

ヲレヲレWebフレームワーク
「Web Flavor」を作ってみよう



西本 圭佑 NISHIMOTO Keisuke
keisuken@cappuccino.ne.jp



わんくま
同盟

わんくま同盟 大阪勉強会 #26

お品書き

- 自己紹介
- Webフレームワークとは
- Web Flavor概要
- デモ
- まとめ

自己紹介

- 緒言 
 - 西本圭佑, NISHIMOTO Keisuke
 - Twitter: @keisuke_n
- 趣味
 - プログラミング (Java/Ruby/Scala)
 - 工作(電子など)
- 仕事 (Java/C#/PHP/JavaScript/ActionScript)
 - Webアプリケーション
 - マルチメディア関連

自己紹介

• 現況

- わんくま同盟 富山勉強会 #01
 - JavaからScalaへ
- ついこの間はASP.NET(C#)
- 最近はPHP+JavaScript (泣)

• 今後

- サプリメントではない鉄分補給
 - 鉄道博物館へ行きたい！
 - 去年も言ったorz

Webフレームワークとは

• 主なWebフレームワーク

- .NET: ASP.NET
- PHP: CakePHP/Smarty/Zend/Ethna
- Perl: Catalyst/Sledge/Tripletail
- Python: Django
- Ruby: Rails
- Java: Struts/Spring/Seasar/Wicket
- 他にもたくさん、今まさにカオス

Webフレームワークとは

• 最近の傾向

– 全部入り

- ORマッパ, Web API (XML RPC/REST)
- テスト (TDD/BDDなど)

– GUI的アプローチ

- ステートフル(セッションレス)
- イベントアクションモデル

– スケーラビリティ

- セッションの同期化、負荷分散
- 依存性の回避(DIコンテナなど)

Webフレームワークとは

• 問題点

– 開発

- 定義が複雑 (DRY/CoCではない)
- 枠組みにはめすぎ (柔軟性がない)
- 分業がしにくい (デザインとロジック, 開発の粒度/依存)
- 学習コストが高い

– 運用

- スケーラビリティ (負荷への対応)
- メンテナンス (修正, 改良, セキュリティ)

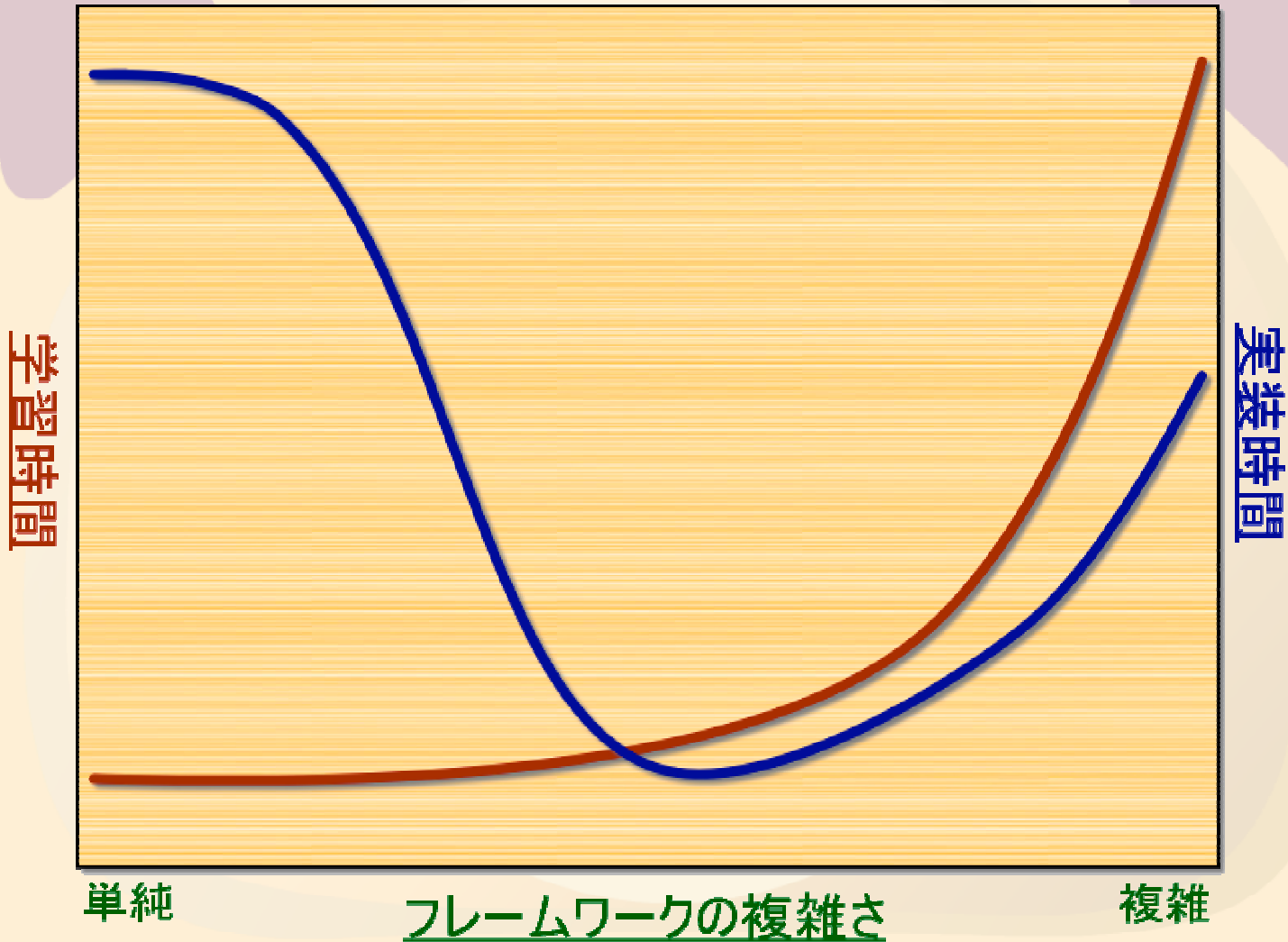
– その他いろいろ

Webフレームワークとは

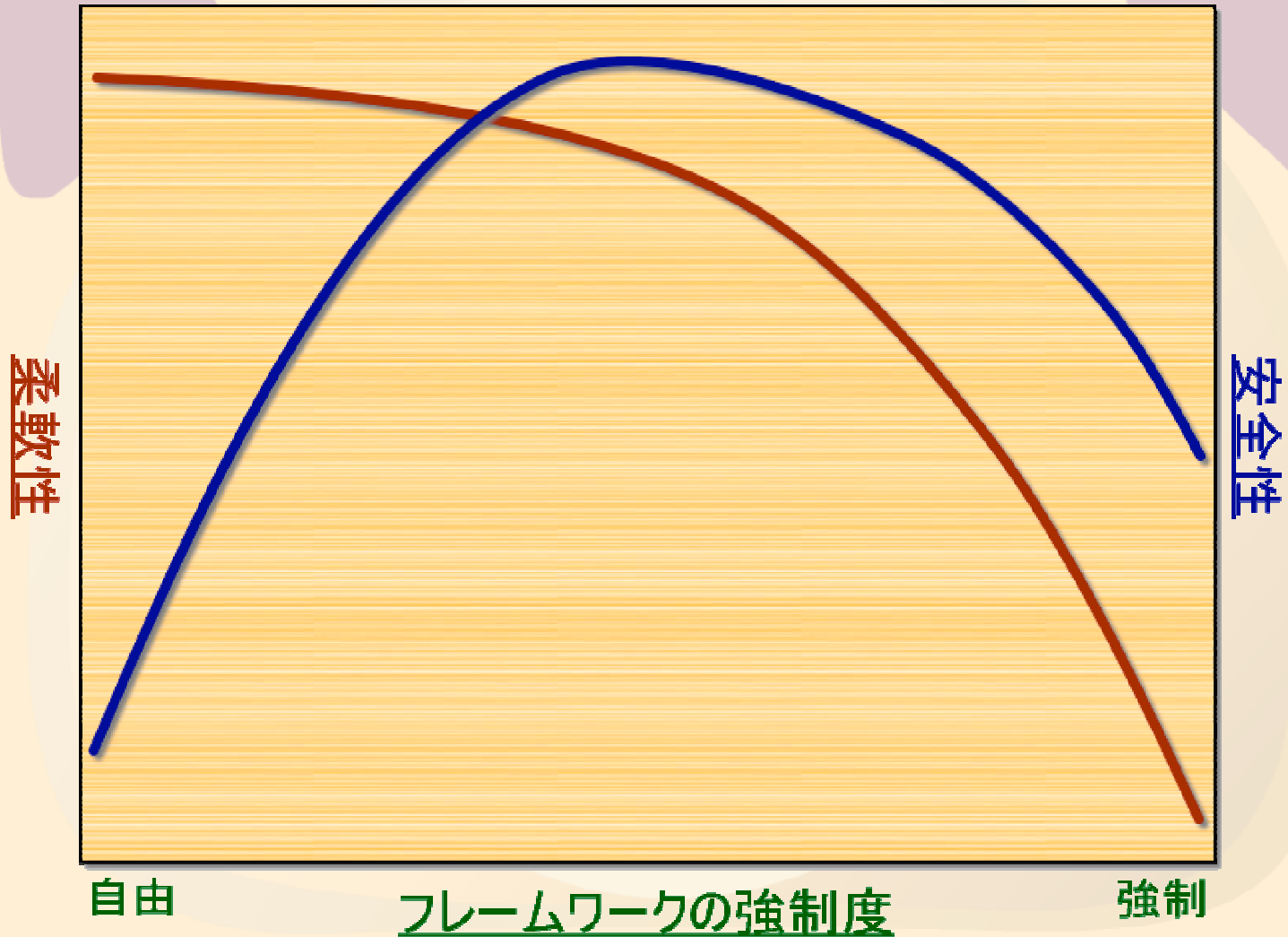
- 考察

- 学習時間
- 実装時間
- 柔軟性
- 安全性
- 分業
- メンテナンス

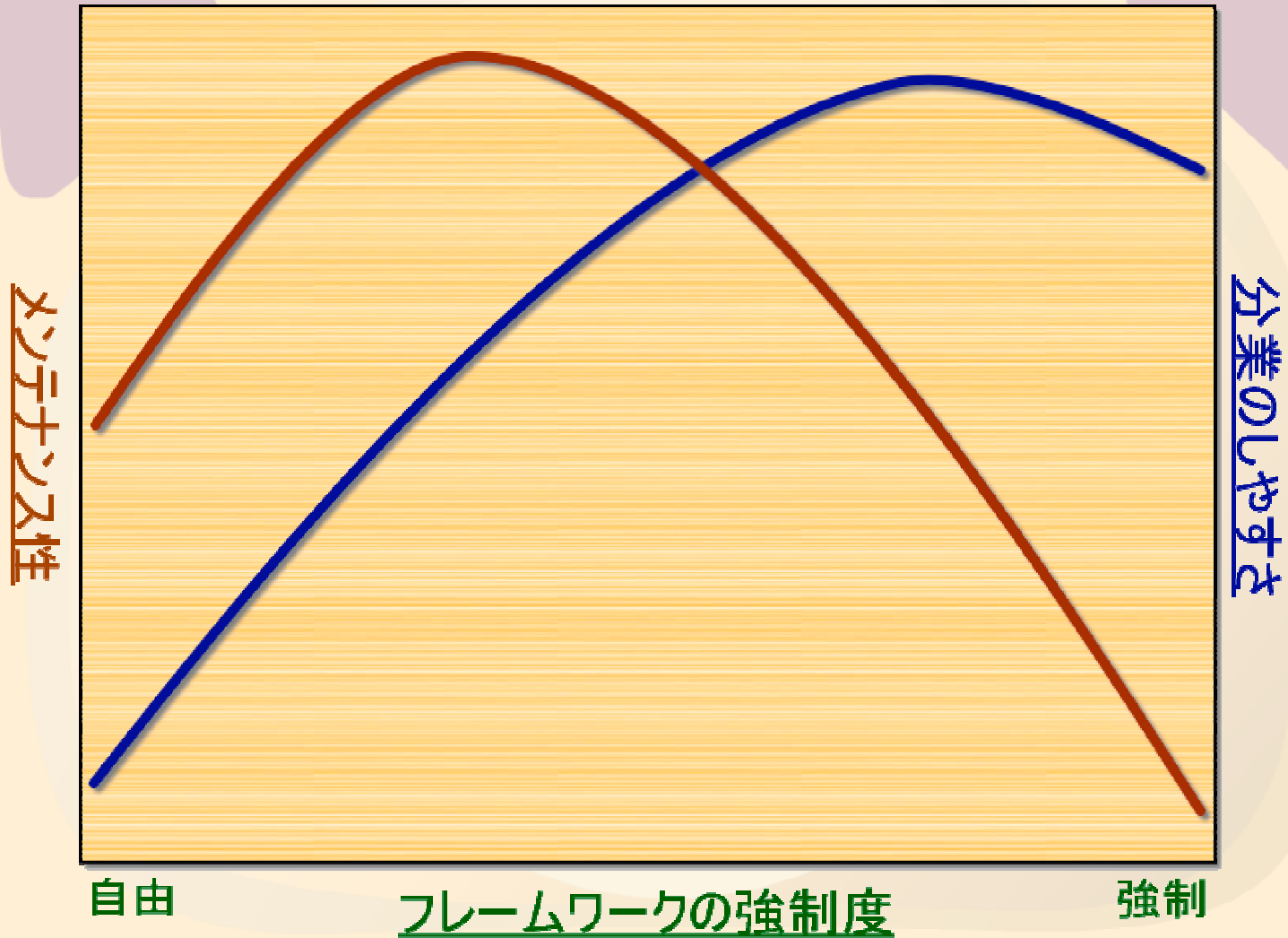
Webフレームワークとは



Webフレームワークとは



Webフレームワークとは



Webフレームワークとは

- 考察まとめ

- Webフレームワークは汎用ではない
- Webフレームワークは複雑になってはいけない
- 機能面はなるべくライブラリにする
- 特殊な処理は使い捨て(再利用を考えない)

Web Flavor概要

- 主な特徴

- <http://webflavor.sourceforge.net/>
- Scalaで作られ、Scalaで開発
 - 記述/柔軟性が高い, パフォーマンスもいい
- 自由である
 - 無料, MITライセンス, 枠組みを強制しない
- ハードルが低い
 - デプロイ不用, シンプルな枠組み
- 日本人が開発している
 - keisukenという人が開発している



Web Flavor概要

- Scalaとは?

- <http://www.scala-lang.org/>
- オブジェクト指向と関数型言語を統合
- 静的型付け, コンパイラ型言語
- インタプリタ装備(対話環境含む)
- Java VM上で動く(.NET CLR上でも動く)
- OSS(Scalaライセンス(修正BSD))
- EPFL(スイス連邦工科大学)を中心に開発

Web Flavor概要

- Scala: Hello, world!

```
object HelloWorld {  
  def main(args: Array[String]) {  
    println("Hello, world!")  
  }  
}  
  
object HelloWorld extends Application {  
  println("Hello, world!")  
}  
  
println("Hello, world!")
```

Web Flavor概要

- Scala: フィボナッチ数列

```
def fib(n: Int): Int =  
  if (n == 0) 0  
  else if (n == 1) 1  
  else fib(n-1) + fib(n-2)  
  
println(fib(30))
```


Web Flavor概要

- 方針

- シンプルな概念と構造
- 強制的な枠組みにしない
- 機能はライブラリで提供
- (実装はなるべく少なく簡単に)

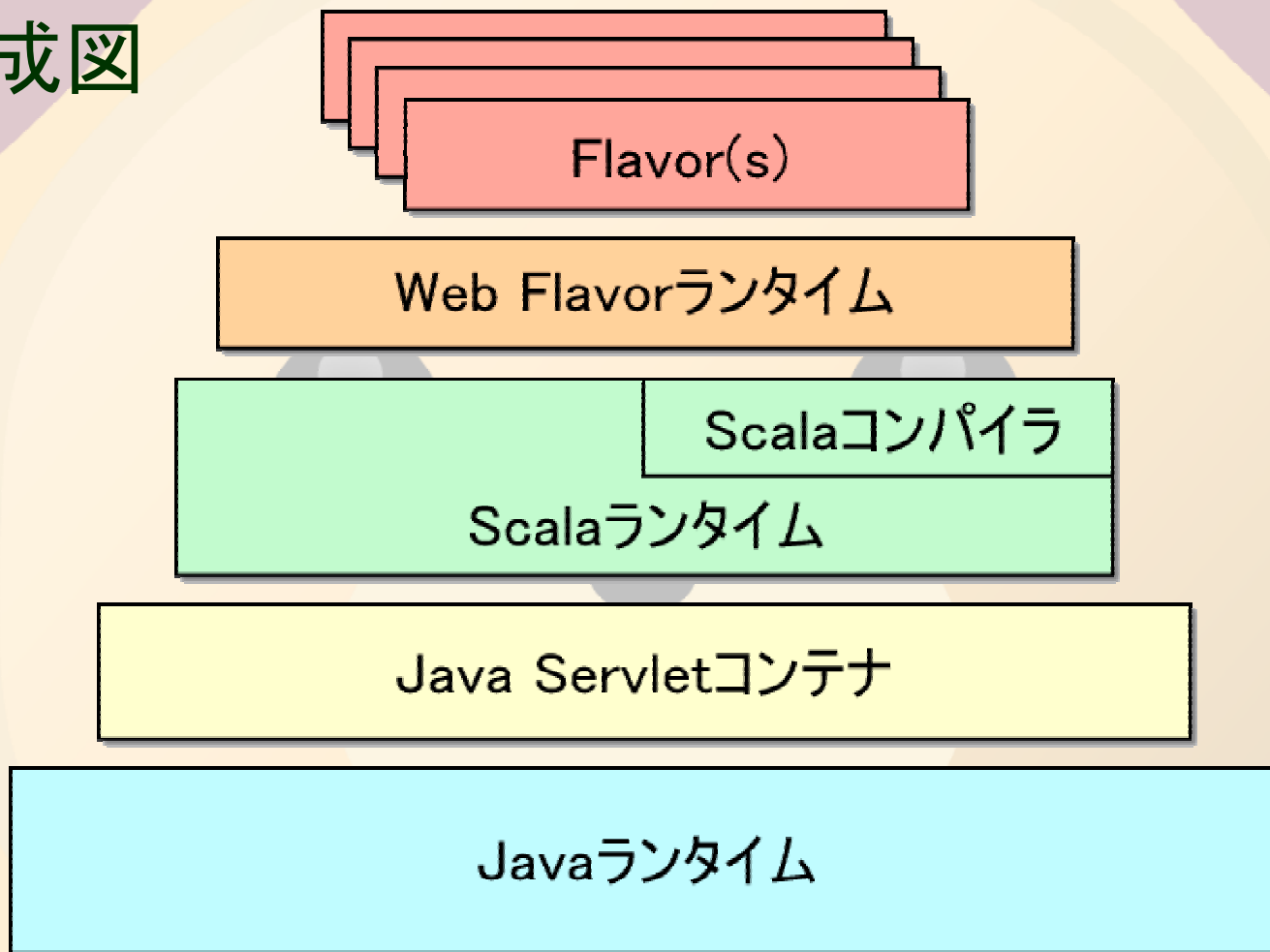
Web Flavor概要

- 主な特徴

- Scalaを使う (**超**重要)
- Java Servletをベースに
- デプロイ不用 (ファイルを置くだけ)
- XMLリテラルでView

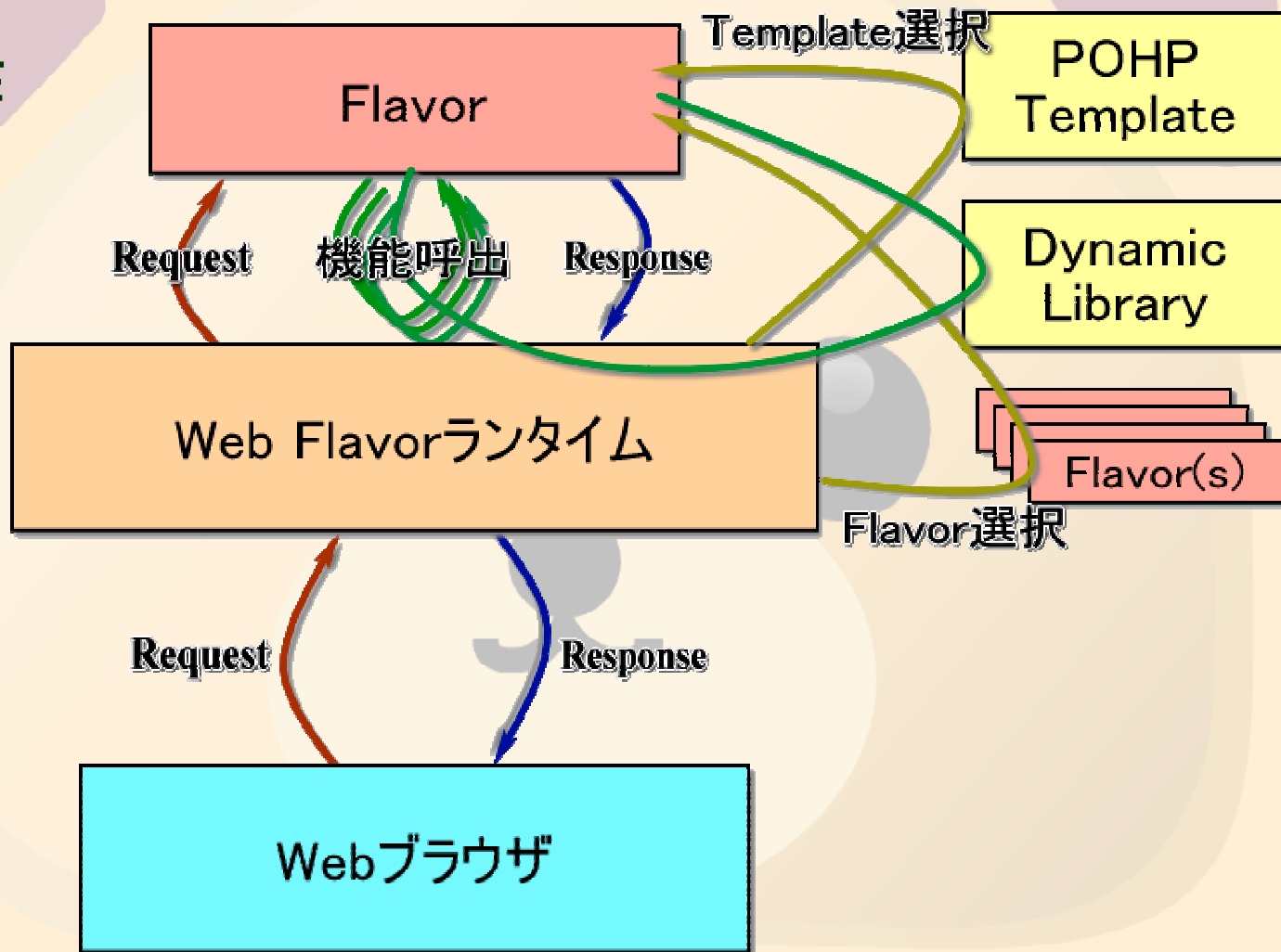
Web Flavor概要

- 構成図



Web Flavor概要

• 動作



Web Flavor概要

- Hello, world!

```
val TITLE = "Hello, world!";  
<html>  
<head>  
<title>{TITLE}</title>  
</head>  
<body>  
<h1>{TITLE}</h1>  
</body>  
</html>
```

Web Flavor概要

- PrintParameters

```
val TITLE = "Hello, world!";  
<html>  
<head><title>{TITLE}</title></head>  
<body>  
<h1>{TITLE}</h1>  
<dl>{  
    for ((name,value) <- params)  
        yield <dt>{name}</dt><dd>{value}</dd>  
}</dl>  
</body>  
</html>
```



Web Flavor概要

- POHP
- Plain Old HTML Page
- Pure HTMLをテンプレートにする
- Pure HTMLだからWebデザイナーと連携化
- 埋め込むHTMLはXMLリテラルを挿入する
- 繰り返しは結局XMLリテラルになってデザインとの分離はしにくいがまあそれはよしとする
(CSSをうまく使ってくれ(^^;)

Web Flavor概要

- POHP例(テンプレート)

```
<html>
```

```
<head>
```

```
<title><span flavor:id="title"/></title>
```

```
</head>
```

```
<body>
```

```
<h1><span id="title"/></h1>
```

```
<span flavor:id="message">Hello</span>
```

```
</body>
```

```
</html>
```


Web Flavor概要

- POHP例(Flavor)

```
Template("",  
  "title" -> Text("Hello, world!"),  
  "message" -> <p>Hello, POHP world!</p>  
)
```

デモ

- Hello, world!もの
- リクエストパラメータの出力
- BBS
- Wiki
- ライブコーディング

まとめ

- フレームワークの方針
 - フレームワークはシンプル
 - ライブラリを充実する
 - DRY/CoCを踏襲する

まとめ

- 今後
 - ステートフルGUI(Ajax含む)
 - ORマッピング対応
 - 国際化
 - テスト対応
 - スケーラビリティ

おわり

ご静聴ありがとうございました！

Any Questions?

