

# ANDROID 292

## IMPLEMENTING INVOKEDYNAMIC IN ANDROID

Jérôme Pilliet, Rémi Forax and Gilles Roussel  
University of Paris-Est Marne-la-Vallée

# JAVA SPECIFICATION REQUEST 292

2006: start of the JSR

2011: included in Java 7

- new instruction: *invokedynamic*
  - flexible method invocation
  - very late linking (at runtime) and dynamic re-linking
- new API: *java.lang.invoke*
  - MethodHandle:
    - type safe function pointers
    - combiners which do arguments fiddling

# ANDROID

Google's OS for smartphones and tablets

- uses the JVM: *Dalvik*
  - almost Java 6 specification
  - registers based
  - *DEX* format: read-only
- limited battery and calculating power

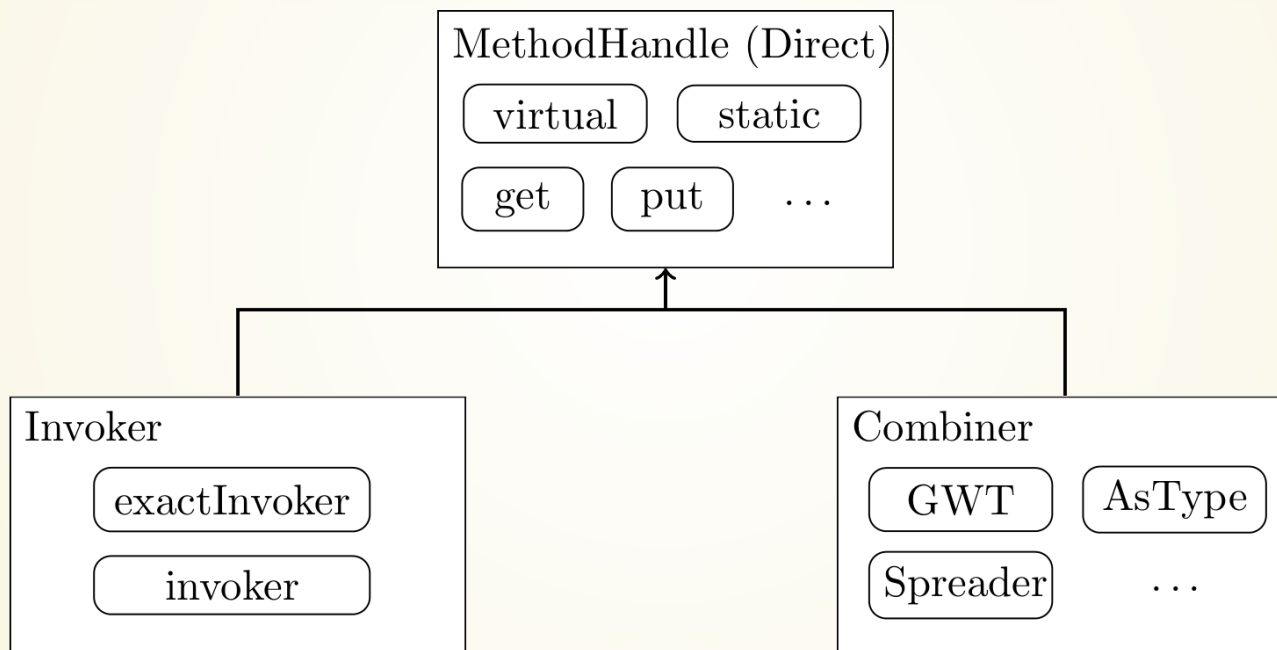
**WHY ?**

**ANDROID 292**

# A NEW SPECIFICATION

- add constant pools and call site count in class format
- one instruction for each new kind of method invocation
  - *MH.invokeExact* → *invoke-exact*
  - *MH.invoke* → *invoke-generic*
  - *invokedynamic* → *invoke-dynamic*
- all these instructions run in the C interpreter

# METHODHANDLE HIERARCHY

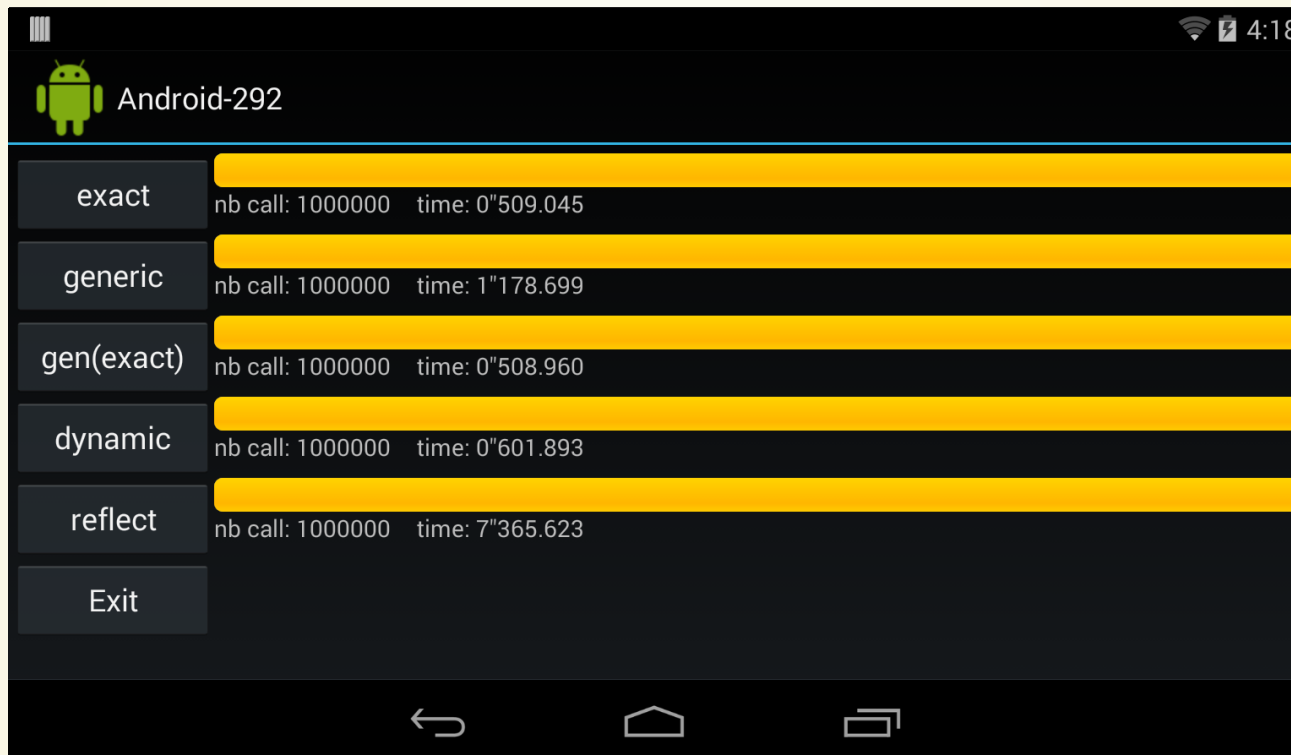


# COMBINERS INTERPRETER

- written in Java
- call by the C interpreter
- all arguments are boxed
- one stack frame for the whole tree
- unboxing when calling a method handle leaf



# RESULTS: INSTRUCTIONS VS REFLECTION

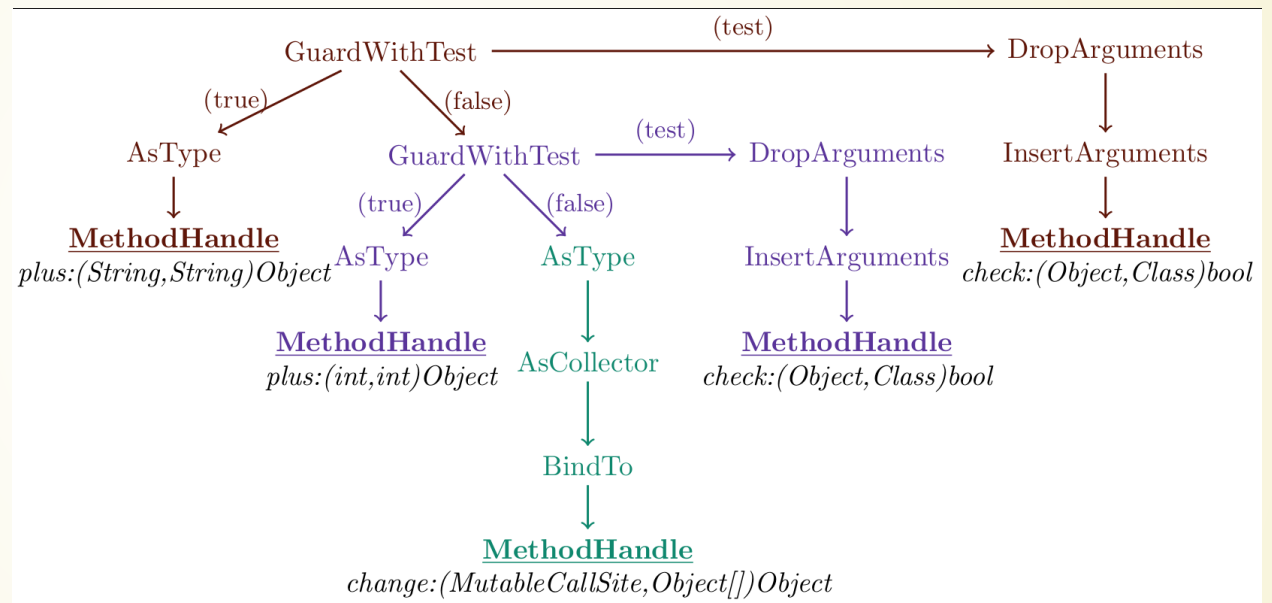


# RESULTS: INSTANCEOF VS COMBINERS INTERPRETER

```

def plus(a, b):
    return a + b

def choose():
    if randint(0, 10) % 2 == 0:
        return plus(40, 2)
    else:
        return plus('foo', 'bar')
    
```



# CONCLUSION

- it works !
- retro-compatibility

# FUTUR

- improve the dual stack prototype
- stack storage the arrays

# ANDROID 292

## QUESTIONS ?

<https://bitbucket.org/jpilliet/android-292>

<https://github.com/brachior/Android-292>