网易APM Hook方案探索

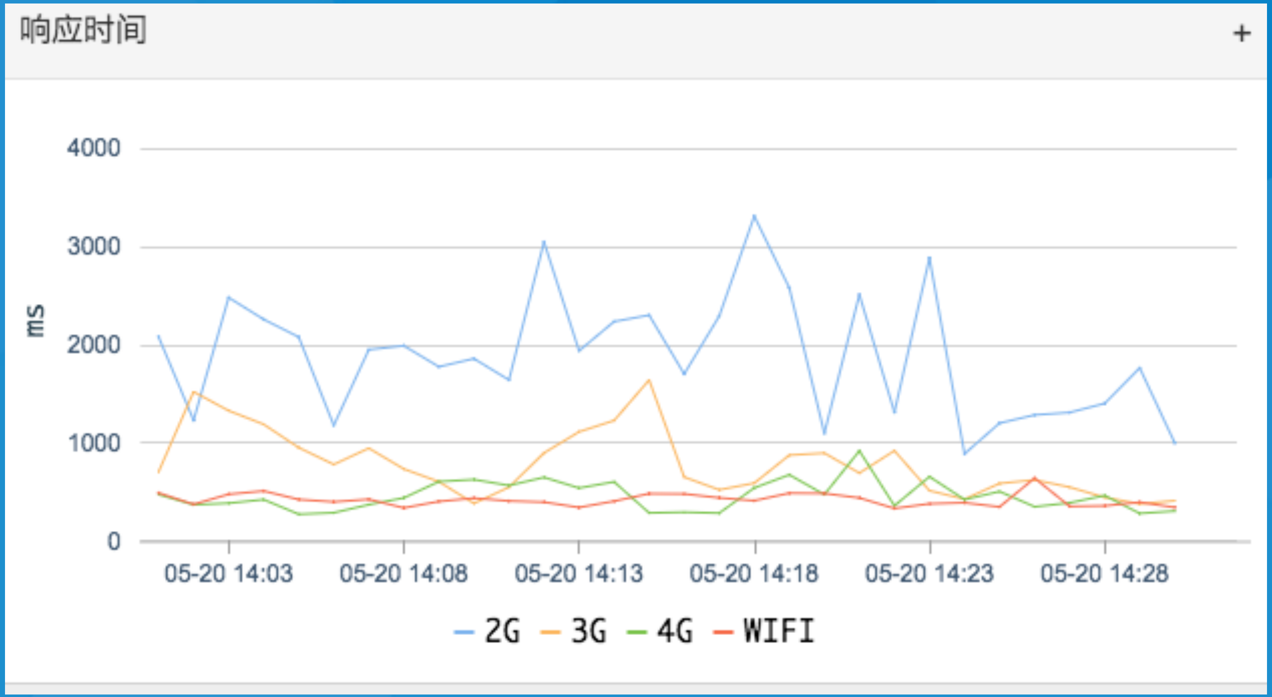
郑文@网易前端技术部

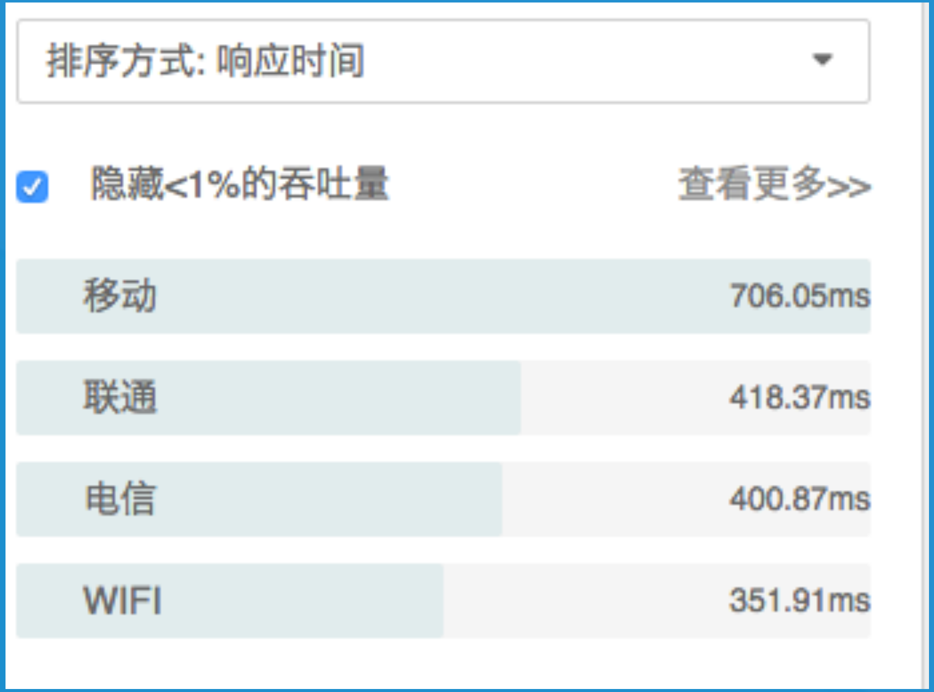
APM

Application Performance Management

- 最小侵入\使用透明
- 更多的收集点

APM



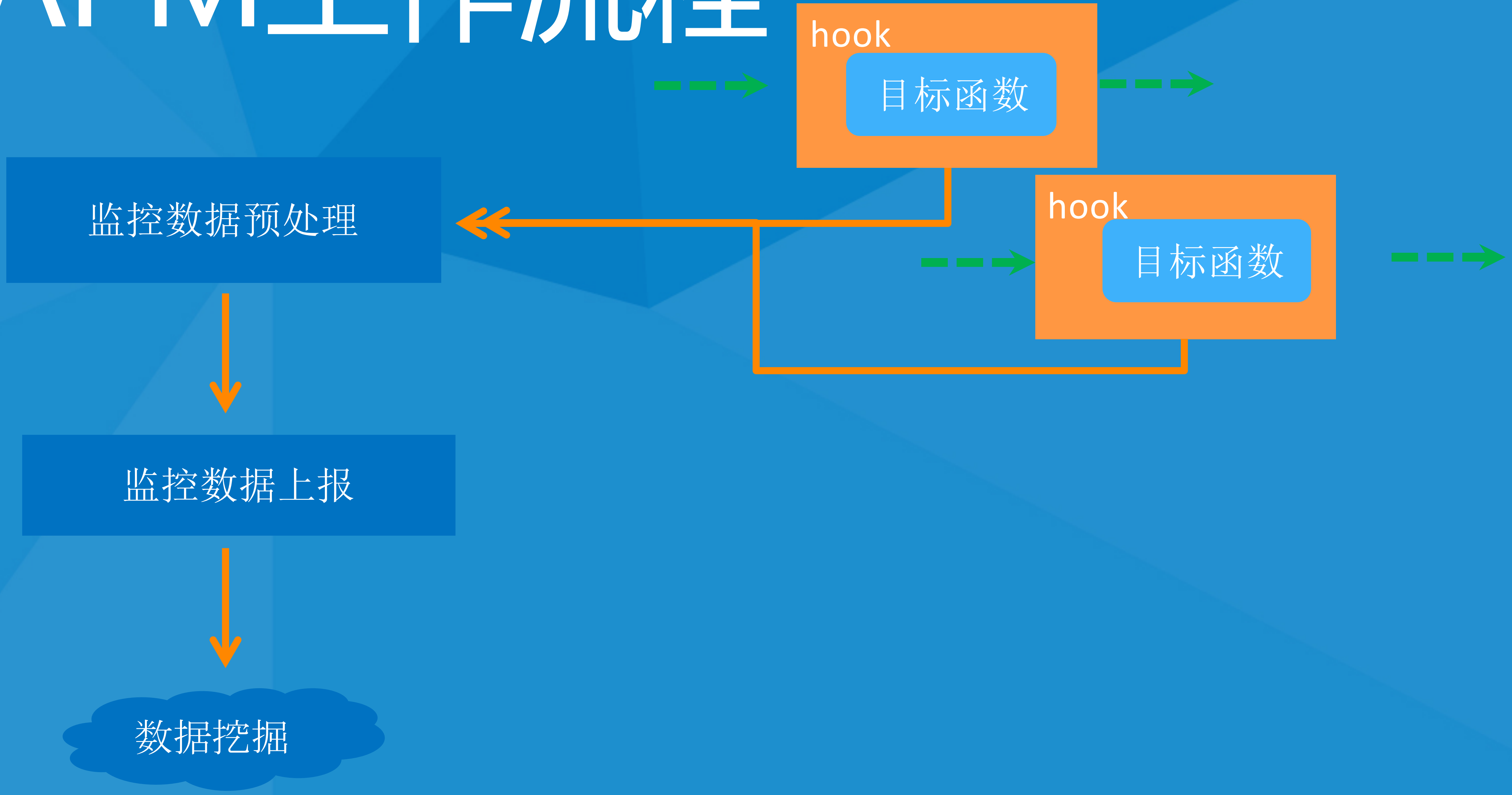


排序方式: 响应时间 ▾

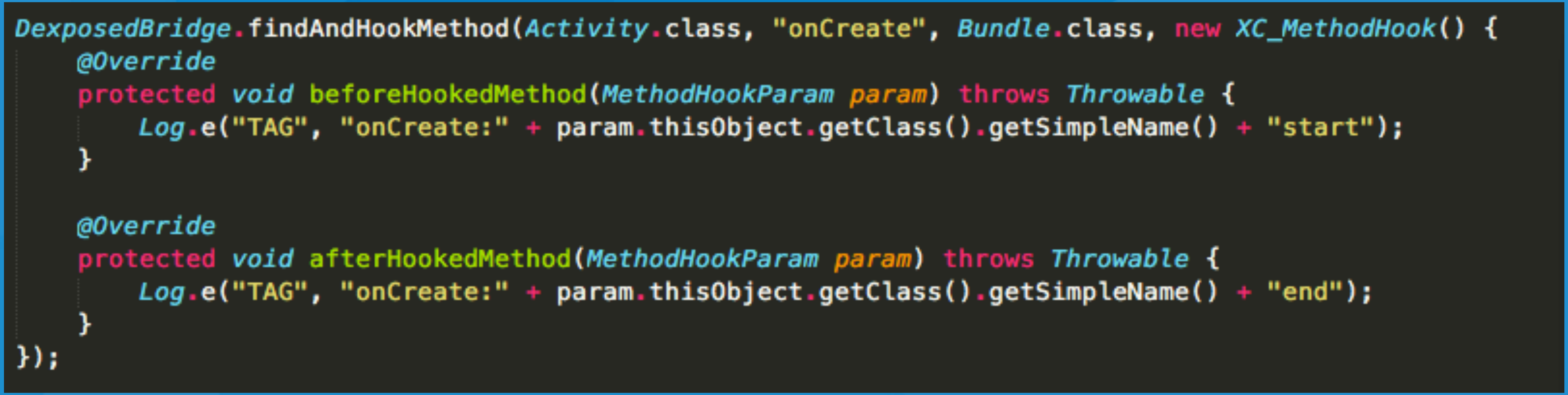
✓ 隐藏<1%的吞吐量 查看更多>>

移动	706.05ms
联通	418.37ms
电信	400.87ms
WIFI	351.91ms

APM工作流程



运行期Hook



```
DexposedBridge.findAndHookMethod(Activity.class, "onCreate", Bundle.class, new XC_MethodHook() {
    @Override
    protected void beforeHookedMethod(MethodHookParam param) throws Throwable {
        Log.e("TAG", "onCreate:" + param.thisObject.getClass().getSimpleName() + "start");
    }

    @Override
    protected void afterHookedMethod(MethodHookParam param) throws Throwable {
        Log.e("TAG", "onCreate:" + param.thisObject.getClass().getSimpleName() + "end");
    }
});
```

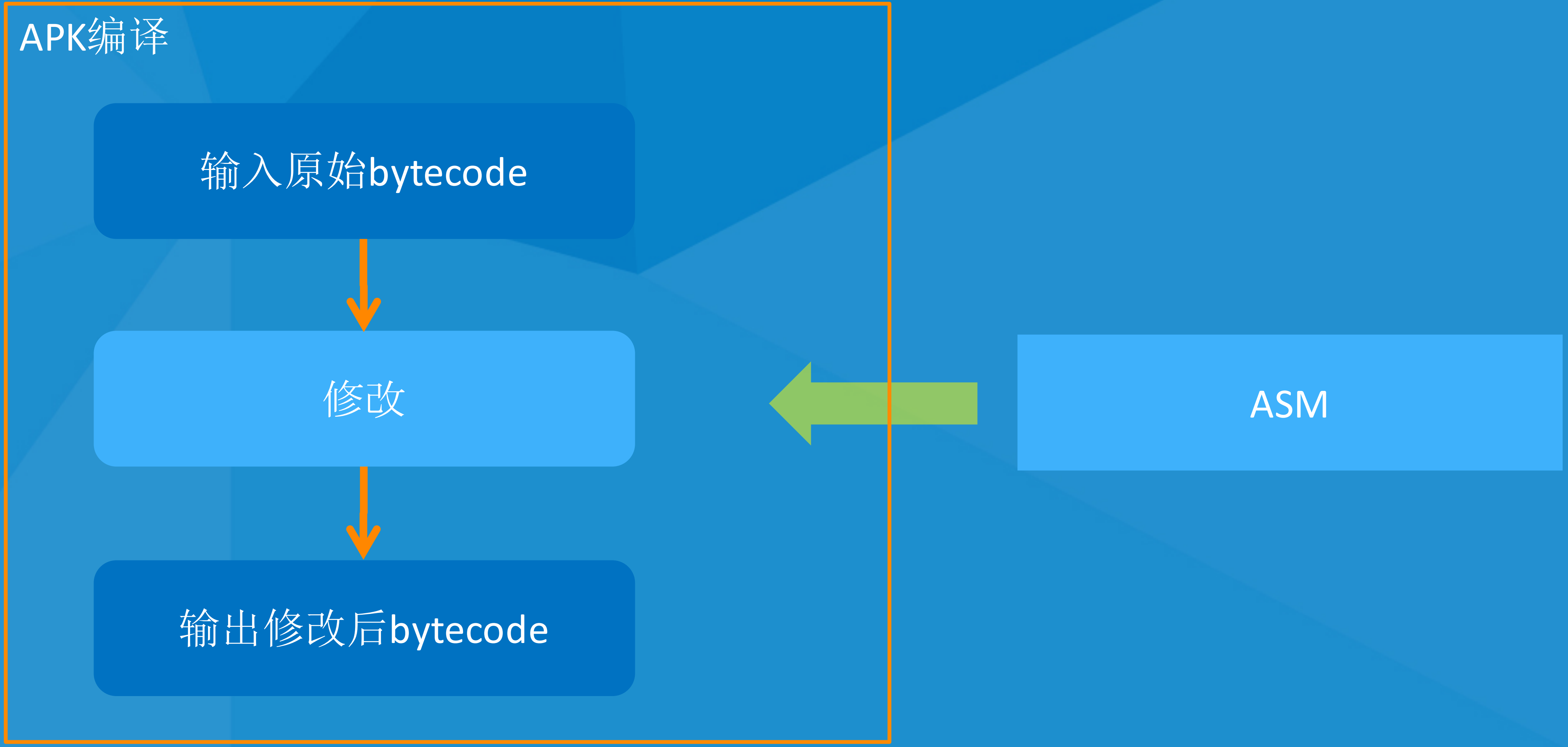
- 动态化
- 私用API的hook
- Hack->版本兼容性？

编译期hook

优缺点：

- 丧失动态化能力，事先埋点
- 无hack，兼容性好

hook流程



ByteCde Instrumetion

```java
public class HelloWorld {
    public void  main(int i){
        foo(i);
    }

    public int foo(int i){
        System.out.println(i);
        return i;
    }
}

public class HelloWorldProxy {
    public static int wrapFoo(HelloWorld hello,
        int i){
         //获取调用时间
        int j = hello.foo(i);
        //获取调用时间
        return j;
    }
}
```

```
aload_0
iload_1
invokevirtual #3 //Method foo:(I)I
```

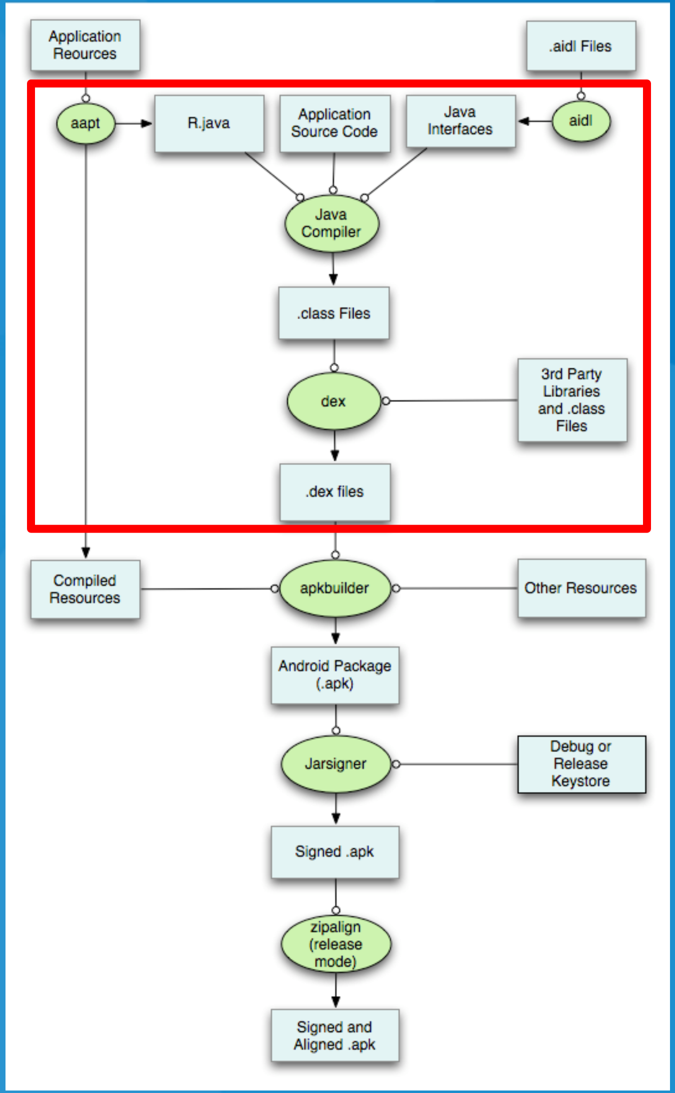
ASM

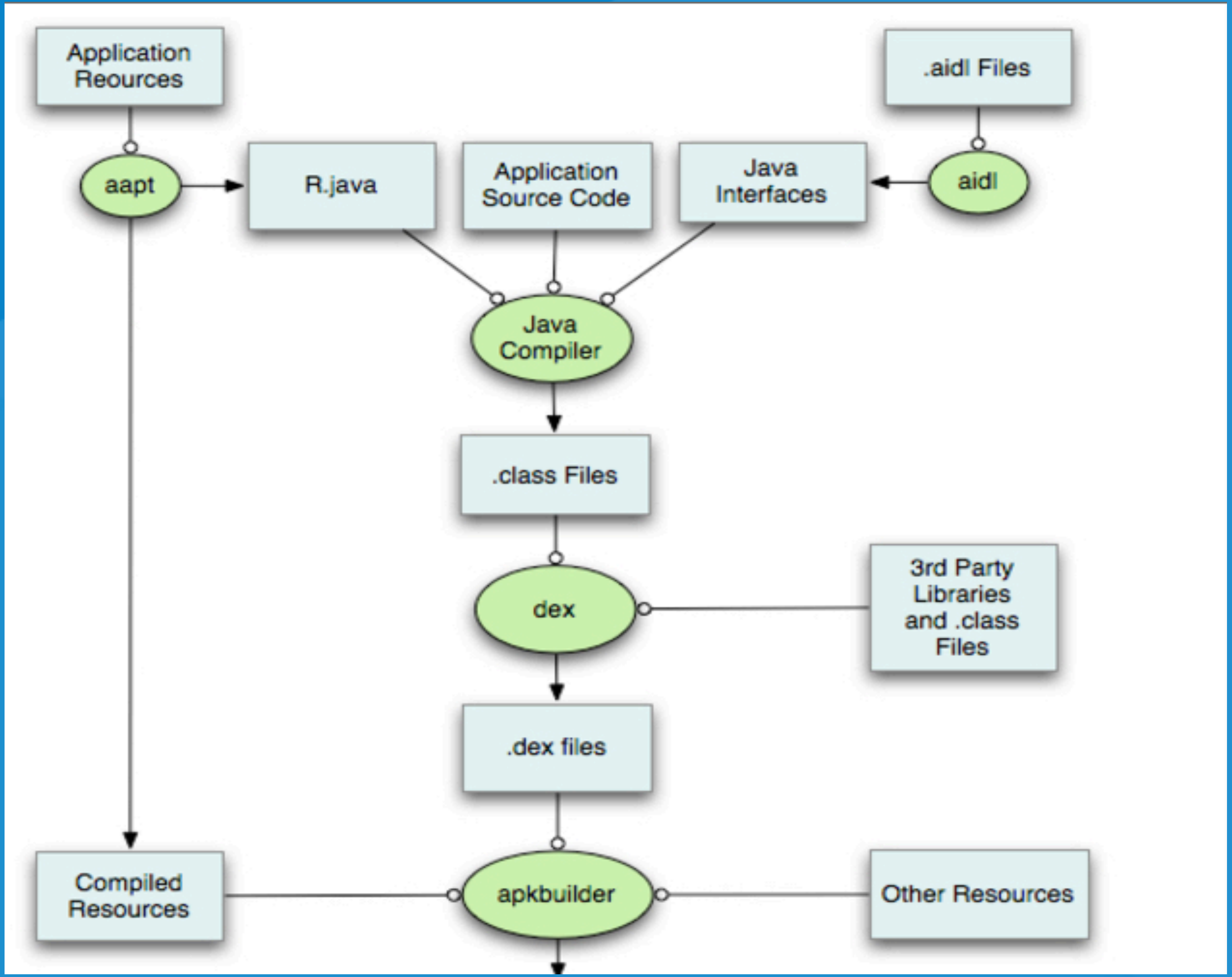
```
aload_0
iload_1
invokestatic #17//Method HelloWorldProxy.wrapFoo(HelloWolrd;I)I
```

构建工具

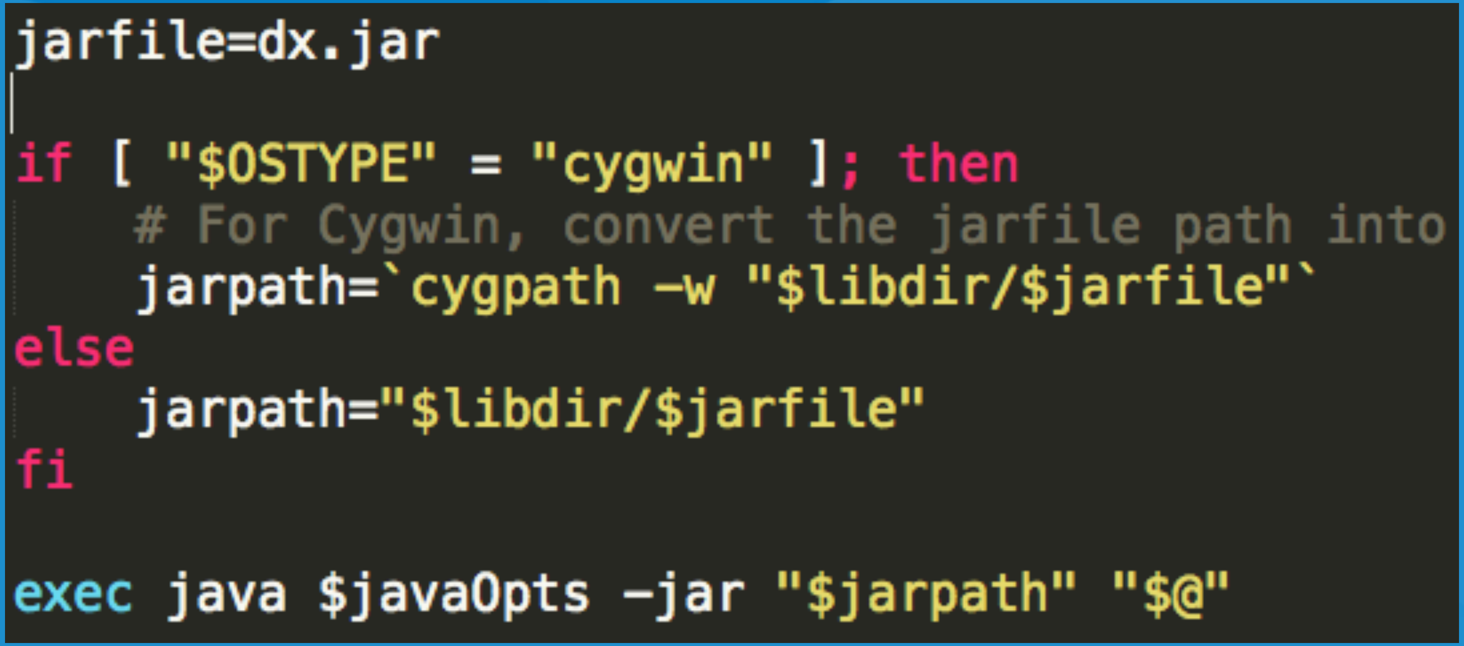
- Ant
- Gradle

构建工具之Ant


Application Reources
.aidl Files
aapt
R.java
Application Source Code
Java Interfaces
aidl
Java Compiler
.class Files
dex
3rd Party Libraries and .class Files
.dex files
Compiled Resources
apkbuilder
Other Resources
Android Package (.apk)
Jarsigner
Debug or Release Keystore
Signed .apk
zipalign (release mode)
Signed and Aligned .apk

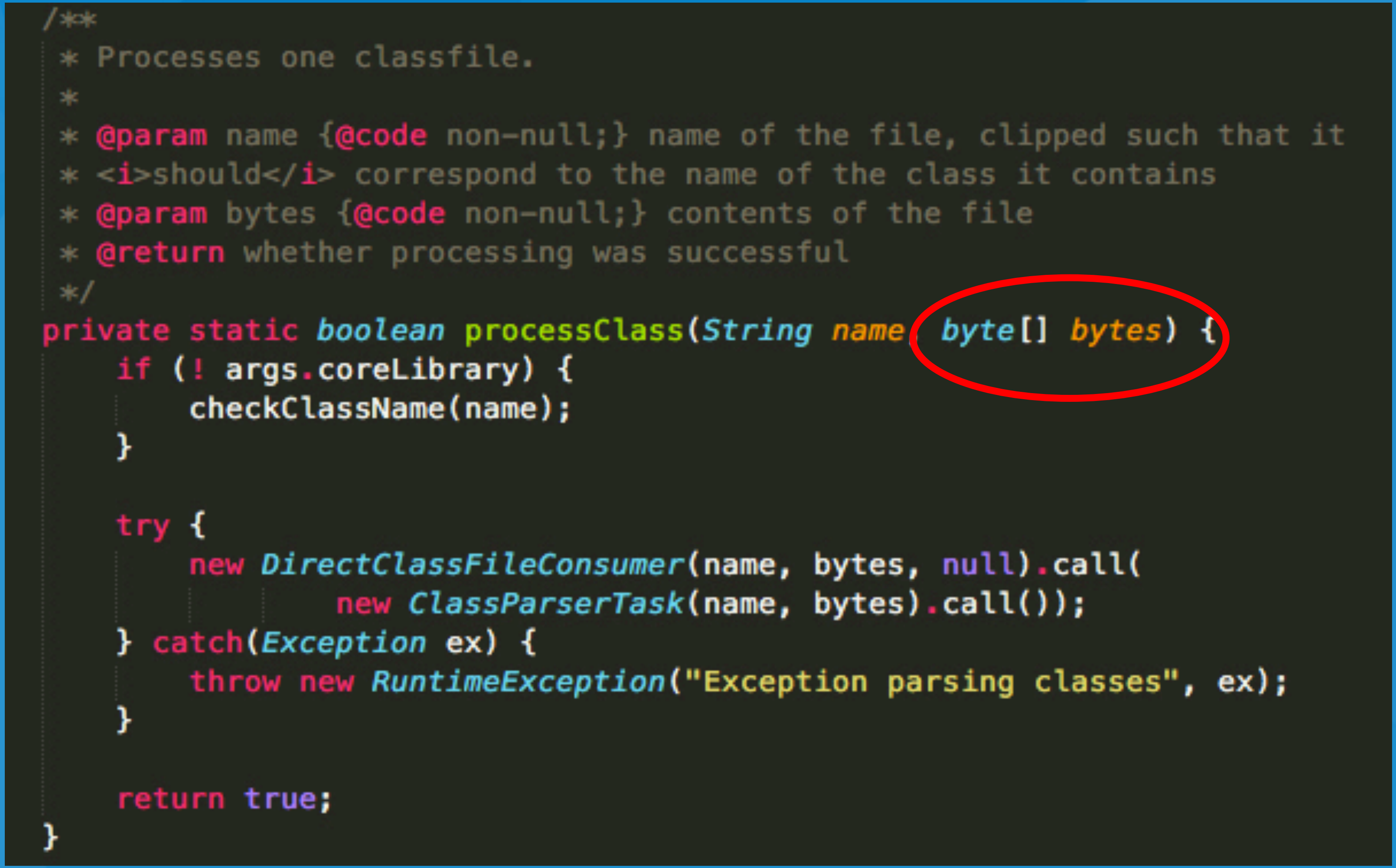

Application Reources
.aidl Files
aapt
R.java
Application Source Code
Java Interfaces
aidl
Java Compiler
.class Files
dex
3rd Party Libraries and .class Files
.dex files
Compiled Resources
apkbuilder
Other Resources

dx is a bash



```
jarfile=dx.jar

if [ "$OSTYPE" = "cygwin" ]; then
    # For Cygwin, convert the jarfile path into
    jarpath=`cygpath -w "$libdir/$jarfile"`
else
    jarpath="$libdir/$jarfile"
fi

exec java $javaOpts -jar "$jarpath" "$@"
```

com.android.dx.command.dexer.Main



```java
/**
 * Processes one classfile.
 *
 * @param name {@code non-null;} name of the file, clipped such that it
 * <i>should</i> correspond to the name of the class it contains
 * @param bytes {@code non-null;} contents of the file
 * @return whether processing was successful
 */
private static boolean processClass(String name, byte[] bytes) {
    if (! args.coreLibrary) {
        checkClassName(name);
    }

    try {
        new DirectClassFileConsumer(name, bytes, null).call(
                new ClassParserTask(name, bytes).call());
    } catch(Exception ex) {
        throw new RuntimeException("Exception parsing classes", ex);
    }

    return true;
}
```

hook Main.processClass ?

Agent

- Agent是main方法之前的拦截器
  - 静态Agent：-javaagent:/path/agent.jar
  - 动态Agent:：VirtualMachine.loadagent
- 通过Instrumentation参数改变内存中的字节码

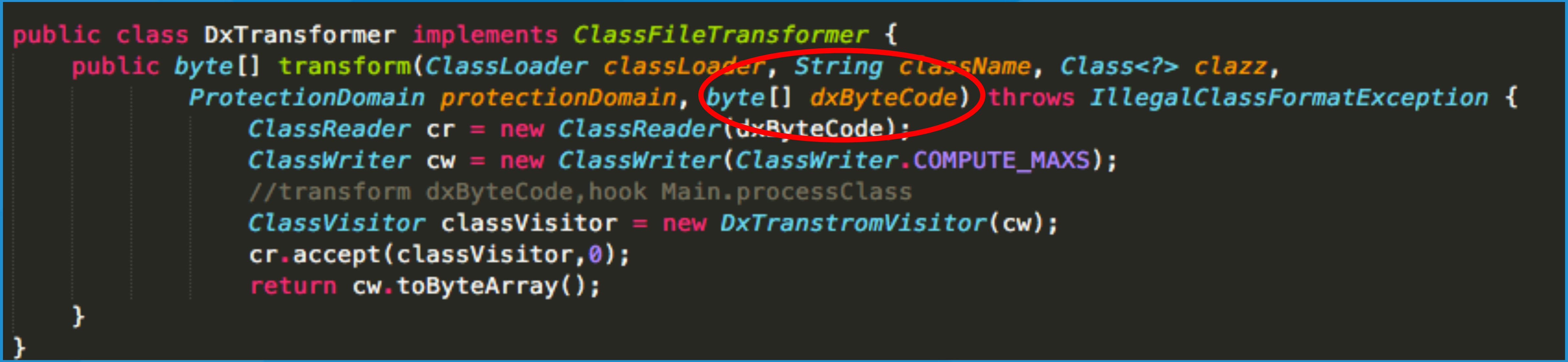
Agent and Intrumentation

```
dx  -javaagent:/path/agent.jar
```

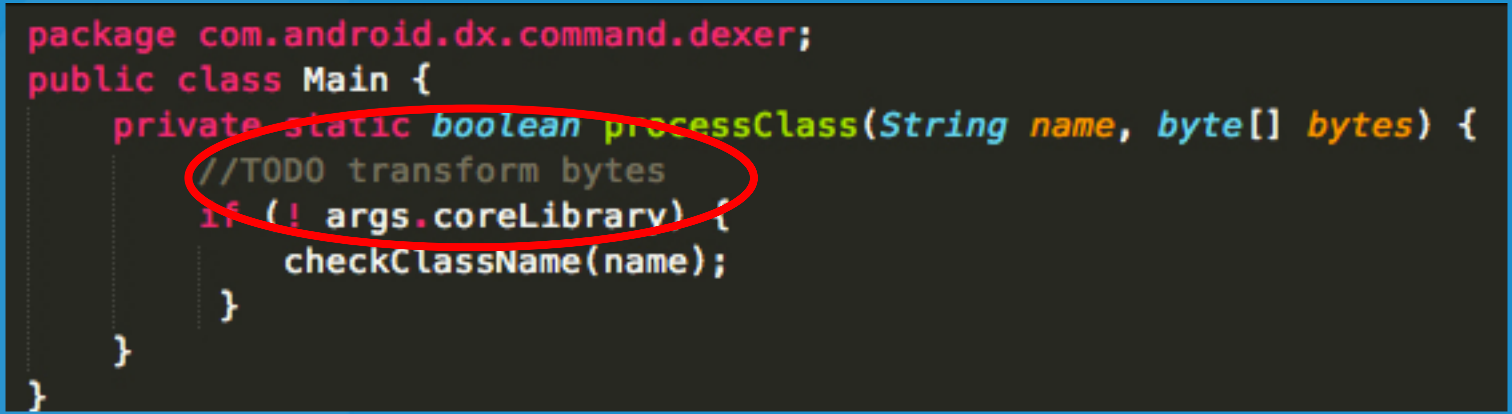
```
META-INF/MAINIFEST.MF
Premain-Class:com.netease.apm.Agent
```

```java
public class Agent {
    public static void premain(String agentArgs, Instrumentation inst) {
        inst.addTransformer(new ClassFileTransformer() {...});
    }

    public static void agentmain(String agentArgs,
        Instrumentation instrumentation) {
        premain(agentArgs, instrumentation);
    }
}
```

ClassTransformer



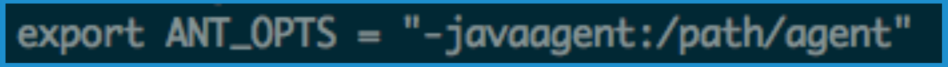
```
public class DxTransformer implements ClassFileTransformer {
    public byte[] transform(ClassLoader classLoader, String className, Class<?> clazz,
            ProtectionDomain protectionDomain, byte[] dxByteCode) throws IllegalClassFormatException {
                ClassReader cr = new ClassReader(dxByteCode);
                ClassWriter cw = new ClassWriter(ClassWriter.COMPUTE_MAXS);
                //transform dxByteCode,hook Main.processClass
                ClassVisitor classVisitor = new DxTranstromVisitor(cw);
                cr.accept(classVisitor,0);
                return cw.toByteArray();
    }
}
```



```
package com.android.dx.command.dexer;
public class Main {
    private static boolean processClass(String name, byte[] bytes) {
        //TODO transform bytes
        if (! args.coreLibrary) {
            checkClassName(name);
        }
    }
}
```

ANT构建工具

1. 利用agent机制hook dx-processClass
2. 修改processClass的输入



```
export ANT_OPTS = "-javaagent:/path/agent"
```

构建工具之Gradle

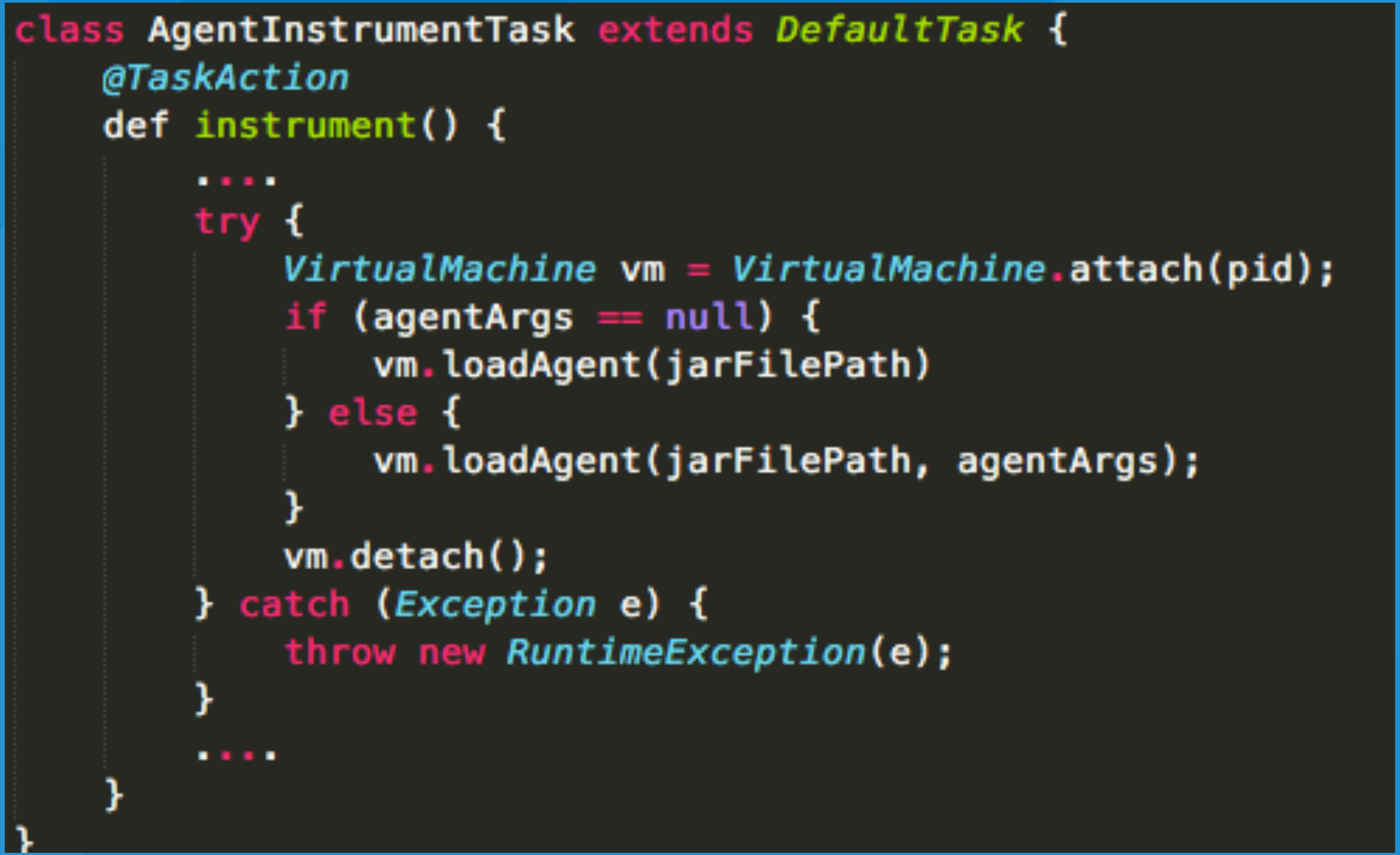
特点

- Gradle是任务依赖系统
- Gradle是守护进程
- Transform API

Transform API

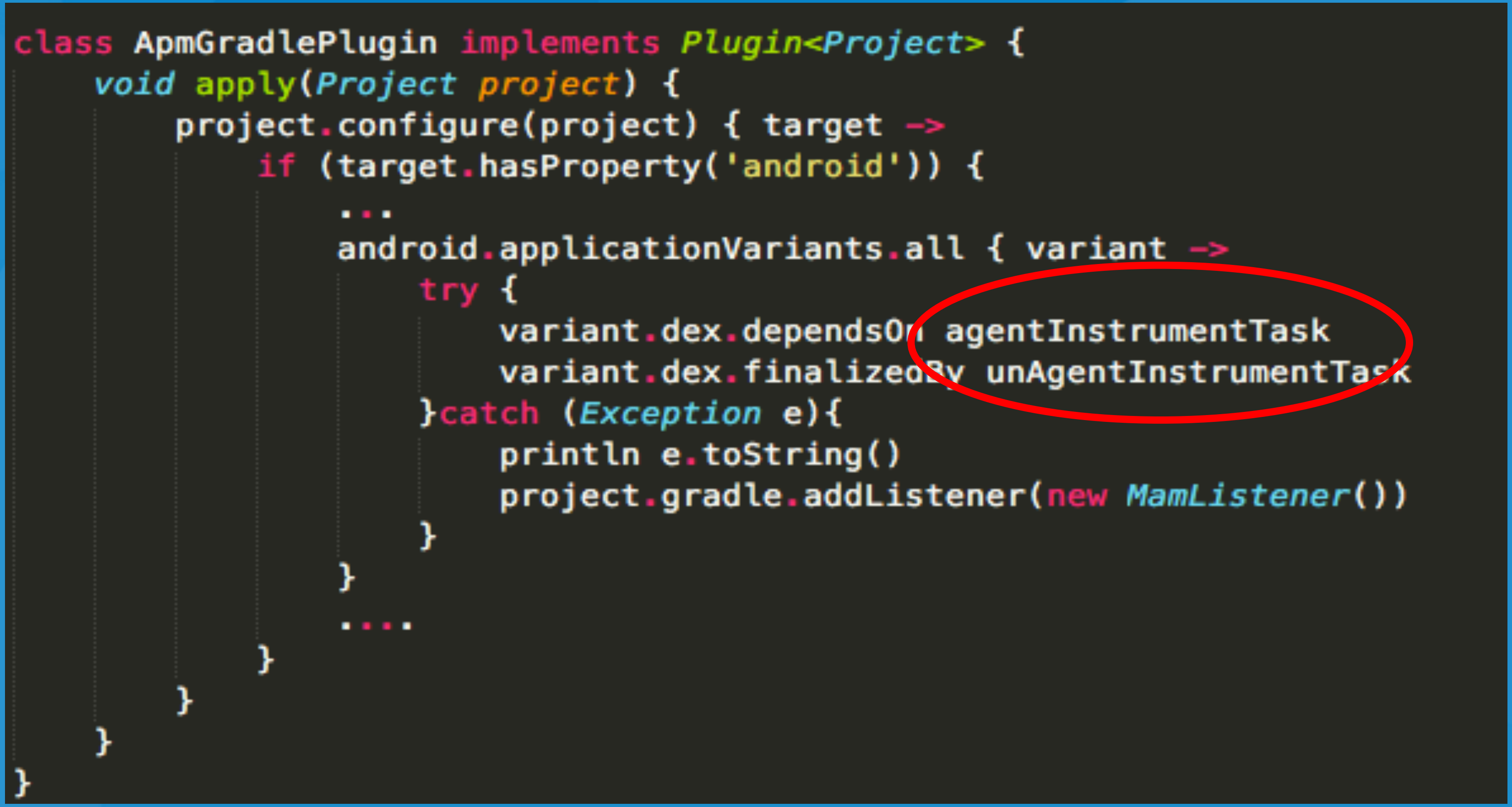
- Starting with 1.5.0-beta1
- The dex class is gone. You cannot access it anymore through the variant API

gradle < 1.5



```
class AgentInstrumentTask extends DefaultTask {
    @TaskAction
    def instrument() {
        ....
        try {
            VirtualMachine vm = VirtualMachine.attach(pid);
            if (agentArgs == null) {
                vm.loadAgent(jarFilePath)
            } else {
                vm.loadAgent(jarFilePath, agentArgs);
            }
            vm.detach();
        } catch (Exception e) {
            throw new RuntimeException(e);
        }
        ....
    }
}
```

gradle < 1.5

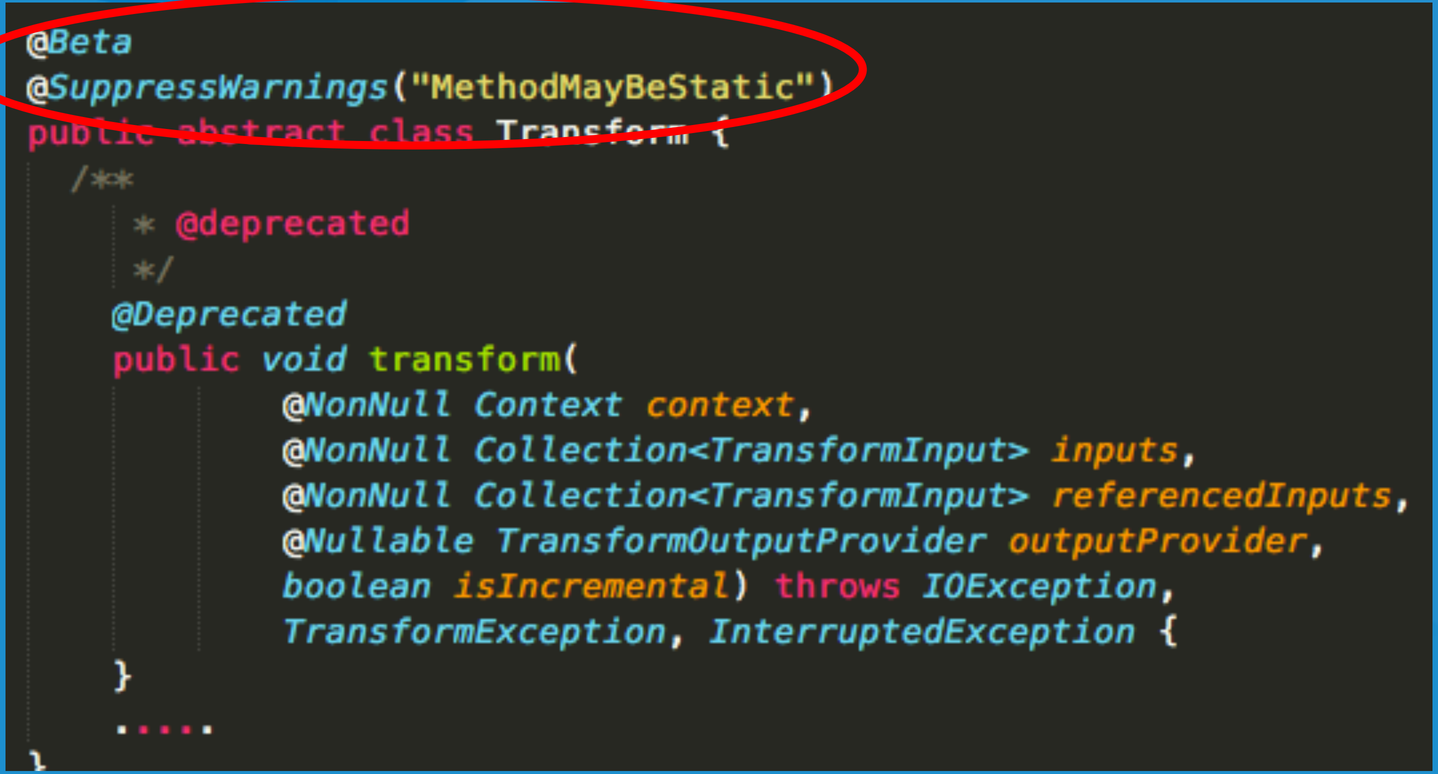


```
class ApmGradlePlugin implements Plugin<Project> {
    void apply(Project project) {
        project.configure(project) { target ->
            if (target.hasProperty('android')) {
                ...
                android.applicationVariants.all { variant ->
                    try {
                        variant.dex.dependsOn agentInstrumentTask
                        variant.dex.finalizedBy unAgentInstrumentTask
                    }catch (Exception e){
                        println e.toString()
                        project.gradle.addListener(new MamListener())
                    }
                }
                ....
            }
        }
    }
}
```

gradle > 1.5

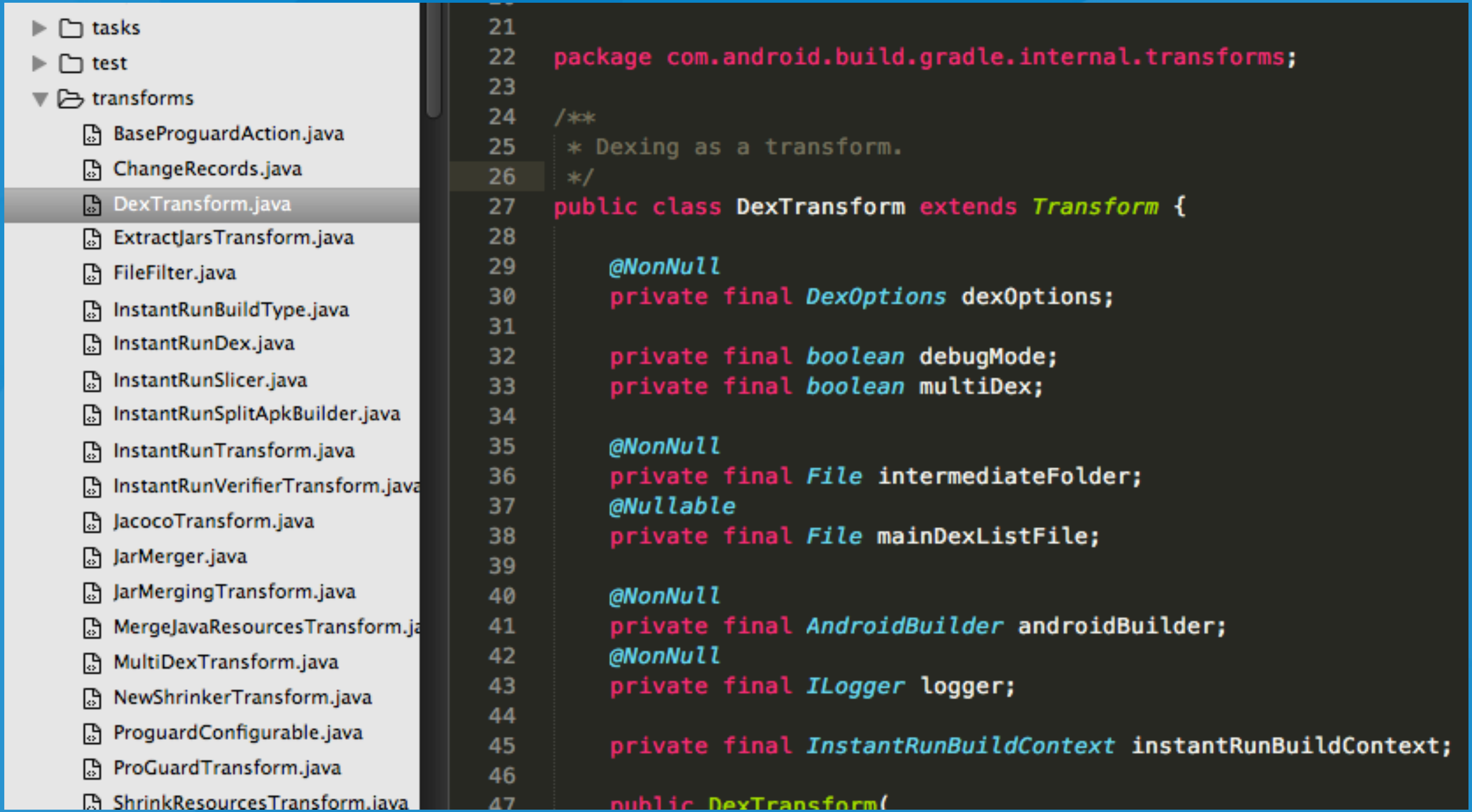
Transform API?

But Transform is not final



```
@Beta
@SuppressWarnings("MethodMayBeStatic")
public abstract class Transform {
  /**
     * @deprecated
     */
    @Deprecated
    public void transform(
                @NonNull Context context,
                @NonNull Collection<TransformInput> inputs,
                @NonNull Collection<TransformInput> referencedInputs,
                @Nullable TransformOutputProvider outputProvider,
                boolean isIncremental) throws IOException,
                TransformException, InterruptedException {
    }
    .....
}
```

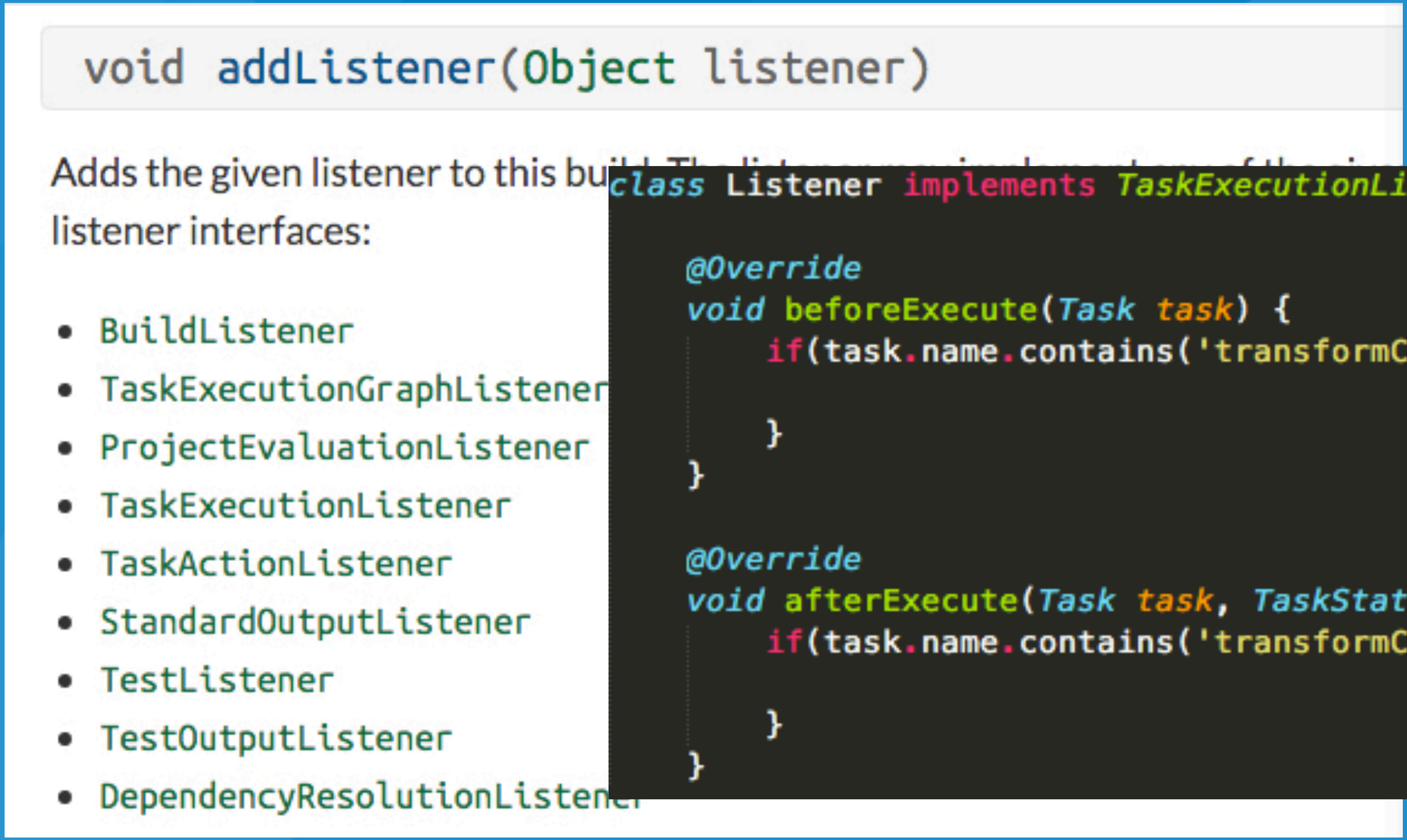
Dexing as a transform



- tasks
- test
- transforms
  - BaseProguardAction.java
  - ChangeRecords.java
  - DexTransform.java
  - ExtractJarsTransform.java
  - FileFilter.java
  - InstantRunBuildType.java
  - InstantRunDex.java
  - InstantRunSlicer.java
  - InstantRunSplitApkBuilder.java
  - InstantRunTransform.java
  - InstantRunVerifierTransform.java
  - JacocoTransform.java
  - JarMerger.java
  - JarMergingTransform.java
  - MergeJavaResourcesTransform.j
  - MultiDexTransform.java
  - NewShrinkerTransform.java
  - ProguardConfigurable.java
  - ProGuardTransform.java
  - ShrinkResourcesTransform.java

```java

package com.android.build.gradle.internal.transforms;

/**
 * Dexing as a transform.
 */
public class DexTransform extends Transform {

    @NonNull
    private final DexOptions dexOptions;

    private final boolean debugMode;
    private final boolean multiDex;

    @NonNull
    private final File intermediateFolder;
    @Nullable
    private final File mainDexListFile;

    @NonNull
    private final AndroidBuilder androidBuilder;
    @NonNull
    private final ILogger logger;

    private final InstantRunBuildContext instantRunBuildContext;

    public DexTransform(
```

exception

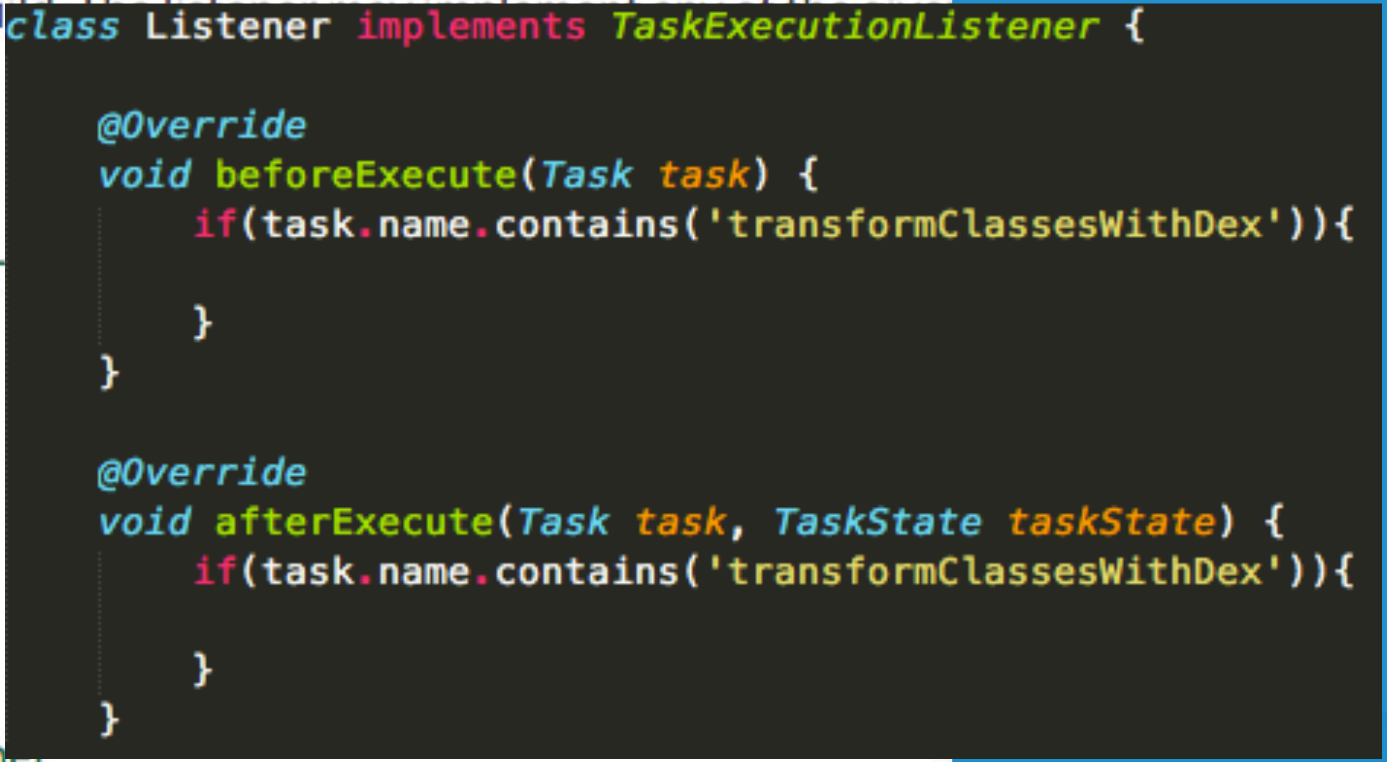


```
void addListener(Object listener)
```

Adds the given listener to this bu

listener interfaces:

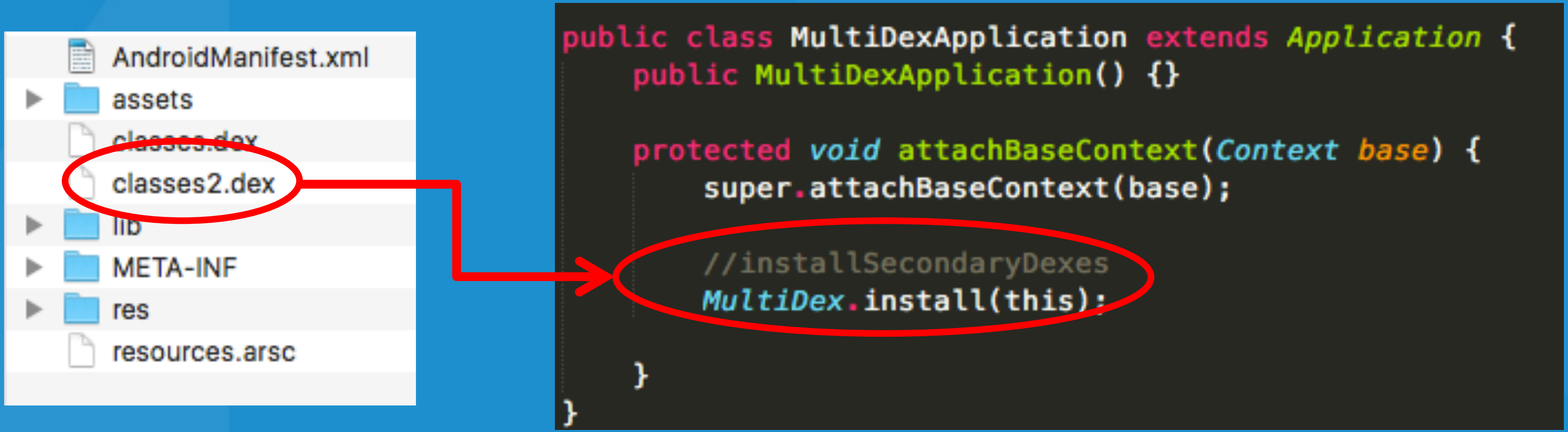
- BuildListener
- TaskExecutionGraphListener
- ProjectEvaluationListener
- TaskExecutionListener
- TaskActionListener
- StandardOutputListener
- TestListener
- TestOutputListener
- DependencyResolutionListen



```
class Listener implements TaskExecutionListener {

    @Override
    void beforeExecute(Task task) {
        if(task.name.contains('transformClassesWithDex')){

        }
    }

    @Override
    void afterExecute(Task task, TaskState taskState) {
        if(task.name.contains('transformClassesWithDex')){

        }
    }
```

But

- MultiDex : noClassDefFountError

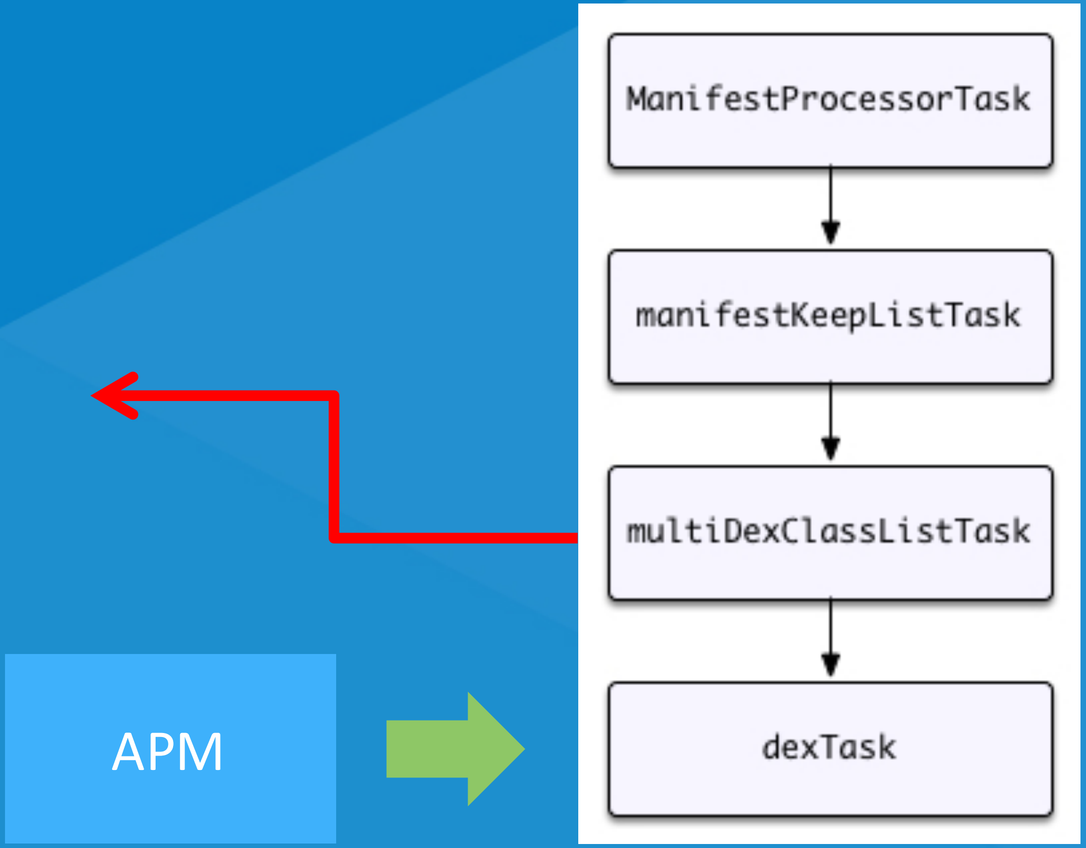


```
AndroidManifest.xml
assets
classes.dex
classes2.dex
lib
META-INF
res
resources.arsc
```

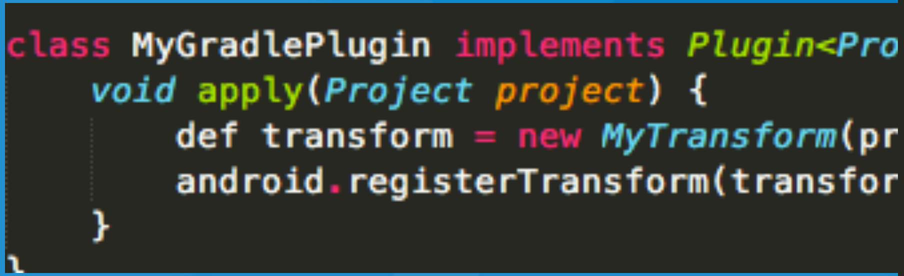
```java
public class MultiDexApplication extends Application {
    public MultiDexApplication() {}

    protected void attachBaseContext(Context base) {
        super.attachBaseContext(base);

        //installSecondaryDexes
        MultiDex.install(this);

    }
}
```

But

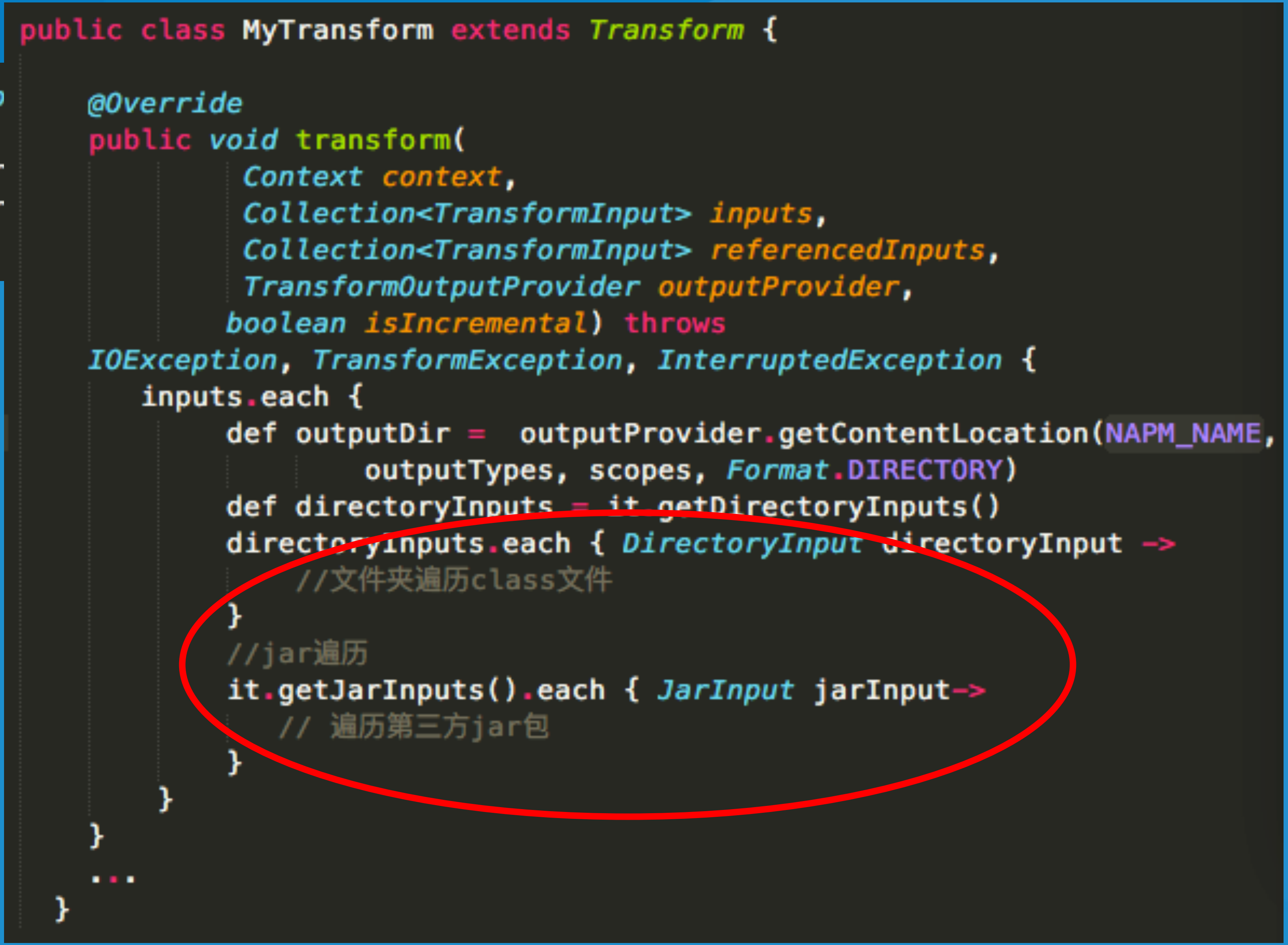
- MultiDex
  - maindexlist.txt
- 多插件冲突



Support Transform API



```
class MyGradlePlugin implements Plugin<Pro
    void apply(Project project) {
        def transform = new MyTransform(pr
        android.registerTransform(transfor
    }
}
```



```
public class MyTransform extends Transform {

    @Override
    public void transform(
                Context context,
                Collection<TransformInput> inputs,
                Collection<TransformInput> referencedInputs,
                TransformOutputProvider outputProvider,
               boolean isIncremental) throws
    IOException, TransformException, InterruptedException {
        inputs.each {
            def outputDir =  outputProvider.getContentLocation(NAPM_NAME,
                    outputTypes, scopes, Format.DIRECTORY)
            def directoryInputs = it.getDirectoryInputs()
            directoryInputs.each { DirectoryInput directoryInput ->
                //文件夹遍历class文件
            }
            //jar遍历
            it.getJarInputs().each { JarInput jarInput->
               // 遍历第三方jar包
            }
        }
    }
    ...
}
```

Support Transform API

- APM Tranform执行优先系统Transform
  - 代码混淆
  - 代码精简
  - Instant Run
- 兼容gradle 1.5，2.0，2.1

我们的尝试

CPU密集/IO

- IO：File/SqlOpenHelper/Http
- Bitmap
- Activity/Application

代理类/接口

```java
URL url = new URL(urlString);
URLConnection conn = (HttpURLConnection) url.openConnection();
```

↓

```java
URLConnection connectionProxy =   HttpInstrumentation.openConnection(url);
```

```java
public class HttpsURLConnectionProxy
                extends HttpsURLConnection {
    private HttpsURLConnection impl;

   public void connect() throws IOException {
        //探针代码
        try {
            this.impl.connect();
        } catch (IOException e) {
            throw e;
        }
        //探针代码
    }
}
```

代理方法

```java
org.apache.http.impl.client.AbstractHttpClient

public final HttpResponse execute(HttpUriRequest request) throws IOException, ClientProtocolException;
```

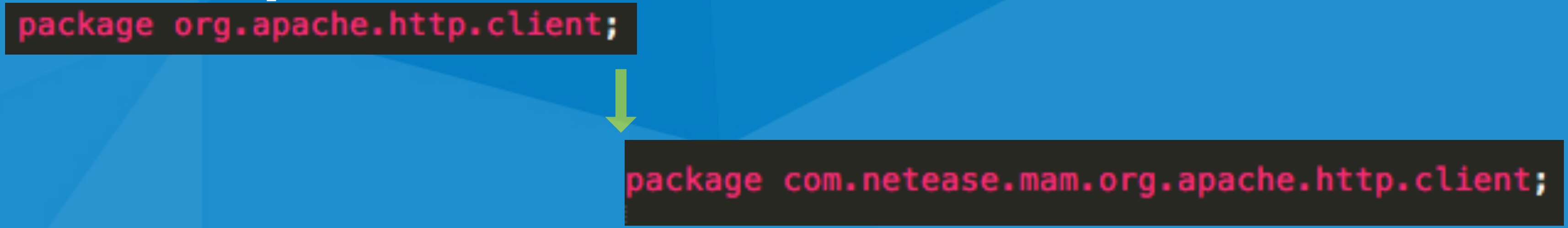
```java
  HttpResponse response = httpClient.execute(postMethod);
```

```java
HttpResponse response =  HttpInstrumentation.execute(httpClient,postMethod);
```

```java
public static HttpResponse execute(HttpClient httpClient, HttpUriRequest request)
        throws IOException {
    try {
        //探针代码
        HttpResponse response = httpClient.execute(request);
        //探针代码
        return response;
    } catch (IOException e) {
        Tracer.exception(e);
        throw e;
    }
}
```

更深的监控

- 内置ApacheClient



```
package org.apache.http.client;
```

```
package com.netease.mam.org.apache.http.client;
```

- 内置URLConnection: custom okhttp-urlconnection
- 外置OKHttp

更多的切入点

- Time To First Byte
- Connect time/Read time
- DNS
- …

其他在做的

- 网络诊断
- 私有DNS解析代替local DNS
- …

接入产品

LOFTER
網易
NEWS
1元
夺宝

Thank You

http://blog.aaapei.com