Rubyによる本気のGC

Serious GC with Ruby

@nari3

#sprk2012

2012/9/15

ネットワーク応用通信研究所

提供

うなぎの件

とにかくプログラミング(や鰻)が好きな人ウォンテッド

株式会社ネットワーク応用通信研究所

埋め込む Tweet Like 0 Send SHARE

<URL:https://www.wantedly.com/projects/478>

自己紹介

- ✓ 中村成洋/@nari3/nari/authorNari
- ✓ CRubyのコミッタ
- ✓ GCメンテナ＆デストロイヤ

GCとは（ｒｙ

GoogleTrendでみるGC

Google トレンド

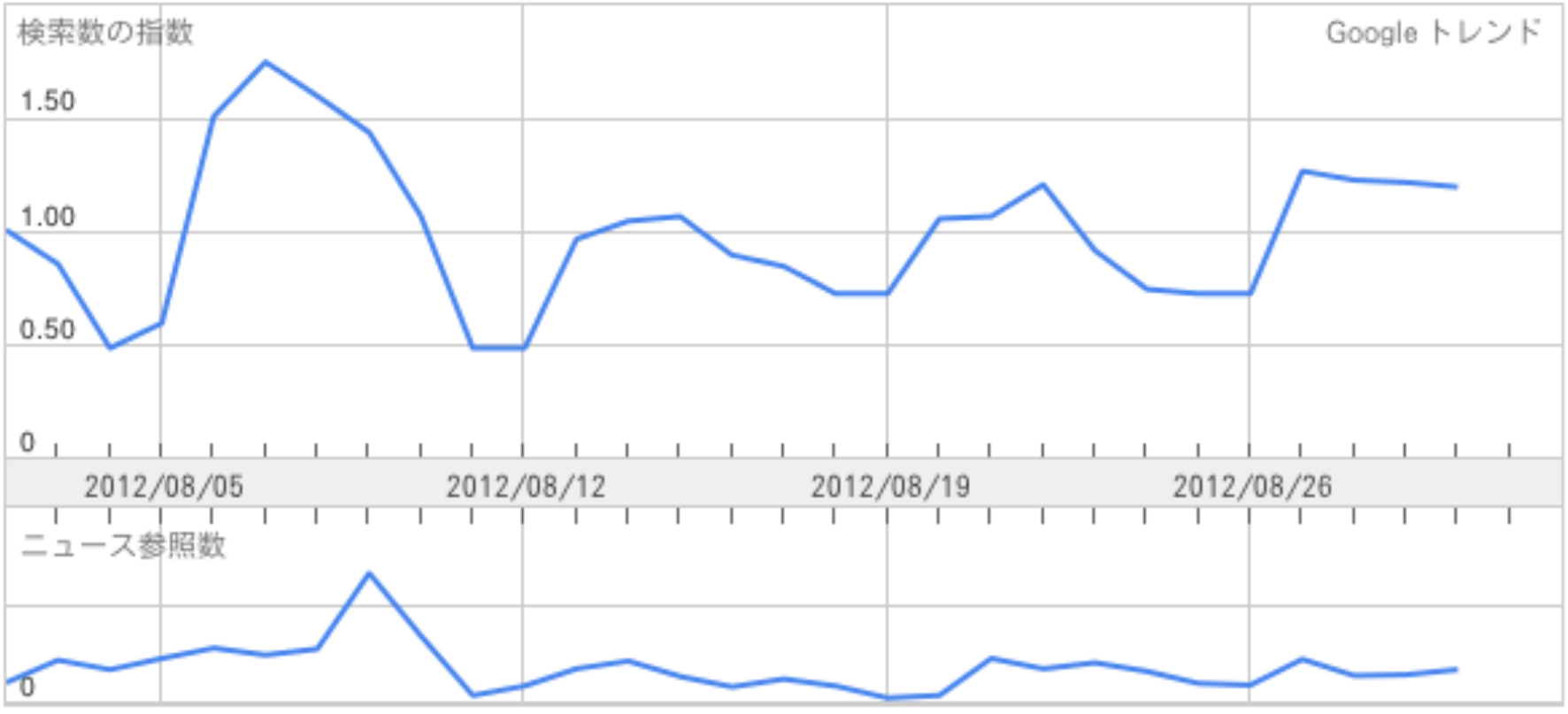
garbage collection

トレンドを検索

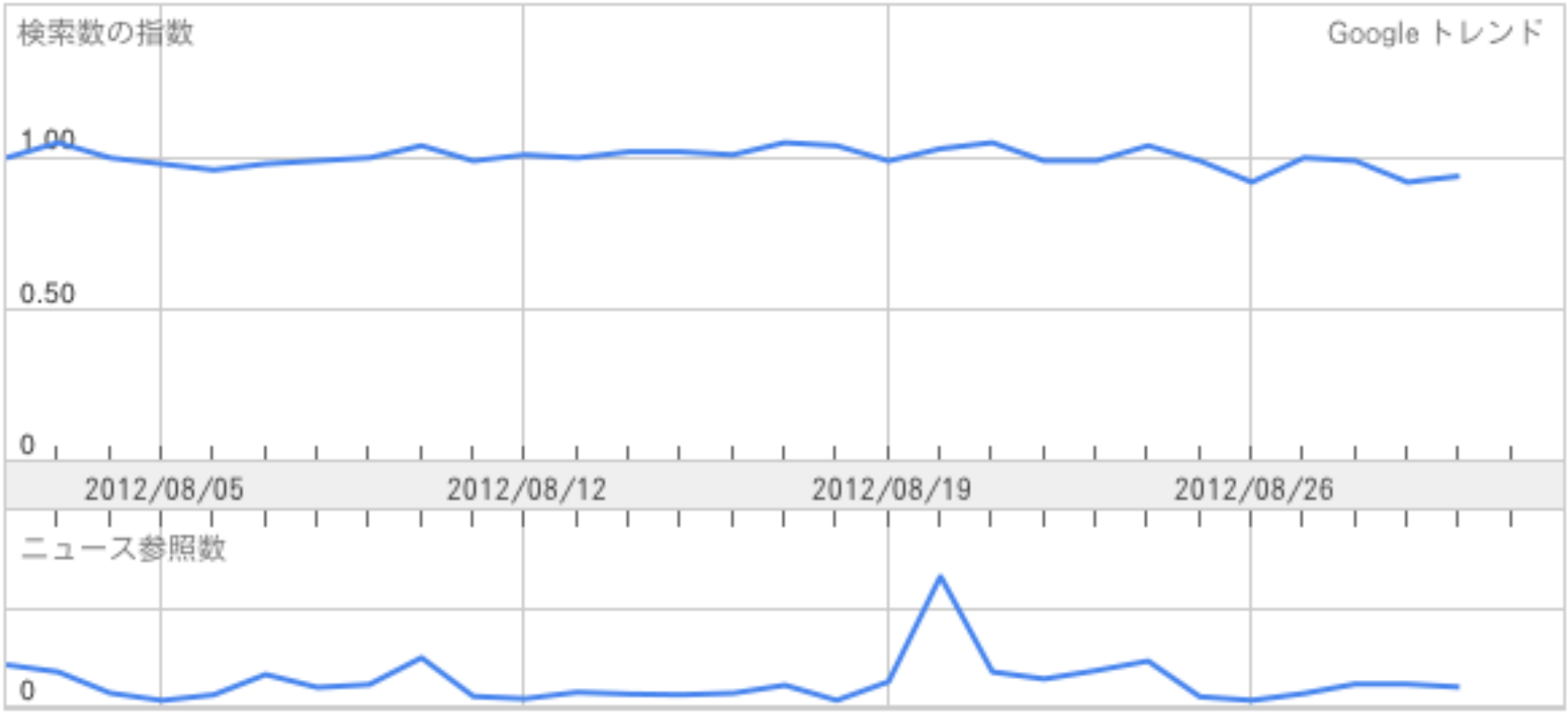
複数のキーワードを比較する場合はカンマか読点で区切ってください。

トレンドの推移

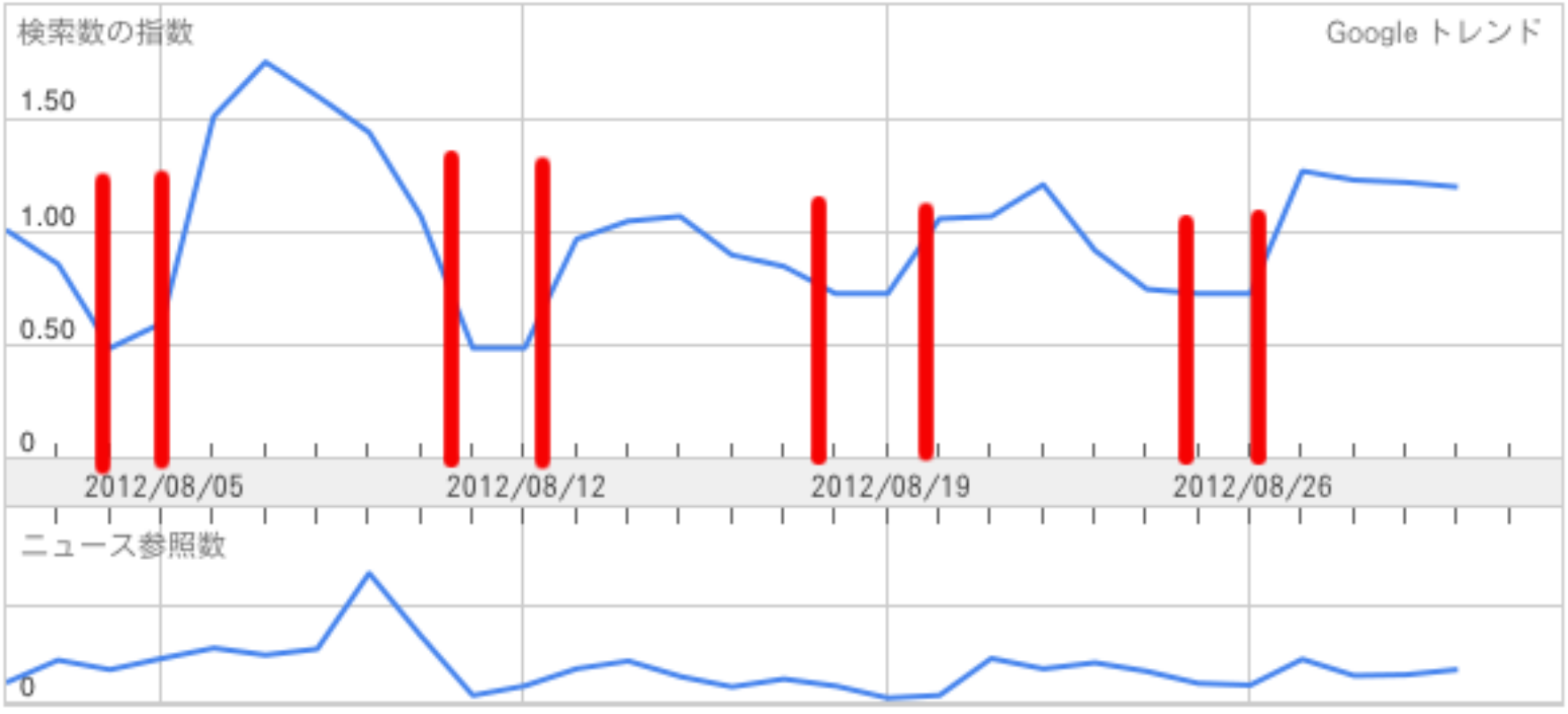
garbage collection 1.00

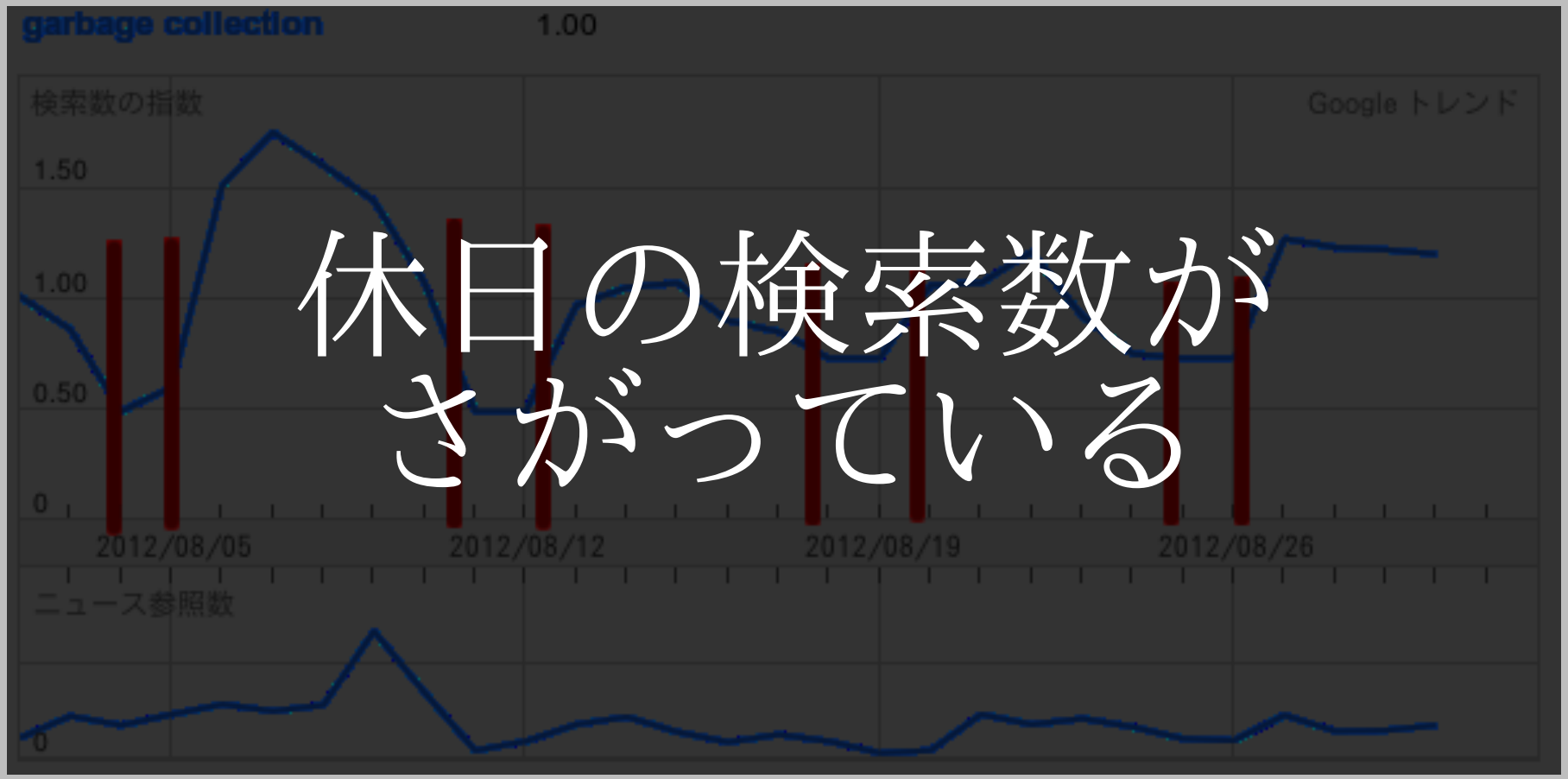


ruby 1.00


検索数の指数
Google トレンド
1.00
0.50
0
2012/08/05
2012/08/12
2012/08/19
2012/08/26
ニュース参照数
0

garbage collection 1.00





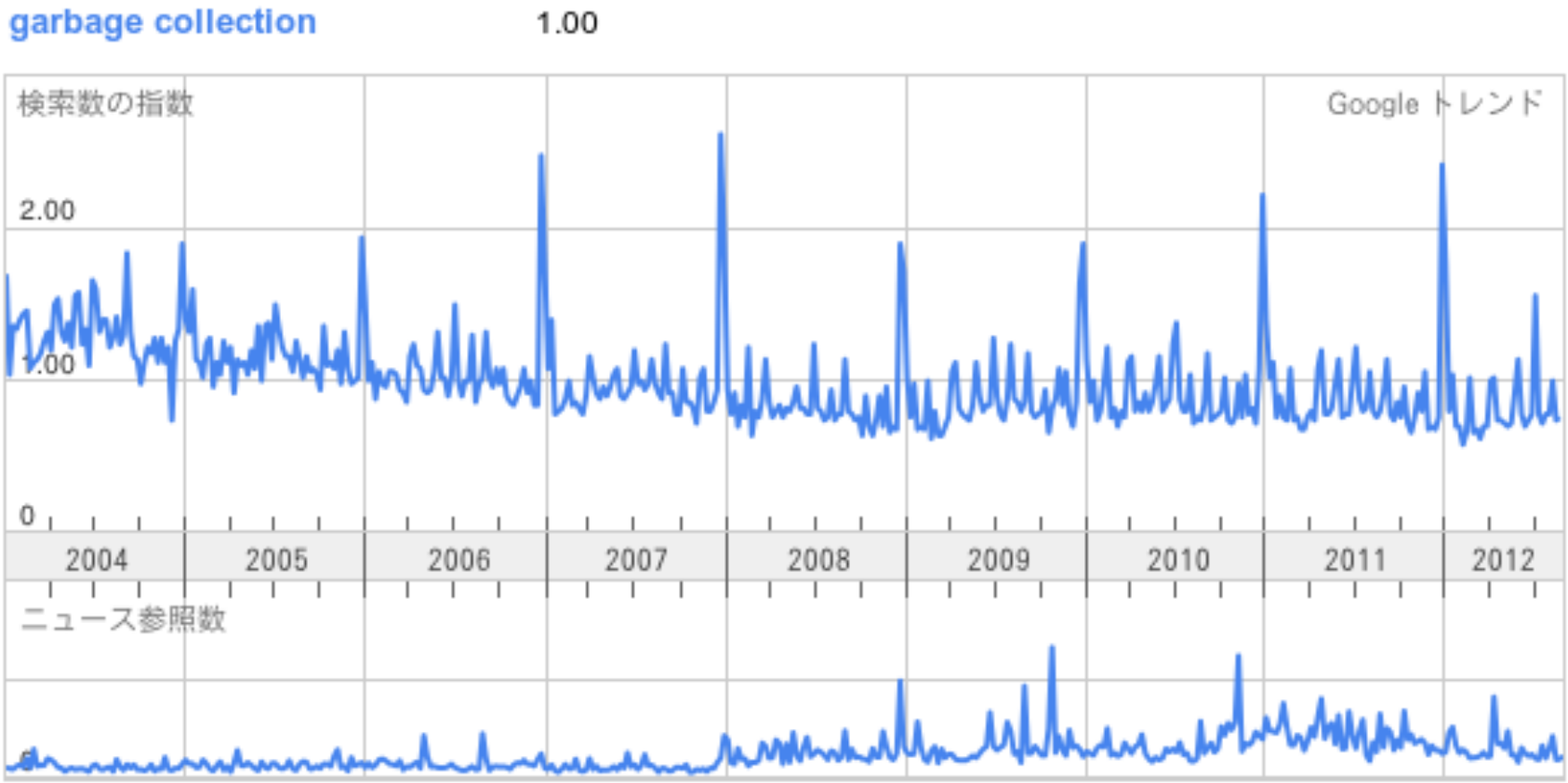
休日の検索数がさがっている

つまり

- ✓ 平日
  - ✓ お仕事でGCを作るリア充
  - ✓ もしくはGCバグに悩まされる一般ピープルが多い
- ✓ 休日
  - ✓ 趣味でGCをいじる人が少ない

GCはまだまだ愛されていない

ちなみに:年末も異常に検索されている



リアルGCの恐れ..

とにかく
GCをもっと
よく知りましょう！

徹底解剖G1GC
実装編
正式版公開！！
（無料配布）

徹底解剖
「G1GC」
実装編

この場を借りて本書のスポンサーのみなさまありがとうございます

m(_ _)m

Ruby2.0 の GC update

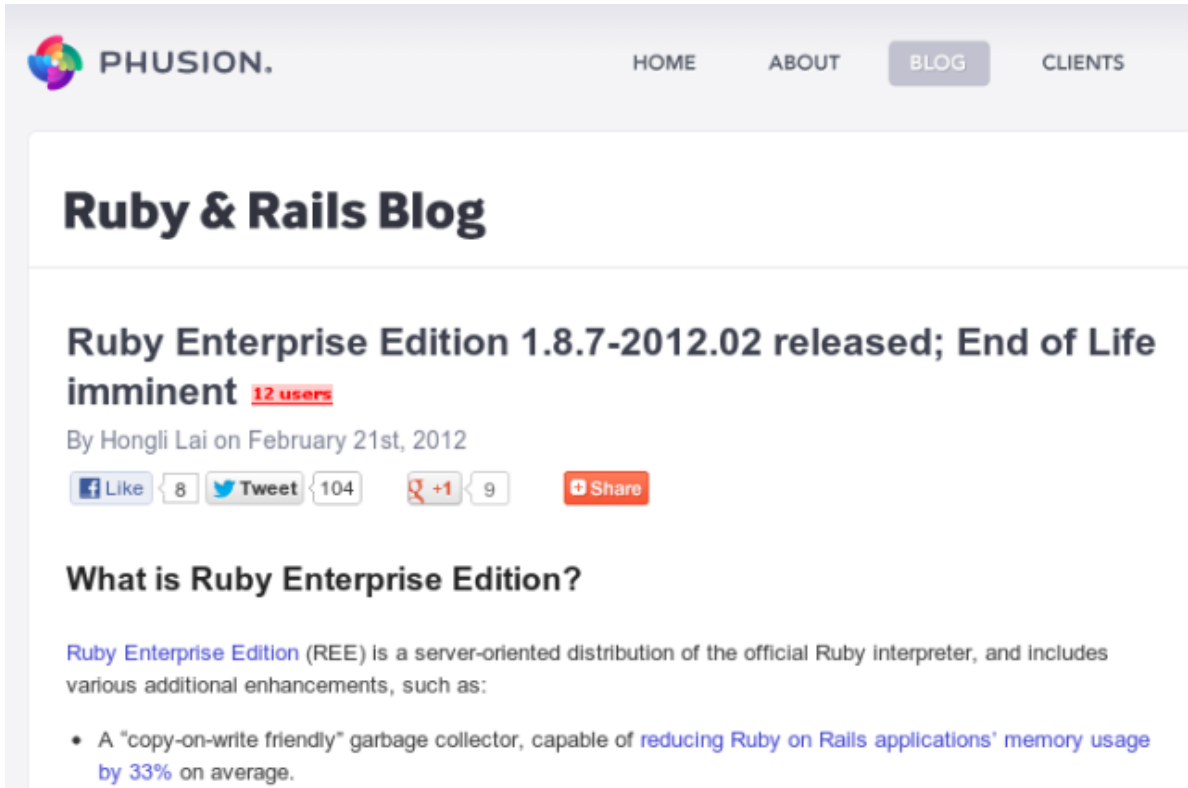
偉い人がいってた

http://www.flickr.com/photos/recompile_net/5952564080

Matz mashes something, so we mash something.

僕もなにか潰したい！

REE is dead



<URL:http://blog.phusion.nl/2012/02/21/ruby-enterprise-edition-1-8-7-2012-02-released-end-of-life-imminent/>

諸君らが愛した**REE**は死んだ。
なぜだ！

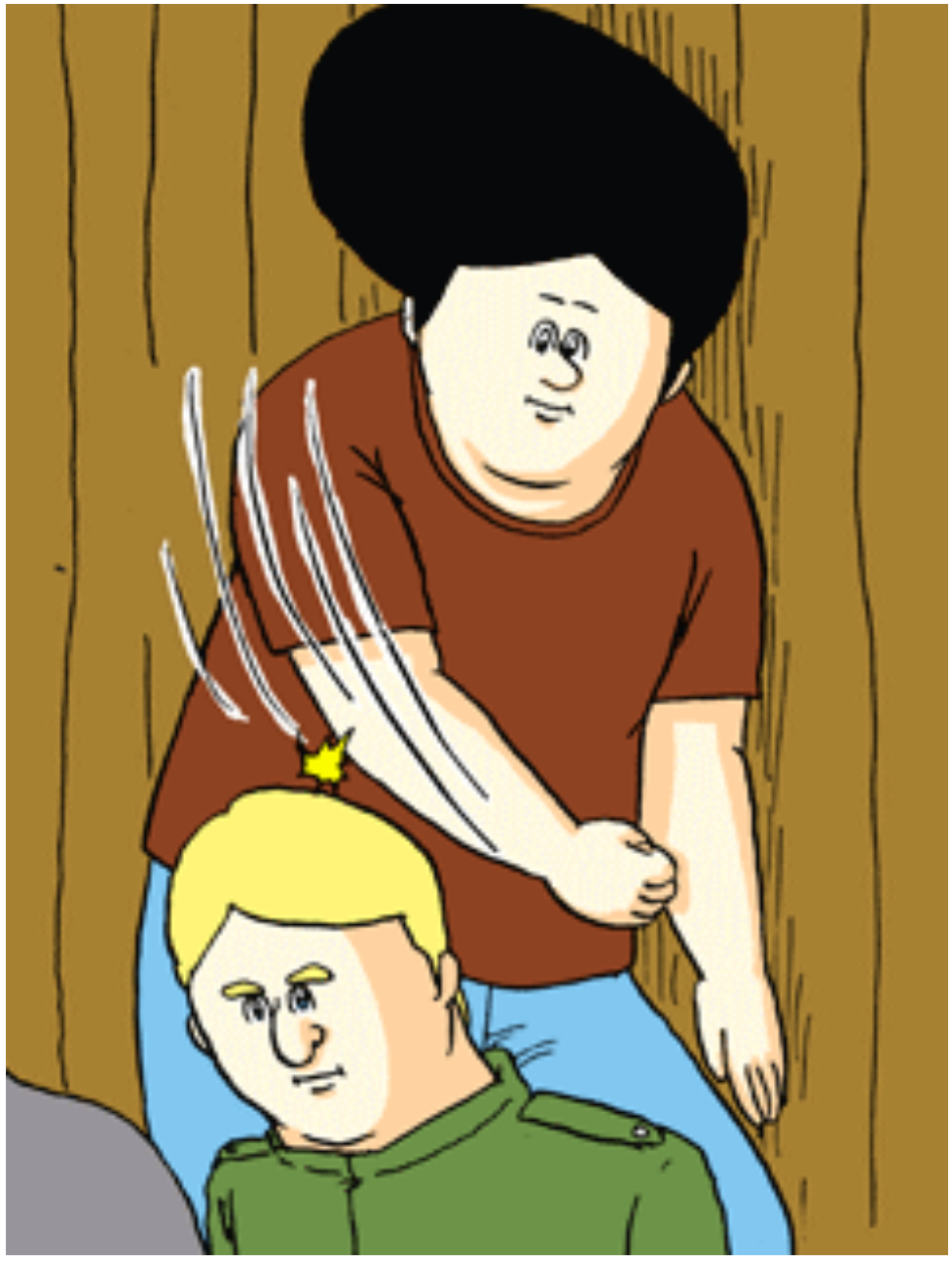
End of Life

Support for Ruby 1.8.x is slowly being dropped by the upstream Ruby core developers in favor of Ruby 1.9 and beyond. The Rails team has recently announced that they will be dropping Ruby 1.8 support in Rails 4. As such, we are also slowly End-of-Life'ing Ruby Enterprise Edition.

We have no plans to create a Ruby 1.9-based version of Ruby Enterprise Edition for the following reasons:

- A copy-on-write patch has recently been checked into Ruby 2.0.

- A copy-on-write patch has recently been checked into Ruby 2.0.



BitmapMarking

- ✓ fork使うようなプログラムで嬉しいはず
- ✓ Passengerなどで恩恵があるらしい

REEつぶしたった
(^o^)

最近の悩み:
Kijiは
どうやって潰そうか...

本題

Rubyで**GC**を書くというお話

動機(1)

RubyKaigi Driven Development

過去の**RubyKaigi**の発表...

- ✓ LazySweepGC - RubyKaigi2008
- ✓ LonglifeGC - 2009
- ✓ LazySweepGC - 2010
- ✓ ParallelMarkingGC - 2011

全部**C**の話やないか！

Rubyの話、させてください

動機(2)

卜部昌平のあまりreblogしないtumblr

どうも周知徹底が不足しているようなので再度のお願いとなりますが、C死ね。

- 確かにCでしか書けない類のプログラムは存在する(例を挙げるならKernel)が、それはCの存在を赦す理由にはならない。
- 確かにCに輪をかけてさらにダメな類のプログラミング言語は存在する(例を挙げるならC++)が、それはCの存在を赦す理由にはならない。
- 確かにCでしか書けないダメプログラマは存在する(例を挙げてほしければここにおまえの名前を入れろ)が、それはCの存在を赦す理由にはならない。

結論:C死ね。

> 結論:C死ね。

[cited from `卜部昌平のあまりreblogしないtumblr']

ほんとどの**GC**は
C or C++で書かれているしなぁ...

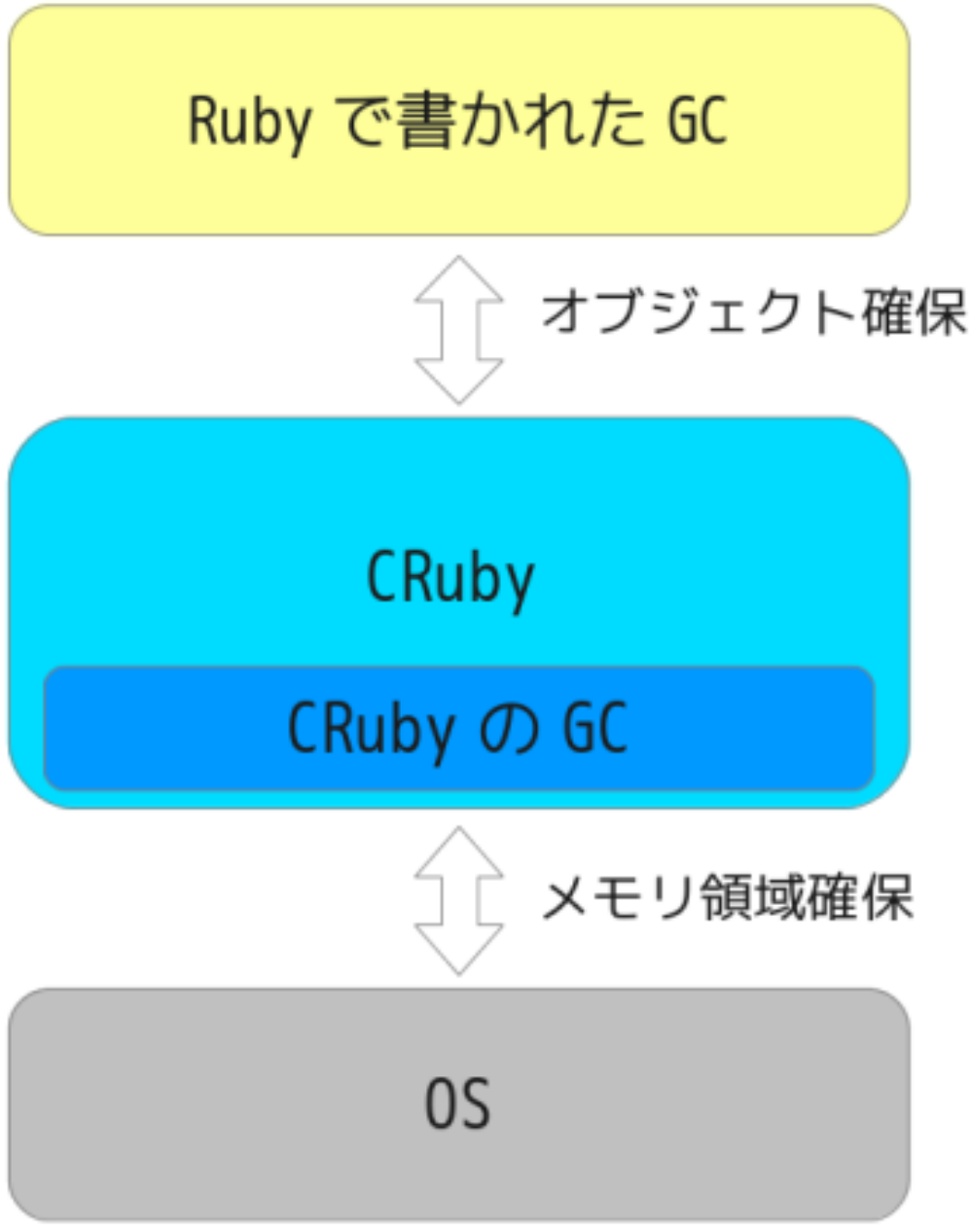
> なぜCで書く必要がある？ないはずだよ。ちゃんと調査すればJavaとかC#とか、ひょっとしたらOCamlやScalaでも用は足りる場合がほとんどなんだ。

[cited from `卜部昌平のあまりreblogしないtumblr']


お前を試した

RubyでGC
書いてみよう！

素直に考えると...


Ruby で書かれた GC
オブジェクト確保
CRuby
CRuby の GC
メモリ領域確保
OS


マジ飽きたわー
ほんと飽きた
あれまだやってる奴いるの？
ほんと飽きたわー！
俺が一番先に飽きたわー

これはあまり意味がない

- 下にいるGCの性能に影響をうける
  - 性能評価しづらい
  - 下のGCが遅いと、上のGCも遅くなる

これはあまり意味がない

- 言語処理系にGCは1つあればいいですよ感
  - 無駄に話が複雑になっているような気がする

もっと違うアプローチを
考えよう！

Meta-circular evaluator

> *An evaluator that is written in the same language that it evaluates is said to be* ***metacircular***.

[cited from `4.1 The Metacircular Evaluator - SICP']

簡単な例:
LISPでLISP書いたった

他の具体例を調べてみる

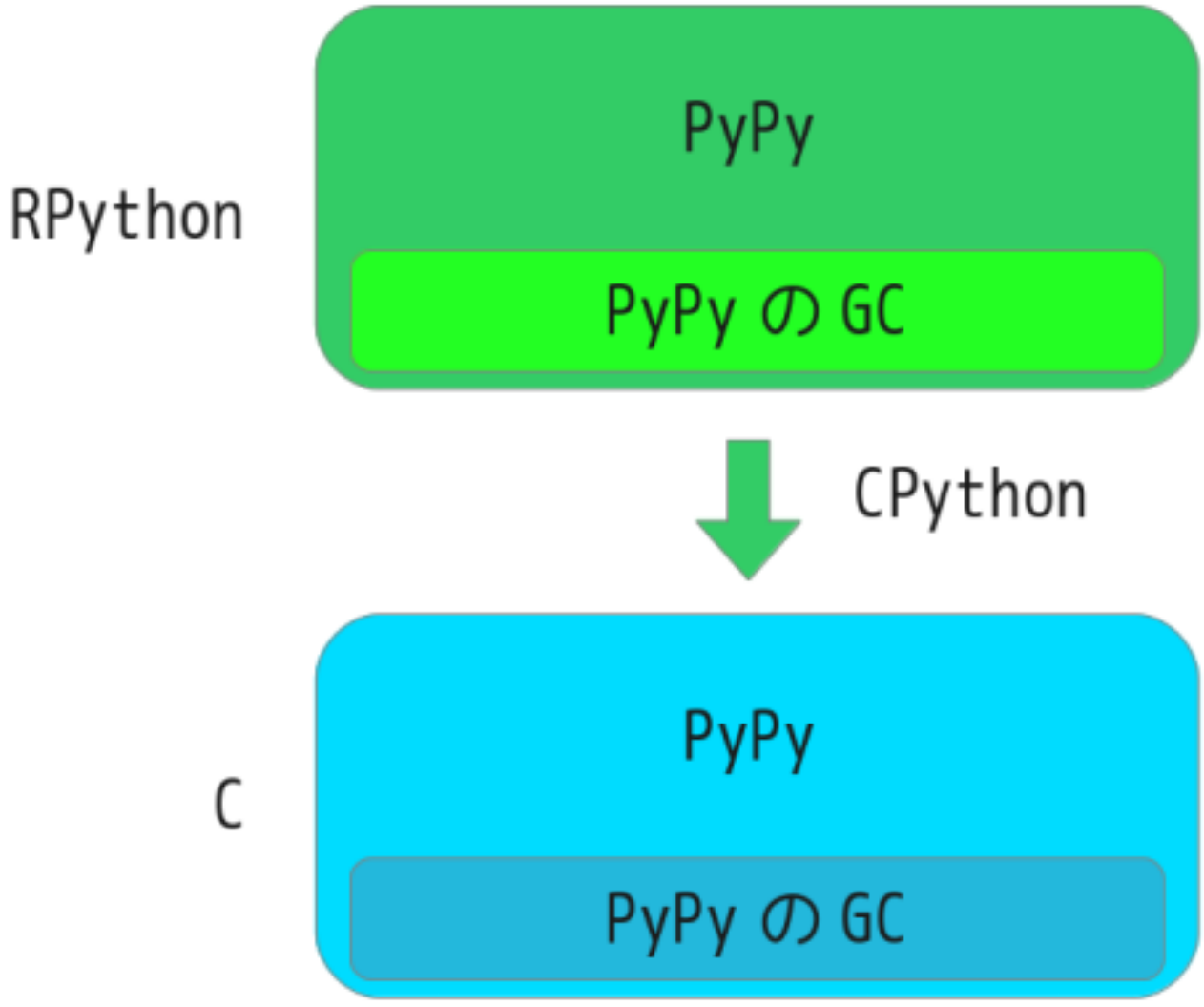
pypy

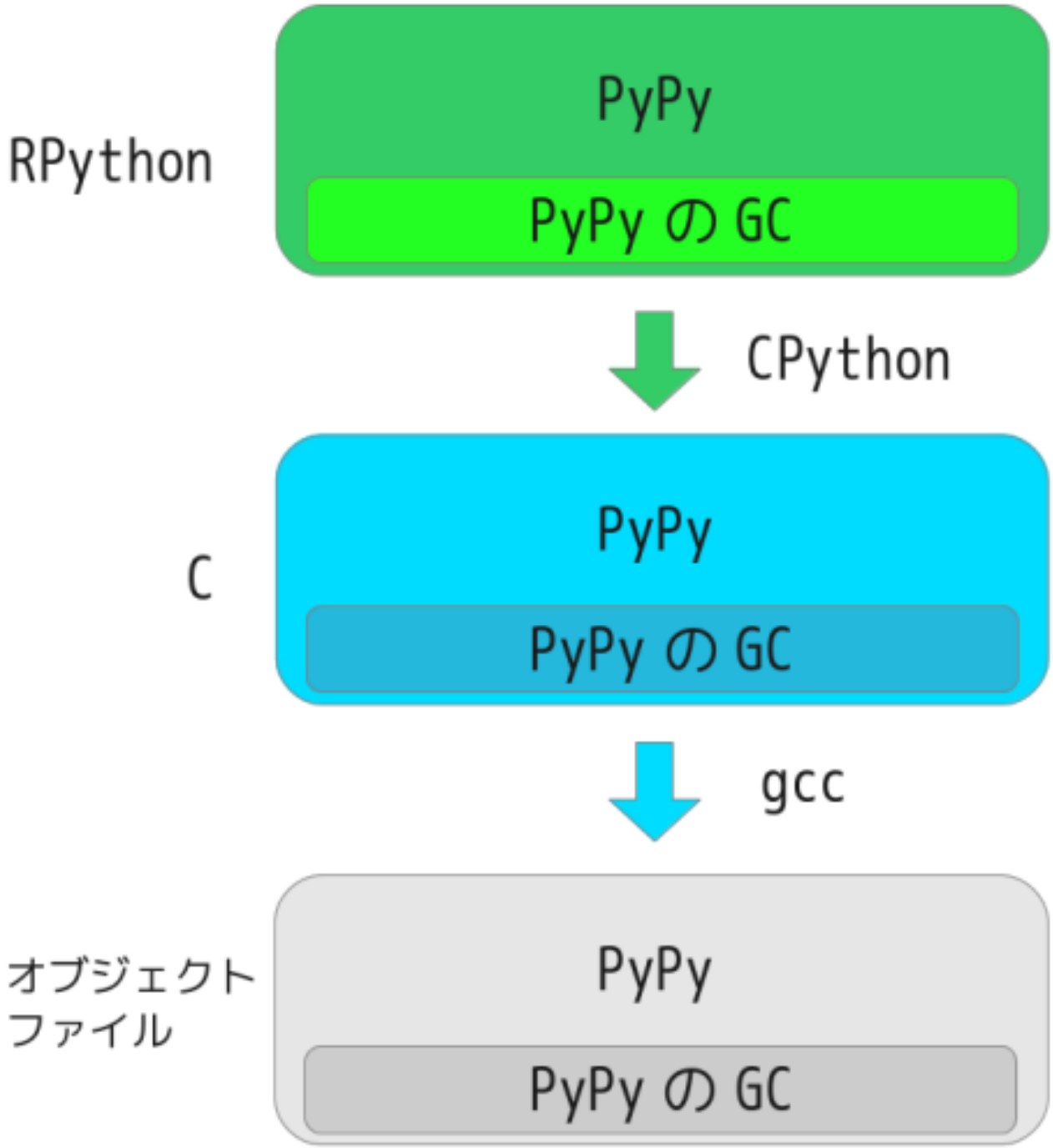
もう少し詳しく

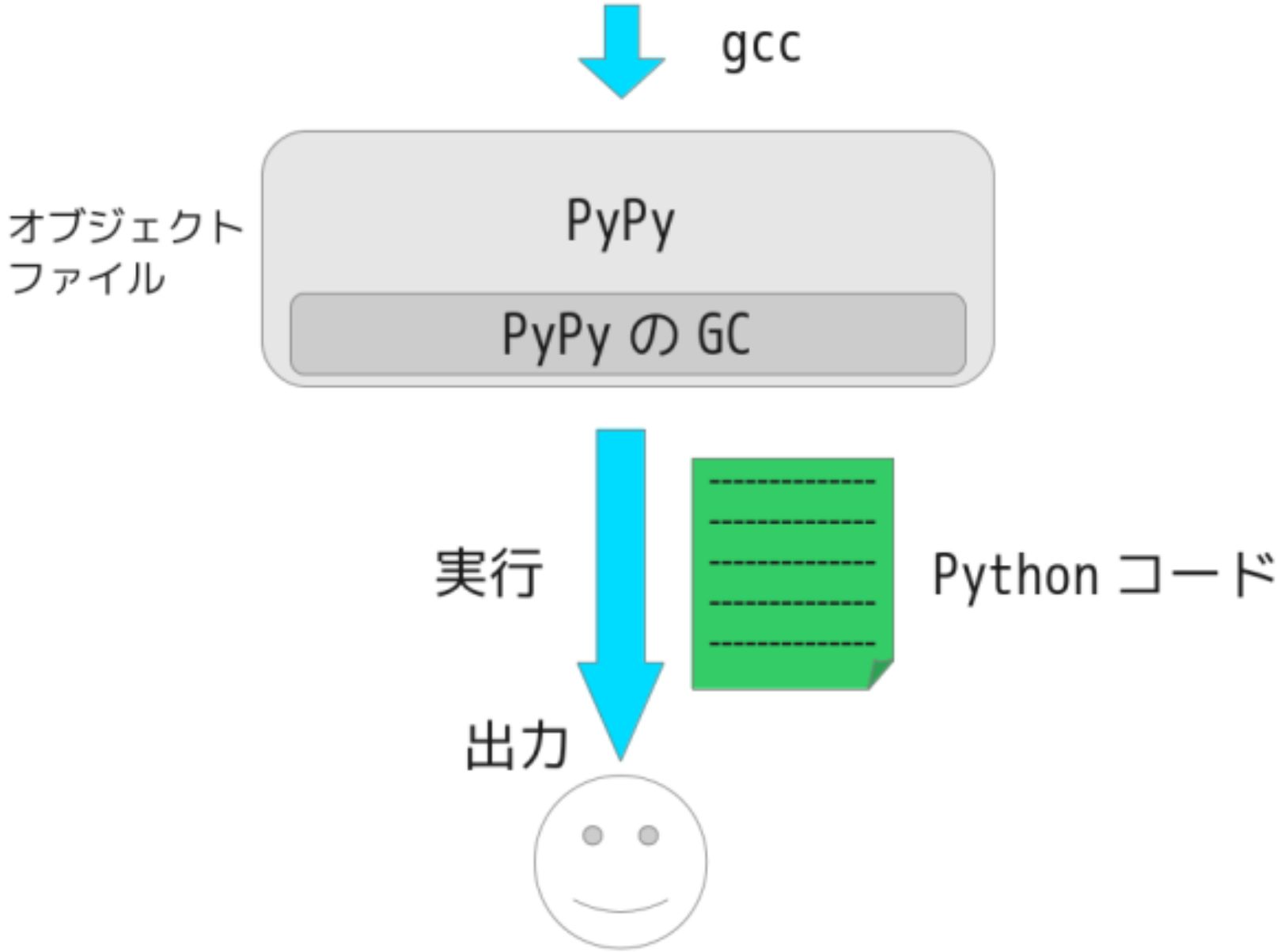
- ✓ RPython(Restricted Python)で実装
  - ✓ Pythonの言語サブセット
- ✓ RPython -> C/LLVM/Java..

変換イメージ

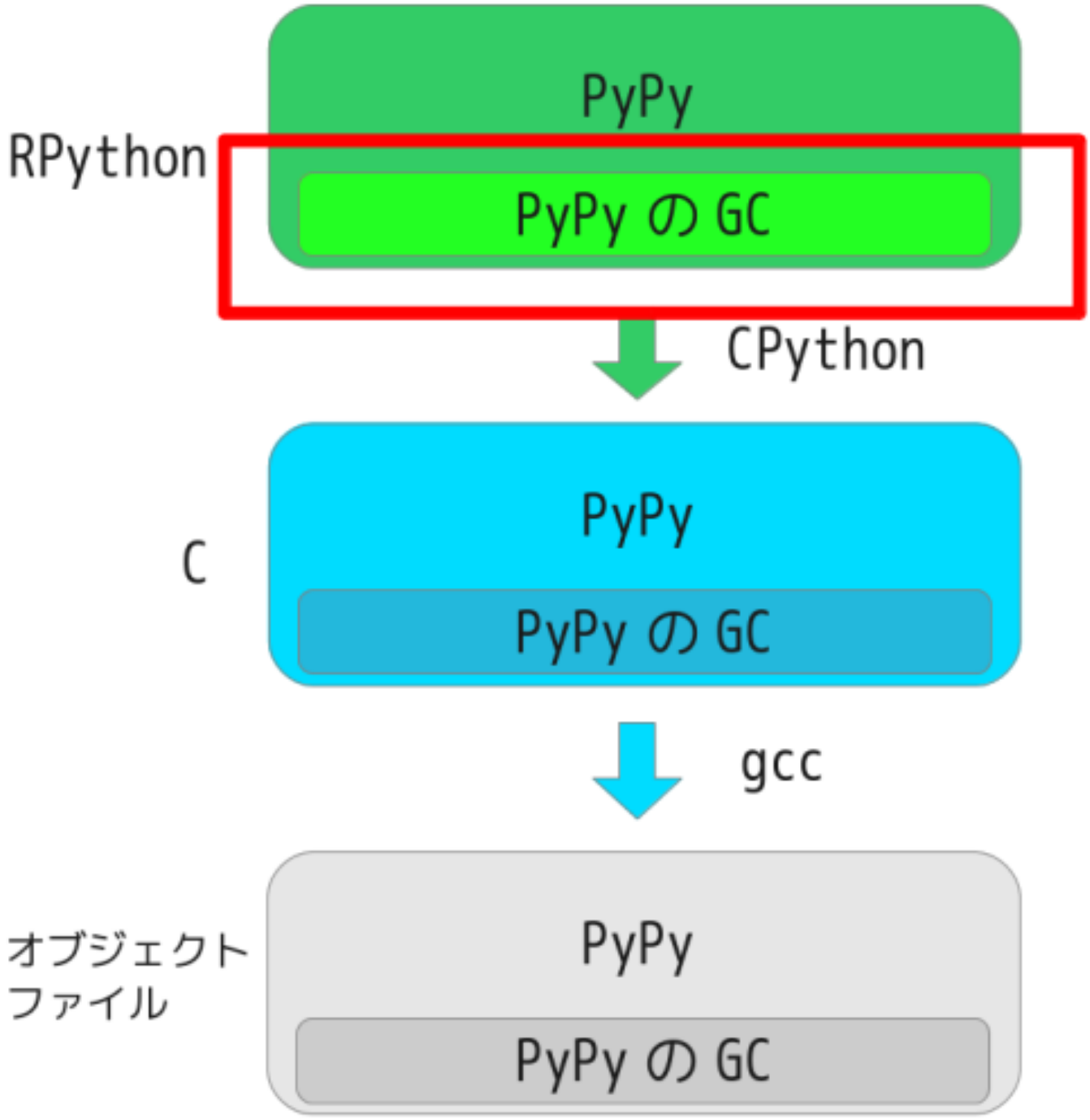
RPython

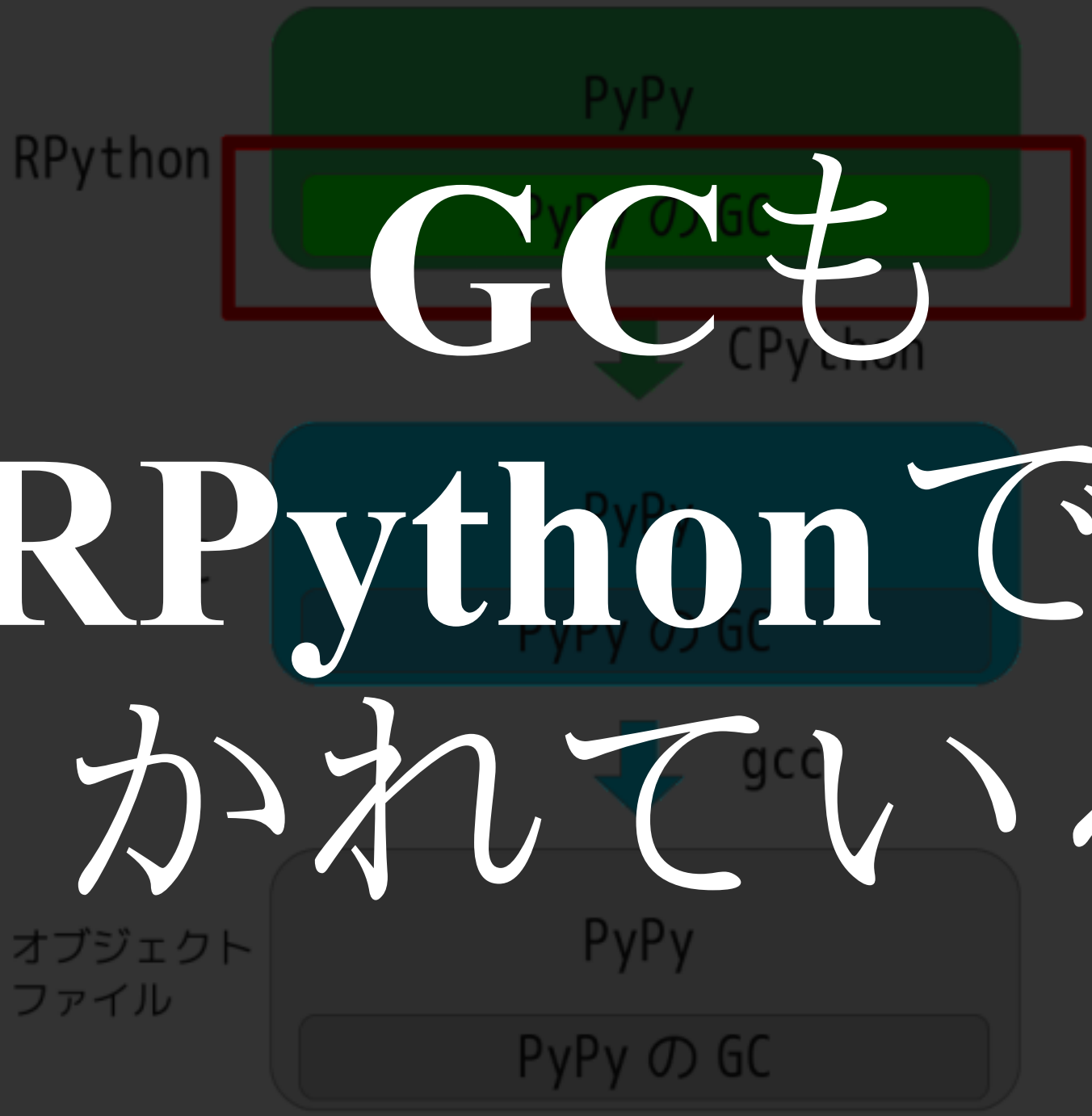

RPython
PyPy
PyPy の GC
CPython
C
PyPy
PyPy の GC


RPython
PyPy
PyPy の GC
CPython
C
PyPy
PyPy の GC
gcc
オブジェクト
ファイル
PyPy
PyPy の GC


gcc
オブジェクト
ファイル
PyPy
PyPy の GC
実行
Python コード
出力

重要なポイント


RPython
PyPy
PyPy の GC
CPython
C
PyPy
PyPy の GC
gcc
オブジェクト
ファイル
PyPy
PyPy の GC



GCも RPythonで書かれている

疑問

GC動作中のGCはどうするのか？

- GC動作中にGCが起きて、その動作中にGCが起きて…
  - GCの無限ループ...?

RPython

- GC中はGC対象のオブジェクトを作らないように気を付ける（たぶん）
  - mallocなどで直接メモリを切り出す
    - libffiを利用

GC中に**GC**が走ることはない


Rubinius
Use Ruby™

仕組み（ｒｙ

ただし

- GCをRubyで書く仕組みがない
  - 今考えればRubiniusをいじったほうがよかったかも…？
    - まぁいいか...


Jikes RVM

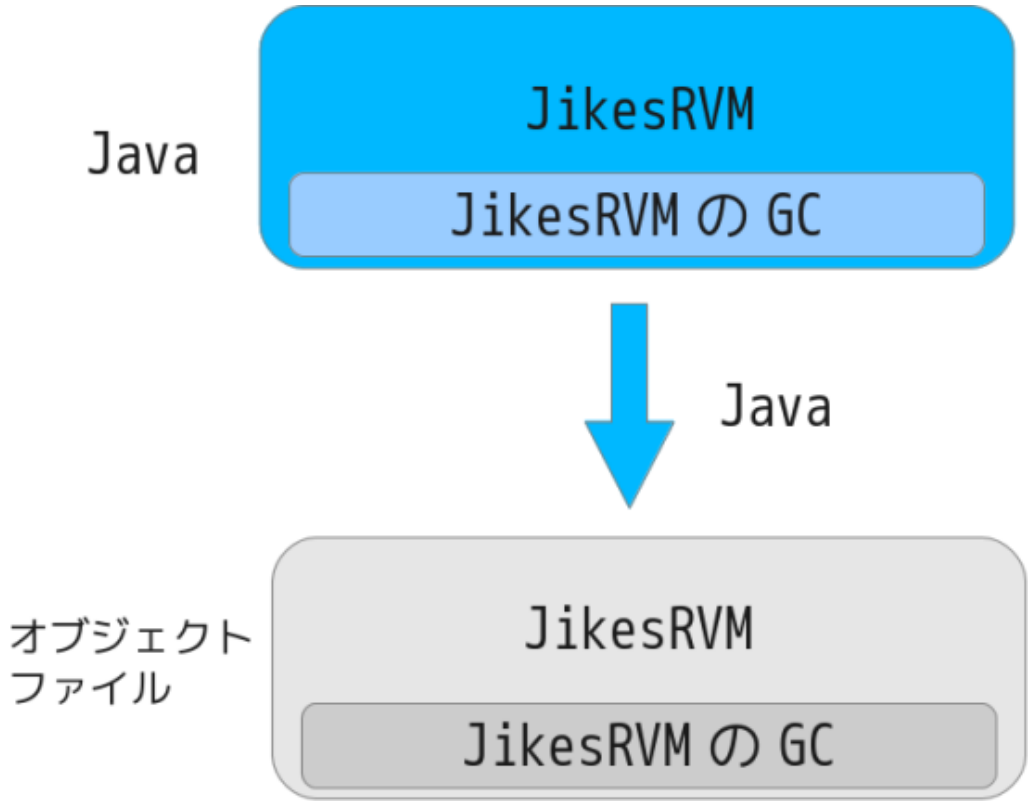
JikesRVM

- RVM(Research Virtual Machine) = 研究用VM
  - GC,VM周りでたくさん論文が書かれている

JikesRVM

- ✓ GCもJavaで書かれている
- ✓ プロジェクト委員の一人が Rechard Jones（RJGC著者）

直接オブジェクトファイルを吐く



GCもJavaで書ける

- ✓ 一緒にオブジェクトファイルに変換される
- ✓ GC中はGCが発生しない
  - ✓ PyPyと同じ理由

JikesRVMの功績は
「**GC**を**Java**でかける」
だけではない

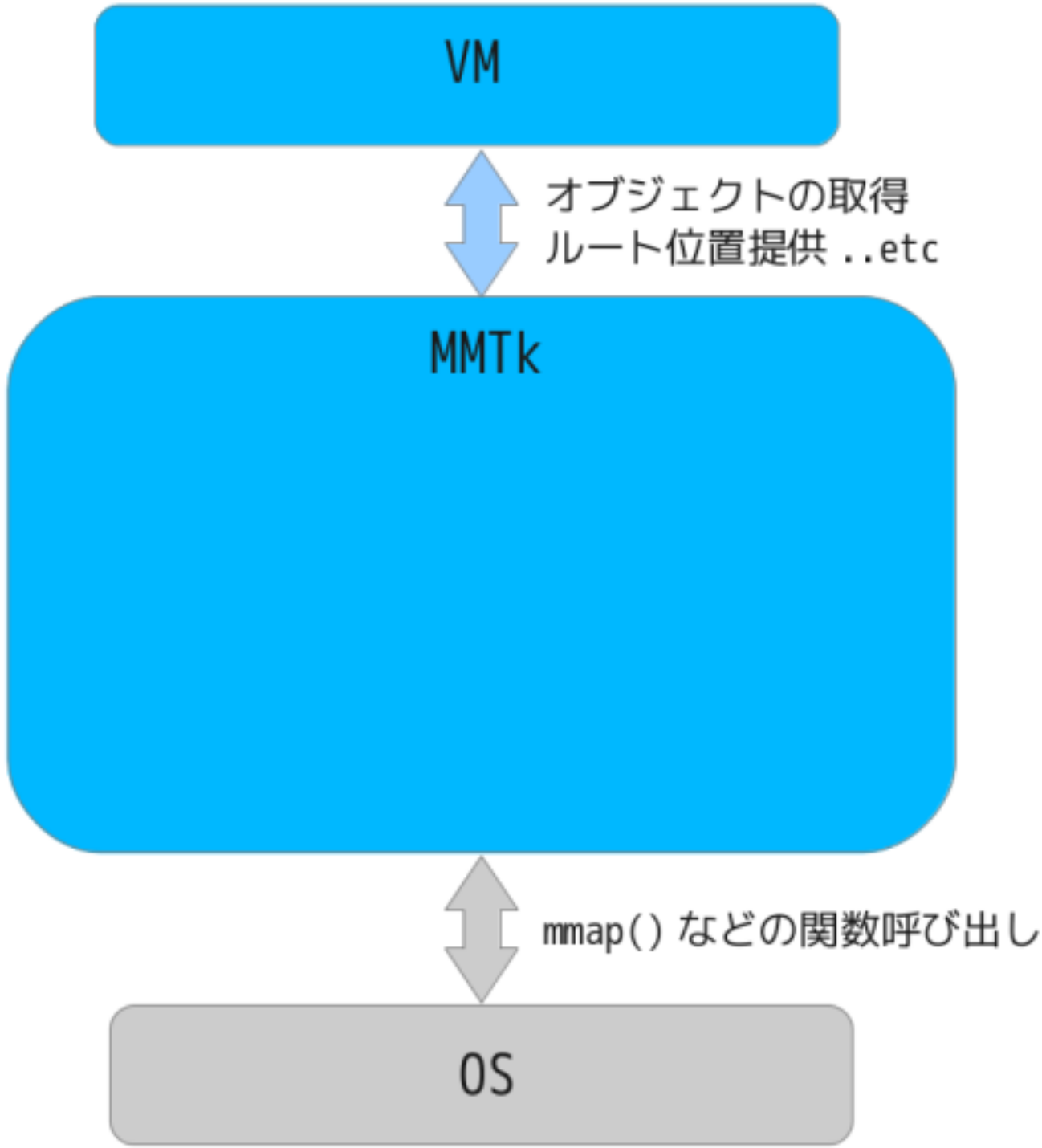
GC部分が**MMTk**という別部品に切り離されている

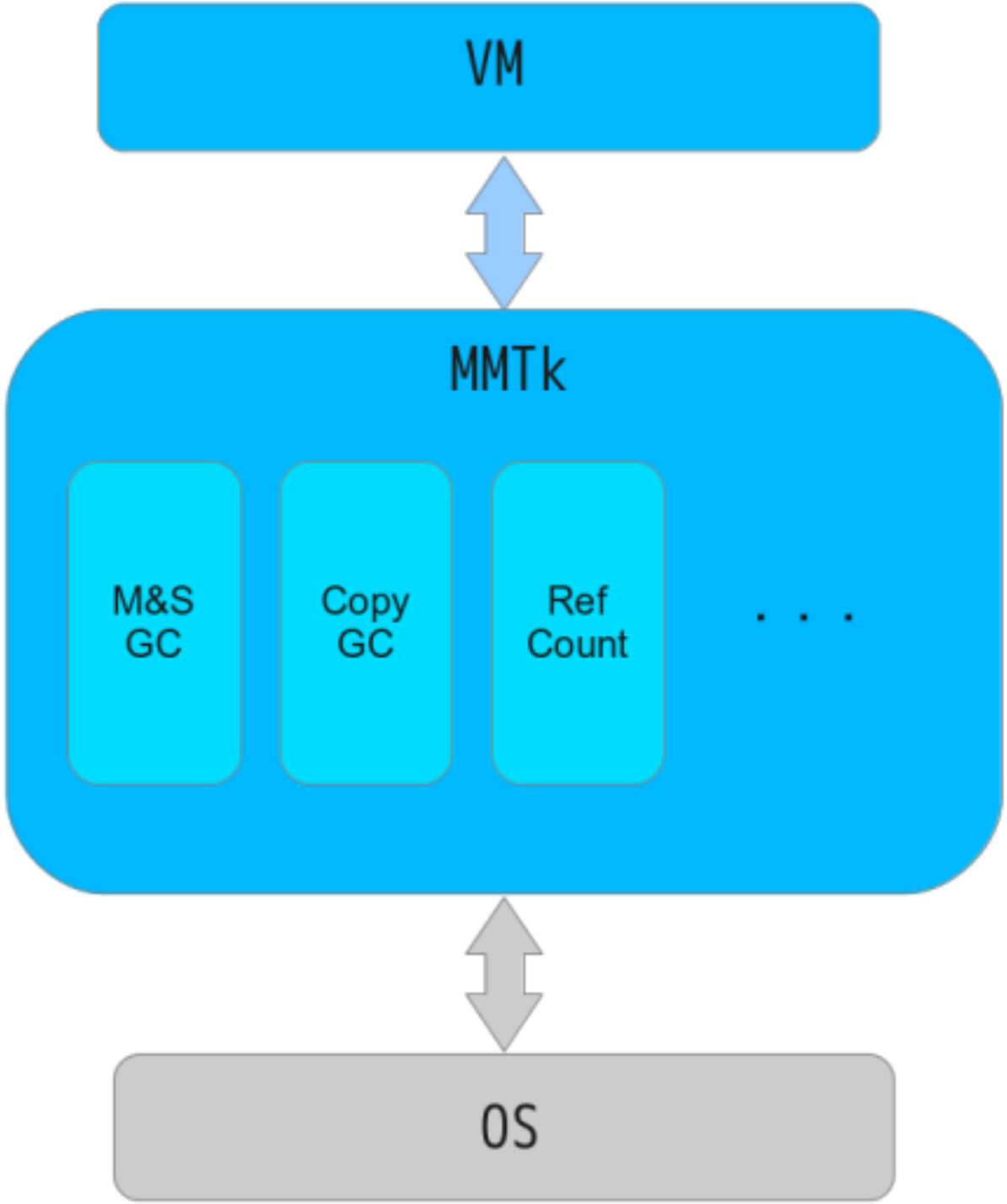
MMTk(Memory Management Tool kit)

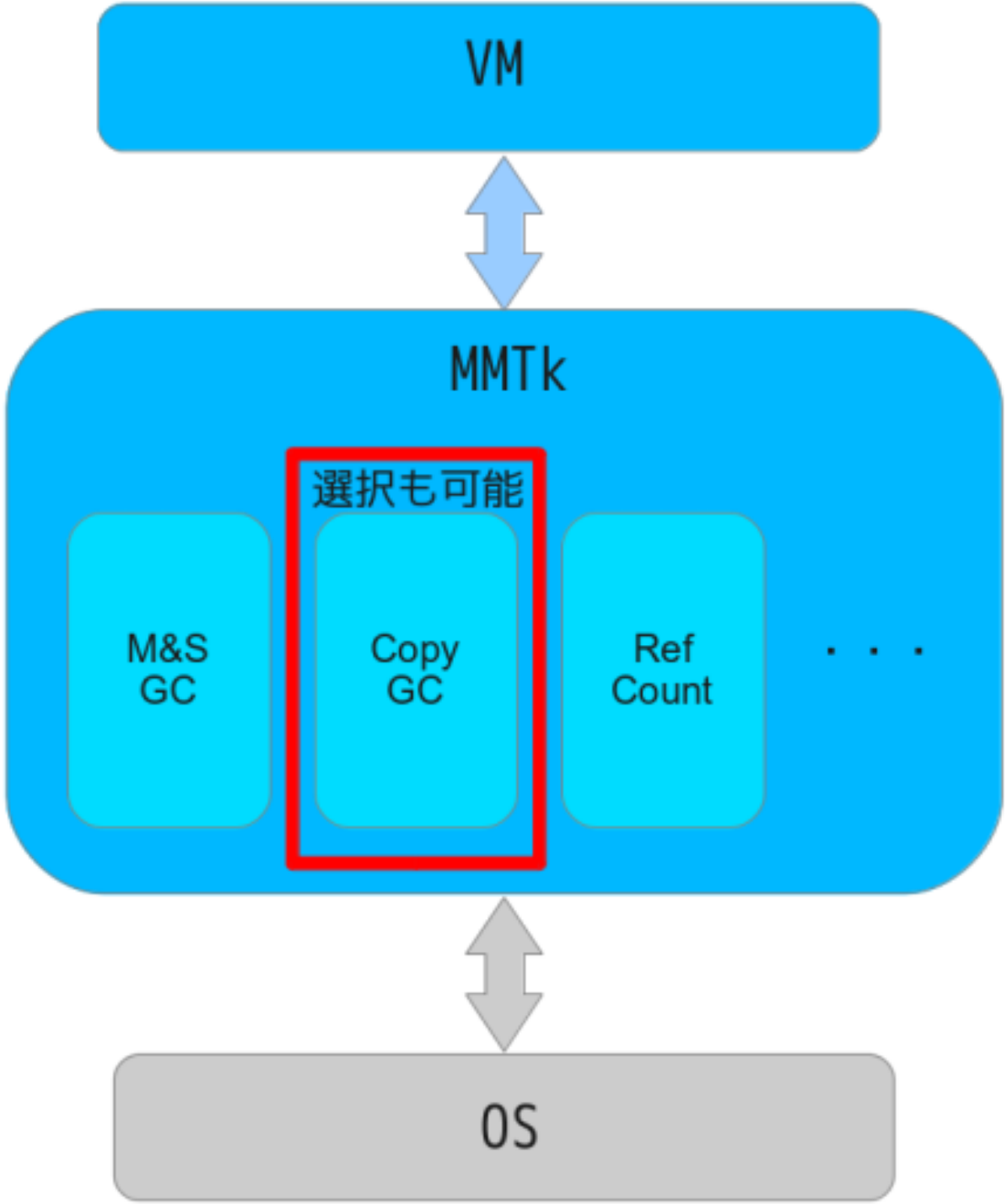
- ✓ GCを11個保有
- ✓ ツールキットなのでGC用の便利メソッドが沢山ある
  - ✓ GCづくりが簡単

MMTk(Memory Management Tool kit)

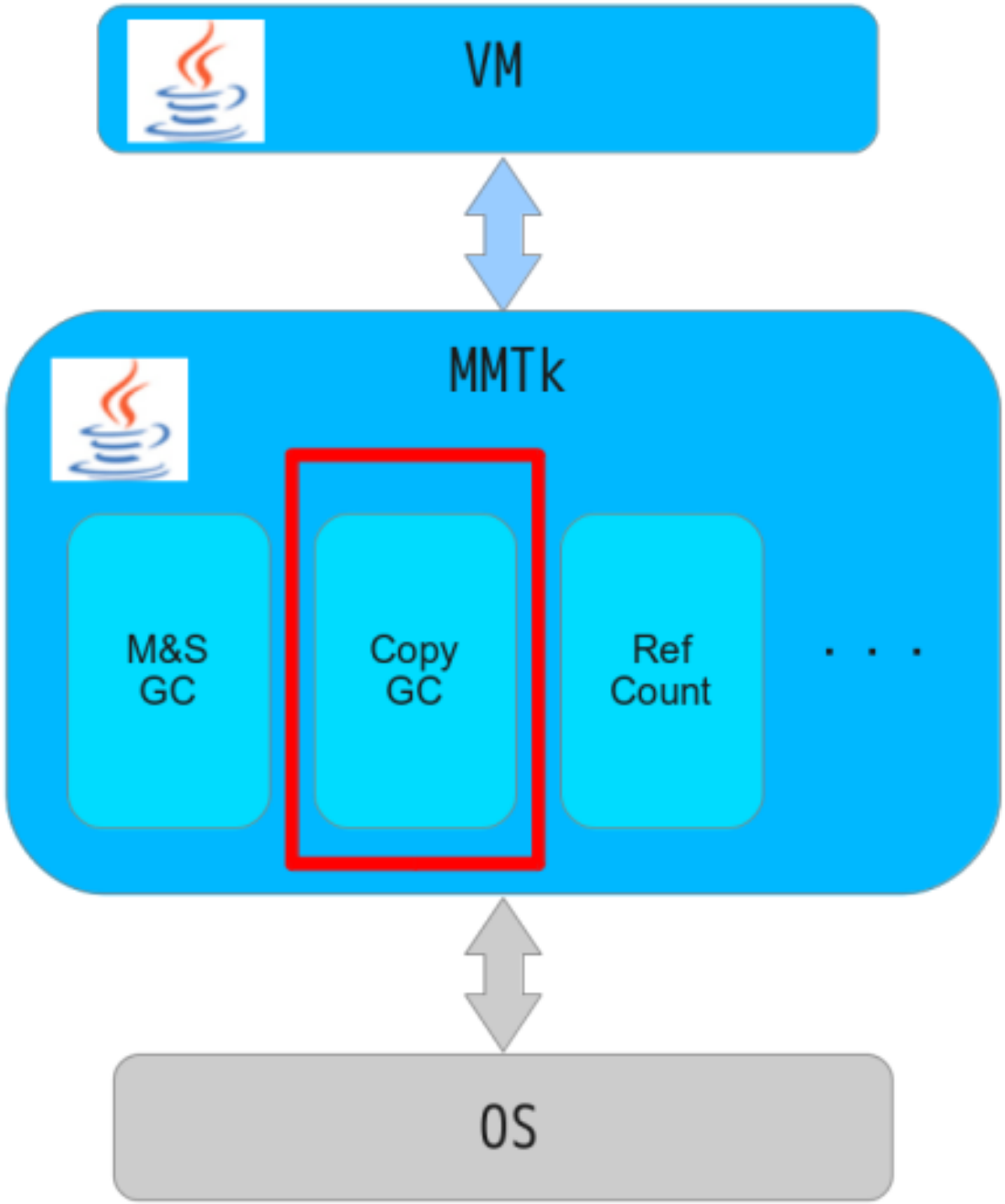
- VMからメモリ管理を別の部品として切り離せる
  - VMは取り替え可能
    - JikesRVM
    - Harness


VM
オブジェクトの取得
ルート位置提供 ..etc
MMTk
mmap() などの関数呼び出し
OS


VM
MMTk
M&S GC
Copy GC
Ref Count
. . .
OS

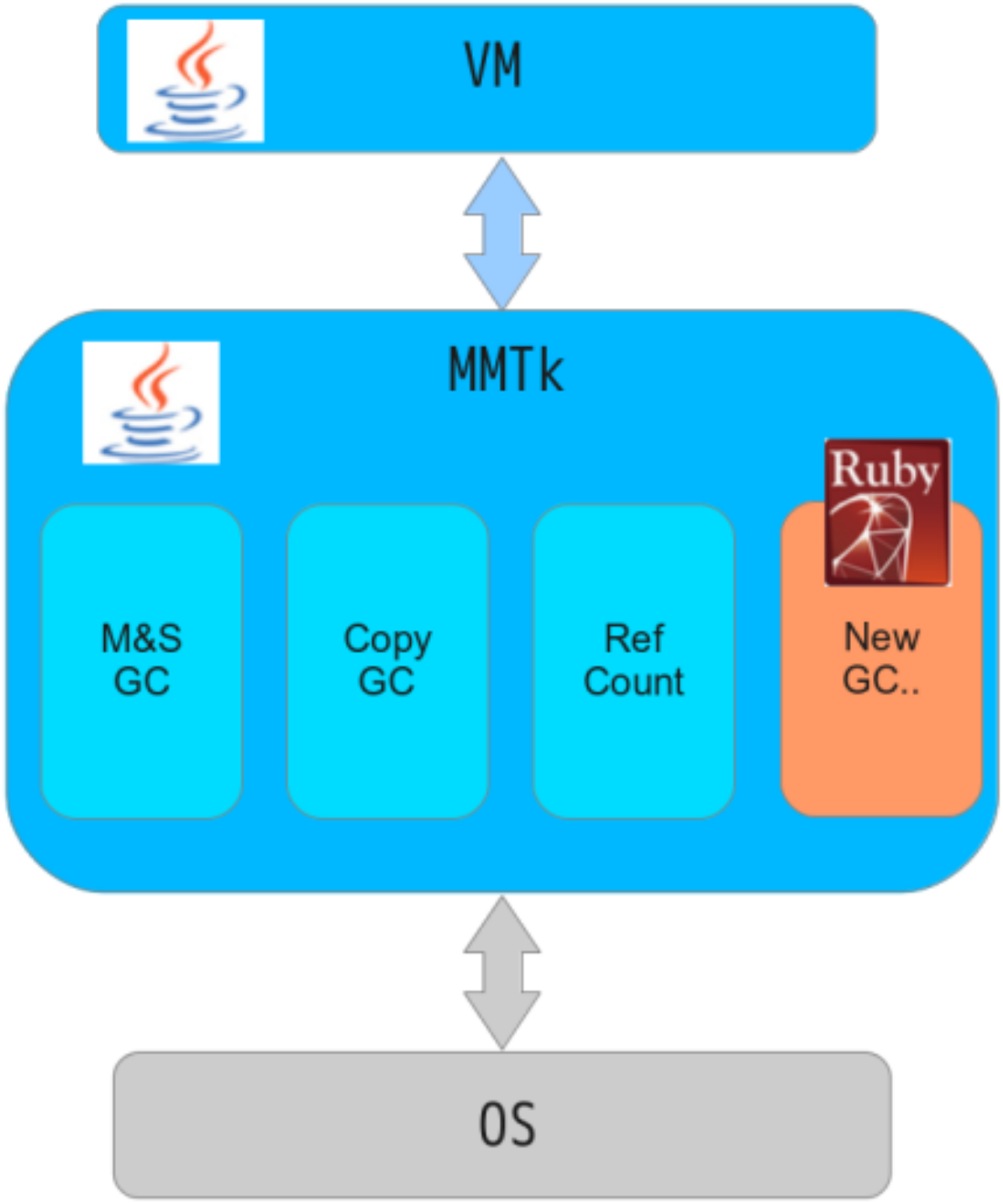

VM
MMTk
選択も可能
M&S
GC
Copy
GC
Ref
Count
・・・
OS

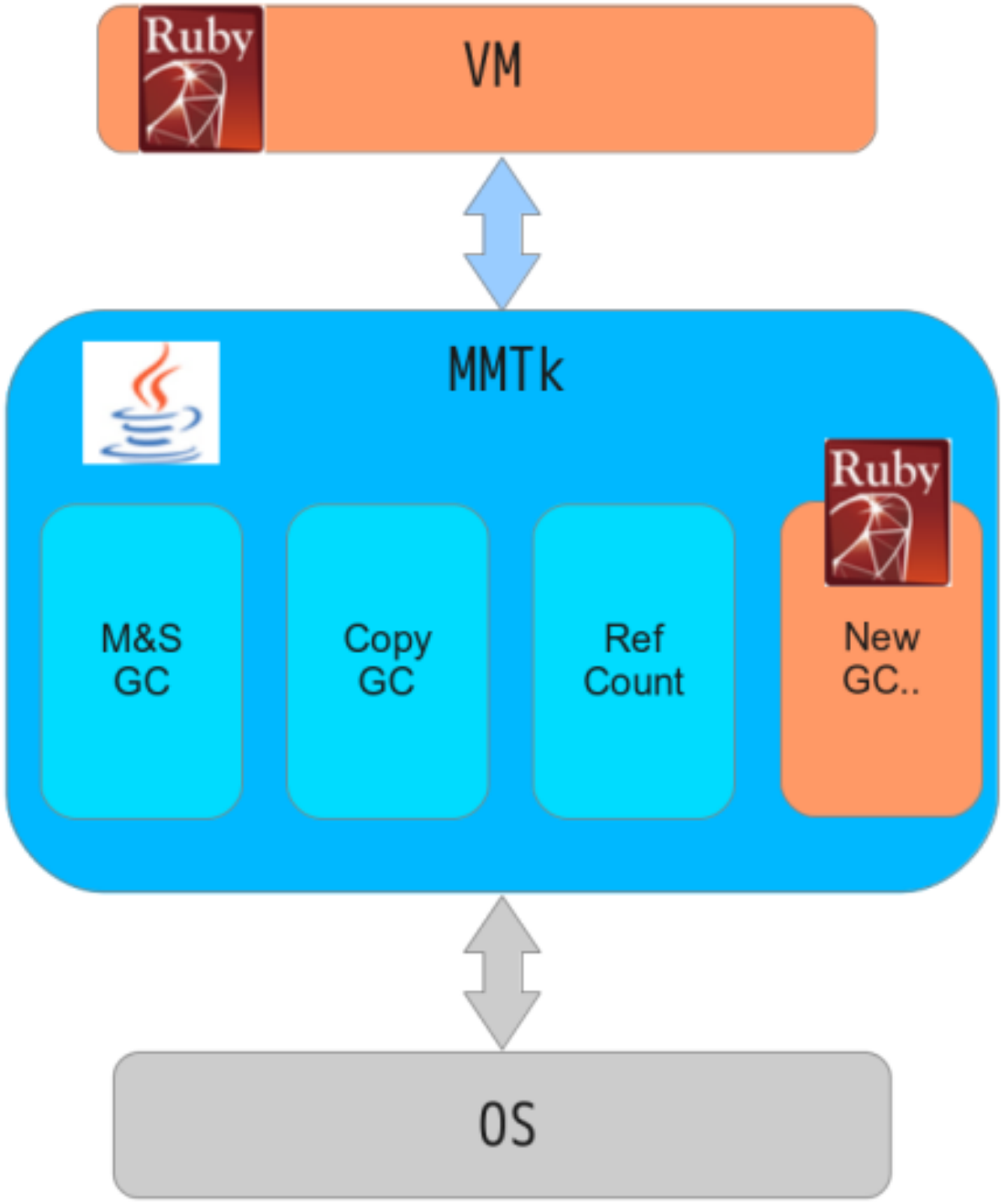
やたー！
GC書き放題！


VM
MMTk
M&S
GC
Copy
GC
Ref
Count
. . .
OS

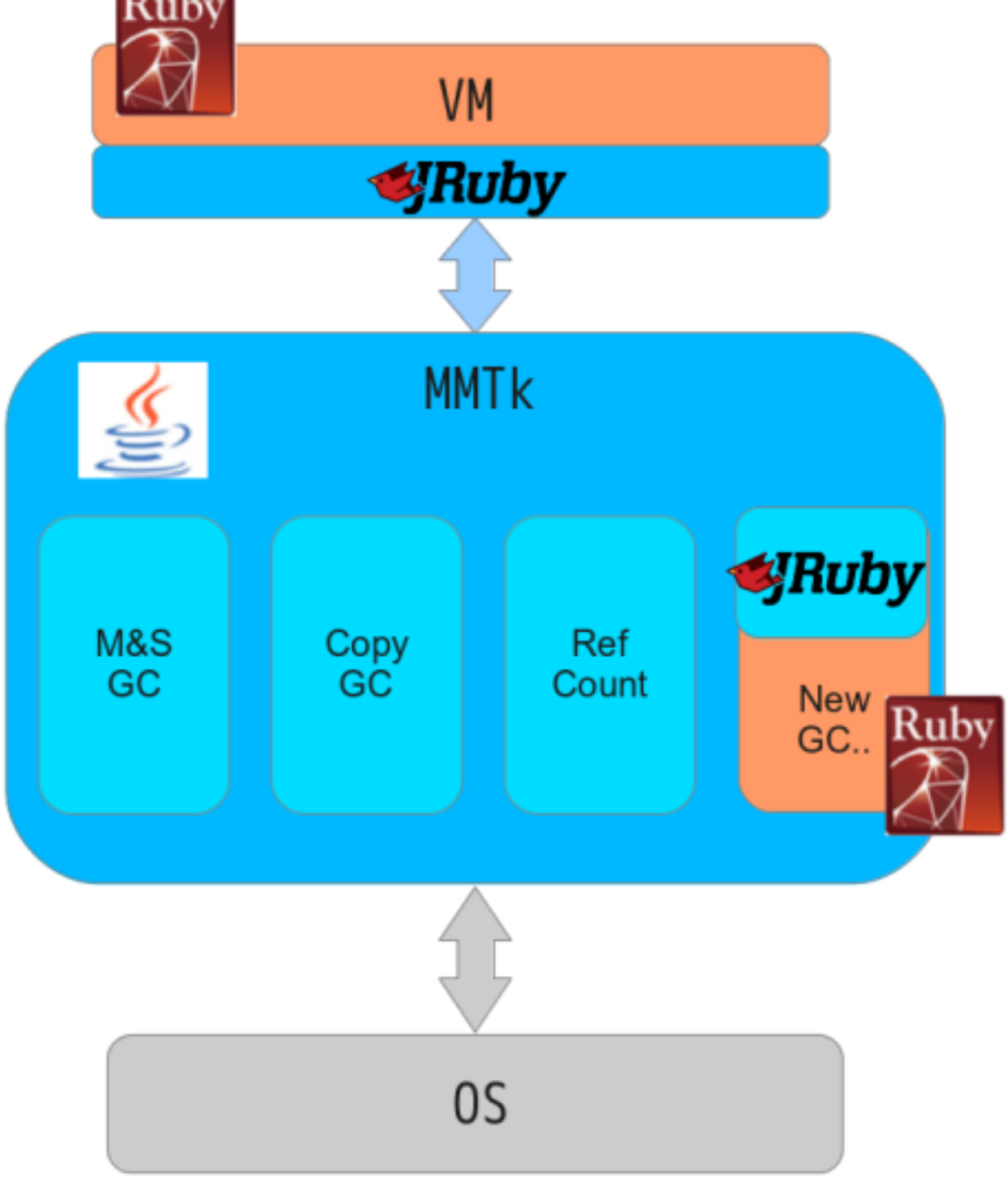
Javaで…

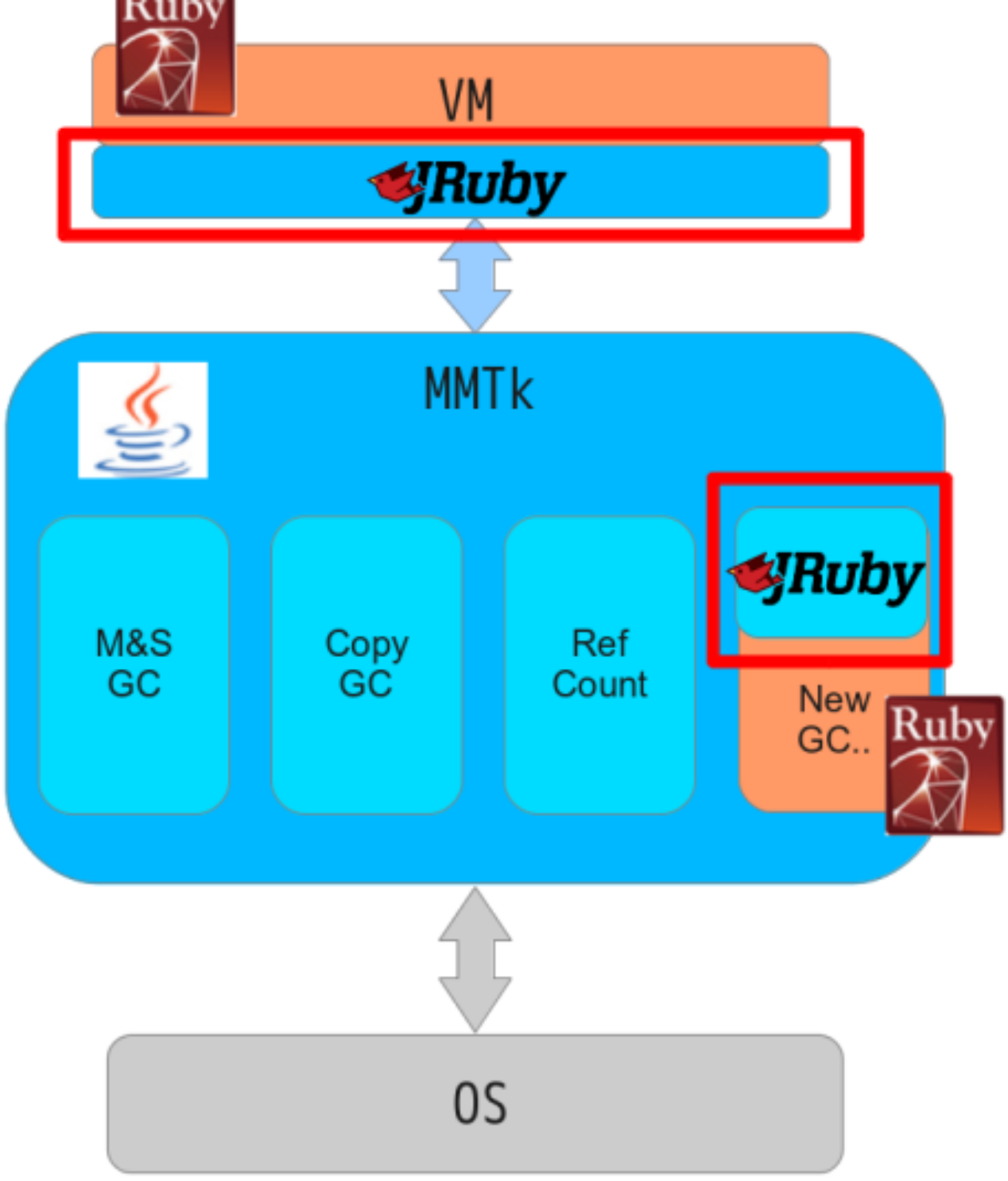
Rubyで書きたい…ッ！！


VM
MMTk
Ruby
M&S GC
Copy GC
Ref Count
New GC..
OS


Ruby
VM
MMTk
M&S
GC
Copy
GC
Ref
Count
Ruby
New
GC..
OS

われわれには**JRuby**があるじゃないか

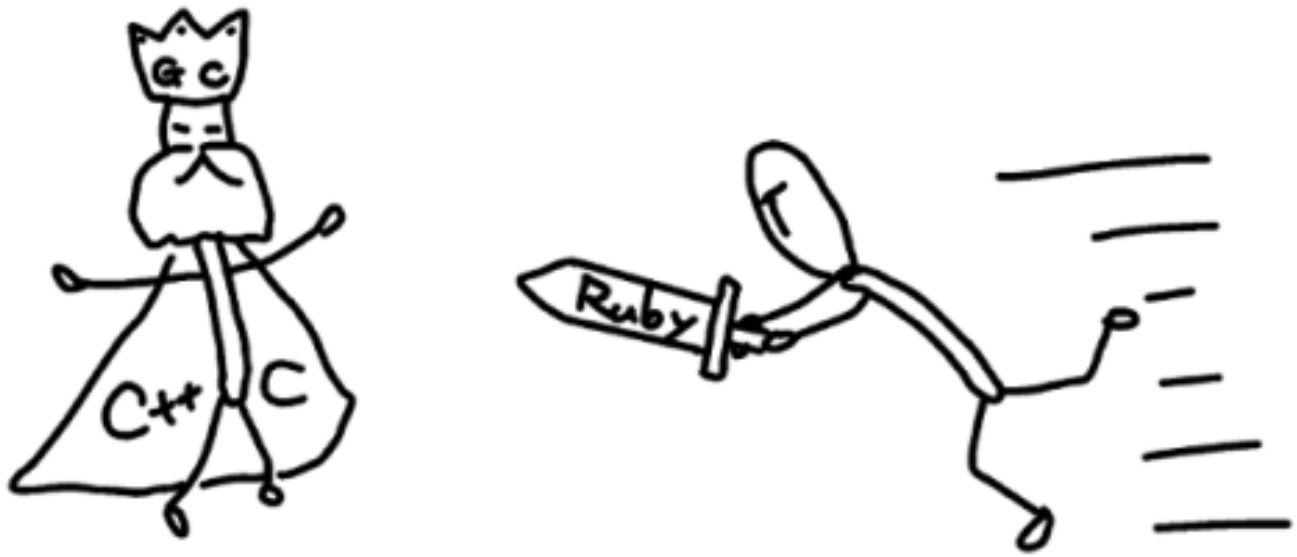

VM
JRuby
MMTk
M&S GC
Copy GC
Ref Count
JRuby
New GC..
Ruby
OS

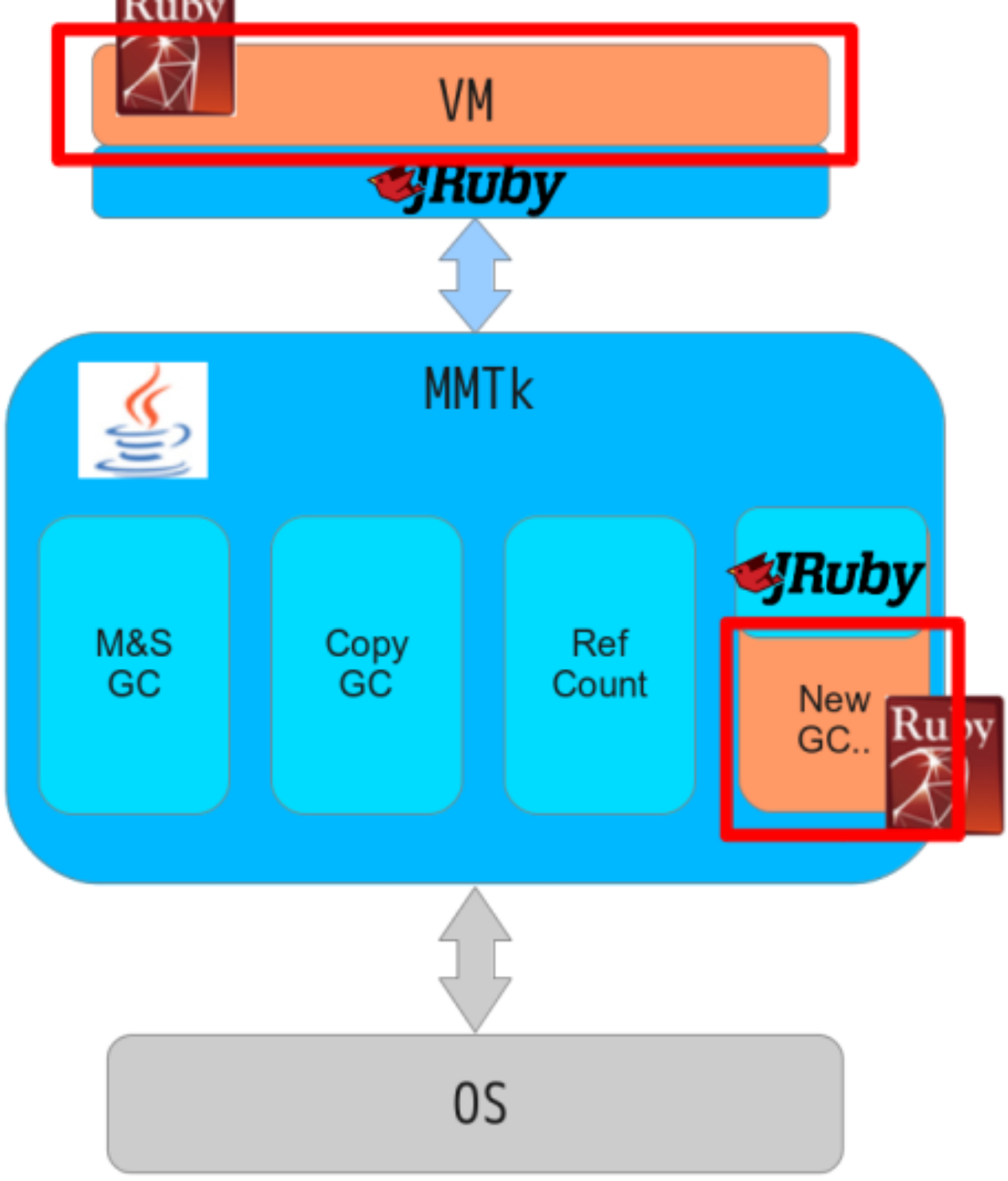

VM
JRuby
MMTk
M&S GC
Copy GC
Ref Count
JRuby
New GC..
Ruby
OS

この部分を
ライブラリ化

Regicide

Regicde = 国王殺し




VM
JRuby
MMTk
M&S GC
Copy GC
Ref Count
JRuby
New GC..
OS

サンプルコード

オブジェクトの定義

```
class FixnumValue < Regicide::Mutator::ObjectValue
  # ...
end
```

Regicide::Mutator::ObjectValueを継承

オブジェクト生成

```
class FixnumValue < Regicide::Mutator::ObjectValue
  # Fixnumをフィールドに持つ
  # オブジェクトを割り当てるメソッド
  def self.from_i(mutator, i)
    ...
  end
end
```

from_i

```
# メモリ割り当て
ptr = mutator.alloc(0, 1, mutator.current_stack.pc)
# オブジェクト生成
v = self.new(ptr)
# Fixnumを格納
mutator.store_data_field(v.object_value, 0,
  org.vmmagic.unboxed.Word.from_long(i))
return v
```

M&S GCは4行で書ける

```
class MSConstraints < org.mmtk.plan.marksweep.MSConstraints; end
class MS < org.mmtk.plan.marksweep.MS; end
class MSCollector < org.mmtk.plan.marksweep.MSCollector; end
class MSMutator < org.mmtk.plan.marksweep.MSMutator; end
```

動作するサンプル

↓

<URL:https://github.com/authorNari/regicide/tree/master/sample>

苦労話

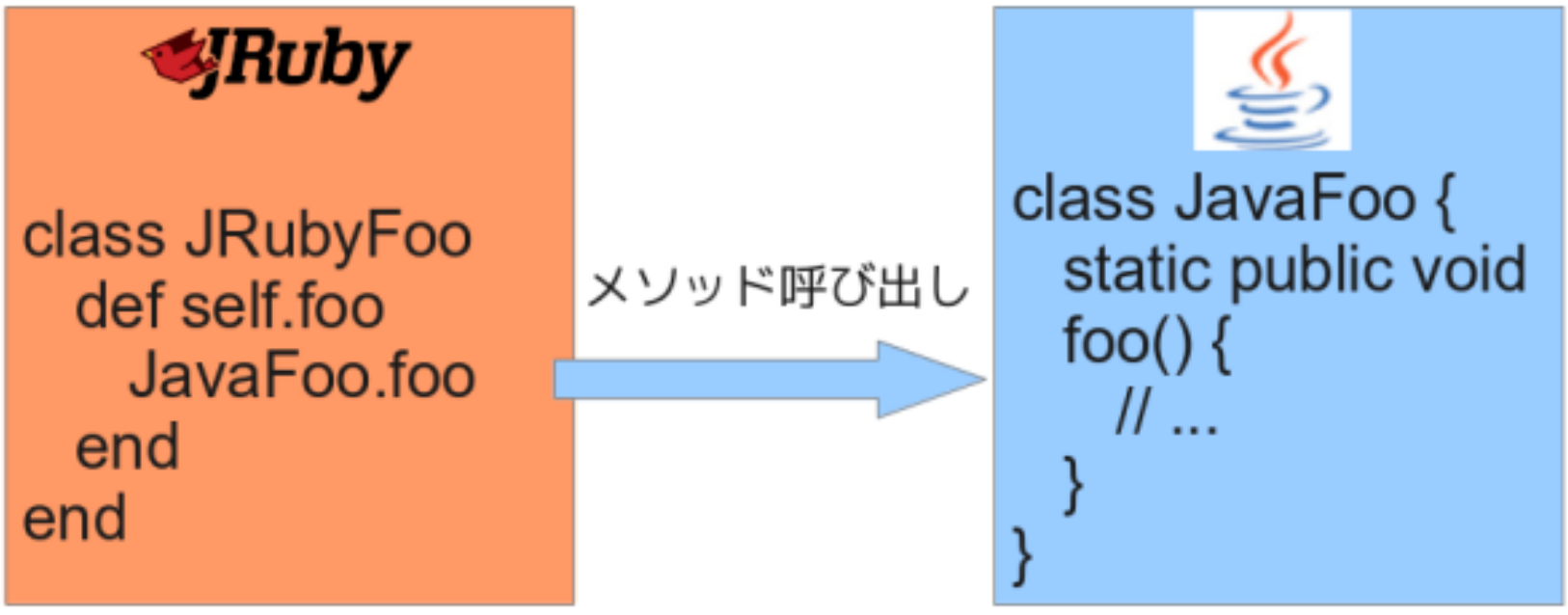
Java+JRuby

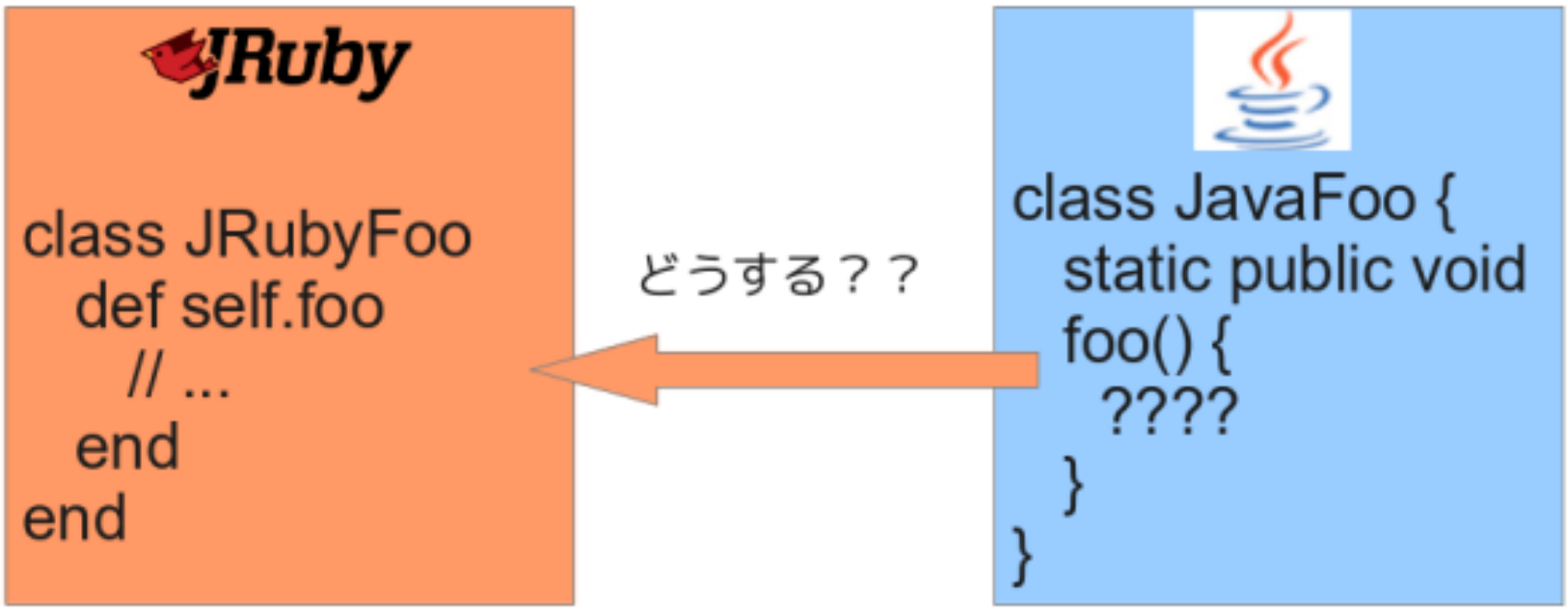
- ✓ JRuby側からJavaを使うのは簡単
  - ✓ メソッド呼び出し
  - ✓ 継承とかとか

Java+JRuby

- ✓ Java側からJRubyを使うのは難しい
  - ✓ けっこう意外だった

どういうことか？


JRuby
class JRubyFoo
def self.foo
JavaFoo.foo
end
end
メソッド呼び出し
class JavaFoo {
static public void
foo() {
// ...
}
}


JRuby
class JRubyFoo
def self.foo
// ...
end
end
どうする？？
class JavaFoo {
static public void
foo() {
????
}
}

どうやるの
（？＿？）

RedBridge

» JRuby Project Wiki Home Page

Embedding JRuby

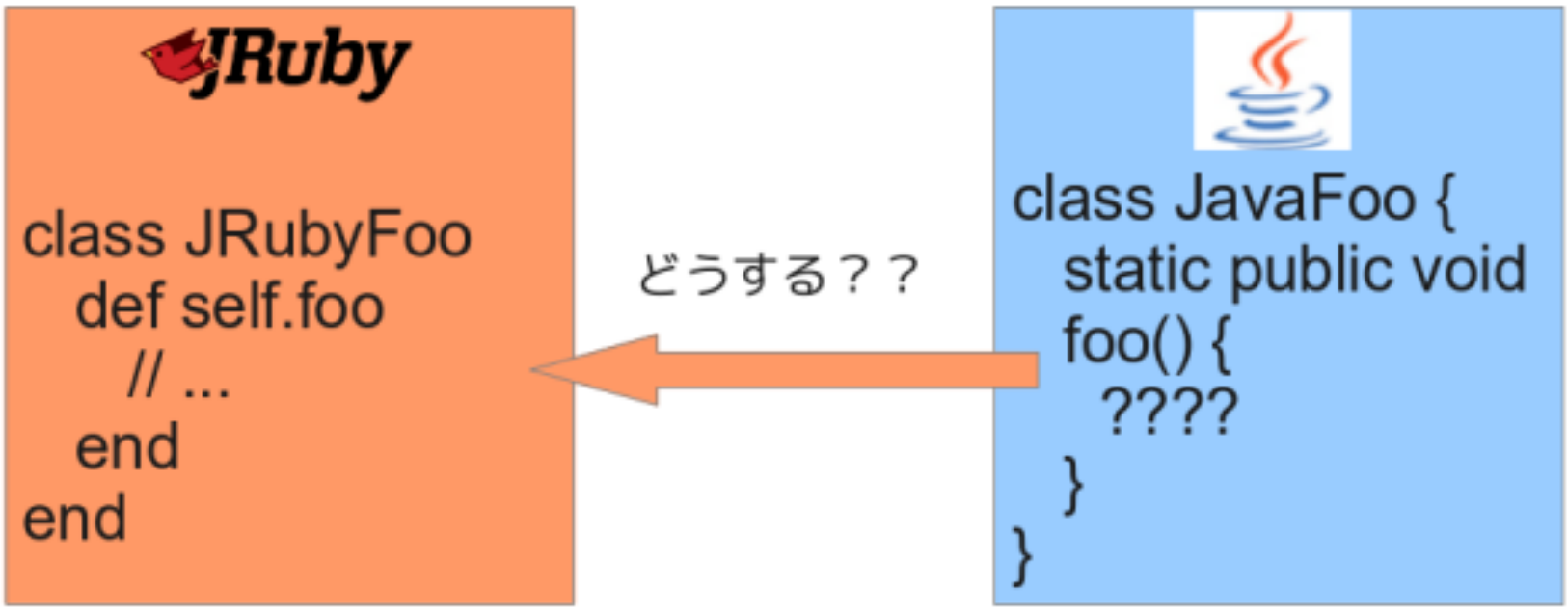
Using Java from Ruby is JRuby's best-known feature--but you can also go in the other di
different way
allowing you
embed JRuby in your Java project.

RedBridge

Table of Contents

RedBridge

- @yokoletさん作（感謝！）
- Javaの中でJRubyの実行環境（コンテナ）を作る
  - コンテナにソースコードぶちこんだり、Rubyコード片を評価できる


JRuby
class JRubyFoo
def self.foo
// ...
end
end
どうする？？
class JavaFoo {
static public void
foo() {
????
}
}

イメージ

```
public static void foo(void) {
  // JRuby実行環境作って
  ScriptingContainer sc =
    new ScriptingContainer();

  // さっきのJRubyFooクラスを読み込み
  Object jrubyFooClass =
    sc.runScriptlet(PathType.ABSOLUTE, "foo.rb");

  // JRubyFoo.fooを呼び出す
  sc.callMethod(fooClass, "foo", Object.class);
}
```

Regicideでは
この辺の技術を
なんやかんや
使ってます

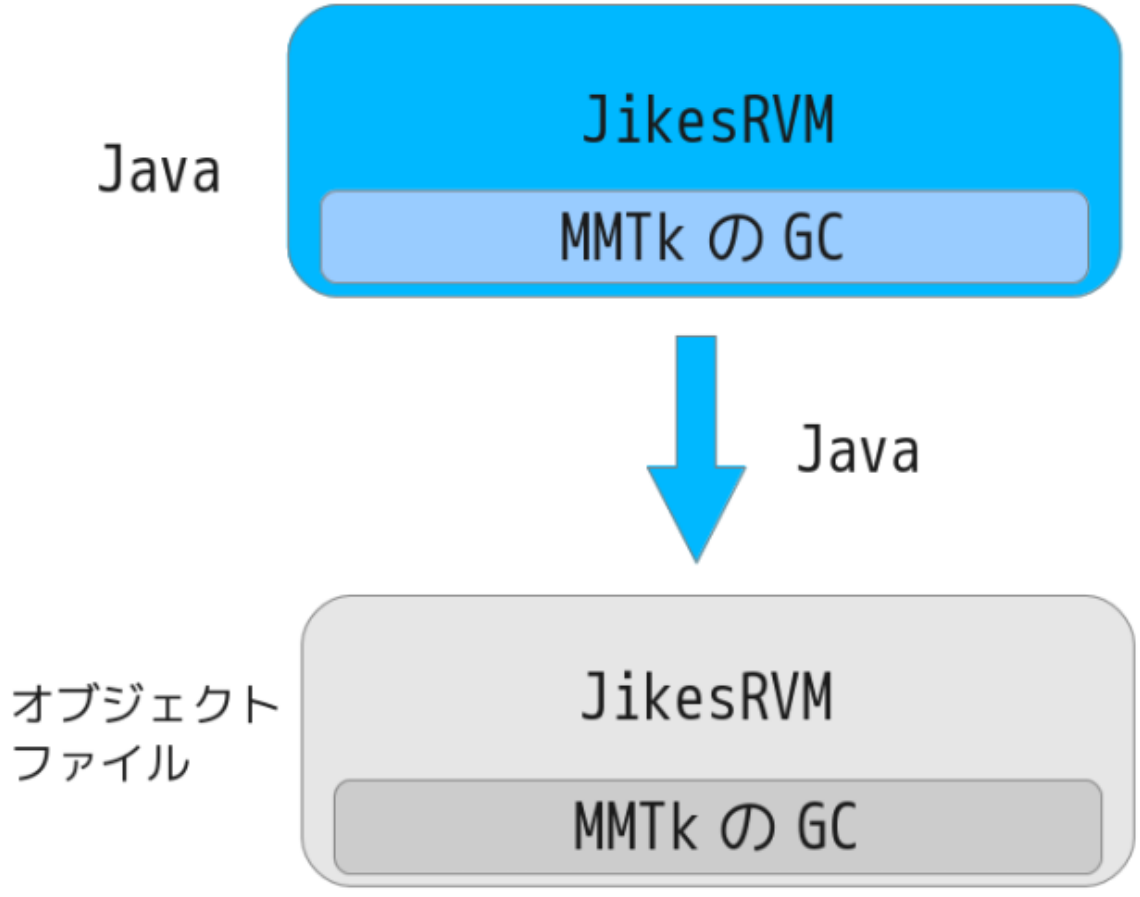
Regicideの悪いところ

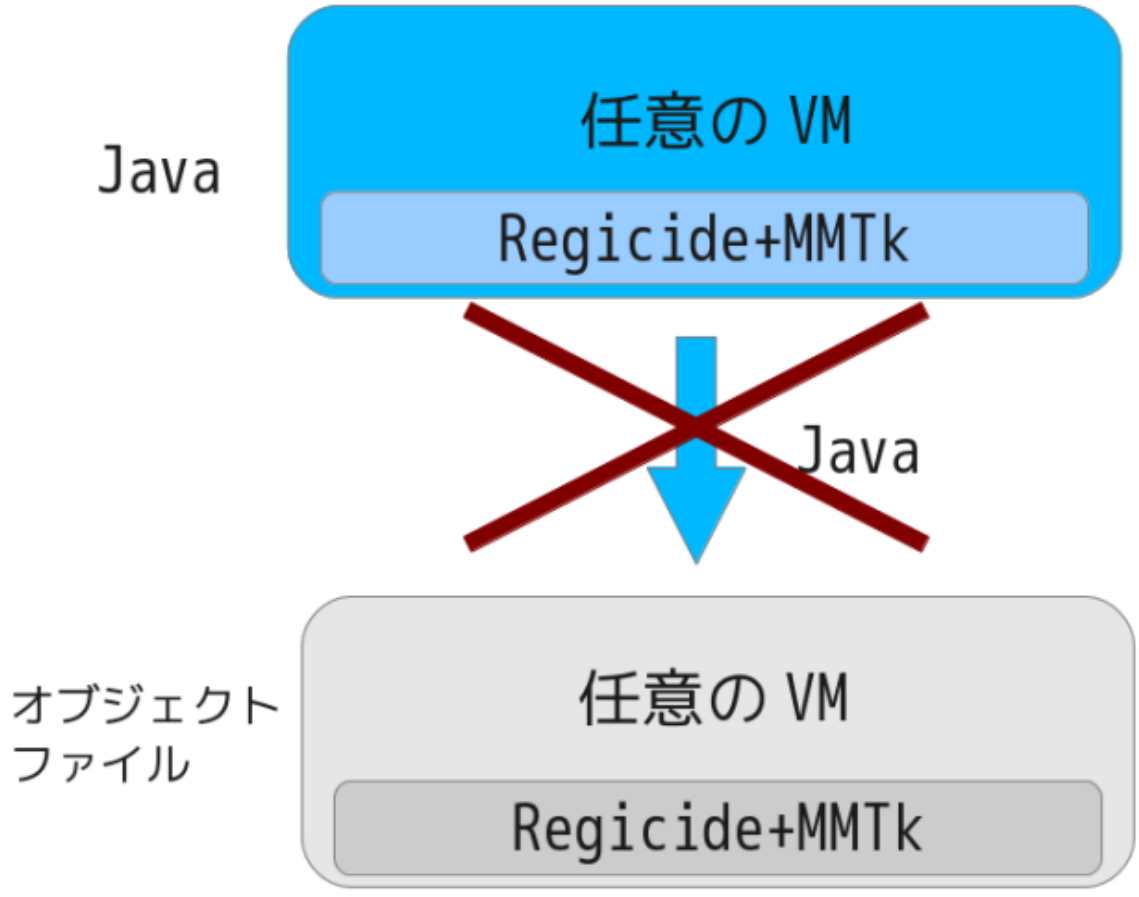
VM作るのめんどい

けっきょくVMは書かないといけない

- GCを評価するためにはある程度ちゃんとしたものが必要
  - でもVMとか興味ないしさぁ...
- LISPすら作るのが億劫
- 飽きてきた

JVMのGCの影響を受ける


Java
JikesRVM
MMTk の GC
Java
オブジェクト
ファイル
JikesRVM
MMTk の GC


Java
任意の VM
Regicide+MMTk
Java
オブジェクト
ファイル
任意の VM
Regicide+MMTk

ふつうにJavaのコードとして動かす感じ

JVMのGCに引っ張られる

- ✓ 性能が引っ張られる
  - ✓ GC停止も不可能だしきちんと性能評価しづらそう
    - ✓ 論文とか書きづらそう

変換ができればいいのに...

- ✓ JikesRVMのブートストラップ部分にJRubyを突っ込む
  - ✓ うまく動かなかった
    - ✓ JRubyがJikesRVMの秘孔を付いているらしい
    - ✓ JikesRVMがまだ未熟

Regicideの良いところ

VMが自分で書ける

言語処理系が自前で書ける

- ✓ GCにすごく優しい処理系とか書ける
- ✓ 言語処理系ひっくるめて検討できる

おいしいところは全部**Ruby**で書ける

- ✓ VM, GC, 全部Rubyで書ける
- ✓ Cみたいに阿鼻叫喚しなくていい（かもしれない）
- ✓ 生産性が100倍（らしい）

勝手に処理系が速くなる感

- ✓ JVM速くなる＝Regicide速くなる
  - ✓ WIN-WIN（笑）
    - ✓ Cでも一緒か…

ほかのGCと比較可能

- ✓ 十数個のGCと性能比較ができる
- ✓ 起動時のオプションを少しいじるだけでOK
  - ✓ 論文とかかけるかもしれない！

githubにあげています
authorNari/regicide

今後の展開

作ってみて気づいたこと

- GC部分は外っ面がRubyなだけで実はCとあまり変わらない感じ...
  - これならCで書いても一緒なのでは...

作ってみて気づいたこと

- ✓ 実はGC実装に特化したミニ言語が欲しいだけなのか…？
  - ✓ 今後はそちら方面で攻めてみたい

まとめ

まとめ

- ✓ やっぱりRubyの話できなかった
- ✓ C死ねと思って作ったRegicideがすでに死にそう

ご清聴
ありがとう
ございました


ふぅ～
やっぱ理解ってる奴が
居ると楽だわ