# PHP at Yahoo! JAPAN

ヤフー株式会社　荻原　一平

PHPカンファレンス2007
2007/09/01

# トピックス

Yahoo! JAPANについて

Yahoo!とPHPの関係

なぜPHPを採用したか

Yahoo!における現在のPHP



YAHOO! JAPAN

# Yahoo! JAPANについて

YAHOO! JAPAN

# Yahoo! JAPAN




YAHOO! JAPAN

# Yahoo! JAPANの規模

## 1日平均ページビュー

# 13億38百万PV

(2007年7月分 月次発表)

# Yahoo! JAPANの規模

## 月間アクティブユーザID数

# 1,996万ID

(当月中にログインしたYahoo! JAPAN ID数)

(2007年7月分 月次発表)

# Yahoo! JAPANの規模

## サービス数

# 約140

(2007年8月時点)

# 言いたいこと！

# きっと、
# 日本一PHPを
# 使ってる…

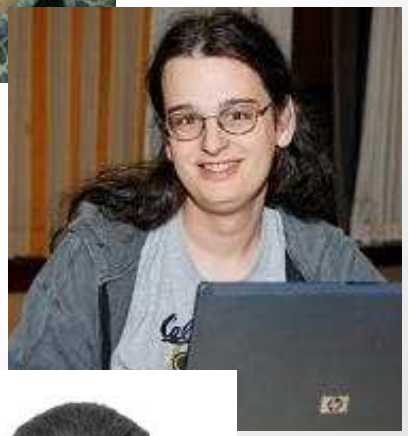
# Yahoo!とPHPの関係

YAHOO! JAPAN

# Yahoo! Incに在籍するPHP開発者

多くのPHP開発者が在籍

- **Rasmus Lerdorf**
  - PHPをつくった人
- **Sara Golemon**
  - runkitなどのPECL拡張メンテナ
- **Dustin Whittle**
  - symfonyエバンジェリスト





YAHOO! JAPAN

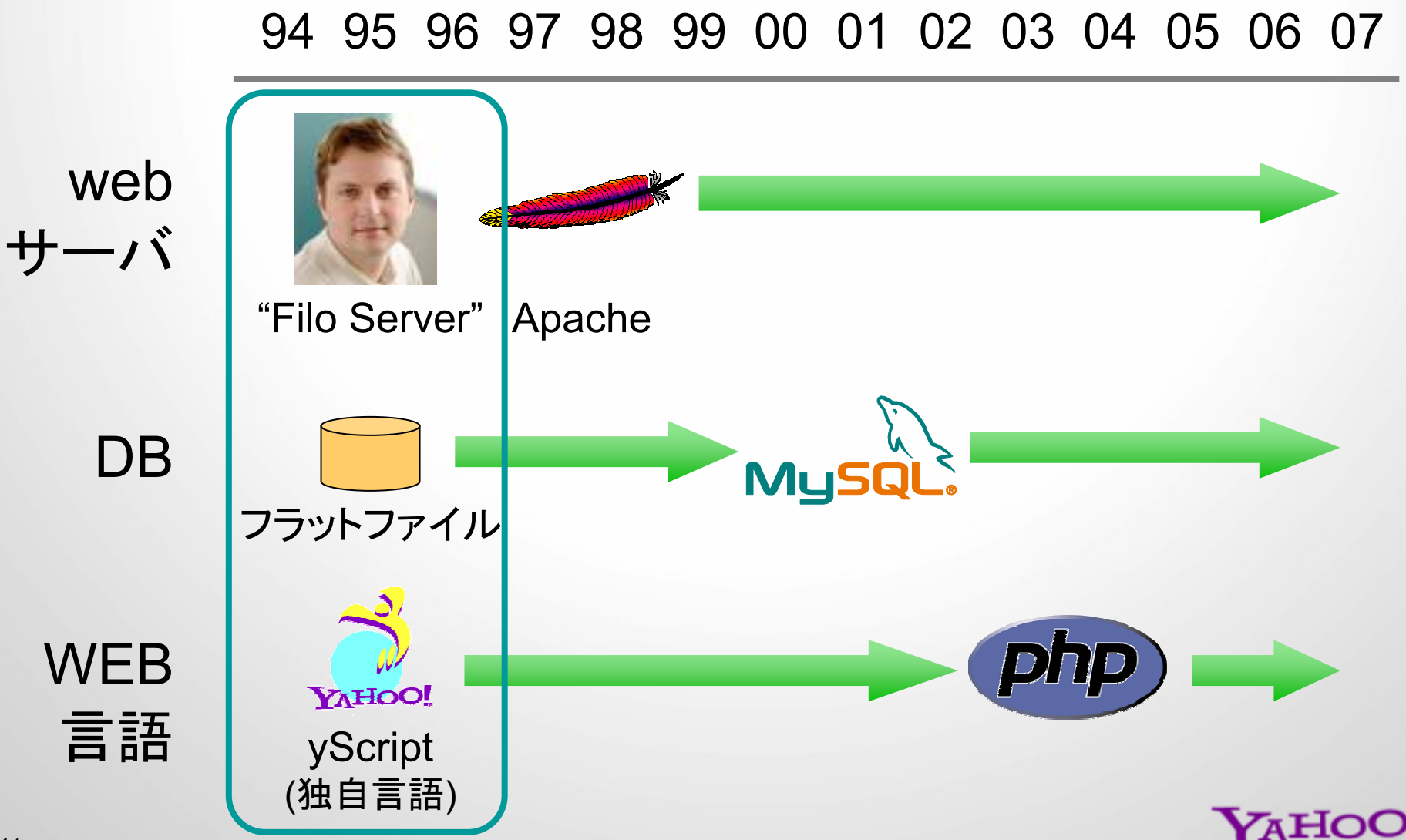
# Yahoo!のPHPコミュニティへの貢献

- 機能追加や拡張モジュールのcontribute
  - 社内で開発されたもので、一般的に適用できる追加機能はオフィシャルのPHPに還元
  - apcやfilter拡張など
- バグレポート、改修
  - 社内から上がってきたバグの検証を実施、改修やバグレポート本家bugzillaに上げる
- www.php.net (公式ミラー)のサーバ提供

# PHP採用までの流れとオープンソース

YAHOO! JAPAN

# 独自ソフトからオープンソースへ






YAHOO! JAPAN

# 初期の独自ソフトウェア

- Filo Server (ファイロサーバ)
  - Yahoo!  Incの共同創業者の1人が作ったwebサーバ
- フラットバイナリファイルDB
  - MySQLのようなリレーショナルデータベースでなく、DBM的な、PHPでいう連想配列のようなもの
- yScript
  - Yahoo!独自で開発されたweb言語の総称

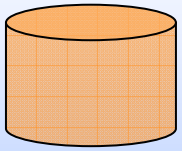
# Yahoo!スタート当初(1994)

- Yahoo! Incは1994年にスタート
- 当初は、独自のC/C++ソフトウェアで構成

**webサーバ**

“Filo Server”

**DB**



フラットファイル

**WEB言語**


yScript



YAHOO! JAPAN

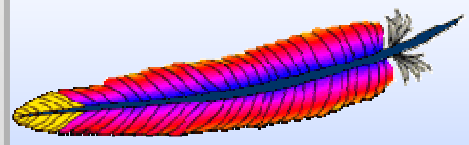
# オープンソースの利用にスイッチ

- オープンソースが盛り上がってきた
- 時代の流れにのり、オープンソースのソフトウェアを採用し始める

# 1996年

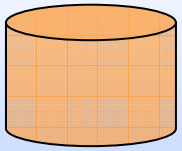
- webサーバの"filo server"をApacheに移行

**webサーバ**



Apache

**DB**



フラットファイル

**WEB言語**

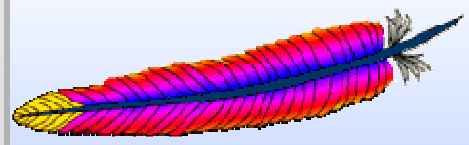

yScript



YAHOO! JAPAN

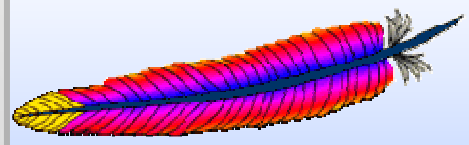
# 1999年

- webサーバの"filo server"をApacheに切替
- フラットバイナリファイルDBをMySQLに切替

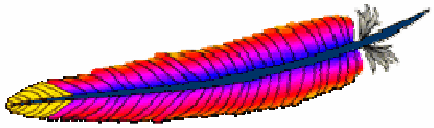
| webサーバ | DB | WEB言語 |
| --- | --- | --- |
|  |  |  |
| Apache | MySQL | yScript |

# 2002年9月

- webサーバの"filo server"をApacheに切替
- フラットバイナリファイルDBをMySQLに切替
- **Web言語をPHPに切替**

| webサーバ | DB | WEB言語 |
| --- | --- | --- |
|  |  |  |
| Apache | MySQL | PHP |

# オープンソースの利用



- **F**reeBSD / **A**pache / **M**ySQL / **P**HP
  - OSがLinuxの代わりにFreeBSDが入っているだけでLAMP的な普通の環境だと思われるかも
- だが、全てをオープンソースのみで構築しているわけではない



YAHOO! JAPAN

# オープンソース以外の利用

- 用途に応じて有償ソフトウェアも利用する
  - データベースにOracleなど
- オープンソースも手を入れる
  - 素で使わずにパッチを大なり小なり入れているものも多い。FreeBSDやApache、PHP等
- 独自技術も豊富

# オープンソース×独自ソフトウェア

- オープンソースと独自技術の親和性を高め相乗効果を得ている
- 例えば、
  - 独自技術をPHPから呼び出せるようにして扱いやすく

# 日本では

- 構成はYahoo! Incと基本的に同じ
- 2004年頃からPHP（当時4.3)に切替を開始
- 現時点では、フロントエンドはほぼPHP
  - 一部はApaheコンテンツハンドラ

# なぜPHPを採用したか

YAHOO! JAPAN

# PHPを選択した流れ

- 2001年10月: 3つの独自言語 (→yScript)
  - それぞれをメンテナンスするのが大変
  - 機能が限られていた (サブルーチンが無かった！)
- 新しいものを採用するため調査を実施
  - 機能やパフォーマンスを比較
  - 自社開発 / 有償ソフト / オープンソース
- 2002年5月にPHPを選択

# 言語を選択するための条件

- 高いパフォーマンス
- 堅牢で安全な環境 (sand-boxed)
- C/C++による拡張
- FreeBSD上での動作
- 動的コンパイル言語
- 国際化をサポート

FreeBSD®



YAHOO! JAPAN

# なぜPHPを選択したか

- Webスクリプティング向けにデザイン
  - HTMLの中にコードが書ける
- 高いパフォーマンス
- 大きなオープンソースコミュニティ
- ライブラリの充実、拡張性がある
- ツールが充実: IDE, デバッガ, プロファイラ
- トレーニングコストが低い

# トレーニングコストの低さが大事

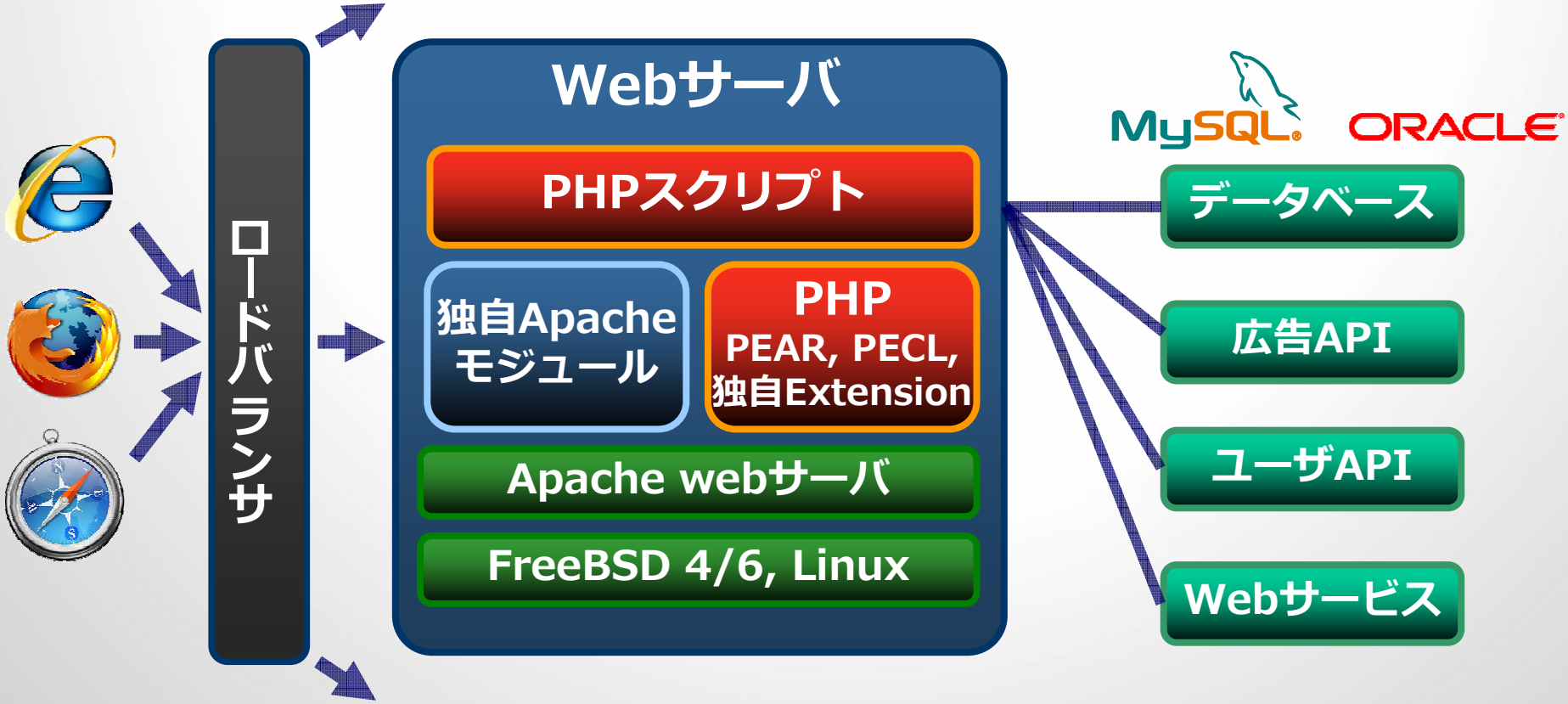
- PHPは敷居が低い
  - もともと使える人も多い、知らなくてもプログラマならそれなりに使える
  - 関数が豊富、ドキュメントも充実
- PHPを使うのはプログラマだけではない
  - HTML・CSSと、ちょっとPHPを書く人とか
  - 画面の出し分けやデータの整形など、補助的にPHPを使う

# Yahoo!における現在のPHP

# トピックス

- サーバ構成
- アプリケーションレイアウト
- 依存関係の管理
- PHP5
- セキュリティ
- パフォーマンス
- 国際化対応
- フレームワーク、CMS

# サーバ構成






YAHOO! JAPAN

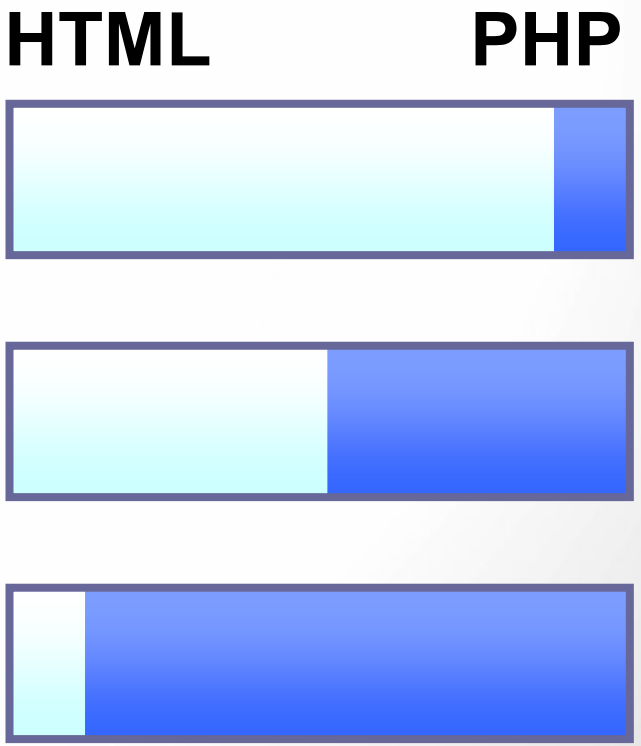
# アプリケーションレイアウト

**HTMLテンプレート**
/usr/local/share/htdocs/*.php

**テンプレートヘルパー**
/usr/local/share/htdocs/*.php

**ビジネスロジック**
/usr/local/share/pear/*.php

**データアクセス、ネットワーク、暗号化等コアロジック**




*The no-framework PHP MVC framework – Rasmus' Toys Page*
*http://toys.lerdorf.com/archives/38-The-no-framework-PHP-MVC-framework.html*

# 依存関係の管理

- PHPは必要最低限の構成でビルド
- 全ての拡張(Extension)はデフォルトですべて無効にして、必要最低限のものだけenableにする

```
./configure --disable-all
```

# 依存関係の管理

- 拡張モジュールは、共有モジュール(Self-contained Extension) としてビルド
- PHP本体とは別にインストール
  - 独自のパッケージ管理ツールでバイナリ(.so)で配布
- 依存関係のあるパッケージはインストール時に合わせてインストール

# 依存関係の管理

- 共有ライブラリにすることで
  - ○不必要な依存を避ける
  - ○メモリ利用量を下げる
  - ○構成変更で再コンパイルが不要
    - インストールはファイルを配置して、php.iniに extension=foo.so と1行書くだけ
  - ×起動時に若干オーバーヘッド

# 依存関係の管理

- Yahoo!では、多種多様のサービスがあり、最大公約数的な環境というものは作れない
- ベースを用意し、ニーズに合わせてパーツ(モジュール)を選択、最適な環境を組み立てる

# PHP5

PHP4からPHP5への移行

YAHOO! JAPAN

# PHP4の終了アナウンス

## PHP 4 end of life announcement

*[13-Jul-2007]* Today it is exactly three years ago since PHP 5 has been released. In those three years it has seen many improvements over PHP 4. PHP 5 is fast, stable & production-ready and as PHP 6 is on the way, PHP 4 will be discontinued.

The PHP development team hereby announces that support for PHP 4 will continue until the end of this year only. After 2007-12-31 there will be no more releases of PHP 4.4. We will continue to make critical security fixes available on a case-by-case basis until 2008-08-08. Please use the rest of this year to make your application suitable to run on PHP 5.

For documentation on migration for PHP 4 to PHP 5, we would like to point you to our migration guide. There is additional information available in the PHP 5.0 to PHP 5.1 and PHP 5.1 to PHP 5.2 migration guides as well.

*http://www.php.net/archive/2007.php*



YAHOO! JAPAN

# PHP4は2007年末サポート終了

- PHP4系のリリースはPHP4.4で終了
- バグフィックス等は2007年末まで
- 重大なセキュリティ修正は2008年8月8日まで




YAHOO! JAPAN

# Yahoo!も今年でPHP4は終了

- 現在では、PHP4.4系と5.1・5.2が混在
- 日本では、現時点でメインが4.4系
- 2007年末までに5系へ移行する
  - –2008年には基本全てPHP5.2以降へ

# セキュリティ

セキュリティ対策 / PHP設定

フィルタ関数 / CSRF対策

YAHOO! JAPAN

# セキュリティ対策

- セキュリティ規則、開発ルールによる予防
  - 個人情報の取り扱いルール
  - 実装方法の指定
  - 脆弱性スキャナによるチェック
  - チェックリストでのリリース前確認
- プラットフォームの整備 (Apache, PHPなど)
  - フィルタ拡張などの機能提供
  - フェイルセーフの視点からの標準設定

# Scanmus

- Yahoo! 社内で開発された XSS (Cross Site Scripting) 脆弱性検出ツール
- Rasmus Lerdorfが開発
  - Rasmus の ”mus” をとって "Scanmus"

ウェブセキュリティ最前線--「パラノイド」:それはヤフーのセキュリティチーム
http://japan.cnet.com/special/story/0,2000056049,20352554-4,00.htm



YAHOO! JAPAN

# php.ini設定

- **open_basedir を設定**
  - /etc/passwdなどを読まれる攻撃を避ける
- **allow_url_fopen, allow_url_include = off**
  - リモートファイルインクルード、オープンプロキシ攻撃を避ける
  - libcurlを代わりに利用
- **safe_mode = Off**
  - 共有ホスティングの環境でのセキュリティ対策

# php.ini設定

- **display_errors = Off**
  - エラーをブラウザ画面に表示しない
  - 但し、log_errors = On で、エラーはログに残す
- **error_reporting = E_ALL (|E_STRICT)**
  - 全てのエラーと警告を出力
  - "Undefined variable"のNOTICEなどが出ないようにコードを書く

# ユーザ入力のフィルタリング

- XSS・SQLインジェクション攻撃
  - ブラウザから来るデータを信頼してはならない
- input_filterフックを利用
  - ユーザから送られたデータは一旦全てサニタイズ
- フィルタ拡張を使う
  - PHP5.2.0より標準で有効 (それ以前はPECLで)
  - Yahoo!のフィルタリングライブラリから派生
  - 社内では独自のものを利用しています

# フィルタ拡張

- filter_default = special_chars
  - デフォルトフィルタを設定、入力を自動でフィルタリングする
  - この場合、特殊文字(< > 等)をHTMLエンティティに変換
- 必要な場合に限り、適切なフィルタを使ってfilter_input, filter_input_array関数でデータを取得

# フィルタ拡張

- filter_input関数でのデータ取得例
- 何処でどのフィルタでデータを取得しているか、ソースの検索がしやすい

```
// タグ文字を除去
$body = filter_input(INPUT_POST, 'body',
        FILTER_SANITIZE_STRING);
// メールアドレスチェック
$mail = filter_input(INPUT_POST, 'mail',
        FILTER_VALIDATE_EMAIL);
```

# フィルタ拡張

- フィルタ拡張は万能ではない
- 例えば、stringやspecial_charsフィルタは、属性インジェクションに無力

```
<a href="<?php echo $_GET['foo'] ?>">
```

▼

```
<a href="javascript:alert(
                    &#039;XSS!&#039;);">
```

- 上の例だと、FILTER_VALIDATE_URLでURLかどうか検証する

# CSRF(XSRF)脆弱性対策

- 投稿や削除などの処理のURLにユーザを誘導し、意図しないコマンドを実行させる攻撃
  - 踏んでしまったユーザの権限で動作する
- **Crumbを利用する**
  - 入力ページや確認ページで、ユーザ固有のダイジェストを作る

*Rasmus Lerdorf: Bigger and Faster – OSCON 2007*
*http://talks.php.net/show/oscon07/25/*

# crumbの生成

```
<?php
$user   = 'rasmus';
$secret = 'foo';
$crumb  = sha1($user . $secret);
?>
<form action="post.php">
  <input type="hidden" name="crumb"
         value="<?php echo $crumb?>"/>
  <input type="text" name="foo"/>
  <input type="submit" name="submit"/>
</form>
```

# crumbの検証

```php
<?php
$user   = 'rasmus';
$secret = 'foo';
$crumb  = sha1($user . $secret);

if ($_POST['crumb'] !== $crumb)) {

  echo "XSRF Detected";
  exit;
}

// CSRFじゃないので、処理続行
?>
```

# パフォーマンス

Opcode Cache / Profiler

PHP Extension / Session

YAHOO! JAPAN

# Opcode Cacheの利用

- PHPは実行時に、スクリプトを中間言語(Opcode)に変換してから、実行する
- 解析したopcodeを共有メモリにキャッシュ
  - ソースコードに手を入れることがなく、手っ取り早く性能を向上させることができる
  - また、実行時最適化もされる

# Opcode Cacheの利用

- 各種キャッシュモジュール
  - APC
  - eAccelerator
  - XCache など
- Yahoo!ではAPCを採用
  - 安定し、よく検証されている
  - Rasmusが開発に携わっている点も大きい

# プロファイラ・デバッガ

- ボトルネックとなっている箇所の洗い出しにプロファイラを利用することが多い
- Yahoo!では、xdebugをメインで利用
- WinCacheGrind やKCacheGrind で可視化

# PHPエクステンションの利用

- C/C++で書いて、PHPに機能を組み込む
- Yahoo!では社内で利用するエクステンションを開発している
- Y!のエンジニアはC/C++に慣れている
  - PHP以前は、よくApacheのDSOモジュールなどを書いていた
  - C++でバックエンドのアプリを書くこともある

# PHPエクステンションの利用

○メリット

- 高速に動作する
- C/C++ライブラリへアクセスできる

×デメリット

- 開発が面倒（コーディング→コンパイル→リンク→デバッグ）
- メモリリーク、リソースリークなどの問題
- 書き方が分からなかったり、間違えていたり…

# PHP埋め込み (PHP embed)

- webアプリではないですが
- C/C++言語のアプリケーションに、PHPを埋め込む
- PHPの柔軟さ、手軽さをCのアプリにも
  - テンプレートプロセッサ代わりにしたり、複雑で変更が激しいところに利用する

# 実際の本番サーバの拡張導入例

- mbstringやapcなど標準の拡張 4つ
- Yahoo!独自の拡張 9つ
  - 暗号化、広告挿入等、独自技術のwrapper
- Y!J社内の他サービス連携用の拡張 2つ
  - API呼び出し用
- 自サービス専用の拡張 5つ

**20拡張**

# セッションの利用を避ける

- 基本ステートレスで構築
  - 複数台で運用する場合などにネック
- Cookieを利用し、よく利用するデータ(ユーザIDや、タイムスタンプ等)を暗号化、署名をしたフォーマットで格納する
- 画面遷移に必要なデータはPOSTで引き継ぐ
  - CSRF対策にはcrumbを利用

# 国際化対応

# Yahoo! Incの国際化対応

- 多言語、多地域の対応を考慮して開発
  - 日本は特殊なのですが...
- たとえば
  - 文字コードはUTF-8
  - 翻訳、ローカライズしやすい実装

# 国際化(i18n)対応

- テンプレート管理ツール r3 を開発
  - –「テンプレートエンジン」ではない
  - – 地域ごとのPHPテンプレートを生成
- 「継承」と「掛け合わせ」の概念で、多言語、多地域のサイトのテンプレートを効率よく生成
- 異なる言語やマーケット、利用方法に対しUIをカスタマイズ、翻訳することができる

# テンプレート管理ツール r3

- 今年4月にオープンソースでβ公開
  - 7月末にversion1.0.0がリリース
- sourceforgeからダウンロード、PEARでインストール
  - http://sourceforge.net/projects/rthree
  - http://sourceforge.net/projects/stickleback
- 日本語の解説あります
  - http://www.slideshare.net/ogi/r3

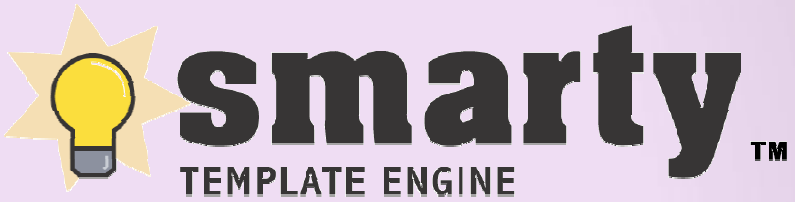
# フレームワーク・CMS

YAHOO! JAPAN

# フレームワークの利用

- オープンソースのフレームワークはYahoo!ではあまり使われていない
  - 汎用的ゆえにパフォーマンスが劣る
- スクリプトレベルの実装方法(フレームワークの利用など)はエンジニアに委ねられている
  - 悪く言うと社内で結構ばらつきがある
- 内製のフレームワークやCMS
  - プロダクトに特化させた形のものも多い

# Smarty




- テンプレートエンジンSmartyはYahoo!では推奨されていない
- 実行にかかるオーバーヘッドがある
  - テンプレートはキャッシュされるが、実行にかかるオーバーヘッドは減らない
- Yahoo!レベルのトラフィックでは、このオーバーヘッドが無視できなくなる



YAHOO! JAPAN

# symfony

- オープンソースのPHPフレームワーク
- 最近、symfonyを利用しはじめた
- Yahoo!用にカスタマイズし導入
  - 今後社内導入事例が増えていくのでは

# symfonyを利用しているサービス

- Yahoo! Bookmarks

  http://bookmarks.yahoo.com
- del.icio.us

  http://del.icio.us
- Yahoo! Answers

  http://answers.yahoo.com





YAHOO! JAPAN

# symfonyの導入

そのまま使っているもの

- コアとなるMVCアーキテクチャ
- 設定システム
- View層
- ユーザセキュリティ

# symfonyの導入

独自のものに置き換えたもの

- Model層とORマッピング
  - Yahoo!のニーズ、スケールにはマッチしない
- 設定システム
  - より柔軟に拡張
- 国際化
  - テンプレート管理ツール r3 との統合

# CakePHP

- 評価中の模様
- プロダクションへの投入はされていない (今のところ予定も無し)

# Drupal

- オープンソースのCMS
- 評価中。本番環境での採用はまだ行われていない
- 社内のID認証システムに対応させ、社内サイトのCMSとして利用されている事例は多い

# まとめ

# まとめ

- Yahoo!がPHPを採用して5年
- いろいろカスタマイズして使っています
  - セキュリティ対策
  - パフォーマンスチューン
- PHP Extensionを多用
- フレームワークは今はあまり使ってない
  - 一部symfonyを使ってます

# ありがとうございました！

YAHOO! JAPAN DEVELOPER NETWORK

http://developer.yahoo.co.jp

*Original slides from Federico Feroldi (Senior Architect Engineer, Yahoo! Europe) speech at PHPDay 2007 in Italy.*
*Special thanks to Federico for the kind permission.*
*http://www.phpday.it/site/phpday-2007/calendario-conferenze/canale-enterprise/federico-feroldi-php-in-yahoo/*



YAHOO! JAPAN

LIFE ENGINE™