

# Bilan de l'utilisation du Framework Symfony2

pour la refonte complète de  
2 applications Web

# Contexte: les applis web au labo

- Importante base classique PHP-MySQL
  - Applis publiques: php\_myresa, php\_conges, glpi
  - Applis "maison": 2002-2007
    - » annuaire interne, gestion thèses/stage, réservations, missions
- Applis difficiles à maintenir et faire évoluer
  - => Utiliser un framework pour améliorer la situation
    - JRES 2011: <https://2011.jres.org/archives/184/index.htm>
    - « La création et l'utilisation des **frameworks web** est issue du besoin de **développer rapidement des applications**, en favorisant la réutilisation de code. Cette philosophie « Don't Repeat Yourself » (DRY) doit **faciliter la maintenance, le test et les évolutions** d'une application »

# les Frameworks Web

- « cadriciels »: cadres de travail + composants logiciels
  - Boite à outils + briques de base + Plan-type
  - Offrent des solutions éprouvées pour aider au dev. d'une appli web
  - Apparus vers 2005
- Plusieurs technologies:
  - PHP: Symfony, Zend, CakePHP
  - Python: Django
  - Java: Struts
  - Ruby: Rails
- Avec des points communs:
  - Architecture type MVC (modèle/vue/contrôleur)
    - Place centrale du modèle de données
    - Séparation claire des composants dédiés à la visu et à l'interaction
  - Automatisation de la persistance des données (ORM)
  - Composants pour faciliter les tâches récurrentes:
    - » Formulaires, Routage, Authentification

# Pourquoi, Comment Symfony2

- C'est un framework qui utilise la technologie PHP
  - il en existe d'autres: Zend, CakePHP,...
- Recherche de synergies au sein in2p3:
  - École in2p3 (octobre 2008): « Développer une application web »
  - Appli *cc-Absence* développée au CC-in2p3
- 1<sup>er</sup> essai: Symfony v1.4
  - auto apprentissage: tutoriel 'Jobeet' (Aout 2010)
  - Prototype pour appli de gestion des thèses et stages
  - Resté sans suite ...
- Symfony v2
  - Présentation positive aux JRES en Novembre 2011
  - Premiers essais encourageants:
    - structuration plus claire, plus grande modularité
    - moteur de templates pour faciliter Affichage (Twig)
- ... prêts pour évaluer Symfoyl2 sur une vraie appli.

# Appli No 1:

- Affichage et Gestion des Nouvelles sur la Page d'Accueil du LAL
  - 800 News, équipe de rédaction: 4 personnes
    - techno: classique SPIP-kit CNRS + code « maison » PHP-MySQL
- Demande explicite de la Direction pour renouveler cette page:
  - Catégories de News, organisation de l'affichage, etc...
  - Mais le cahier des charges reste très approximatif
    - ... il faudra s'attendre à des réajustements
- Partie Gestion et Rédaction:
  - appli développée en 2004, écrite en Perl
  - expertise Perl disparue du labo
  - => il faut refondre le code
  - ... opportunité d'un projet concret mais simple avec Symfony2

# App No1: prototype, évaluation

- Une réalisation assez rapide (2 mois) :
  - Point de départ: tutoriel "Création d'un blog avec Symfony2 »
    - <http://keiruaprod.fr/symblog-fr/>
  - Appli classique de Gestion et de Rédaction de News
    - Proche du tutoriel. Entités: Article, Documents, Tags
    - Formulaire de création, affichage de listes, etc..
    - Grande efficacité du langage de template Twig (>> PHP)
- Les éléments pour un prototype convaincant:
  - Des données réelles:
    - Reverse engineering via Symfony
      - base MySQL en production injectée dans un premier modèle de données
    - Evolution de ce modèle de données facilitée par persistance automatique
  - Une présentation «professionnelle» : style CSS *Bootstrap* de Twitter
    - Contribue fortement à la crédibilité
    - NB: complètement indépendant de Symfony, utilisable en simple HTML
- Evaluation positive: on continue avec Symfony

# App No1: prototype

Anonymous - Login

A propos (README) +

## Les unes

Tri par date, on commence par les plus lointaines dans le futur

← Previous 1 2 3 4 5 6 7 8 ... 198 Next →

titre	Date	hot	brouillon	url	doc-image
<a href="#">Journées Jeunes Chercheurs 2012</a>	07/09/2012	1		<a href="#">url</a>	x
<a href="#">Higgs Hunting 2012 workshop</a>	18/07/2012			<a href="#">url</a>	
<a href="#">Atlas Higgs to ZZ workshop</a>	03/07/2012	1		<a href="#">url</a>	

# App No1: cycle rapide de livraisons

- On se concentre sur la présentation de la page d'accueil
  - Suite du projet prévue sur 4 mois
  - On s'attend à de nombreux changements d'orientation
- **Le prototype évolue en une plateforme de test et démonstrations:**
- de plugins JavaScript et JQuery

- et de maquettes
- présentées et discutées en Comité Web
  - Livraisons et Revues fréquentes
  - 2 semaines
- Revues suivies de réajustements
  - Livraison -> revue -> redéfinitions

... développement de type « agile »





# Appli No2:

## Gestion des demandes de Missions

- Basée sur l'utilisation de formulaires
- 1 appli originelle clonée en 10 applis +/- spécifiques
  - ensemble assez critique: >1000 demandes de mission/an
  - Code PHP-MySQL + library Pear (Formulaires)
- **Très gros problème de maintenance**
- Nous décidons de reprendre ces applis avec Symfony

## Cahier des charges:

- Architecture: 1 seule appli (maintenance, évolution)
- Mode admin: ajouter villes, pays, monnaies, etc..
- Mode user: «moderniser» et unifier les interfaces

# App No2: les Formulaires

- Ce sont les composants essentiels de cette appli.
- Réalisation très rapide d'un prototype avec une «interface simple»:
  - 2 semaines
  - Une appli unique qui remplace les 10 anciennes
  - Tout le code consacré à la persistance est supprimé
- Les composants « Formulaires » de Symfony sont très efficaces
  - et bien documentés.

# Ex. formulaire demande mission

Affichage d'un formulaire classique via Symfony

Entité «Voyage» -> Objet Formulaire -> «HTML»

## Trajet - Aller :

\* Mode de voyage :

\* Pays départ :

\* Ville départ :

\* Date départ :

\* Heure départ :

\* Pays d'arrivée :

\* Ville arrivée :

\* Date arrivée :

\* Heure arrivée :

## Trajet - Retour :

\* Mode de voyage :

# Formulaires dynamiques

Changement du mode de voyage: train -> voiture  
Modifier une partie de la page: JavaScript

## • Trajet - Aller :

\* Mode de voyage :

\* J'ai pris connaissance de l'article ci-dessus:       \* Type voiture :  Vehicule pour convenance personnelle avec remboursement base train  
 Vehicule personnel pour necessite de service  
 Vehicule administratif  
 Vehicule tierce-personne

\* Kilometrage :       \* Puissance :

\* Immatriculation :

\* Pays départ :       \* Ville départ :

# Formulaires dynamiques (suite)

Validation: «HTML» -> Objet Formulaire -> Entité Voyage

Mettre à jour les données de l'entité

Plusieurs stratégies possibles, pas facile de choisir

## • Trajet - Aller :

\* Mode de voyage :

Devrez-vous faire plusieurs aller-retours?

Oui ou Non ?  Oui  
 Non

Nombre de voyages

\* Pays départ :

\* Ville départ :

# Formulaires dynamiques: bilan

- Documentation insuffisante sur le site officiel
  - la situation évolue:
  - <http://webmozarts.com/2012/03/06/symfony2-form-architecture/>
- Dans la communauté:
  - Peu de tutoriels, souvent incomplets.
- On y arrive mais avec beaucoup trop de difficultés:
  - Trop de perte de temps à faire des essais
  - Qualité de la documentation pas au niveau de la qualité du code.

# Bilan Apprentissage Symfony2

- Effort d'apprentissage: plutôt réduit
  - Plus facile que Symfony1
    - Structuration plus simple, plus «logique»
  - Auto-formation: choisir un tutoriel adapté à un projet simple
    - minimum 2 semaines, au calme
  - Démarrage assez rapide, encourageant
    - Prototype qq semaines → Application qq mois
  - Formulaire dynamiques: plus difficile
    - Problème compliqué pour tous les Frameworks
- Élément clé: utiliser un éditeur adapté
  - Par exemple *Eclipse* (c'est un IDE, un env. de développement)
    - Contrôle syntaxe PHP
    - Accès direct aux interfaces et code des composants Symfony

# Bilan utilisation Symfony2

- Bilan très positif:
  - Productivité: beaucoup plus facile, plus rapide
    - Persistance, Routage, Authentification, Templates (Twig),...
  - Permet une grande réactivité (*agilité*)
    - Au niveau des Fonctionnalités
      - développement itératif facilité par la structure MVC.
    - Au niveau de la Présentation:
      - styles, widgets (Twitter, JQueryUI,..)
  - Facilité de maintenance et évolution
    - architecture connue, standard (documentée)



# Conclusion sur les Framework

- Investissement rentable ?
  - Pas toujours possible de disposer de quelques semaines
  - Le seuil n'est pas très élevé, une fois passé, retour arrière impossible: *persistance des données*,...
    - «convaincre les autorités » n'est plus une question.

⇒comment a-t-on pu se passer d'un Framework?
- Des pistes pour faciliter leur utilisation :
  - Régler la question du « bon » Framework:
    - Expérience acquise sera profitable ailleurs, pour un autre projet.
  - Régler la question du degré d'expertise:
    - Fournir des petites applis type, faciles à adapter et modifier.
    - Utilisables par les non experts
    - ... avec un «look» valorisant