

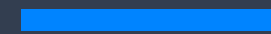
**DataClassGenerate.**

# **Shrinking Kotlin Data Classes**

---

Sergei Rybalkin  
Meta

# THE COST OF “DATA”



# THE COST OF “DATA”

```
data class DC(  
    val i: Int,  
    val s: String?,  
    val a: Any  
)
```

# THE COST OF “DATA”

```
// data class
data class DC(
    val i: Int,
    val s: String?,
    val a: Any
)
```

```
// regular class
class RC(
    val i: Int,
    val s: String?,
    val a: Any
)
```

# THE COST OF “DATA”

/\*

Data Class

after compilation:



bytecode – **2.9 Kb**

DEX code – **2.7 Kb**

\*/

**data class** DC(...)

/\*

Regular Class

after compilation:

bytecode – **1.3 Kb**

DEX code – **1.4 Kb**

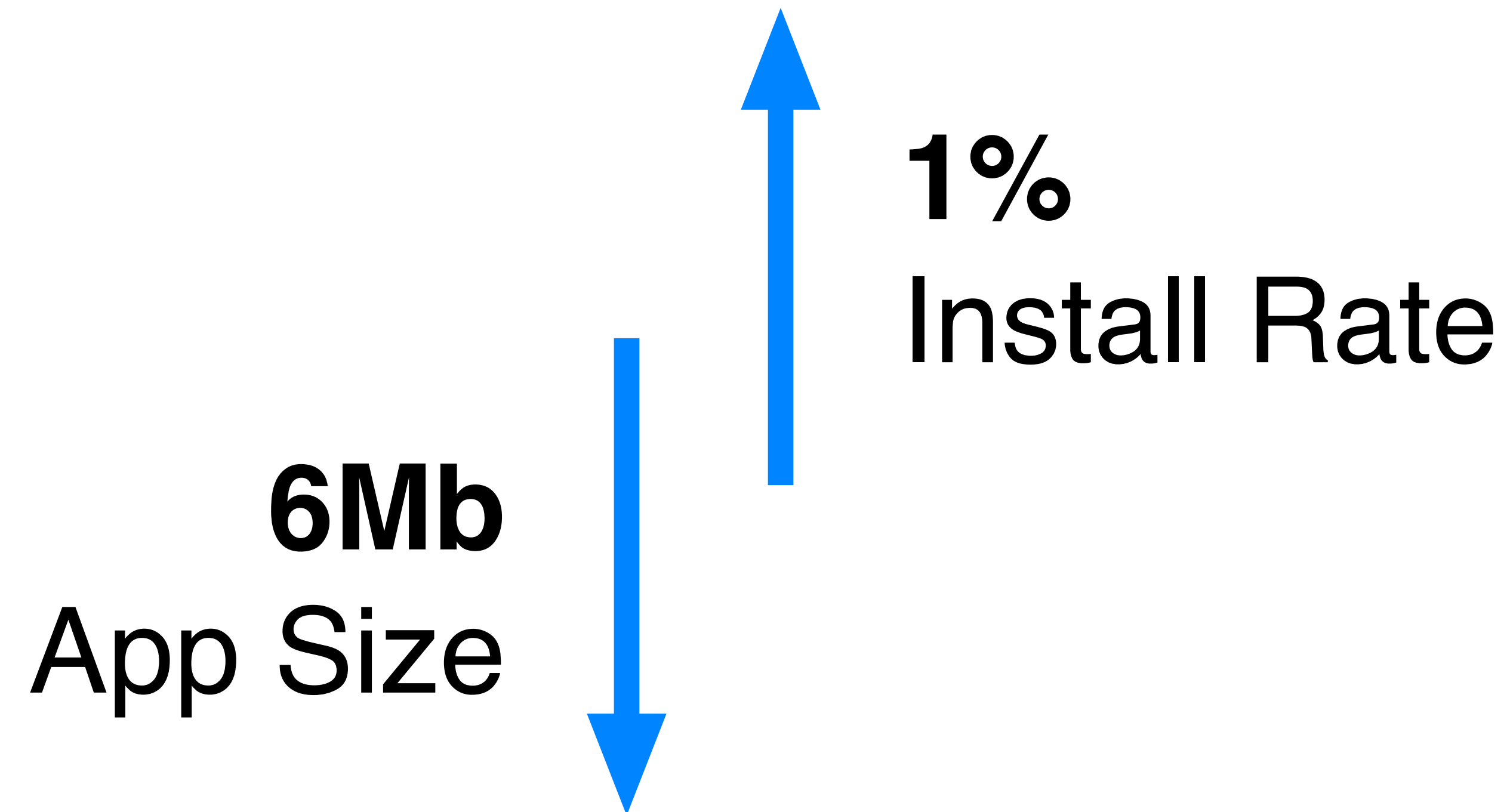
\*/

**class** RC(...)

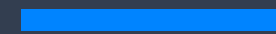
## DATA CLASSES. BYTECODE

```
class DC
  <init>(...)
  getI() | getS() | getA()
  component1()
  component2()
  component3()
  copy()
  synthetic copy$default
  toString()
  hashCode()
  equals(...)
```

# BUT WHO CARES?



# ANDROID PIPELINE





# ANDROID PIPELINE

---

# ANDROID PIPELINE

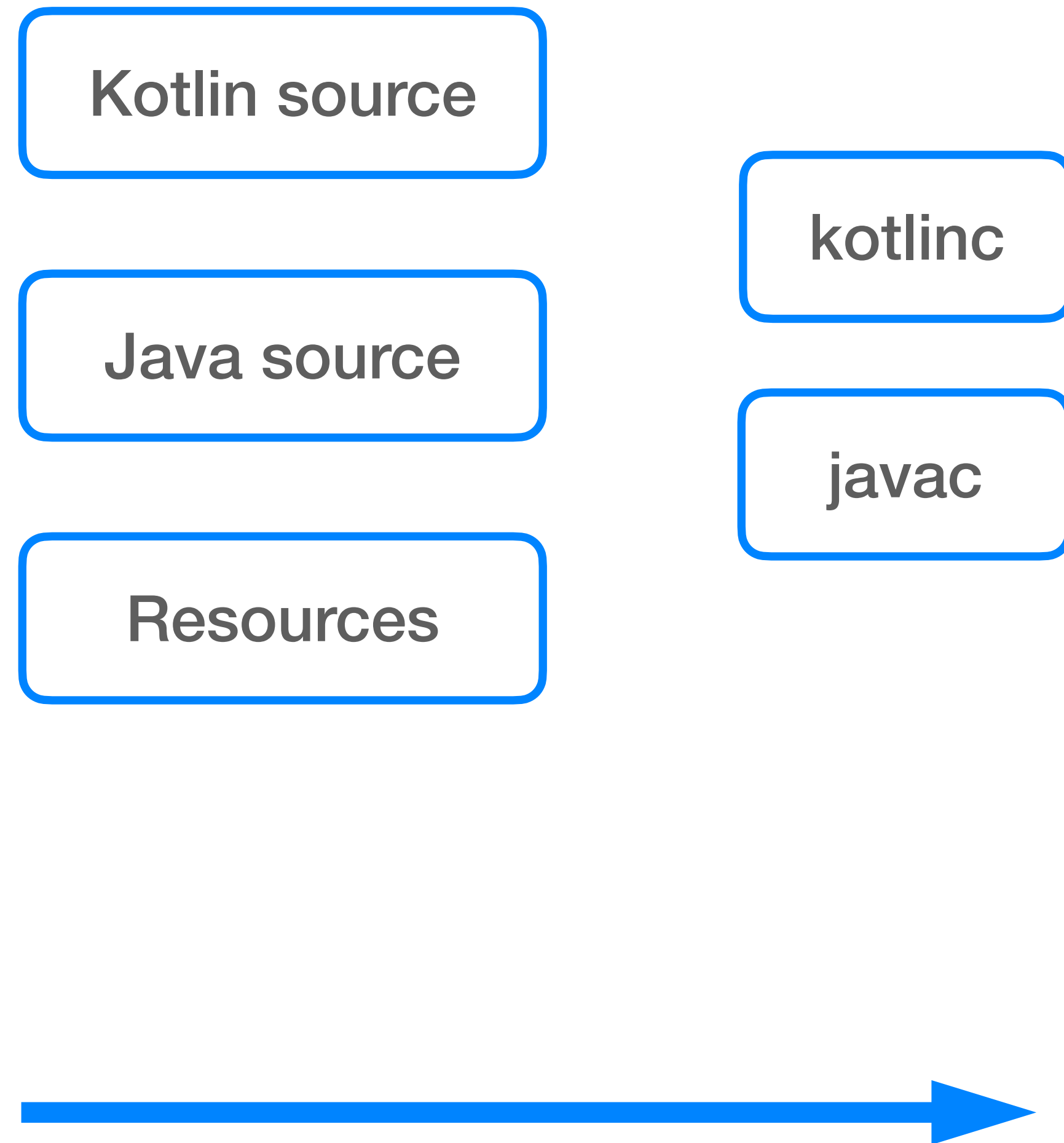
---

Kotlin source

Java source

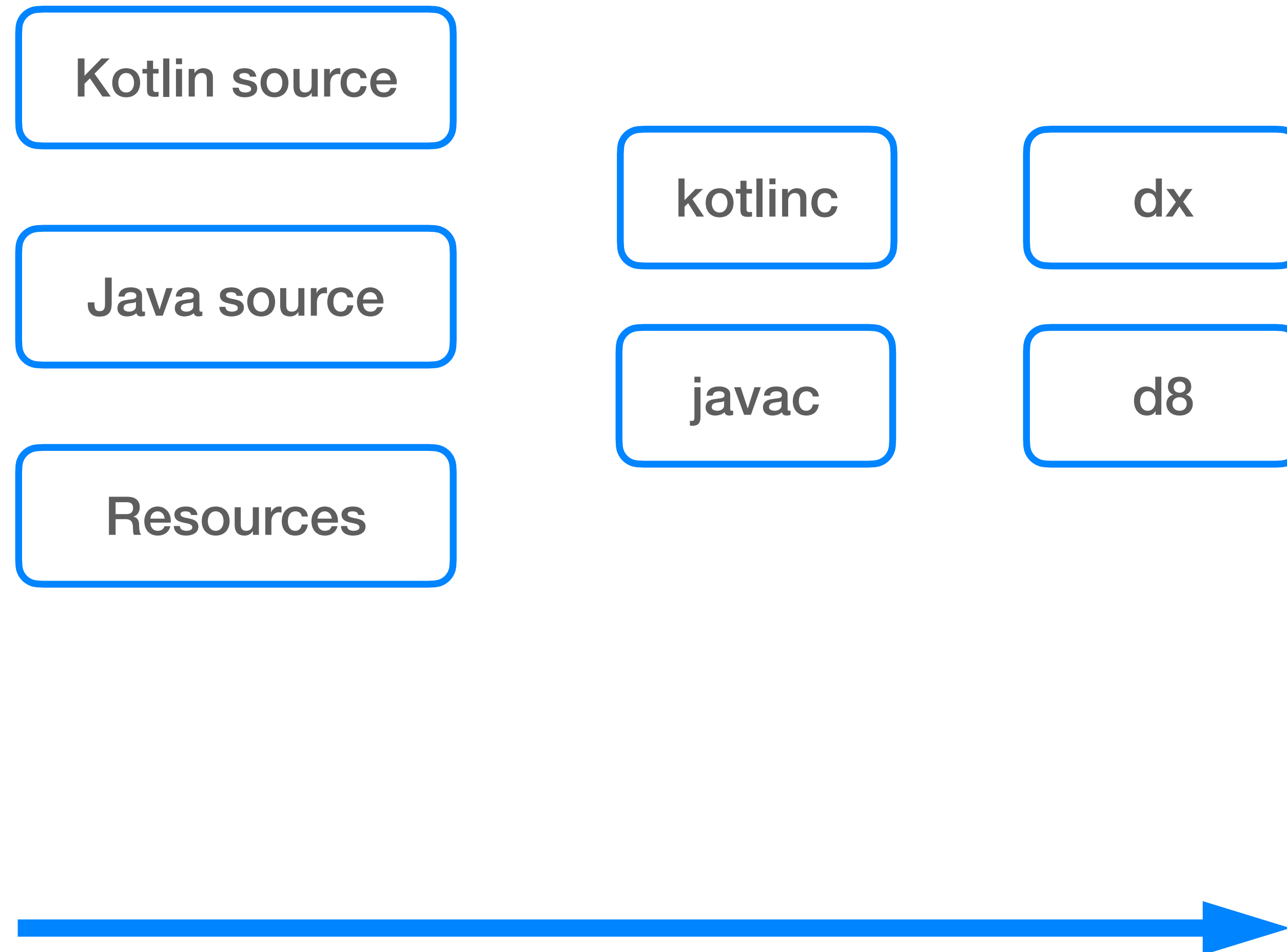
Resources

# ANDROID PIPELINE



# ANDROID PIPELINE

---



# ANDROID PIPELINE



# ANDROID PIPELINE



# ANDROID BYTECODE OPTIMIZERS



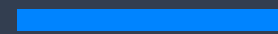
R8

ProGuard



Redex

# BYTECODE OPTIMISATIONS





# BYTECODE OPTIMISATIONS

```
fun foo(): Int {  
    return bar().i  
}
```

```
fun bar(): DC = DC(42, null, Any())
```

```
data class DC(  
    val i: Int,  
    val s: String?,  
    val a: Any  
)
```

# INLINING #1

```
fun foo(): Int {  
    return DC(42, null, Any()).i  
}
```

```
fun bar(): DC = DC(42, null, Any())
```

```
data class DC(  
    val i: Int,  
    val s: String?,  
    val a: Any  
)
```

# REMOVE UNREACHABLE / LOCAL DCE

```
fun foo(): Int {  
    return DC(42, null, Any()).i  
}
```

```
fun bar(): DC = DC(42, null, Any())
```

```
data class DC(  
    val i: Int,  
    val s: String?,  
    val a: Any  
)
```

## INLINING #2

```
fun foo(): Int {  
    return 42  
}
```

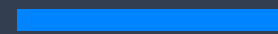
```
data class DC(  
    val i: Int,  
    val s: String?,  
    val a: Any  
)
```

# REMOVE UNREACHABLE / LOCAL DCE #2

```
fun foo(): Int {  
    return 42  
}
```

```
data class DC(  
    val i: Int,  
    val s: String?,  
    val a: Any  
)
```

# DATA CLASS OPTIMISATIONS



## DATA CLASSES. BYTECODE

```
class DC
  <init>(...)
  getI() | getS() | getA()
  component1()
  component2()
  component3()
  copy()
  synthetic copy$default
  toString()
  hashCode()
  equals(...)
```

## DATA CLASSES. BYTECODE

```
class DC  
  getI() | getS() | getA()
```

Inlining Pass (R8 / Redex)



## DATA CLASSES. BYTECODE

```
class DC
  component1()
  component2()
  component3()
```

Inlining Pass

Dead Code Elimination Pass (R8 / Redex)

## DATA CLASSES. BYTECODE

```
class DC  
—copy()  
—synthetic copy$default
```

Dead Code Elimination Pass (R8 / Redex)

## DATA CLASSES. HARD ONES

```
class DC
    toString()Ljava/lang/String;
    hashCode()I
    equals(Ljava/lang/Object;)Z
```

# DATA CLASSES. TOSTRING

```
#9          : (in LDC;)

name        : 'toString'
type        : '()Ljava/lang/String;'

access      : 0x0001 (PUBLIC)
code        -
registers    : 4
ins         : 1
outs        : 3

insns size  : 45 16-bit code units

0004d0:      |[0004d0] DC.toString:()Ljava/lang/String;
0004e0: 2200 0600      |0000: new-instance v0, Ljava/lang/StringBuilder; // type@0006
0004e4: 7010 0f00 0000  |0002: invoke-direct {v0}, Ljava/lang/StringBuilder;.<init>:()V // method@000f
0004ea: 1a01 1b00      |0005: const-string v1, "DC(i=" // string@001b
0004ee: 6e20 1200 1000  |0007: invoke-virtual {v0, v1}, Ljava/lang/StringBuilder;.append:(Ljava/lang/String;)Ljava/lang/StringBuilder; //
0004f4: 5331 0000      |000a: iget-wide v1, v3, LDC;.i // field@0000
0004f8: 6e30 1000 1002  |000c: invoke-virtual {v0, v1, v2}, Ljava/lang/StringBuilder;.append:(J)Ljava/lang/StringBuilder; // method@0010
0004fe: 1a01 0a00      |000f: const-string v1, ", s=" // string@000a
000502: 6e20 1200 1000  |0011: invoke-virtual {v0, v1}, Ljava/lang/StringBuilder;.append:(Ljava/lang/String;)Ljava/lang/StringBuilder; //
000508: 5431 0200      |0014: iget-object v1, v3, LDC;.s:Ljava/lang/String; // field@0002
00050c: 6e20 1200 1000  |0016: invoke-virtual {v0, v1}, Ljava/lang/StringBuilder;.append:(Ljava/lang/String;)Ljava/lang/StringBuilder; //
000512: 1a01 0900      |0019: const-string v1, ", a=" // string@0009
000516: 6e20 1200 1000  |001b: invoke-virtual {v0, v1}, Ljava/lang/StringBuilder;.append:(Ljava/lang/String;)Ljava/lang/StringBuilder; //
00051c: 5431 0100      |001e: iget-object v1, v3, LDC;.a:LObject; // field@0001
000520: 6e20 1100 1000  |0020: invoke-virtual {v0, v1}, Ljava/lang/StringBuilder;.append:(Ljava/lang/Object;)Ljava/lang/StringBuilder; //
000526: 1a01 0800      |0023: const-string v1, ")" // string@0008
00052a: 6e20 1200 1000  |0025: invoke-virtual {v0, v1}, Ljava/lang/StringBuilder;.append:(Ljava/lang/String;)Ljava/lang/StringBuilder; //
000530: 6e10 1300 0000  |0028: invoke-virtual {v0}, Ljava/lang/StringBuilder;.toString:()Ljava/lang/String; // method@0013
000536: 0c00          |002b: move-result-object v0
000538: 1100          |002c: return-object v0
```

## DATA CLASSES. BYTECODE

**// Saves up to 900 bytes (60% of overhead)**

```
class DC
  <init>(...)
  getI() | getS() | getA()
  component1()
  component2()
  component3()
  copy()
  synthetic copy$default
  toString()
  hashCode()
  equals(...)
```

# DATACLASSGENERATE

## DCG



# DATACLASSGENERATE. DCG

```
@DataClassGenerate(  
    toString = Mode.OMIT,  
    equalsHashCode = Mode.KEEP  
)  
data class DC(  
    val i: Int, val s: String?, val a: Any  
)
```

# DCG. ANNOTATION

```
annotation class DataClassGenerate(  
    val toString: Mode = Mode.OMIT,  
    val equalsHashCode: Mode = Mode.KEEP  
)  
  
enum class Mode {  
    /** Express an intent to keep/generate a method(s)  
    DataClassGenerate. */  
    KEEP,  
    /** Express an intent to omit method(s) generation.  
    */  
    OMIT  
}
```



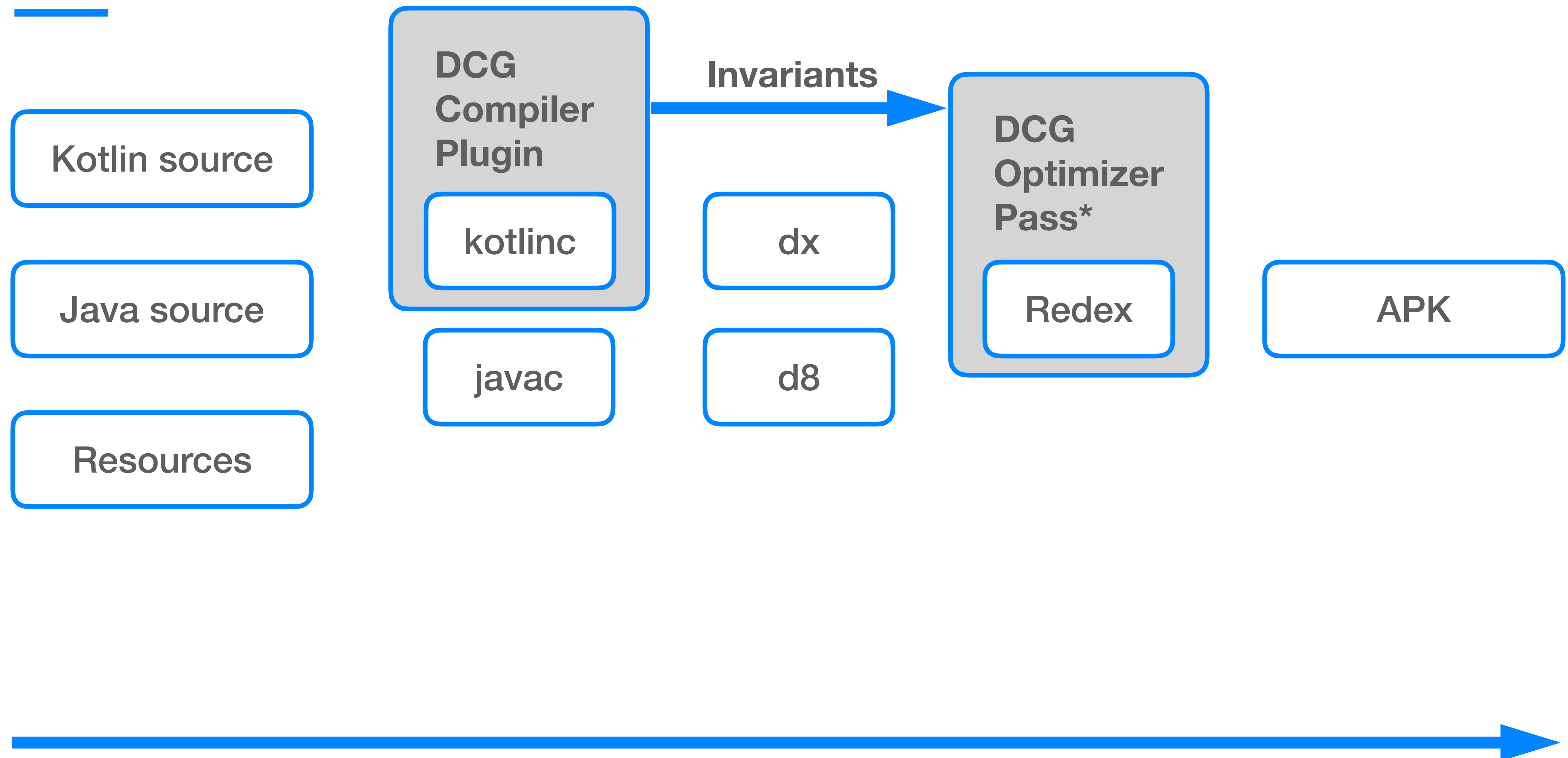
## DCG. SIMPLIFIED EQUIVALENT

```
data class DC(  
    val i: Int, val s: String?, val a: Any  
) {  
    // Saves 300 bytes (~20% of overhead)  
    // @DataClassGenerate(toString = Mode.OMIT)  
    override fun toString(): String {  
        return super.toString()  
    }  
}
```

# DCG. SIMPLIFIED EQUIVALENT

```
data class DC(  
    val i: Int, val s: String?, val a: Any  
) {  
    // Saves 200 bytes (~13% of overhead)  
    // @DataClassGenerate(equalsHashCode = Mode.OMIT)  
    override fun equals(other: Any?): Boolean {  
        return super.equals(other)  
    }  
    override fun hashCode(): Int {  
        return super.hashCode()  
    }  
}
```

# DCG IN ANDROID PIPELINE



# DCG. IMPLEMENTATION



# DCG. IMPLEMENTATION

```
class DataClassGenerateBuilder(  
    val classBuilder: ClassBuilder,  
    private val mode: PluginMode,  
) : DelegatingClassBuilder() {  
  
    override fun defineClass(args) {  
        ...  
    }  
  
    override fun newMethod(args): MethodVisitor {  
        ...  
    }  
}
```

# DCG. IMPLEMENTATION

```
class DataClassGenerateBuilder(...) {  
  
    override fun defineClass(args) {  
        1. Check if declaration is a data classes.  
        2. Decide if you want to process synthetic  
classes.  
        3. Check if DCG is compatible with other  
attached plugins.  
        4. Check @DataClassGenerate annotation.  
        5. Generate readable Errors.  
    }  
}
```

# DCG. IMPLEMENTATION

```
class DataClassGenerateBuilder(...) {  
  
    override fun newMethod(args): MethodVisitor {  
        1. Check if we need to apply DCG.  
        2. Check hand-written method overrides.  
        3. Redefine a method visitor:  
            0: aload_0  
            Method com/facebook/kotlin/compilerplugins/dataclassgenerate/superclass/.  
              DataClassSuper.toString:()Ljava/lang/String;  
            1: invokespecial  
            4: areturn  
    }  
}
```

# IMPLEMENTATION. DCG

```
class DataClassGenerateInterceptorExtension(  
    val pluginMode: PluginMode,  
) : ClassBuilderInterceptorExtension {  
  
    override fun interceptClassBuilderFactory(...  
): ClassBuilderFactory {  
        return interceptor {  
            override fun newClassBuilder(...) =  
                DataClassGenerateBuilder(..., pluginMode)  
        }  
    }  
}
```



# IMPLEMENTATION. DCG

---

```
enum class PluginMode {  
    /** Plugin applies only to data classes annotated with [@DataClassGenerate] */  
    EXPLICIT,  
  
    /**  
     * Same as [EXPLICIT], but Plugin will report a compilation error whenever it  
    see a data class  
     * without [@DataClassGenerate] annotation.  
    */  
    STRICT,  
  
    /**  
     * Plugin applies to all data classes, if data class is not annotated with  
    [@DataClassGenerate] it  
     * will be treated as annotated with [@DataClassGenerate]  
     *  
    */  
    IMPLICIT  
}
```

# THANK YOU



Sergei Rybalkin  
@lightdelay