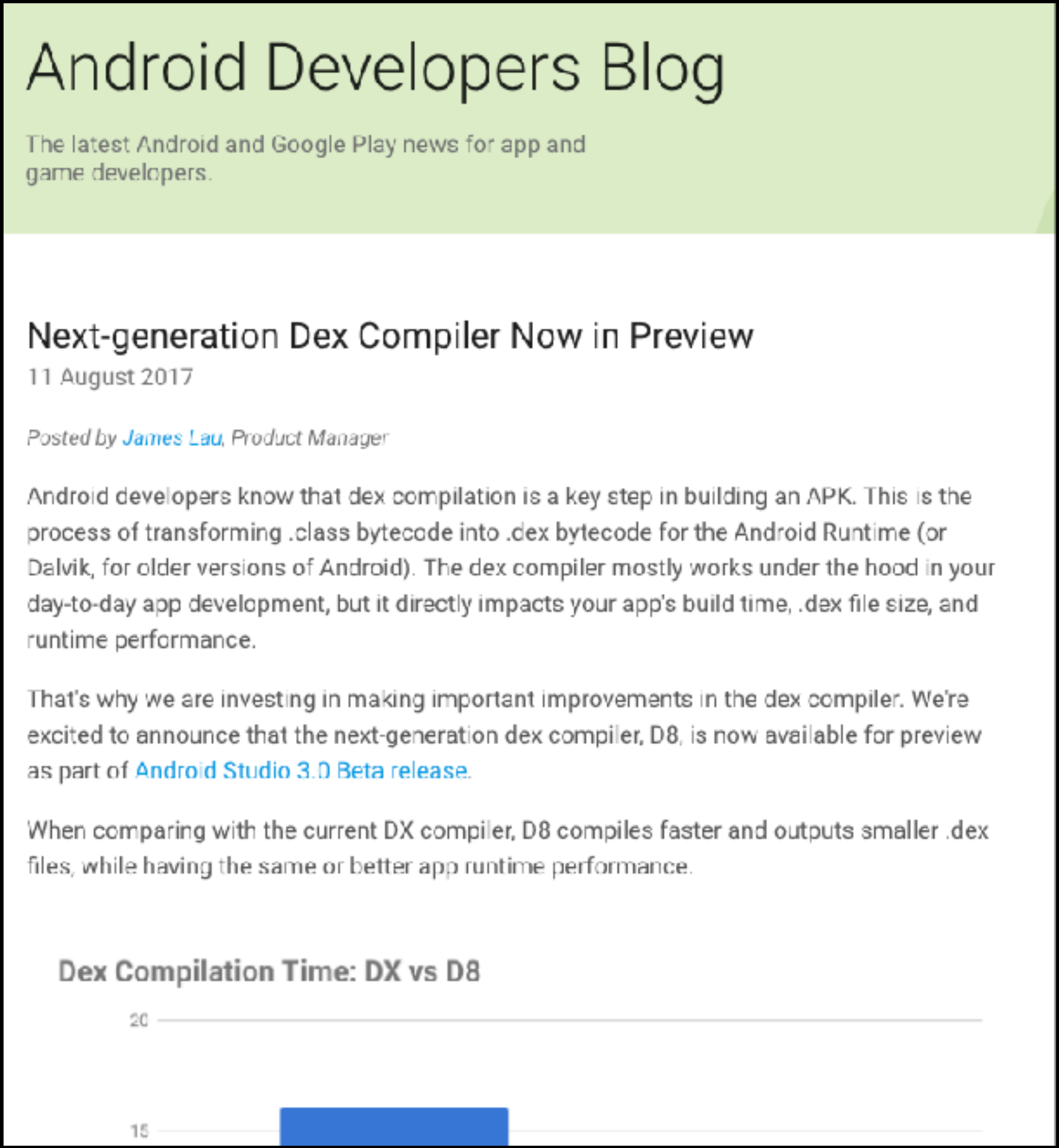
./gradlew [task] --debug


inputs
tools
outputs
.java
java compiler
.class
libraries
ProGuard
.class
dx
.dex
proguard-rules.pro
proguard configs
apkbuilder
.apk
resources
aapt


inputs
tools
outputs
.java
java compiler
.class
libraries
ProGuard
.class
dx
.dex
proguard-rules.pro
proguard configs
apkbuilder
resources
aapt
.apk


inputs
tools
outputs
.java
java compiler
.class
libraries
ProGuard
.class
dx
.dex
proguard-rules.pro
proguard configs
apkbuilder
resources
aapt
.apk


inputs
tools
outputs
.java
java compiler
.class
libraries
ProGuard
.class
dx
.dex
proguard-rules.pro
proguard configs
apkbuilder
resources
aapt
.apk



Android Developers Blog

The latest Android and Google Play news for app and game developers.

Next-generation Dex Compiler Now in Preview

11 August 2017

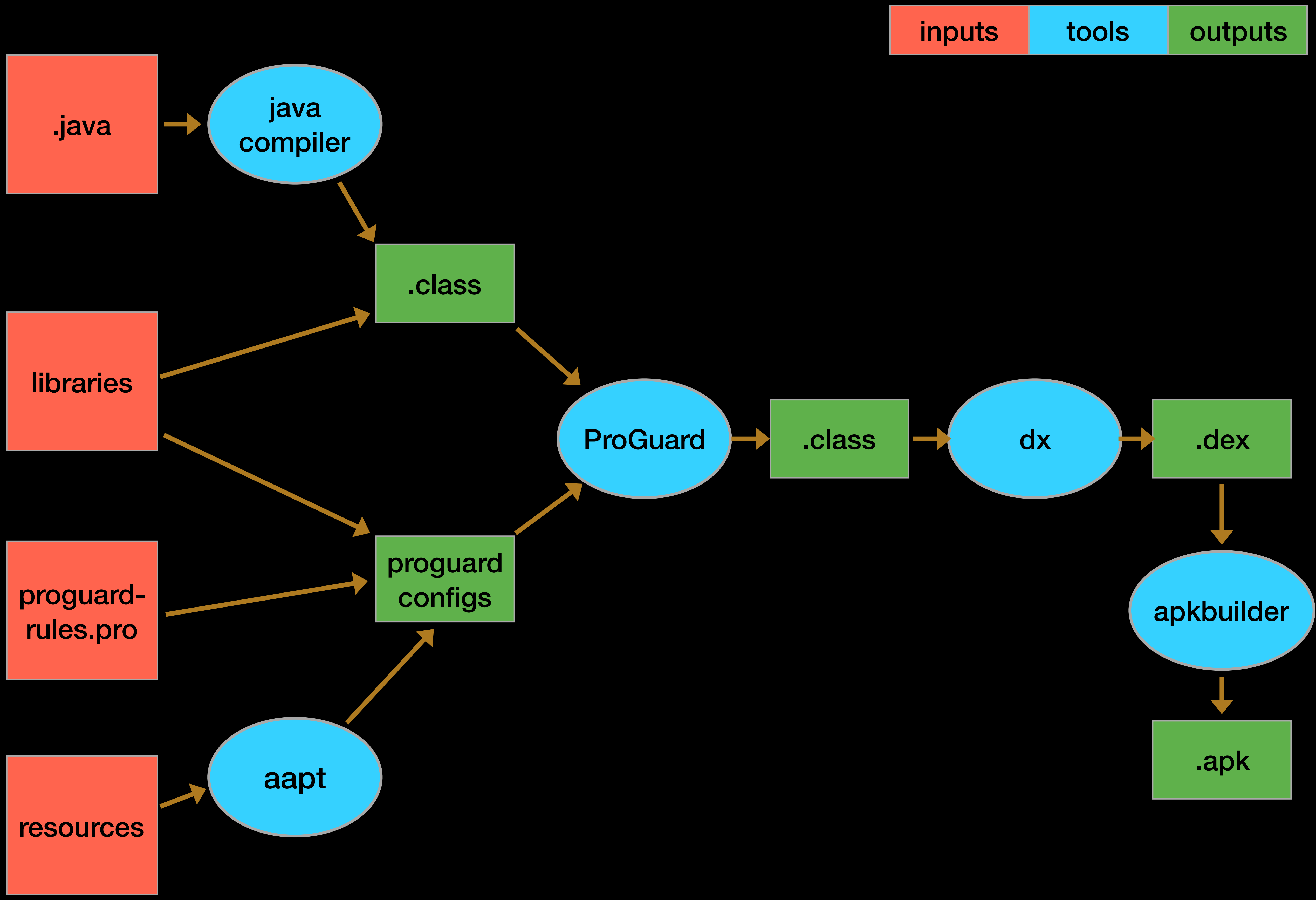
Posted by James Lau, Product Manager

Android developers know that dex compilation is a key step in building an APK. This is the process of transforming .class bytecode into .dex bytecode for the Android Runtime (or Dalvik, for older versions of Android). The dex compiler mostly works under the hood in your day-to-day app development, but it directly impacts your app's build time, .dex file size, and runtime performance.

That's why we are investing in making important improvements in the dex compiler. We're excited to announce that the next-generation dex compiler, D8, is now available for preview as part of Android Studio 3.0 Beta release.

When comparing with the current DX compiler, D8 compiles faster and outputs smaller .dex files, while having the same or better app runtime performance.

Dex Compilation Time: DX vs D8



inputs

tools

outputs

.java

java compiler

.class

libraries

proguard-rules.pro

resources

aapt

proguard configs

ProGuard

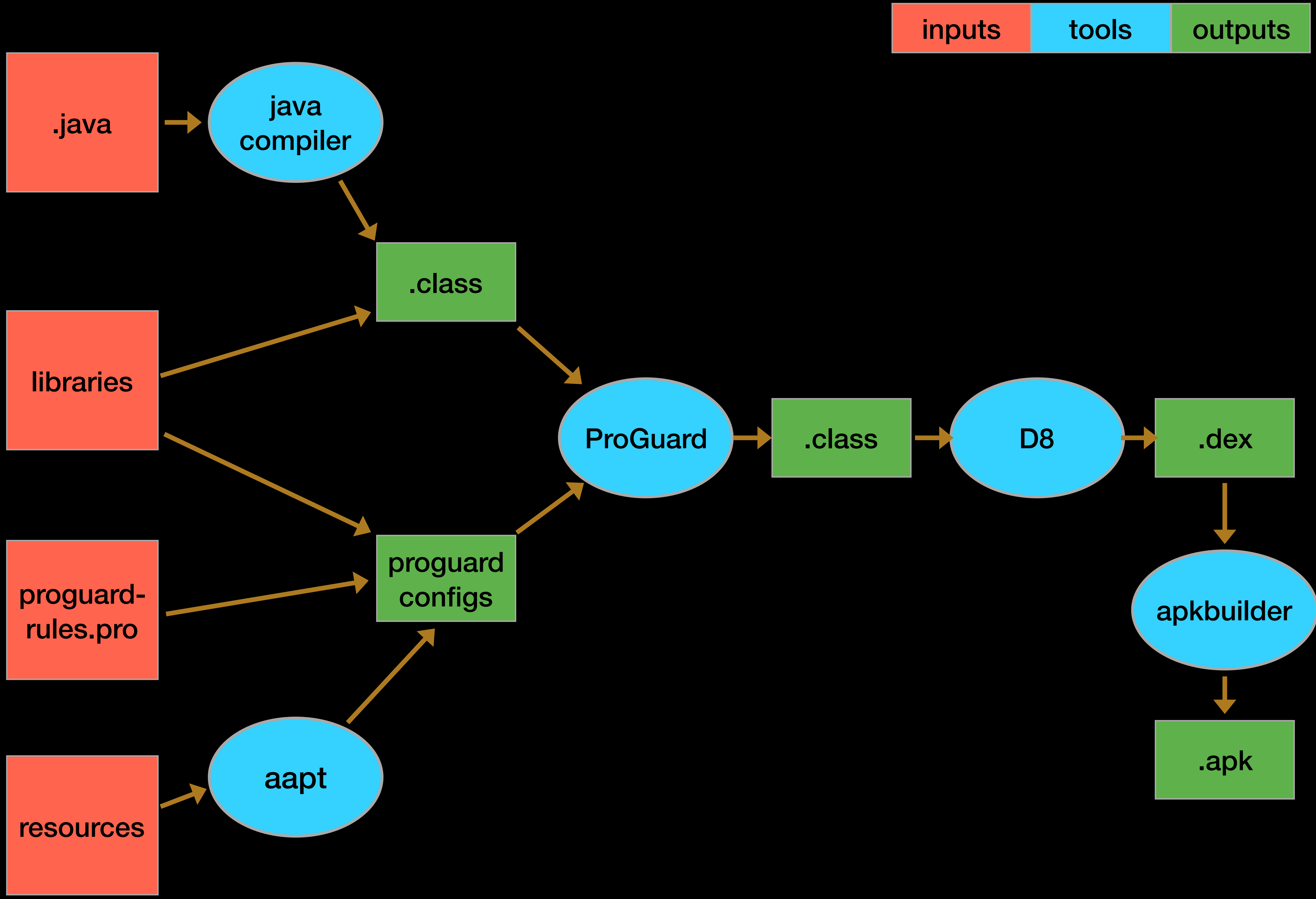
.class

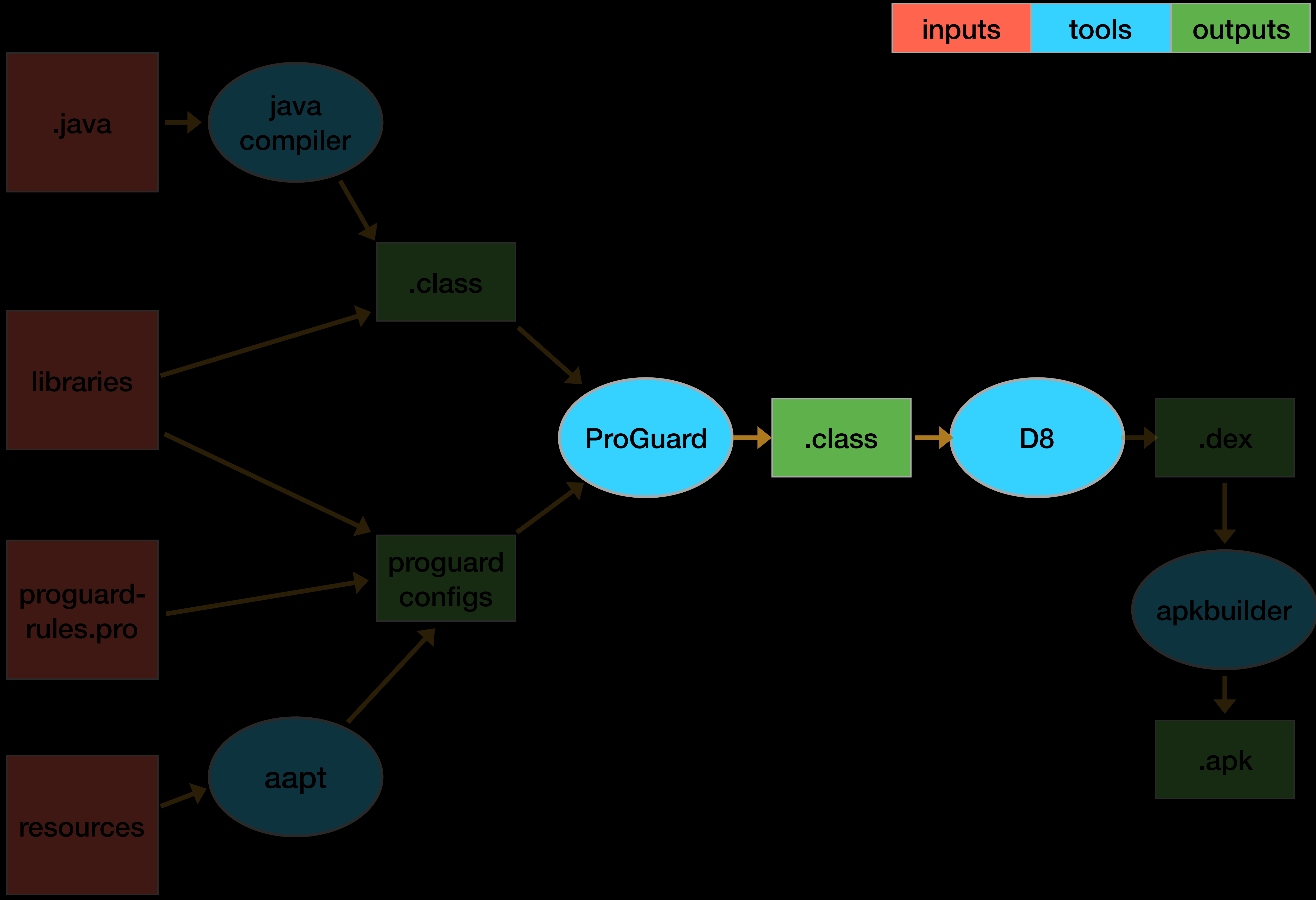
dx

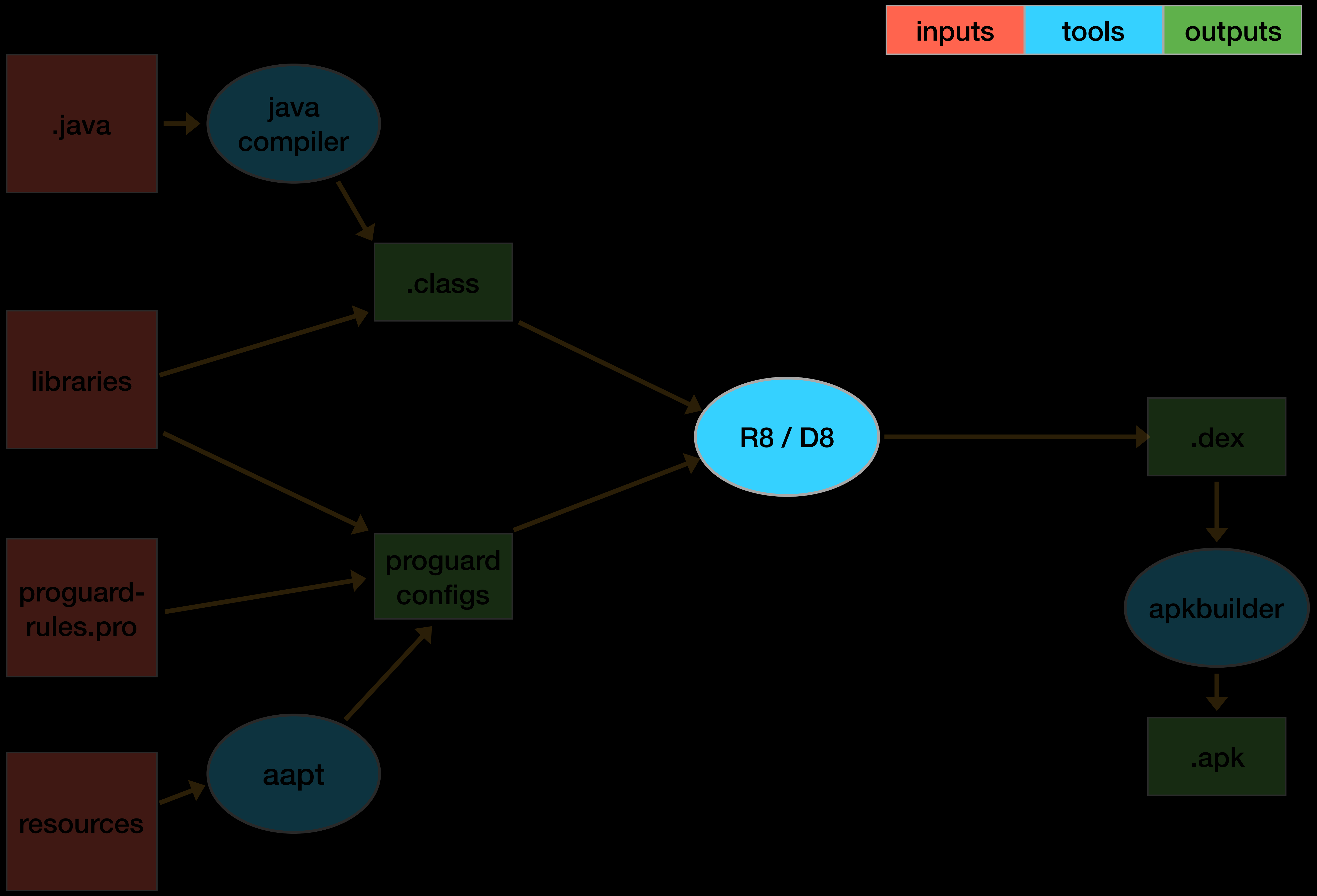
.dex

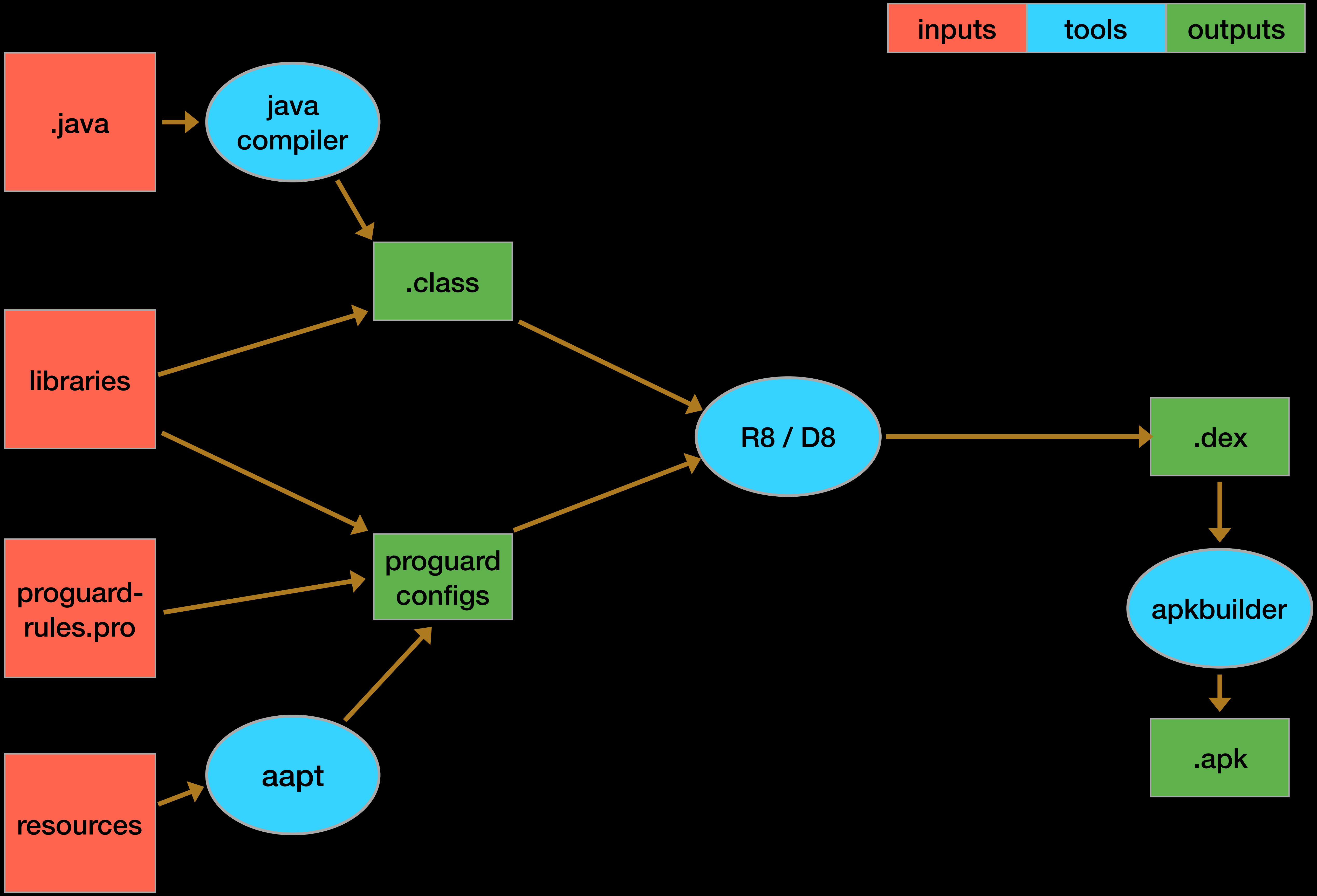
apkbuilder

.apk


inputs
tools
outputs
.java
java compiler
.class
libraries
ProGuard
.class
D8
.dex
proguard-rules.pro
proguard configs
apkbuilder
resources
aapt
.apk


inputs
tools
outputs
.java
java compiler
.class
libraries
ProGuard
.class
D8
.dex
proguard-rules.pro
proguard configs
apkbuilder
resources
aapt
.apk


inputs
tools
outputs
.java
java compiler
.class
libraries
R8 / D8
.dex
proguard configs
proguard-rules.pro
apkbuilder
.apk
aapt
resources


inputs
tools
outputs
.java
java compiler
.class
libraries
R8 / D8
.dex
proguard configs
proguard-rules.pro
apkbuilder
aapt
.apk
resources

R8

https://r8.googlesource.com/r8

https://issuetracker.google.com/issues?q=componentid:326788

https://groups.google.com/forum/#!forum/r8-dev

ProGuard ➜ R8

** PREVIEW **

```
${project_root}/gradle.properties

android.enableR8=true
```

** PREVIEW **

ProGuard ➜ R8

```
${project_root                  properties

android.enableR
```

ProGuard → R8

** PREVIEW **

```
${project_root              properties

android.enable
```

** PREVIEW **

ProGuard ➜ R8

```
${project_roo                 properties

android.enable
```

ProGuard → R8

** PREVIEW **

```
${project_root          properties

android.enableR
```

ProGuard ➜ R8

- mapping.txt - only renamed elements

ProGuard ➜ R8

- mapping.txt - only renamed elements
- -printseeds seeds.txt 🪲
- -printusage usage.txt 🪲
- -printconfiguration 🪲

** PREVIEW **

ProGuard ➜ R8

- mapping.txt - only renamed elements
- -printseeds seeds.txt 🪲
- -printusage usage.txt 🪲
- -printconfiguration 🪲
- ~~-dump dump.txt~~

ProGuard ➜ R8

- same -keep[x] rule format
- “shrinking” ➜ “tree-shaking”
- “obfuscation” ➜ “minification”

Go forth!

L0: obfuscate your code

L1: read default rules and AAPT-generated rules

L2: NARROW, TARGETED keep rules

L3: try out R8

http://ware.to/proguard