

Seasar Conference 2008 Autumn



GroovyでProjectZero 軽量+Javaな世界

Javaと軽量言語はどう交わっていくのか?

Java読書会:根本和郎

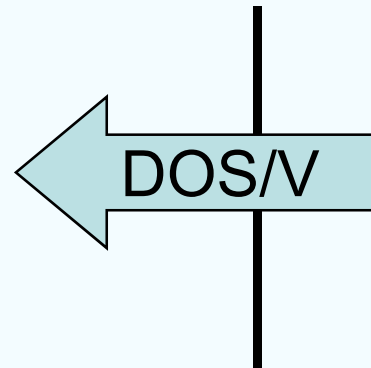
- 所属はIBMですが、個人参加です
- 記述内容は個人的見解に基づきます

- Web界で多用されている軽量言語が、エンタープライズ界にどう導入されていくのかを予想する
- 考察点
 - 過去のコード資産
 - Javaの為にある環境の移行コスト
 - Webサーバーとの親和性 (WebSphere)
 - 開発環境 (Eclipse)
 - 移植性 (Linux, Windows, iSeries zSeries pSeries,)
 - データの意味の普遍性の重要性、XMLとの親和性
 - 最終的な性能への期待値

- ProjectZeroに見る軽量言語の傾向
- エンタープライズ軽量言語の可能性
- Groovy概略
 - Optional Typing
 - JavaVM言語
 - Web界とEnterprise界の比較
- エンタープライズで軽量言語を適応する場合の評価ポイント

言葉の壁

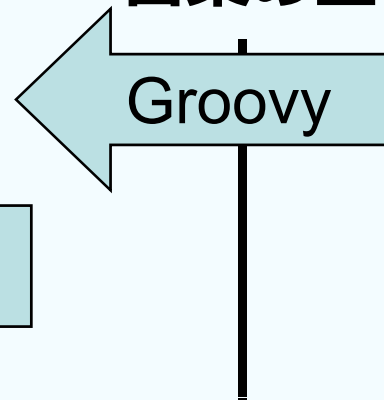
PC98
日本



PC/AT
米国他、多数

言葉の壁

Ruby
Ruby on Rails



Java
Struts, Hibernate, ..

■ 高い忠誠度、高い移行コスト

- Editor : Emacs \leftrightarrow vi
- OS : Linux \leftrightarrow Windows
- 開発環境 : IDE \leftrightarrow Editor
- 言語 : Java \leftrightarrow 軽量言語

■ 言語 (日本語の例)

- 日本人のdefault実装
- 千年以上蓄積された本(豊富なLibrary)
- 1億人以上のuser
- 豊富なコンテンツ (マンガ アニメ ゲーム ...)

■ Rubyへのロイヤリティー

- 言語仕様が好き (嗜好)
- Matzが好き (感情)
- Ruby on Railsを使いたい (アジャイル開発)

■ Javaへのロイヤリティー

- 過去の蓄積したライブラリ (宿命)
- 部門方針 (業)
- 仕事だから (宿命)
- 信頼性を保証するため (仕事)

- ヒルズ族では経済は動かない
見かけの派手さ ≠ 経済インパクト
- トヨタ、銀行、製造、で社会が動く
人気で株価は動かない

—「渋井真帆の日経新聞読みこなし隊」





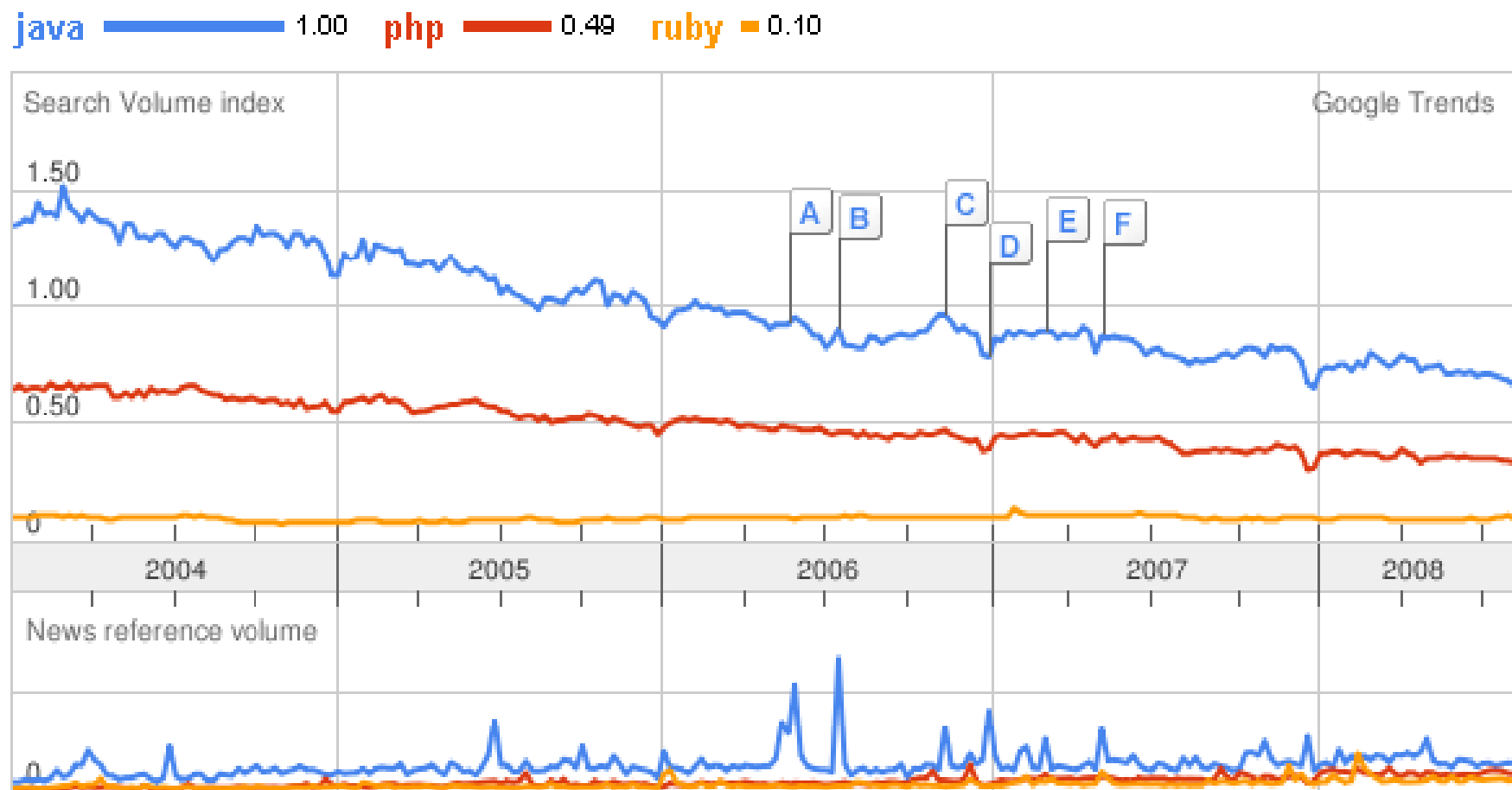
ProjectZero : 手のひらサイズWebサーバー

- JavaSEで動く Webサーバー
- 小型軽量
- www.projectzero.org
- RESTful
- CDCD (Community driven Commercial Development)
- WebとSOAの架け橋能力
- アプリケーション記述言語は Groovy および PHP ← ダブル軽量言語

- Groovy + PHP = ダブル軽量言語?
 - PHPは流行っているから
 - Groovyは何故? Rubyじゃないの?



■ Java 100, PHP 49, Ruby 10
PHP x 2 = Java



■ Community Driven Commercial Development

- ソースを見ることはオープン
- コードの登録は制限あり
 - GPL対策?

■ JavaからRuby：必要前提

- 過去のJavaコード資産の継承がないこと
- 内製であり、SI契約が伴わないこと

■ Java+軽量言語：可能

- Java：高速処理、ライブラリーの継承、Javaプログラマーの登用
- 軽量言語：作り捨てるのユーティリティ、遅くて良いもの、使用期間が短いもの、Programming Firstなプロトタイプ開発用

■ Java補助言語としてのGroovy

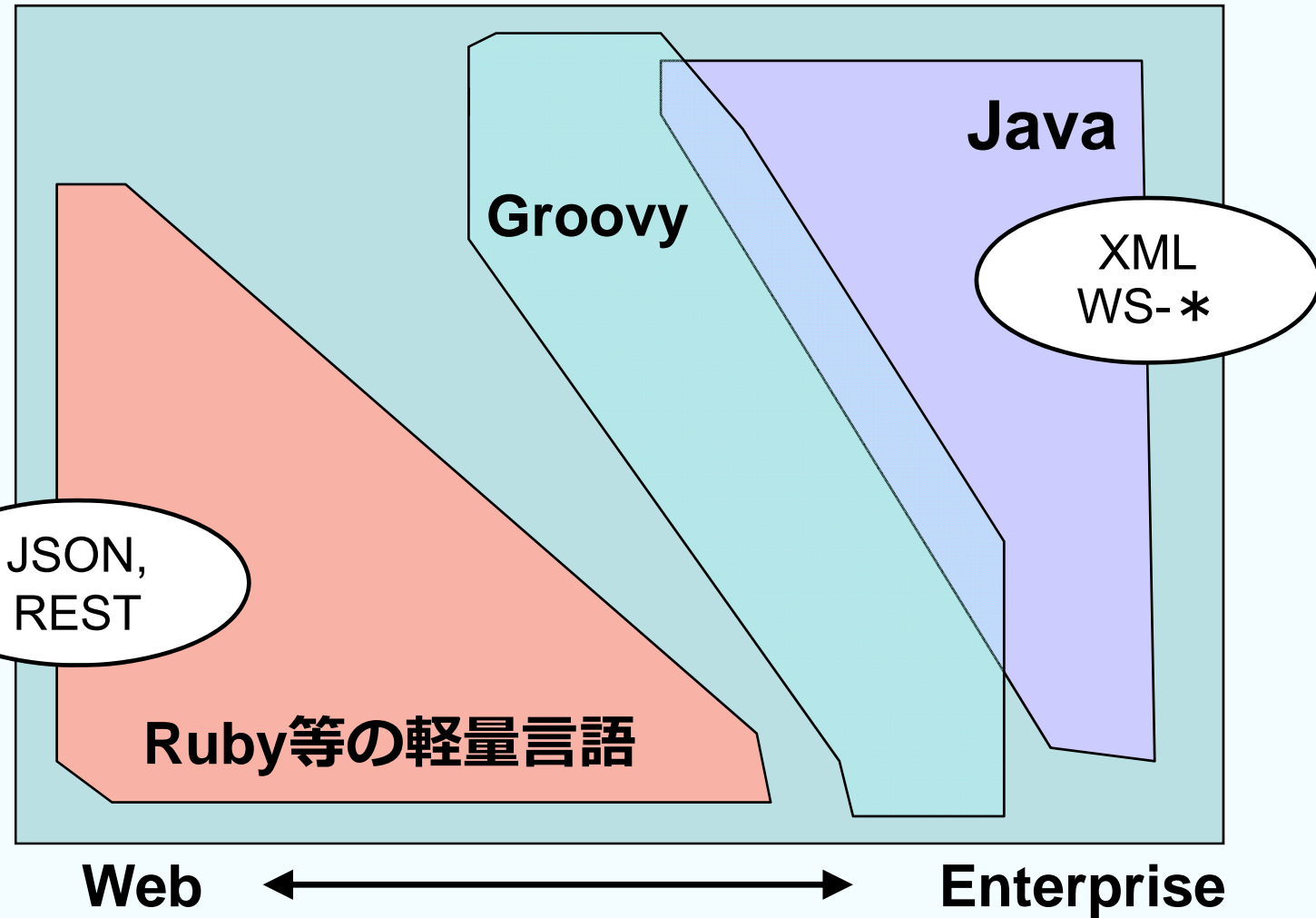
SOA

Complex



Simple

スタート
アップ企業



JSON,
REST

Ruby等の軽量言語

Groovy

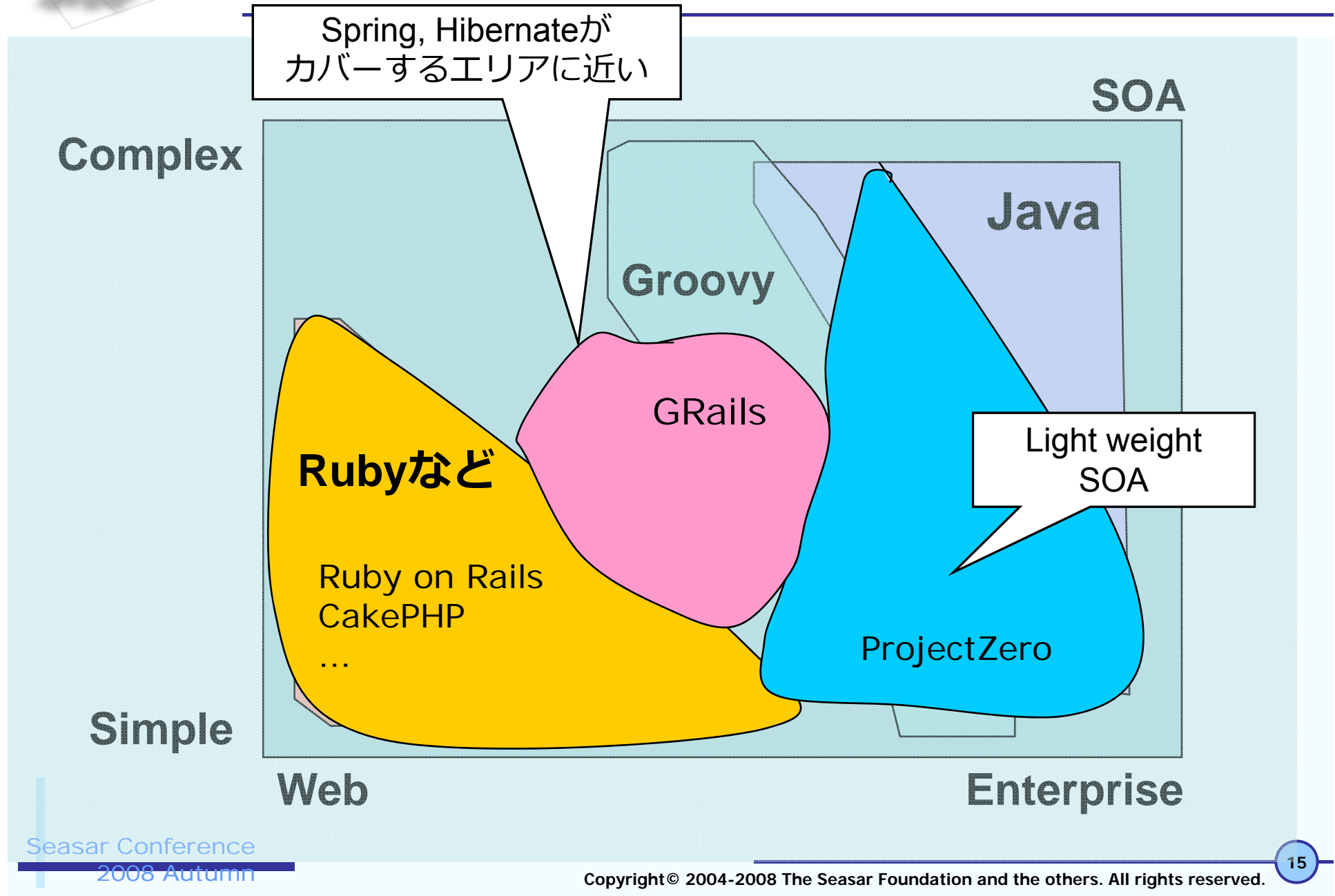
Java

XML
WS-*

Web

Enterprise

軽量言語Framework分布予想図



- JVMで動く
- Javaと親和性が高い、普通のメソッド呼び出しで結合可能
- Rubyの簡潔さをJava風味で表現
- Meta Object Protocol で、全てのアクセスがラップされ、動的変更可能
- Javaにとっての大がかりな糖衣構文 (Syntax sugar)

Foo = 型宣言なし Bar = 型宣言あり

```
class TypeSafeTest {  
    static void main(args) {  
        Foo f = new Foo()  
            f.age = "taro" ← def宣言は、何とでもマッチする  
        Bar b = new Bar()  
            b.age = "hanako" ← X Int宣言したのでcastエラー  
    }  
}
```

```
class Foo {  
    def name  
    def age  
}
```

← def 型だと制約なし
単なるObject型となる

```
class Bar {  
    String name  
    int age  
}
```

← 堅く型宣言することもできる

- 「1,2,3,4,5」という数字の配列から、偶数を抜き出して印刷せよ

Unixっぽい

Ruby

```
p [1,2,3,4,5].select { |i| i%2==0 }
```

Groovy

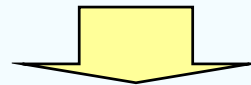
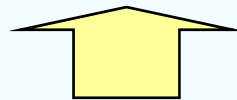
```
println ([1,2,3,4,5].findAll { it%2==0 } )
```

“動詞(find) + 目的語(All)” でJavaっぽい

Groovy側

```
class Hello {  
    static void main(args) {  
        World wd = new World()  
        wd.greet(" world! ")  
    }  
}
```

実行結果:
Hello world Groovy!



相互呼び出し

Java側

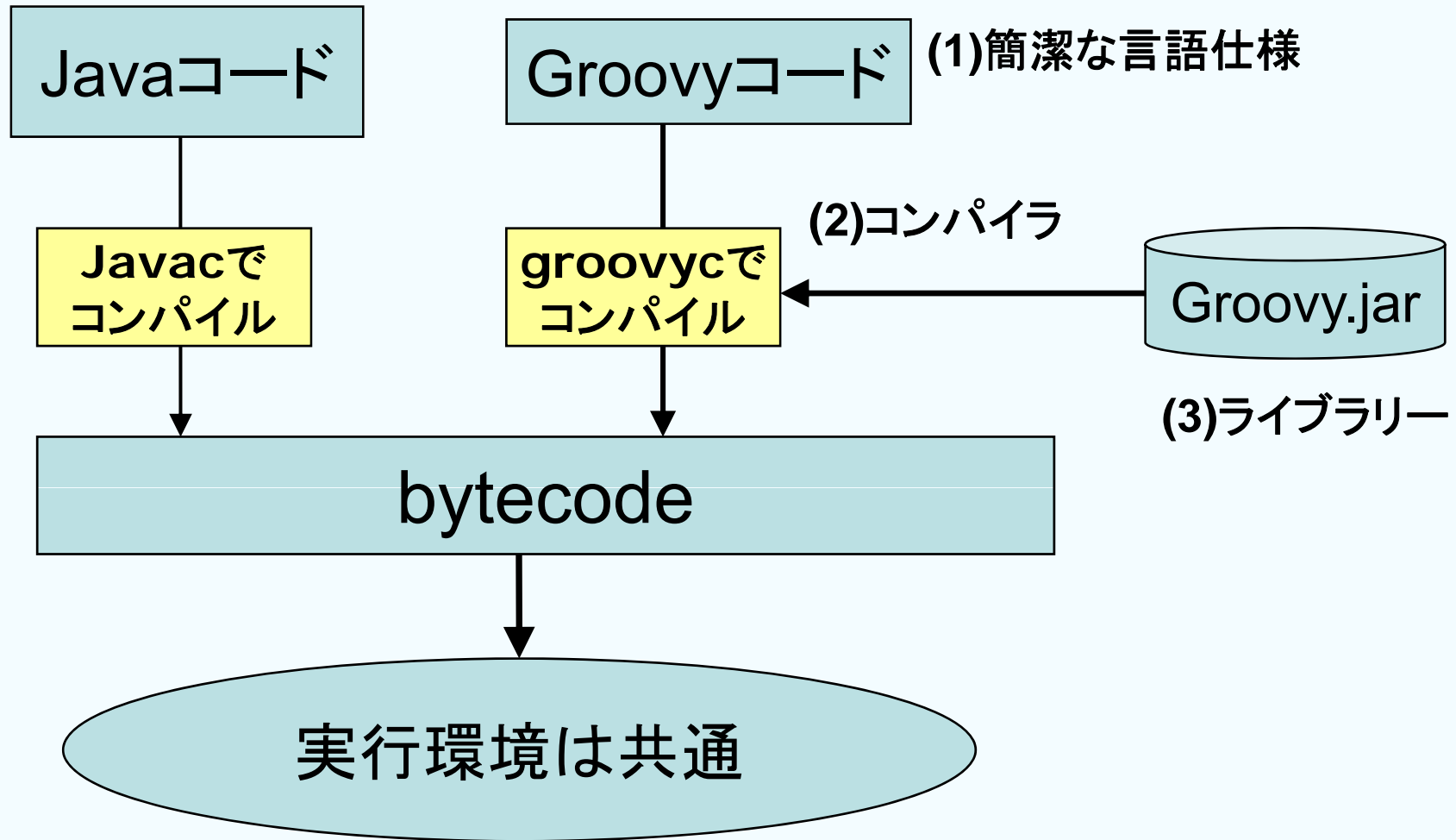
```
public class World {  
    public void greet(String arg) {  
        System.out.println("Hello " + arg);  
    }  
}
```

HelloWorld.groovy

```
public class HelloWorld {  
    public static void main(String[]  
        args) {  
        System.out.println("Hello  
            World");  
    }  
}
```

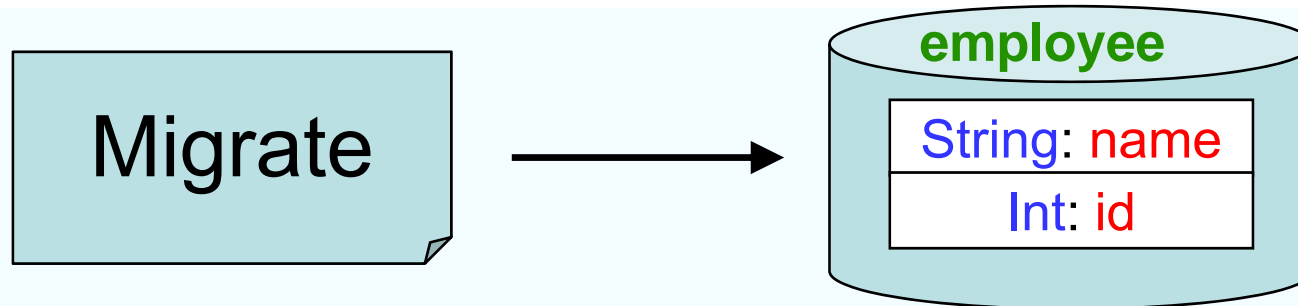
※ このページはアニメーションを含みます

Java + 糖衣構文 = Groovy
Syntax Sugar





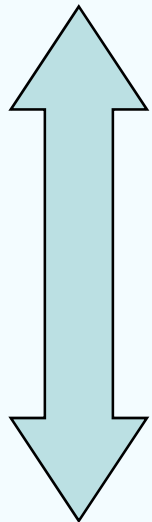
- Javaのプリコンパイラ
- Java風味軽量言語
- Optional Typing



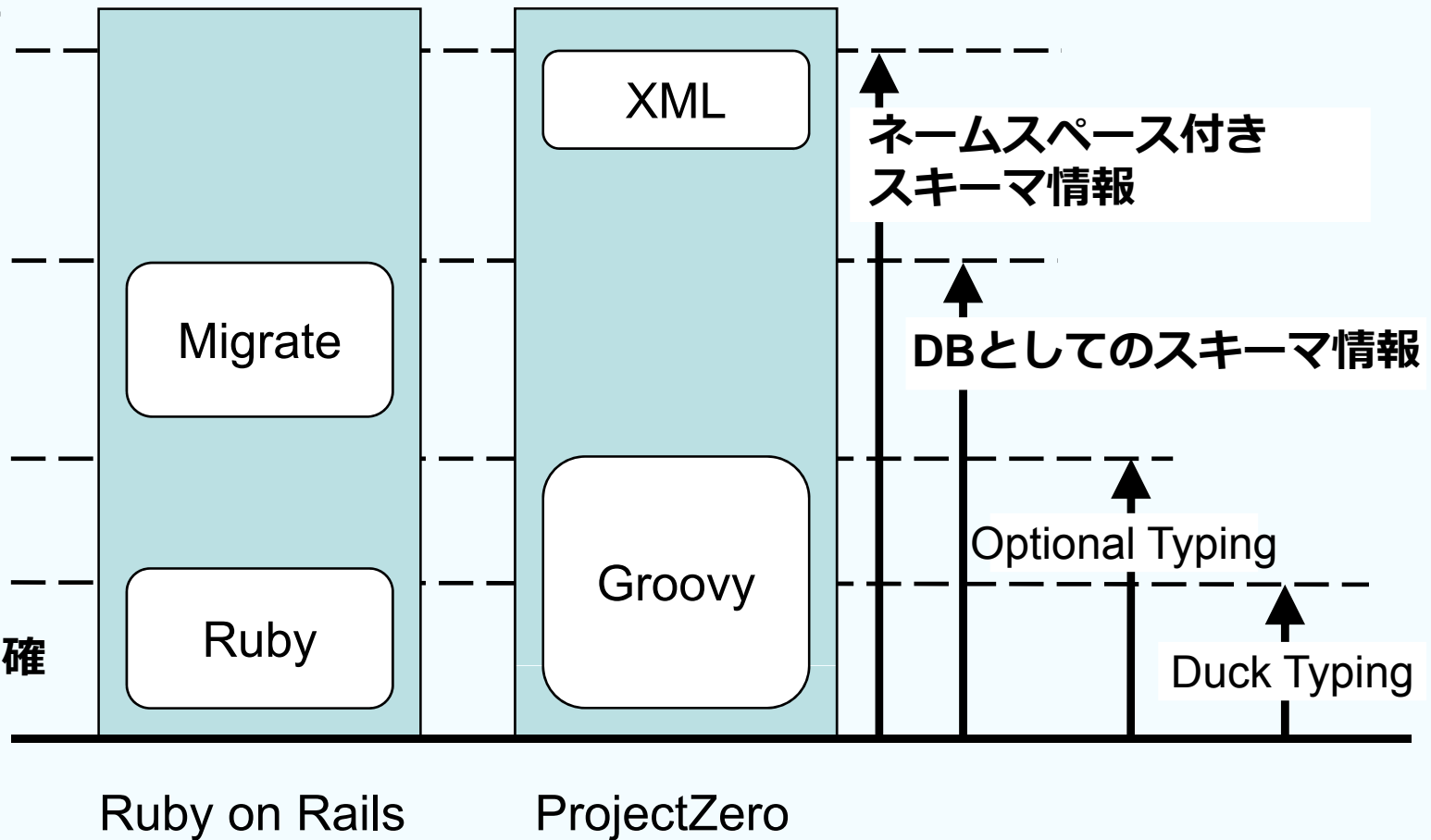
Migrate

```
class MyDatabase < ActiveRecord::Migration
  def self.up
    create_table(:employee) do |t|
      t.column(:name, :string, :limit => 100, :null => false )
      t.column(:id, :integer, :null -> false)
    end
  end
  def self.down
    drop_table :employee
  end
end
```

厳格かつ煩雑



簡潔かつ不正確



方式	記述	表現
XML Schema	<pre><?xml version="1.0"?> <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <xsd:element name="greeting" type="xsd:string"/> </xsd:schema></pre>	完全にユニーク エンタープライス 間に使用可能
Migrate	<pre>t.column(:name, :string, :null => false)</pre>	同一開発 チーム内
Groovy	<pre>String greet = "Hello" --- または --- def greet="Hello"</pre>	同一 Serialized Object間
Ruby	<pre>greet = "Hello"</pre>	Duck Typing式 型推測

■ SeasarConの会場でRubyの普及度を測ってはいけない

- × 渋谷の街頭調査
(既に母集団が偏っている)
- ○ 無作為な番号に電話を掛けての聞き取り

JavaからRubyなんて
簡単と思っている人

Silent majority

SeasarCon

RubyKaigi

- Java : JCP
Java **Community** Process
- Groovy : JSR241
“The Groovy Programming Language”
- 個人ではなく、みんなで決めていること
- 言語はみんなのものであるべき

- 信頼性 \neq 99.99%
- 信頼性 = 「挙動が定義通り」
- 例：嘘つき村の住人は高信頼性
`assert(!言っていること == true)`

- 定義(スペック)重要
「バグではなく仕様です」
動きが定義されているから言える。
- Groovy = JSR241

- 不採用：軽量だから速い
- 採用：重いがフルスペック
全機能が実現可能
 - ○：重くて遅い
 - ×：～ができない
- 例
 - Eclipse = あらゆる開発のPlatform
 - Dojo Toolkit = 万能AJAXフレームワーク



■ JavaVMで動く

- 全OSで動作可能
- Javaの環境が使用可能
 - 64bit、マルチコア、...

■ 二段階動作

- goooovy で直接スクリプト起動
 - サクサク開発
- groovyc でバイトコード生成
 - classやjarにして、しっかり動作

■ 軽量言語

- ジャズ・セッション
- 高い個人の自由度
- Ad-hocな開発に向く

■ Java

- オーケストラ
- 大規模開発の為の強い制約
- 長期的保守を最初から想定

- 実は . . .
開発生産性 < 保守容易性
- 型制約があるコード (Java)
 - 手がかりが多く、読みやすい
 - コードに意外性がない
 - 制約に基づく開発が可能
- Duck typing
 - 手がかりが少ない

■ Eclipse

- みんなのビジネス基盤
- イケてるPlug-in を作ってビジネス可能

■ Ruby

- GPL
- Rubyコードに触れるもの全てGPLになる

■ Ruby != Eclipse



まとめ：エンタープライズ軽量言語としての評価点

- 普及度
- 言語仕様策定プロセス
- 信頼性
- 最終的な性能への期待値
- ライセンス形態
- 真の開発生産性
- 学習コスト