Veni, Vidi, Built:
Android Gradle Plugin

LYK dalinaum@gmail.com

Julius Caesar

BC 100 ~ BC 44

Veni, Vidi, Vici

왔노라, 보았노라, 이겼노라

Julius Caesar

BC 100 ~ BC 44

Veni, Vidi, Built

왔노라, 보았노라, 빌드했노라

- Gradle
- Tasks
- Android custom tasks
- New Android Plugin
- NDK
- Jack & Jill
- Atom Android

Gradle

- Configuration (Static) + Build (Dynamic)

Ant

- Focus on Build
- hard to configure

Maven

- Focus on Configuration
- hard to customise

Alt.

- Gradle
  - Groovy language (Dynamic)
  - DSL (Static)
- Apache Ivy

Groovy

- Runs on Java VM
- The principle of least surprise (JAVA)
- Built-in List, Range, Map, Closure
- Improved Loops, Switches
- Built-in Regex, Improved Strings
- Supports Dynamic typing and Static typing.

Task: Hello World

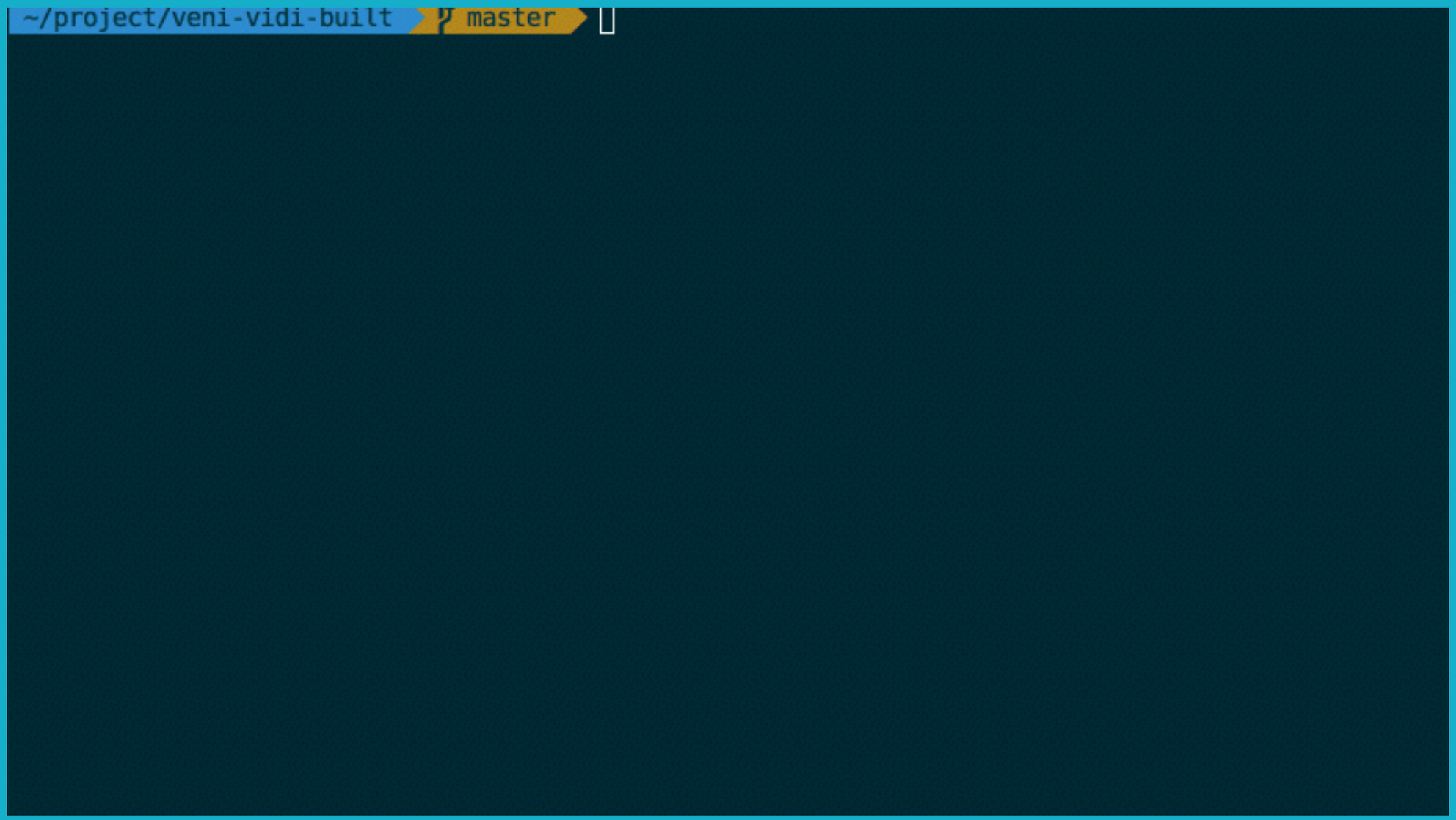
```
task hello {

    doLast {
        println 'Hello world!'
    }
}
```

gradle -q hello

Task: Hello World (short version)

```
task hello << {
    println 'Hello world!'
}
```

gradle -q hello



upper, count

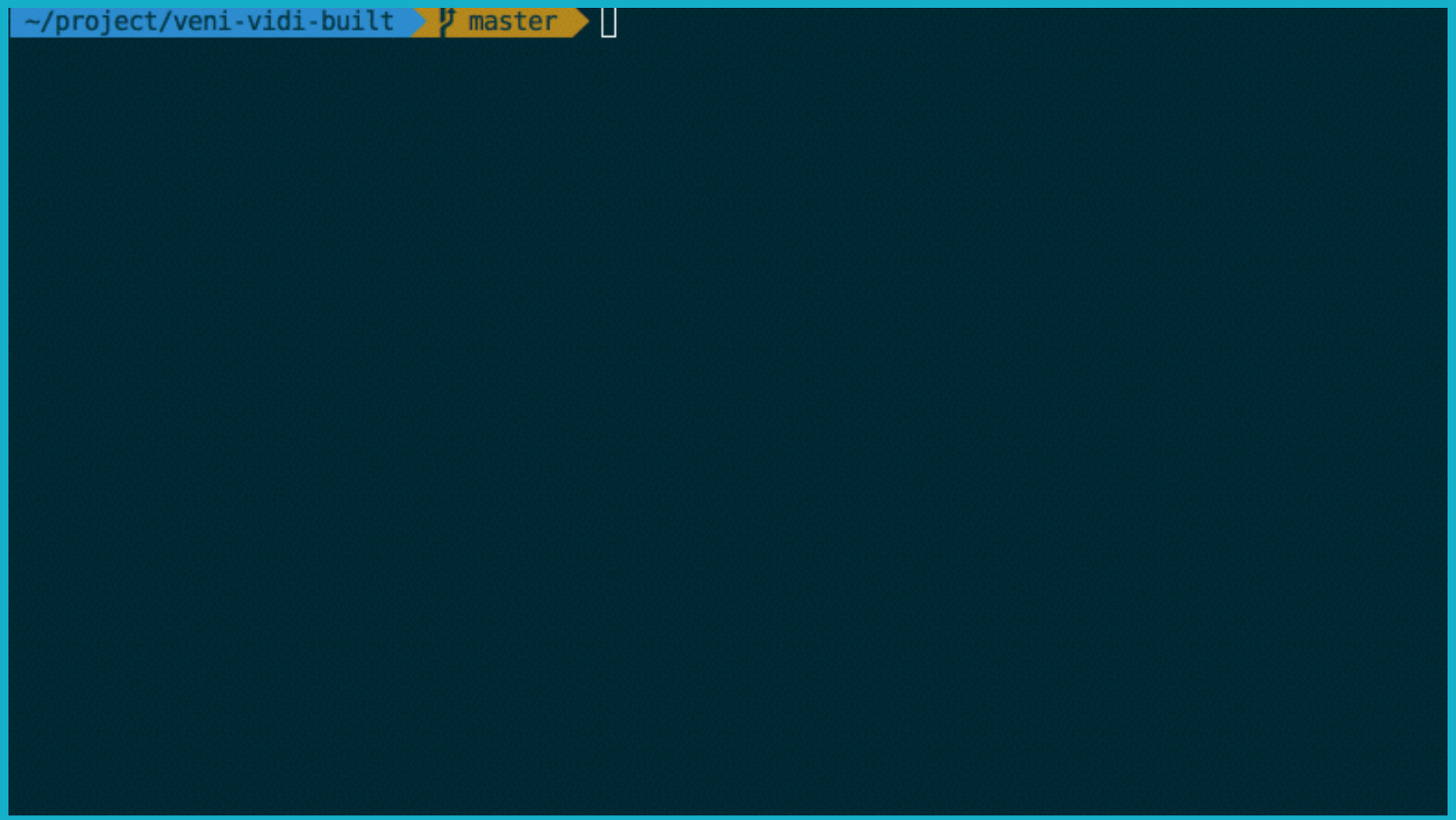
```
task upper << {
    String someString = 'mY_nAmE'
    println "Original: " + someString
    println "Upper case: " + someString.toUpperCase()
}


task count << {
    4.times { print "$it " }
}
```

gradle -q upper count

Dependency

```
task upper << {

    String someString = 'mY_nAmE'

    println "Original: " + someString

    println "Upper case: " + someString.toUpperCase()

}


task count << {

    4.times { print "$it " }

}
```

gradle -q intro



Android Gradle

build.grade (all)

app/build.gradle (app module)

settings.gradle (import app module)

```
~/project/veni-vidi-built/04-hello-android  master  cat settings.gradle
include ':app'
```

jCenter

JCenter is the place to find and share popular Apache Maven packages for use by Maven, Gradle, Ivy, SBT, etc.
For the most comprehensive collection of artifacts, point your Maven at: http://jcenter.bintray.com
Want to distribute your own packages through JCenter? You can link your package by clicking the "Include My Package" button.
And if you're into legacy, you can even synchronize your packages directly to Maven Central.

```
buildscript {
    repositories {
        maven {
          url  "http://dl.bintray.com/android/android-tools"
        }
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle-experimental:0.1.0'
    }
}

allprojects {
    repositories {
        jcenter()
```

Android custom task

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 22
    buildToolsVersion "23.0.0 rc3"

    defaultConfig {...}
    buildTypes {...}
}

task helloWorld(type: Exec) {
    executable './hello-world.sh'
}

gradle.projectsEvaluated {
    preBuild.dependsOn(helloWorld)
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:22.2.0'
}
```

```
~/project/veni-vidi-built/04-hello-android/app  master  cat hello-world.sh
#!/bin/bash
for i in {1..100}
do
     echo "Hello GDG, $i times"
 done
```

gradle -q assemble

New Android Plugin (Experimental)

gradle/wrapper/gradle-wrapper.properties

```
#Sun Jul 12 09:43:24 KST 2015
distributionBase=GRADLE_USER_HOME
distributionPath=wrapper/dists
zipStoreBase=GRADLE_USER_HOME
zipStorePath=wrapper/dists
distributionUrl=https\://services.gradle.org/distributions/gradle-2.5-all.zip
```

build.gradle

```
buildscript {
    repositories {
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle-experimental:0.1.0'

        // NOTE: Do not place your application dependencies here; they belong
        // in the individual module build.gradle files
    }
}

allprojects {
    repositories {
        jcenter()
    }
}
```

app/build.gradle

```
apply plugin: 'com.android.model.application'

model {
    android {
        compileSdkVersion = 22
        buildToolsVersion = "23.0.0 rc3"

        defaultConfig.with {
            applicationId = "gdg.kr.hellogdg"
            minSdkVersion.apiLevel = 11
            targetSdkVersion.apiLevel = 22
            versionCode = 1
            versionName = "1.0"
        }
    }

    android.buildTypes {
        release {
            isMinifyEnabled = false
            proguardFiles += file('proguard-rules.pro')
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:22.2.0'
}
```

component model mechanism (Gradle 2.5)

```
apply plugin: 'com.android.model.application'

model {
    android {
        compileSdkVersion = 22
        buildToolsVersion = "23.0.0 rc3"

        defaultConfig.with {
            applicationId = "gdg.kr.hellogdg"
            minSdkVersion.apiLevel = 11
            targetSdkVersion.apiLevel = 22
            versionCode = 1
            versionName = "1.0"
        }
    }

    android.buildTypes {
        release {
            isMinifyEnabled = false
            proguardFiles += file('proguard-rules.pro')
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:22.2.0'
}
```

```
apply plugin: 'com.android.model.application'

model {
    android {
        compileSdkVersion = 22
        buildToolsVersion = "23.0.0 rc3"

        defaultConfig.with {
            applicationId = "gdg.kr.hellogdg"
            minSdkVersion.apiLevel = 11
            targetSdkVersion.apiLevel = 22
            versionCode = 1
            versionName = "1.0"
        }
    }

    android.buildTypes {
        release {
            isMinifyEnabled = false
            proguardFiles += file('proguard-rules.pro')
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:22.2.0'
}
```


~/project/veni-vidi-built/05-new-gradle
master

NDK support (New Android Plugin)

local.properties

```
sdk.dir=/opt/sdk
ndk.dir=/usr/local/Cellar/android-ndk/r10d/
```

build.gradle

```
1   apply plugin: 'com.android.model.application'
2
3   model {
4       android {
5           compileSdkVersion = 22
6           buildToolsVersion = "23.0.0 rc3"
7
8           defaultConfig.with { ... }
9       }
10
11      android.ndk {
12          moduleName = "hello-jni"
13      }
14
15      android.buildTypes { ... }
```

```
16
17      android.productFlavors {
18          create("arm") {
19              ndk.abiFilters += "armeabi"
20          }
21          create("arm7") {
22              ndk.abiFilters += "armeabi-v7a"
23          }
24          create("arm8") {
25              ndk.abiFilters += "arm64-v8a"
26          }
27          create("x86") {
28              ndk.abiFilters += "x86"
29          }
30          create("x86-64") {
31              ndk.abiFilters += "x86_64"
32          }
33          create("mips") {
34              ndk.abiFilters += "mips"
35          }
36          create("mips-64") {
37              ndk.abiFilters += "mips64"
38          }
39          create("all")
40      }
```

app/src/main/jni/

hello-jni.c

```
/* This is a trivial JNI example where we use a native method
 * to return a new VM String. See the corresponding Java source
 * file located at:
 *
 *   apps/samples/hello-jni/project/src/com/example/hellojni/HelloJni.java
 */
jstring
Java_com_example_hellojni_HelloJni_stringFromJNI( JNIEnv* env,
                                                  jobject thiz )
{
#if defined(__arm__)
  #if defined(__ARM_ARCH_7A__)
    #if defined(__ARM_NEON__)
      #if defined(__ARM_PCS_VFP)
        #define ABI "armeabi-v7a/NEON (hard-float)"
      #else
        #define ABI "armeabi-v7a/NEON"
      #endif
    #else
      #if defined(__ARM_PCS_VFP)
        #define ABI "armeabi-v7a (hard-float)"
      #else
        #define ABI "armeabi-v7a"
      #endif
    #endif
  #else
   #define ABI "armeabi"
  #endif
#elif defined(__i386__)
   #define ABI "x86"
#elif defined(__x86_64__)
   #define ABI "x86_64"
#elif defined(__mips64)  /* mips64el-* toolchain defines __mips__ too */
   #define ABI "mips64"
#elif defined(__mips__)
   #define ABI "mips"
#elif defined(__aarch64__)
   #define ABI "arm64-v8a"
#else
   #define ABI "unknown"
#endif

    return (*env)->NewStringUTF(env, "Hello from JNI !  Compiled with ABI " ABI ".");
}
```

app/src/main/java/
gdg/kr/hellogdg/

```java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    TextView  tv = new TextView(this);
    tv.setText( stringFromJNI() );
    setContentView(tv);
}
```

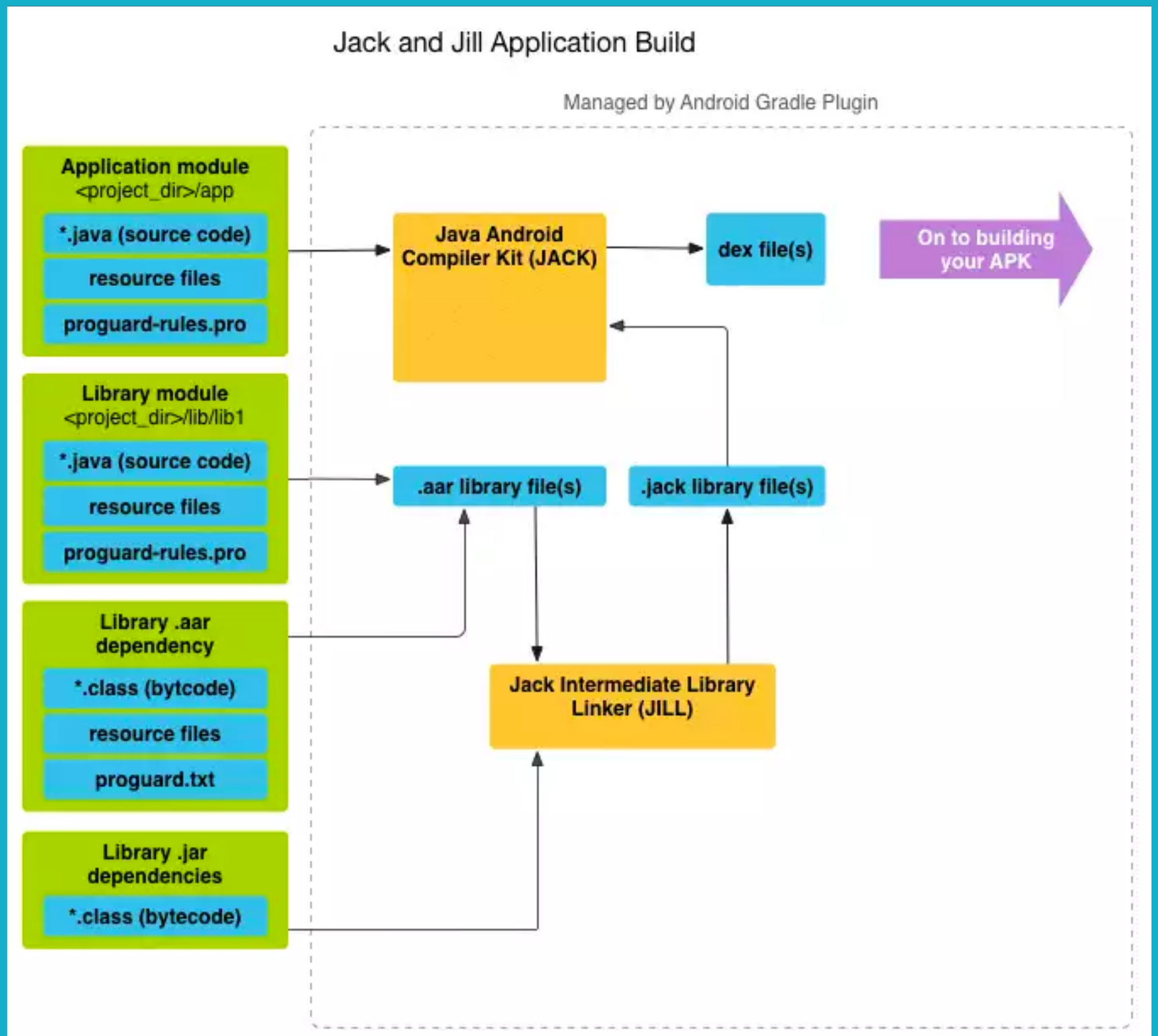
```java
public native String  stringFromJNI();

public native String  unimplementedStringFromJNI();

static {
    System.loadLibrary("hello-jni");
}
```

NDK example:

https://github.com/googlesamples/android-ndk

JACK & JILL

- JACK - Java Android Compiler Kit
- JILL - Jack Intermediate Library Linker


Jack and Jill Application Build
Managed by Android Gradle Plugin
Application module
<project_dir>/app
*.java (source code)
resource files
proguard-rules.pro
Library module
<project_dir>/lib/lib1
*.java (source code)
resource files
proguard-rules.pro
Library .aar
dependency
*.class (bytcode)
resource files
proguard.txt
Library .jar
dependencies
*.class (bytecode)
Java Android
Compiler Kit (JACK)
dex file(s)
On to building
your APK
.aar library file(s)
.jack library file(s)
Jack Intermediate Library
Linker (JILL)

build.gradle

```
defaultConfig {
    applicationId "gdg.kr.hellogdg"
    minSdkVersion 11
    targetSdkVersion 22
    versionCode 1
    versionName "1.0"
    useJack = true
}
```

Atom Android

- Atom plugin for Android
- https://github.com/atom/apm

Fin.