





# **La professionnalisation du monde PHP.**

# HYPertext PREROCessor

Créé en 1994 par Ramus Lerdorf

Très permissif

Longtemps connu comme un langage de "bidouilleur" dépourvu de standards.

Conséquences :

- Pas de collaboration entre projets.

- Reprise de code difficile

- Mauvaise image

***Comment PHP est-il parvenu à maturité ?***

## Les acteurs



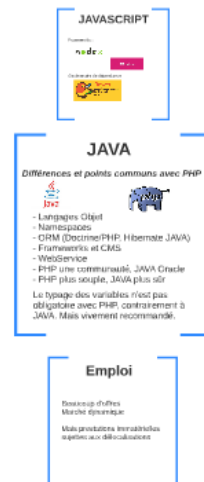
## Les outils



## Quand les Frameworks & CMS Collaborent



## Comparatif avec un autre monde





# Les acteurs

## Un peu d'histoire.

1994 - 1ère version de PHP

2004 - PHP5 permet la programmation orienté objet grâce zend engine

2009 - PHP5.3 -> Namespace



## Normes et organismes

### En général

#### PHPGROUP

PHP est  
une technologie ouverte, libre, collaborative  
pour développer web et mobile

#### PHP-FIG

PHP-FIG est une communauté  
d'entraide et de partage de connaissances  
pour les développeurs PHP

### En France

Association Française des  
utilisateurs de PHP

Culture Conferences

## Frameworks & CMS

- Démocratise chez PHP l'architecture MVC
- Leur notoriété en fait des acteurs majeurs
- Adoptent et font donc appliquer les PSR

"Symfony suit les standards définis par les documents PSR-0, PSR-1 et PSR-2."

"Personally I prefer some mix of Drupal and PSR-2 coding standards, but I need to follow common and modern trends in the PHP world."

"Zend\Loader\StandardAutoloader is designed as a PSR-0-compliant autoloader."

# Un peu d'histoire.

1994 - 1ère version de PHP

2004 - PHP5 permet la programmation orienté objet grâce zend engine

2009 - PHP5.3 -> Namespace

```
// root/fruit.php
// namespace
class Fruit
{
    public function afficherFruit($fruit)
    {
        echo $fruit;
    }
}

// root/namespace
namespace Vroot
class Fruit
{
    public function afficherFruit($fruit)
    {
        echo $fruit;
    }
}

// root/fruit.php
include "root/fruit.php"; //collision on a 2 classes fruit
class Fruit
{
    public function afficherMyfruit($fruit)
    {
        $foo = new Fruit();
        return $foo->afficherfruit($fruit);
    }
}
$foo = new Fruit();
echo $foo->afficherMyfruit("bonjour");

//collision
namespace root\foo;
use Vroot\Fruit as myfruit; // alias pour de collision
class Fruit
{
    public function afficherMyfruit($fruit)
    {
        $foo = new myfruit();
        return $foo->afficherfruit($fruit);
    }
}
$foo = new Fruit();
echo $foo->afficherMyfruit("bonjour");
```

```

2 // root/fruit.php
3 // sans namespace
4 class Fruit
5 {
6     public function afficherFruit($fruit)
7     {
8         echo $fruit;
9     }
10 }
11
12 // avec namespace
13 namespace \root
14 class Fruit
15 {
16     public function afficherFruit($fruit)
17     {
18         echo $fruit;
19     }
20 }
21
22 // root/foo/fruit.php
23 include "/root/fruit.php"; //collision on a 2 classes Fruit
24 class Fruit
25 {
26     public function afficherMyFruit($fruit)
27     {
28         $foo = new Fruit();
29         return $foo->afficherFruit($fruit);
30     }
31 }
32 $bar = new Fruit();
33 echo $bar->afficherMyFruit('banane');
34
35 //solution
36 namespace root\foo;
37 use \root\Fruit as myFruit; // Alias pas de collision
38 class Fruit
39 {
40     public function afficherMyFruit($fruit)
41     {
42         $foo = new myFruit();
43         return $foo->afficherFruit($fruit);
44     }
45 }
46 $bar = new Fruit();
47 echo $bar->afficherMyFruit('banane');

```

# Normes et organismes

## ***En général***

### **PHPGROUP**

PHP.net

Une communauté composé d'une équipe principal

Tout développeur peut y contribuer

### **PHP-FIG**

*Framework Interop Group*



### **PSR**

Propose « Standards Recommendation  
Recommandation « standard »  
Codage pour PSR1 et 2  
Chargement pour PSR0 et 4

## **En France**

Association Française des  
utilisateurs de PHP

Forums

Conférences

**afup**



# ***En général***

## **PHPGROUP**

PHP.net

Une communauté composé d'une équipe principal

Tout développeur peut y contribuer

## **PHP-FIG**

*Framework Interop Group*



### **PSR**

Propose a Standards Recommendation

Recommandation ≠ standard

Codage pour PSR1 et 2

Chargement pour PSR0 et 4

Associa  
utilisateur

Forums

# ***En général***

## ***PHPGROUP***

PHP.net

Une communauté composé d'une équipe principal

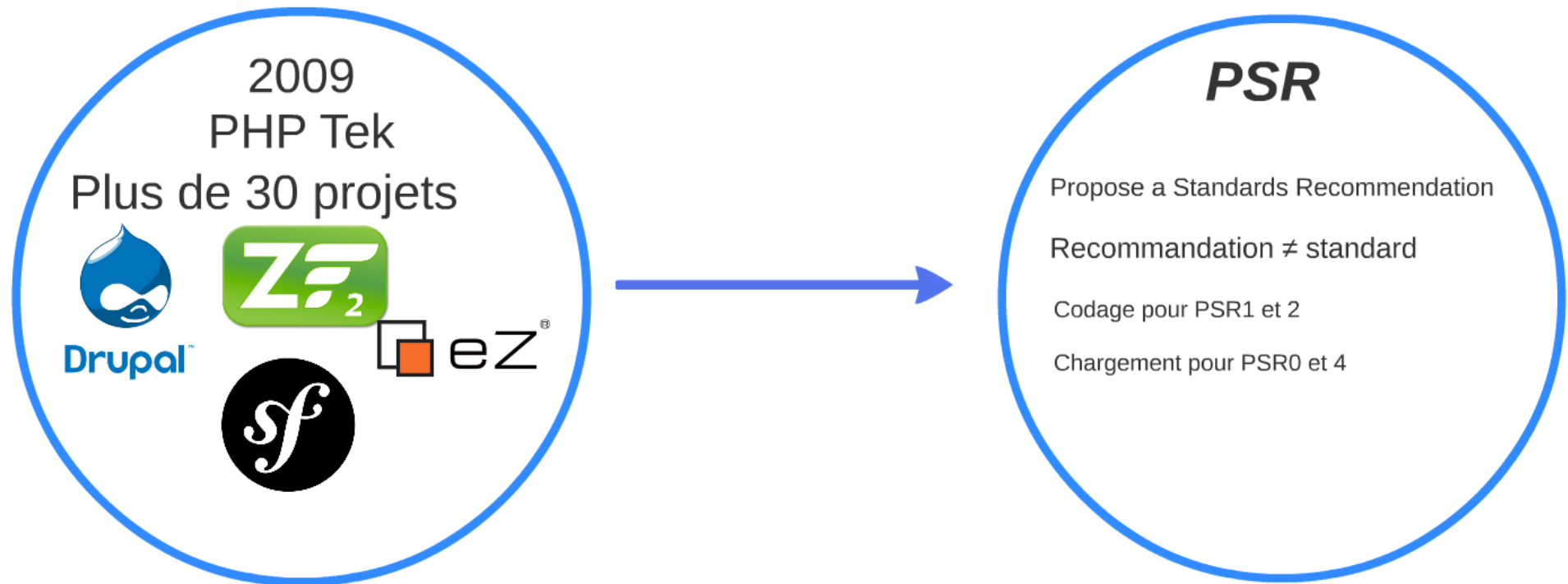
Tout développeur peut y contribuer

## ***PHP-FIG***

***Framework Interop Group***

# PHP-FIG

*Framework Interop Group*



2009  
PHP Tek  
Plus de 30 projets



Drupal™



# ***PSR***

Propose a Standards Recommendation

Recommendation  $\neq$  standard

Codage pour PSR1 et 2

Chargement pour PSR0 et 4



# En France

Association Française des  
utilisateurs de PHP

Forums

Conférences



# Frameworks & CMS

- Démocratise chez PHP l'architecture MVC
- Leur notoriété en fait des acteurs majeurs
- Adoptent et font donc appliquer les PSR

*"Symfony suit les standards définis par les documents PSR-0, PSR-1 et PSR-2."*

*"Personally I prefer some mix of Drupal and PSR-2 coding standards, but I need to follow common and modern trends in the PHP world."*

*"Zend\Loader\StandardAutoloader is designed as a PSR-0-compliant autoloader."*

# Les outils

## Travail d'équipe



## Gestionnaire de dépendances

**Composer**  
The PHP package archivist.  
Dépôt principal de paquets pour Composer

Depuis 2012  
+ 30 milles packages  
Près de 300 millions de téléchargements

**Composer**  
Gestionnaire de dépendance  
Télécharge les paquets qui sont sur Packagist

## Gains de productivité

- Plus de 40 Frameworks
  - Plus de 20 CMS
-   
- Symfony ZF1 Magento
- Open source
  - Gain de temps en développement
  - Réduction des coût

## Programmer plus vite

-   
- NetBeans Eclipse Sublime Text
- Naviguer dans le code
  - Modules (Git, frameworks, etc)
  - Formatage automatique du code

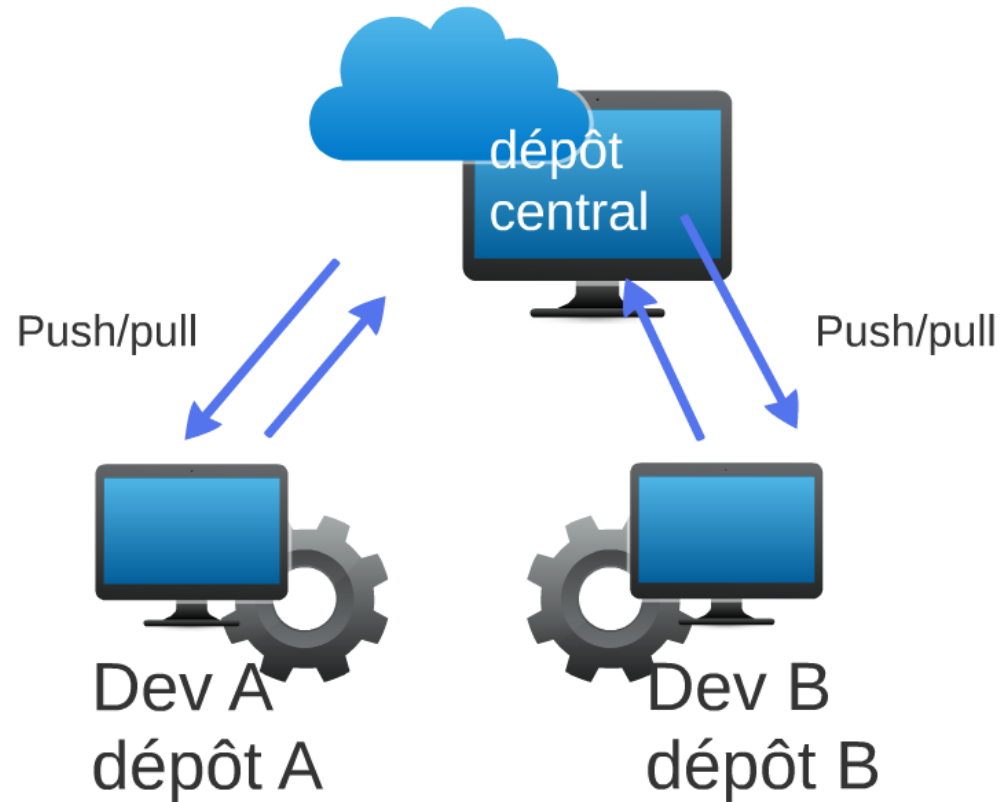


# Travail d'équipe



# git

Logiciel de versionning  
décentralisé



# Gestionnaire de dépendances



**Packagist**

The PHP package archivist.

Dépôt principal de  
paquets pour  
Composer

Depuis 2012

+ 30 milles packages

Près de 300 millions de téléchargements



Gestionnaire de dépendance

Télécharge les paquets qui  
sont sur Packagist

# ***Gains de productivité***

- Plus de 40 Frameworks
- Plus de 20 CMS



Symfony



Magento



Drupal™

- Open source
- Gain de temps en développement
- Réduction des coût

# Programmer plus vite



- Naviguer dans le code
- Modules (Git, frameworks, etc)
- Formatage automatique du code



De nouvelles portes vont s'ouvrir au monde PHP

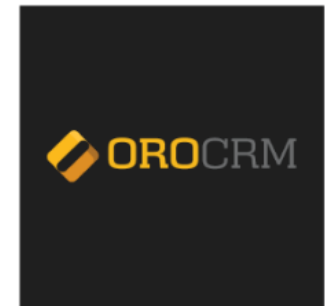
- Les Namespaces
- Le FIG et ses recommandations PSR
- Composer & packagist
- Les Licences

Year	Percentage
2007	13.2%
2008	10.8%
2009	14.5%
2010	12.1%

# Qui partage du code



*Sonata*



*Composants*



*Full stack*

# *Plus loin*

Ces CMS partagent eux même du code

Sonata partage déjà du code

Drupal8 produira des composants compatible avec d'autres projets

On peut donc imaginer Symfony2, ou encore Zend2, utiliser du code produit par d'autres projets.

De nouvelles portes vont s'ouvrir au monde PHP

# Comment cela a il été rendu possible ?

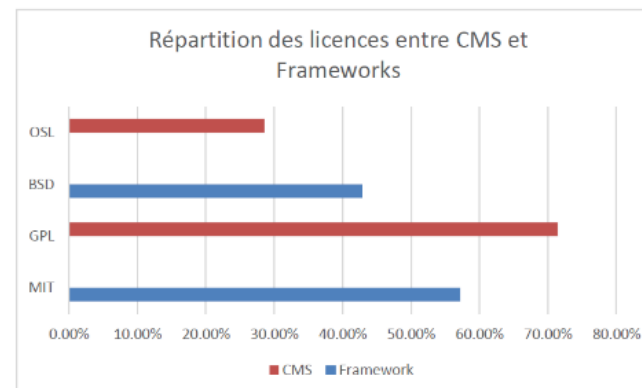
- Les Namespaces
- Le FIG et ses recommandations PSR
- Composer & packagist

## - Les Licences

Projet	MIT	GPL	BSD	OSL
SF2	1			
ZEND2			1	
DRUPAL		1		
EZPblish		1		
Codeigniter			1	
Cakephp	1			
Yii			1	
Magento				1
Prestashop				1
Wordpress		1		
Joomla		1		
FUEL PHP	1			
Typo3		1		
Laravel	1			

Frameworks:  
Licences permissive

CMS:  
Licences virales





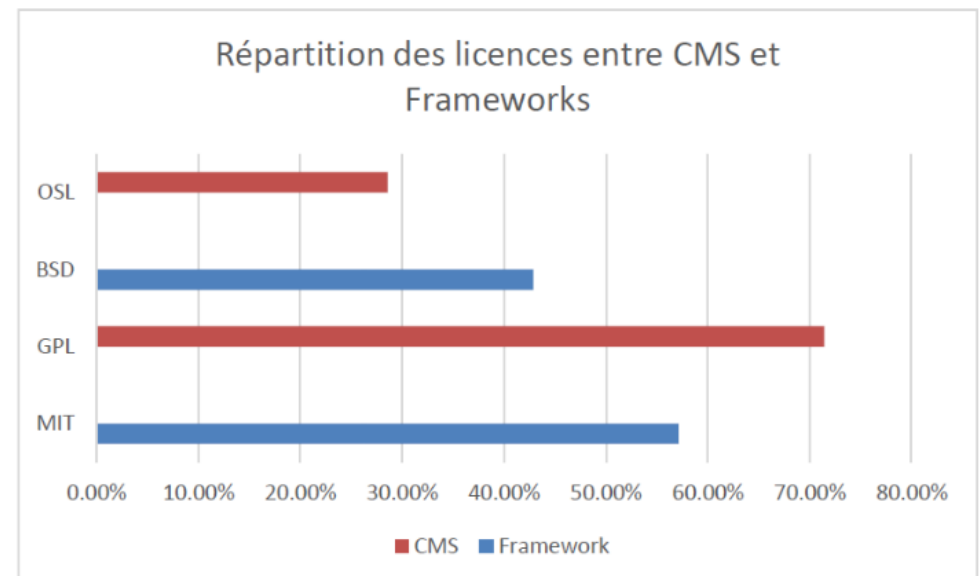
# - Composer & packagist

## - Les Licences

Frameworks:  
Licences permissive

CMS:  
Licences virales

Projet	MIT	GPL	BSD	OSL
SF2	1			
ZEND2			1	
DRUPAL		1		
EZPblish		1		
Codeigniter			1	
Cakephp	1			
Yii			1	
Magento				1
Prestashop				1
Wordpress		1		
Joomla		1		
FUEL PHP	1			
Typo3		1		
Laravel	1			



# Comparatif avec un autre monde

## JAVASCRIPT

Frameworks :



Gestionnaire de dépendance



## JAVA

*Différences et points communs avec PHP*



- Langages Objet
- Namespaces
- ORM (Doctrine/PHP, Hibernate JAVA)
- Frameworks et CMS
- Webservice
- PHP une communauté, JAVA Oracle
- PHP plus souple, JAVA plus sûr

Le typage des variables n'est pas obligatoire avec PHP, contrairement à JAVA. Mais vivement recommandé.

## Emploi

Beaucoup d'offres  
Marché dynamique

Mais prestations immatérielles  
sujettes aux délocalisations

# JAVASCRIPT

Frameworks :

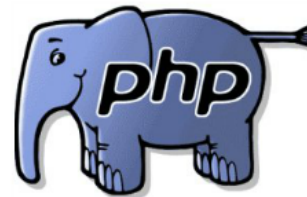


Gestionnaire de dépendance



# JAVA

## *Différences et points communs avec PHP*



- Langages Objet
- Namespaces
- ORM (Doctrine/PHP, Hibernate JAVA)
- Frameworks et CMS
- WebService
- PHP une communauté, JAVA Oracle
- PHP plus souple, JAVA plus sûr

Le typage des variables n'est pas obligatoire avec PHP, contrairement à JAVA. Mais vivement recommandé.

# Emploi

Beaucoup d'offres  
Marché dynamique

Mais prestations immatérielles  
sujettes aux délocalisations

# *Conclusion*



Une image qui s'améliore

Plus de 80% des sites web

Un avenir prometteur

PHP est mature

Peut mûrir d'avantage

