



# PI - GROUPE

Présentation  
Framework PI

[www.pi-groupe.net](http://www.pi-groupe.net)



# **SOMMAIRE**

## ▶ **1. PRÉSENTATION PI**

- **1.1 But Général**
- **1.2 Méthodologie**
- **1.3 Environnement de développement**

## ▶ **2. MÉTHODOLOGIE ET NORMES**

- **2.1 Organisation et choix des outils**
- **2.2 Outils et démarche de développement**
- **2.3 Modélisation et méthodologie d'analyse**

## ▶ **3. STRUCTURE ET CONFIGURATION PI**

- **3.1 Noyau et Squelette du système**
- **3.2 Structure type d'une application**
- **3.3 Configuration et déploiement**



## **SOMMAIRE**

### ▶ **4. FRAMEWORK PI**

- **4.1** Une bibliothèque structuré des ressources
- **4.2** Utilisation PEAR
- **4.3** Programmation orienté objet – Design Patterns

### ▶ **5. EXPLOITATION PI**

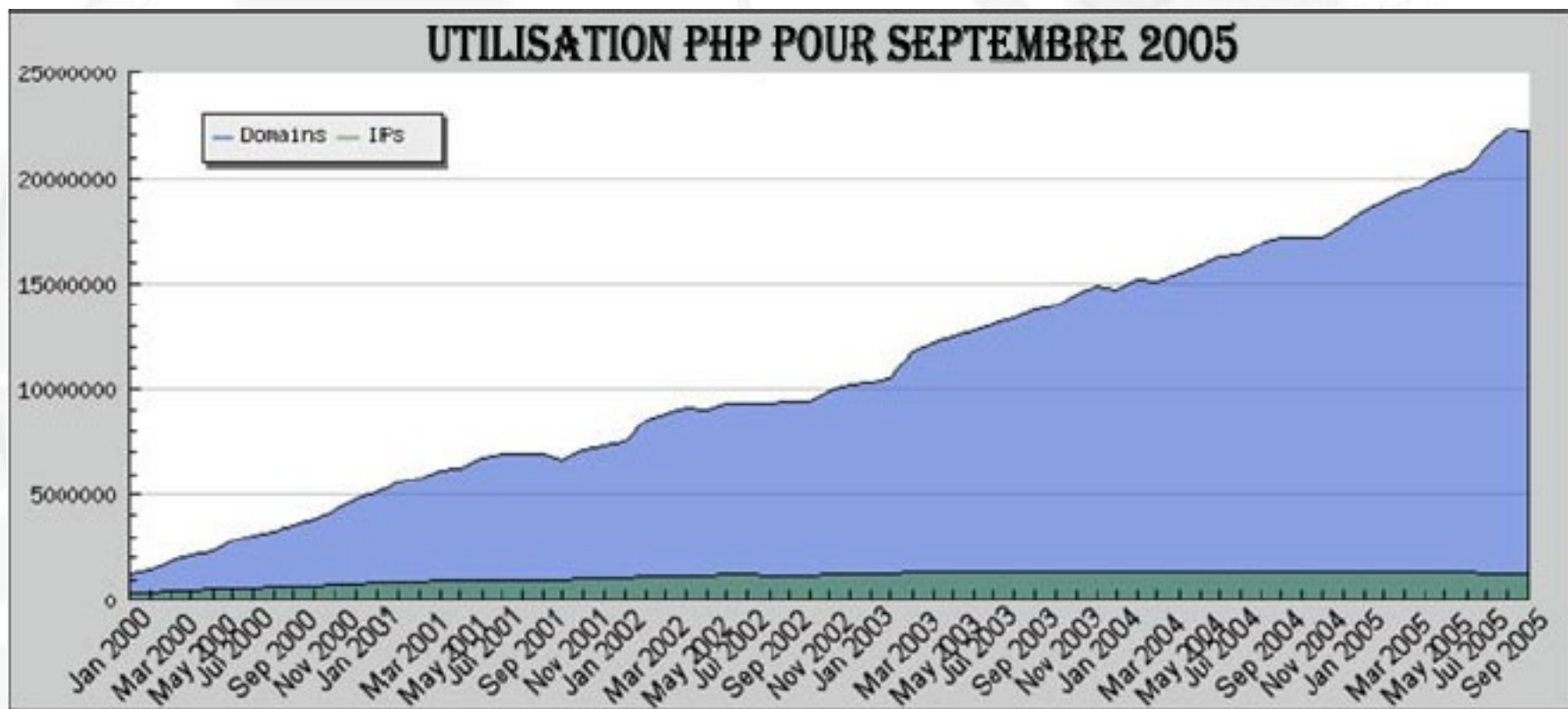
- **5.1** Services d'aide au développement
- **5.2** Gestion de la Sécurité
- **5.3** Versionnage, Sauvegarde et Archivage

### ▶ **6. DOCUMENTATION PI**

- **6.1** Génération de documentation
- **6.2** Documents de gestion
- **6.3** Documents de références



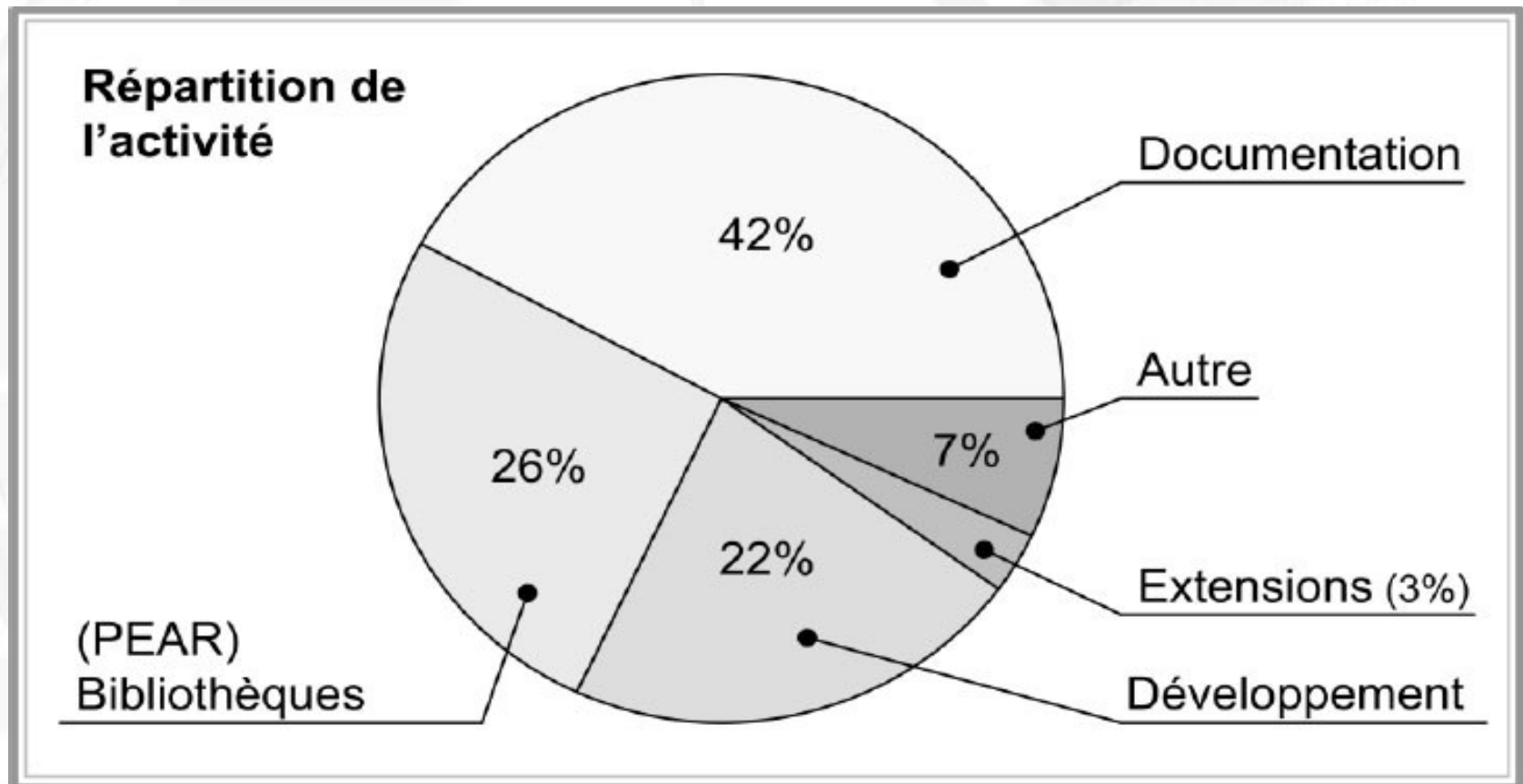
## INTRODUCTION



**Fig 1 - Un site web sur 3 dans le monde**  
(plus de 22 millions de domaine et 1 200 000 adresses IP, 40 % des sites web français)



## INTRODUCTION



**Fig 2 – Documentation et Ressources PHP DE PI**



## ▶ 1. PRÉSENTATION PI

### ▶ 1.1 But Général

- **Proposer une méthodologie de développement**
- **Gérer plusieurs équipes au sein de plusieurs projets**
- **faciliter l'échange d'informations entre les différentes équipes**
- **Garder toujours une qualité de développement bien définie**
- **Répondre plus rapidement aux besoins des développeurs**
- **Déployer un nouveau projet très rapidement**
- **Adapter et faire évoluer la plateforme rapidement et à moindre coût**
- **Utiliser au mieux les dernières technologies de l'information**



## ▶ 1. PRÉSENTATION PI

### ▶ 1.2 Méthodologie

**Au niveau programmation, PI est construit autour d'une normalisation des méthodes et techniques utilisées :**

- une norme de **codage**
- une norme pour la **documentation technique**
- une **ossature type** pour le développement d'une application
- un processus de réalisation construit sur **UML** et proposant un **avancement itératif et incrémentable**.



## ▶ 1. PRÉSENTATION PI

### ▶ 1.3 Environnement de développement

le développement d'un projet PI repose sur une conception professionnelle : **l'architecture 5 couches** (séparation entre l'accès à la base de données, le traitement des tables et l'affichage du contenu) :

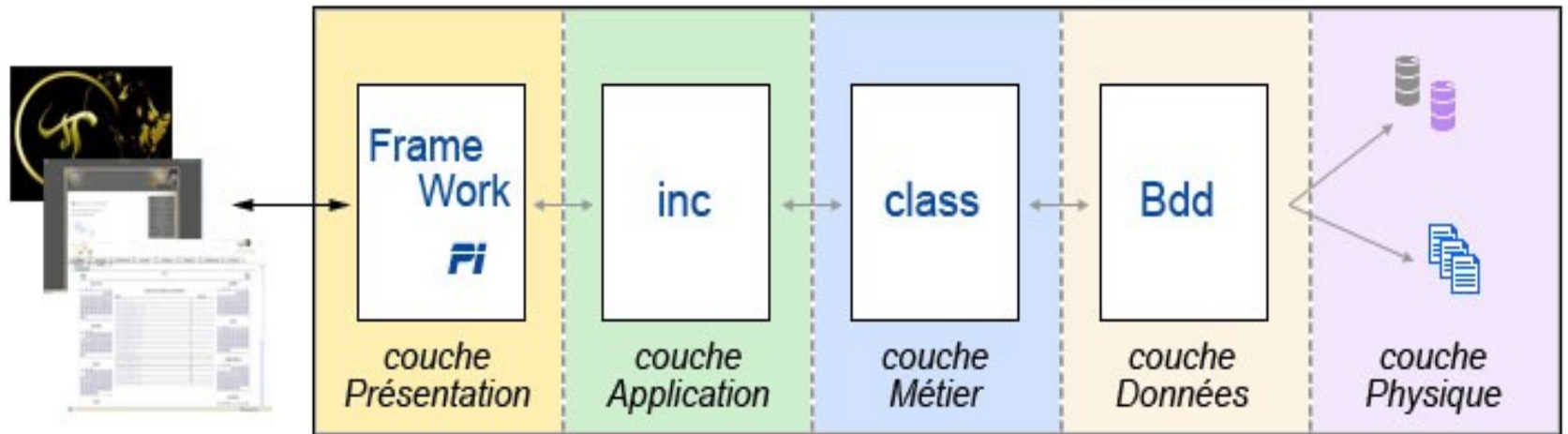
- une couche **physique**
- une couche **d'abstraction** d'accès à la base de données (**DataObject**)
- une couche **métier** correspondant aux classes de PI
- une couche **application**
- une couche **présentation**





## ▶ 1. PRÉSENTATION PI

### ▶ 1.3 Environnement de développement



**Fig 3 – Architecture 5 couches**



## ► 2. MÉTHODOLOGIE ET NORMES

### ► 2.1 Organisation et choix des outils

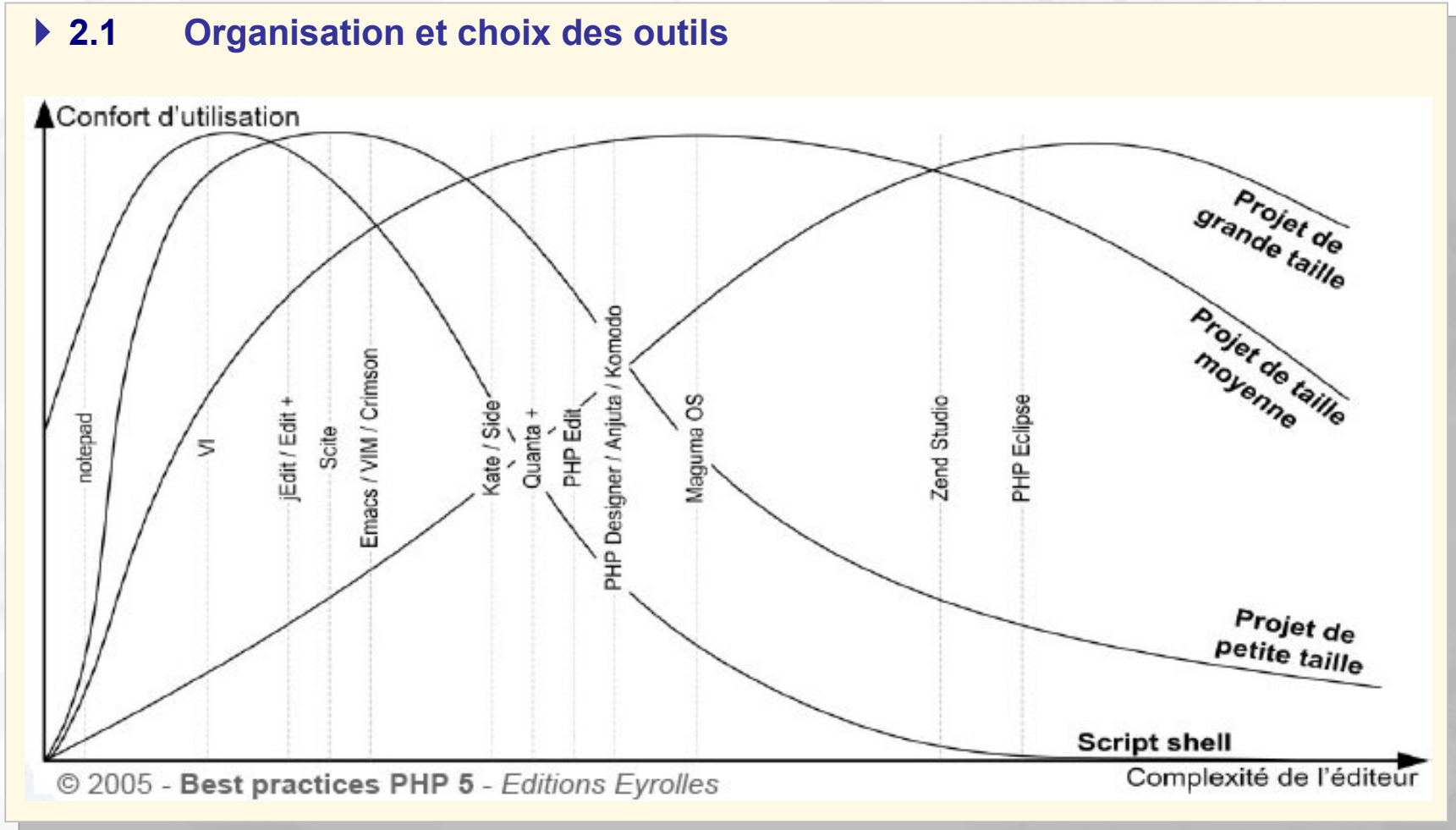


Fig 4 – Un éditeur fonction de vos besoins, habitudes et votre environnement<sub>10</sub>



## ▶ 2. MÉTHODOLOGIE ET NORMES

### ▶ 2.1 Organisation et choix des outils

Le **FRAMEWORK** PI évolue autour de **RESSOURCES** choisies en fonction de vos besoins, votre environnement, vos contraintes, etc.

Quelques critères de choix :

- La popularité de la ressource
- La capacité à évoluer
- La notoriété / motivation des auteurs

Type de ressources :

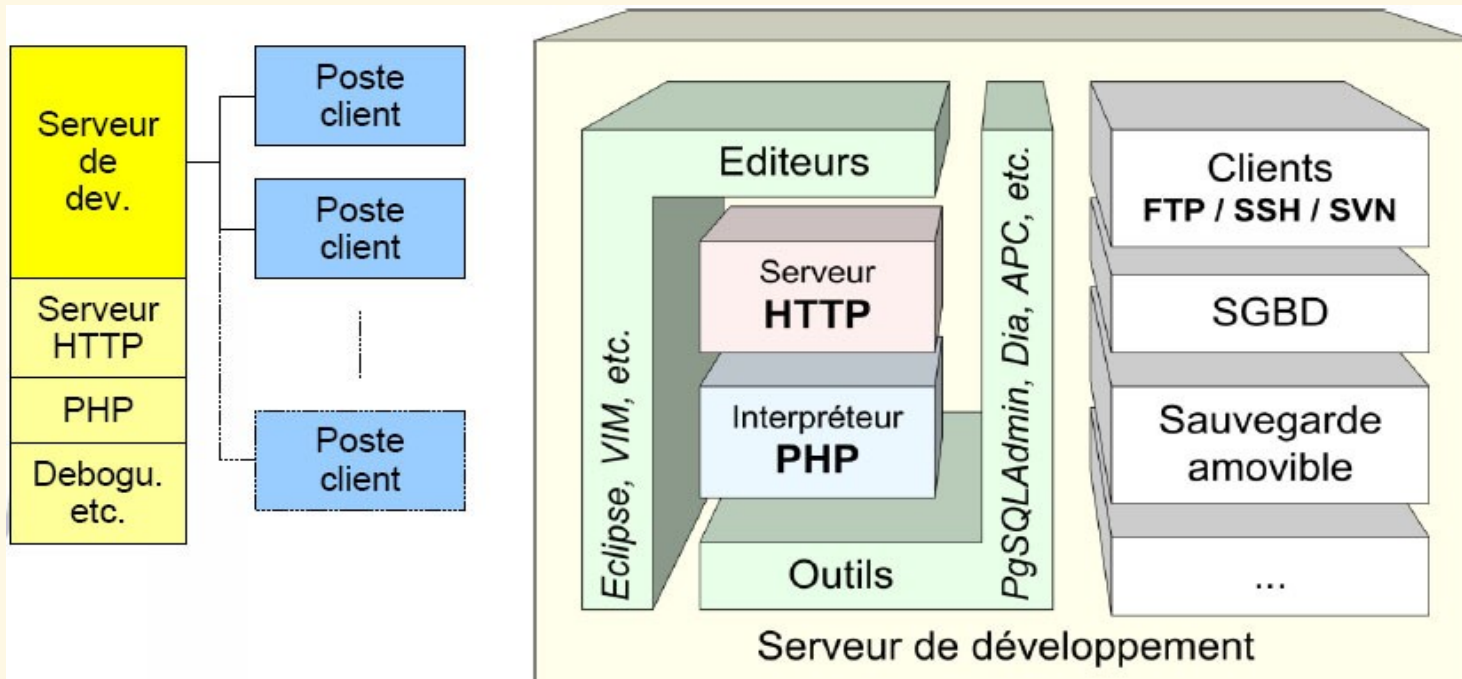
- Les application (fonctionnalités toutes faites)
- Les librairies (fonctionnalités à intégrer, PEAR)
- Les extensions (fonctionnalités développées en C)



## ► 2. MÉTHODOLOGIE ET NORMES

### ► 2.1 Organisation et choix des outils

**L'ENVIRONNEMENT D'EXECUTION** pour le développement optimisé pour la modélisation, l'édition, le débogage, le travail en équipes, etc.





## ▶ 2. MÉTHODOLOGIE ET NORMES

### ▶ 2.2 Outils et démarche de développement

Une méthodologie et une norme de développement est utilisé par PI afin :

- ▶ de pouvoir travailler en groupe
- ▶ de faciliter la maintenance des applications
- ▶ de préparer la génération automatique de la documentation développeur

Cette **normalisation du codage** est une synthèse des règles de codage du projet **PEAR** .

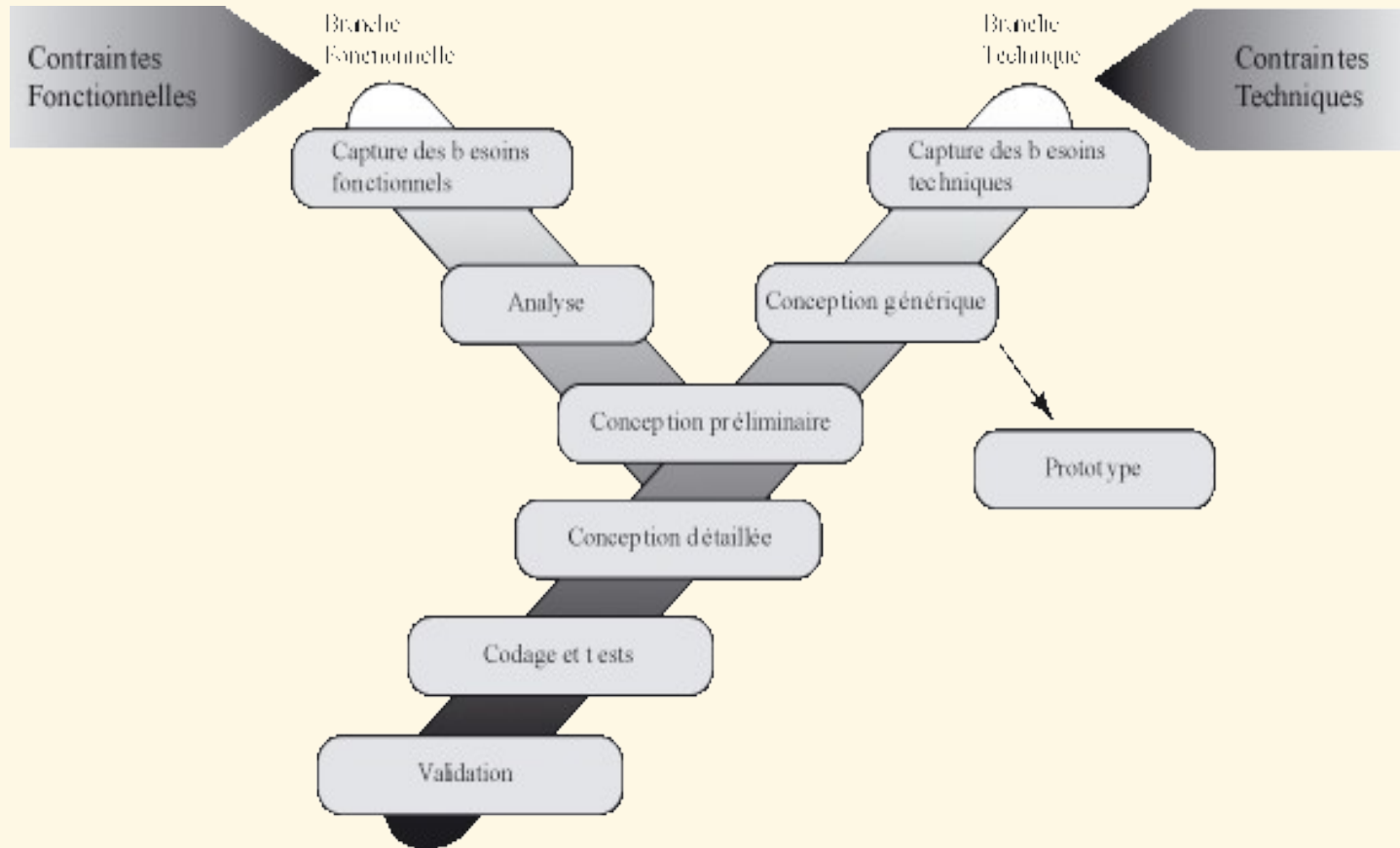
Eclipse est un excellent environnement de développement regroupant tous les outils répondant à nos besoins :

- ▶ outil de **modélisation** (UML)
- ▶ outil de **développement** (PHP, HTML, CSS, SQL)
- ▶ outil de **versionnage** (CVS)



## ► 2. MÉTHODOLOGIE ET NORMES

### ► 2.2 Outils et démarche de développement



**Fig 6 – Démarche de développement en « Y »**



## ▶ 2. MÉTHODOLOGIE ET NORMES

### ▶ 2.3 Modélisation et méthodologie d'analyse

Pour la conception et la **modélisation** de la base de données, **DB designer** reste un bon outil car sous licence GPL, et utilisable sous Linux et Windows

Pour la réalisation des diagrammes de classes, de séquences, etc, **Eclipse UML** a été retenu car en pleine évolution et sous le même environnement de développement choisi. Il permet la réalisation des:

- ▶ diagrammes de classes
- ▶ diagrammes de cas d'utilisation
- ▶ diagrammes d'activités
- ▶ diagrammes de collaboration
- ▶ diagrammes de séquence



## ▶ 2. MÉTHODOLOGIE ET NORMES

### ▶ 2.3 Modélisation et méthodologie d'analyse

L'expression et l'analyse des besoins se matérialise par des documents OpenOffice de différents niveaux, dans lesquels on insère les diagrammes UML produits à l'aide du logiciel de modélisation choisi :

- ▶ **Note de cadrage** (inclut diagrammes de la vue métier)
- ▶ **Exigences fonctionnelles** (inclut diagrammes de la vue système informatique)
- ▶ **Conception de l'interface utilisateur** (inclut diagrammes de la vue applicative)

Dans un contexte professionnel :

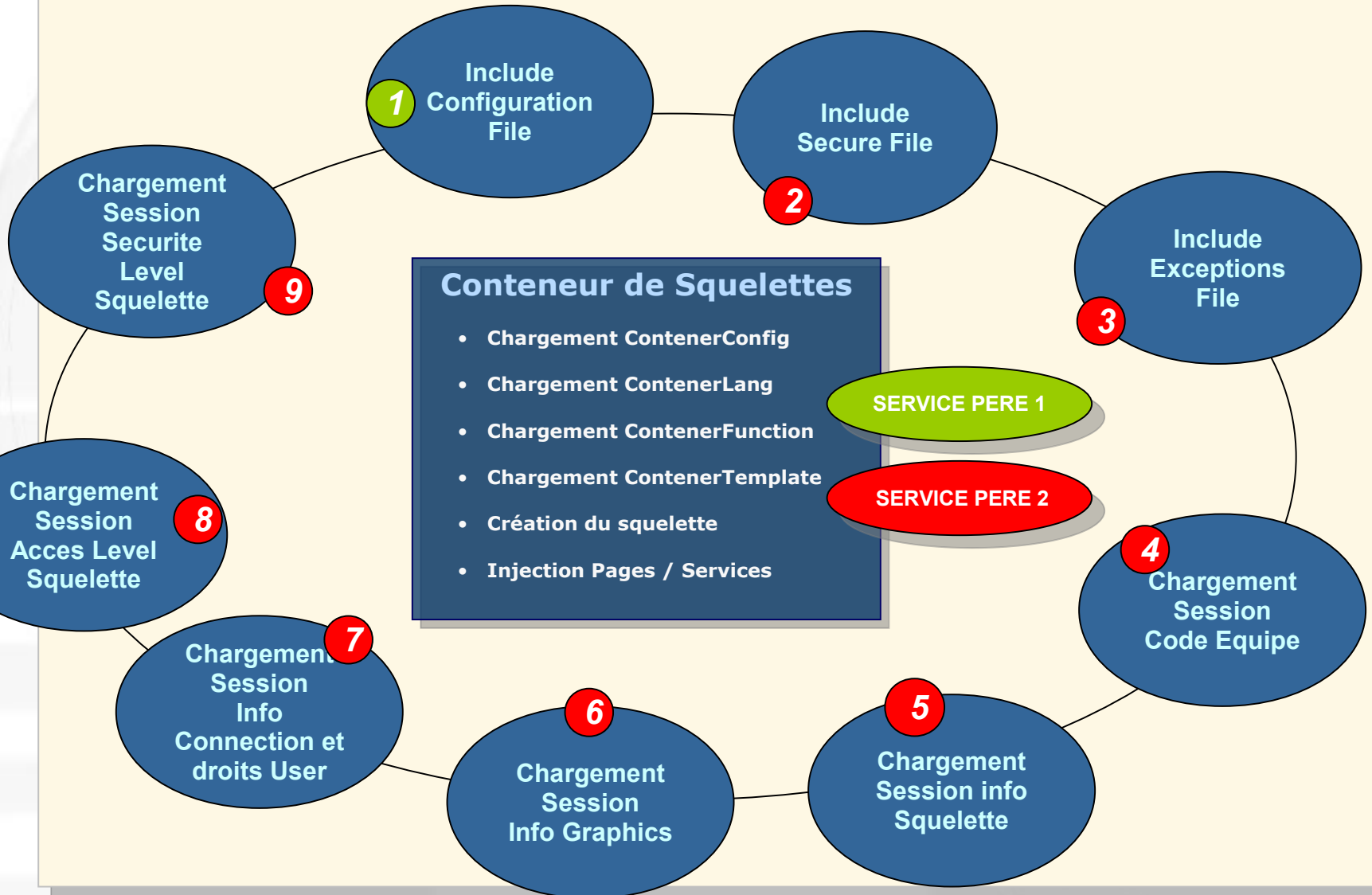
- ▶ **Création de modèle pour la Documentation de Gestion**
- ▶ **Création d'un stéréotype de la modélisation**





## ▶ 3. STRUCTURE ET CONFIGURATION PI

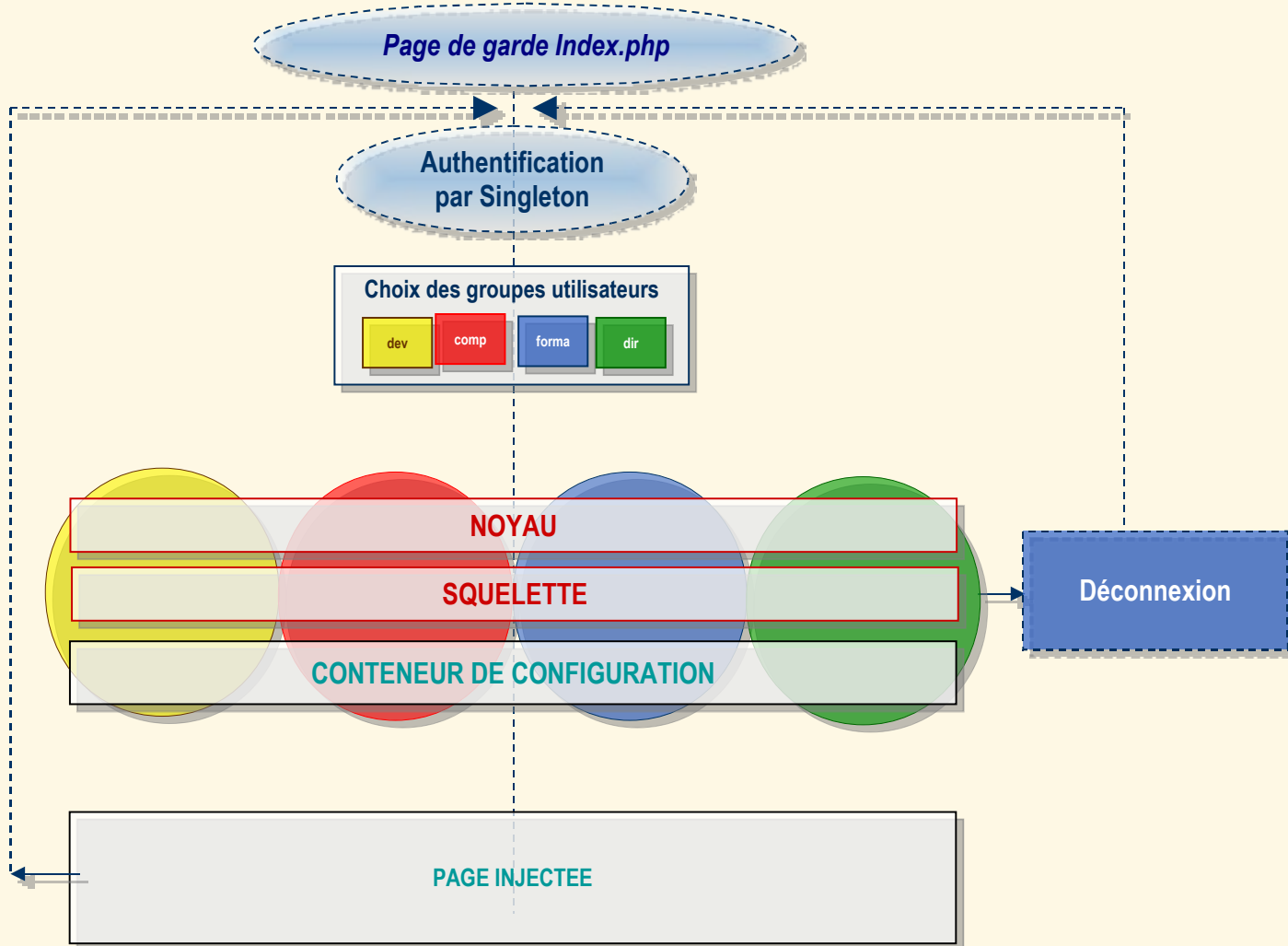
### ▶ 3.1 Noyau et Squelette du framework





## ▶ 3. STRUCTURE ET CONFIGURATION PI

### ▶ 3.2 Structure type d'une application





## ▶ 3. STRUCTURE ET CONFIGURATION PI

### ▶ 3.3 Configuration et déploiement

La configuration se fait lors de l'installation de la plate-forme en créant plusieurs fichiers :

- ▶ `app_cfg.inc.php`
- ▶ `data_graphism.ini.php`
- ▶ `data_object.ini.php`

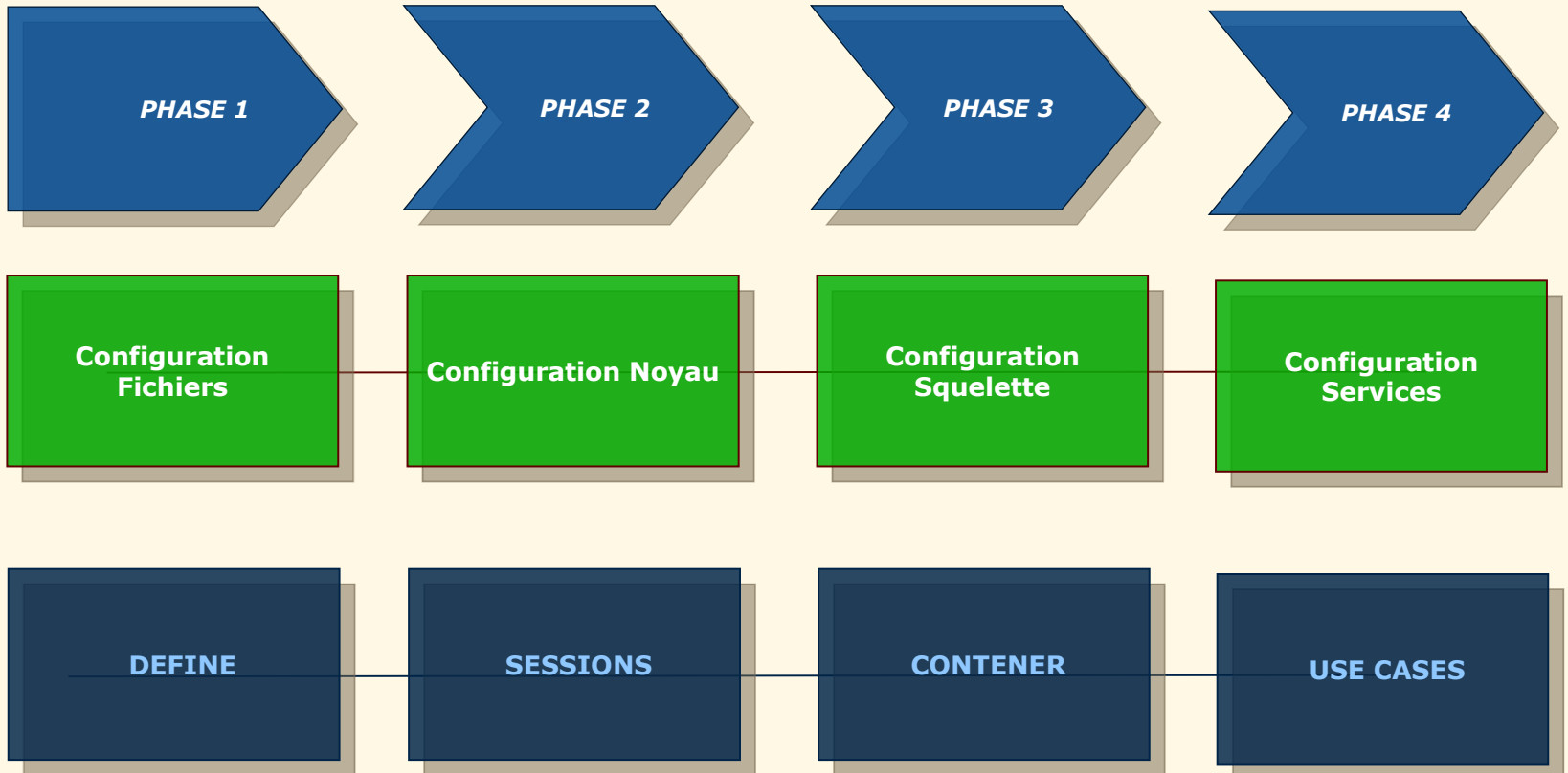
D'autres fichiers de configuration existants dans la plate-forme :

- ▶ `data_object_links.ini`
- ▶ `contener.ini.php`
- ▶ `droits.ini.php`
- ▶ `securite.ini.php`



## ▶ 3. STRUCTURE ET CONFIGURATION PI

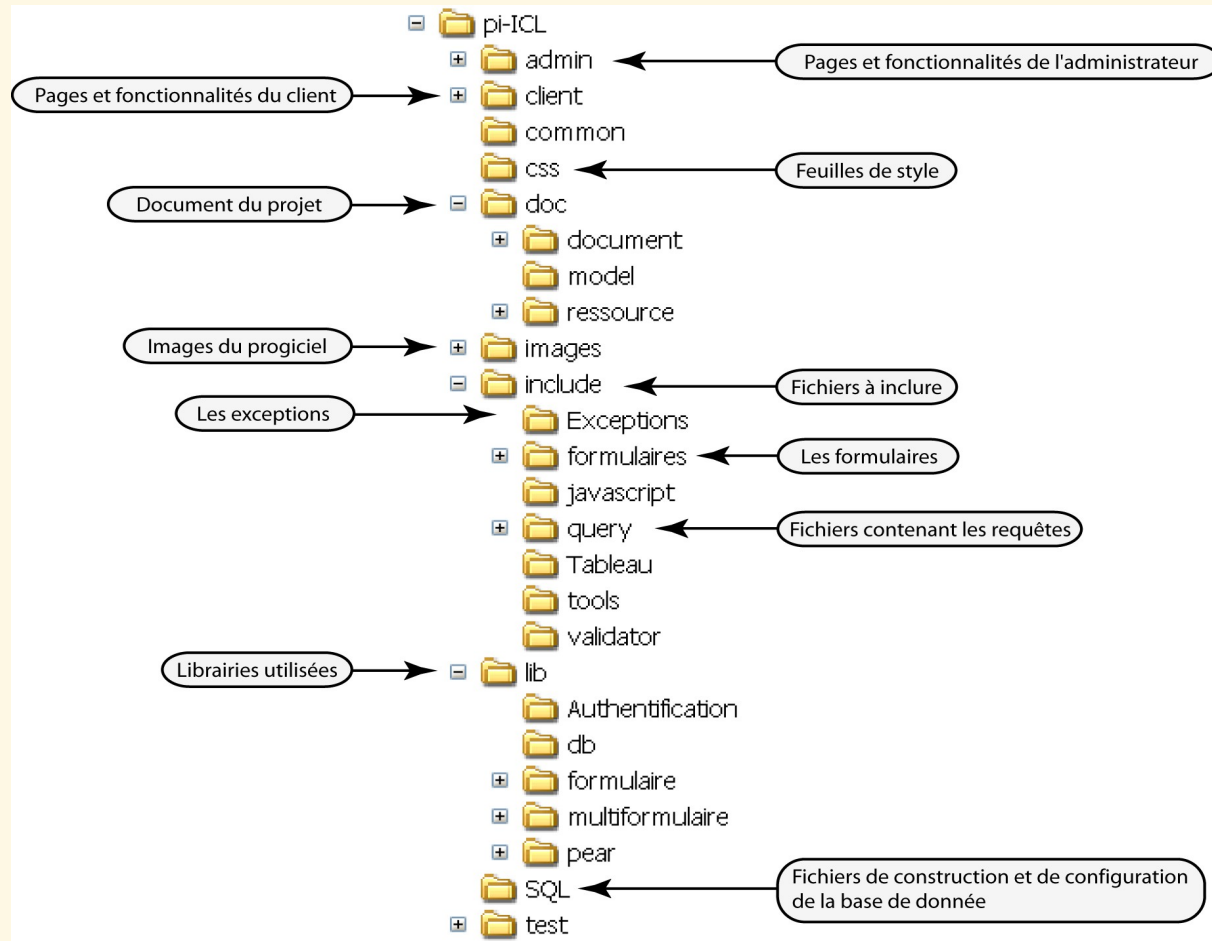
### ▶ 3.3 Configuration et déploiement





## ► 4. FRAMEWORK PI

### ► 4.1 Une bibliothèque structurée des ressources





## ▶ 4. FRAMEWORK PI

### ▶ 4.2 Utilisation PEAR

**L'utilisation de PEAR permet de répondre à la plupart des difficultés rencontrées durant le développement d'un webservice :**

- ▶ **Gestion et vérification des dates**
- ▶ **Création de calendriers et de plannings**
- ▶ **Gestion des bases de donnée et DataObject**
- ▶ **Gestion de la Sécurité et de l'authentification**
- ▶ **Gestion de la Configuration et des langues**
- ▶ **Gestion des formulaires et des règles**
- ▶ **Gestion de Fichiers et de Contener**
- ▶ **Génération de code HTML (comme des tableaux)**
- ▶ **Division du code HTML et du code PHP par la gestion des template**
- ▶ **Large éventail d'outils orientés réseaux**



## ▶ 4. FRAMEWORK PI

### ▶ 4.2 Utilisation PEAR

Les packages PEAR principaux utilisés dans la plate-forme sont :

- ▶ **PEAR ::DB DataObject** : Abstraction objet de la base de données
- ▶ **PEAR ::HTML Quickform** : Création et gestion avancées de formulaires
- ▶ **PEAR ::Auth** : paquetage d'authentification
- ▶ **PEAR ::HTML** : paquetage utilisé pour la gestion des tableaux HTML
- ▶ **PEAR ::HTML Template Xipe** : Séparation du code HTML et du code PHP
- ▶ **PEAR ::Date** : Gestion des date
- ▶ **PEAR ::Calendar** : paquetage pour générer des calendriers
- ▶ **PEAR ::Config** : paquetage pour la gestion des conteners



## ▶ 4. FRAMEWORK PI

### ▶ 4.3 Programmation orienté Objet – Design Patterns

La programmation orienté objet utilisée par PI comporte les avantages suivants :

- ▶ Un code **compréhensible**
- ▶ Un code **réutilisable**
- ▶ Un code **naturel**
- ▶ Un code **modulaire**
- ▶ Un code **protégé**

Ces améliorations nous permettent d'utiliser les design patterns. Avec ses évolutions majeures telles que des membres ou des constructeurs / destructeurs unies, PHP 5 nous autorise la conception par **design pattern qu'intègre PI** :

- ▶ **Singleton** : « design pattern » dont on souhaite créer une classe *qui s'instancie une seule fois*
- ▶ **Factory** : « design pattern » dont on souhaite déléger certains *traitements à d'autres classes, en instanciant les objets correspondants selon le contexte*





## ▶ 5. EXPLOITATION PI

### ▶ 5.1 Services d'aide au développement

Les termes suivants sont utilisés dans la plate-forme :

- ▶ « **Module** » ensemble de services, regroupés dans un repository unique
- ▶ « **Use Case** » ensemble indexé de services, constitué de services fils et d'un unique service Père.
- ▶ « **Service Père** » service principale d'un Use Case dont les services fils héritent.
- ▶ « **Service Fils** » service héritant d'un service père
- ▶ « **Service** » page web identifiable graphiquement via un Nom
- ▶ « **Package** » regroupement d'un ensemble de Use Case.
- ▶ « **Noyau** » Page Coeur de chargement de PI dont le squelette du groupe d'un utilisateur.
- ▶ « **Squelette** » Structure type Conteneur permettant la création du support logiciel de type webservice .
- ▶ « **Contener** » regroupement d'un ensemble de classes selon un thème métier.



## ▶ 5. EXPLOITATION PI

### ▶ 5.1 Services d'aide au développement

**Un module Administrateur a été développé de manière à pouvoir accompagner le développement d'un projet PI. Après identification, l'administrateur peut :**

- ▶ **Consulter les connexions des membres et responsables**
- ▶ **Créer / modifier / supprimer des membres, des responsables, des équipes**
- ▶ **Attribuer un ou plusieurs responsables à une équipe de membres**
- ▶ **Créer / modifier / supprimer nouveaux services ( UsesCase)**
- ▶ **Attribuer un ensemble de services à une équipe ou à un responsable**
- ▶ **Attribuer un ensemble de droits d'accès aux membres et responsables**
- ▶ **Attribuer un squelette spécifique à chaque équipe**
- ▶ **Administrer les services de développement du projet (suivi développement Use Case, Sauvegarde BDD, Création DataObject, Création de Models...)**



## ▶ 5. EXPLOITATION PI

### ▶ 5.2 Gestion de la Sécurité

La plate-forme PI respecte donc plusieurs points concernant la sécurité :

- ▶ La disponibilité de l'application et son bon fonctionnement
- ▶ Non divulgation des données confidentielles (base du client)
- ▶ Intégrité des données
- ▶ Intégrité du progiciel et de son image

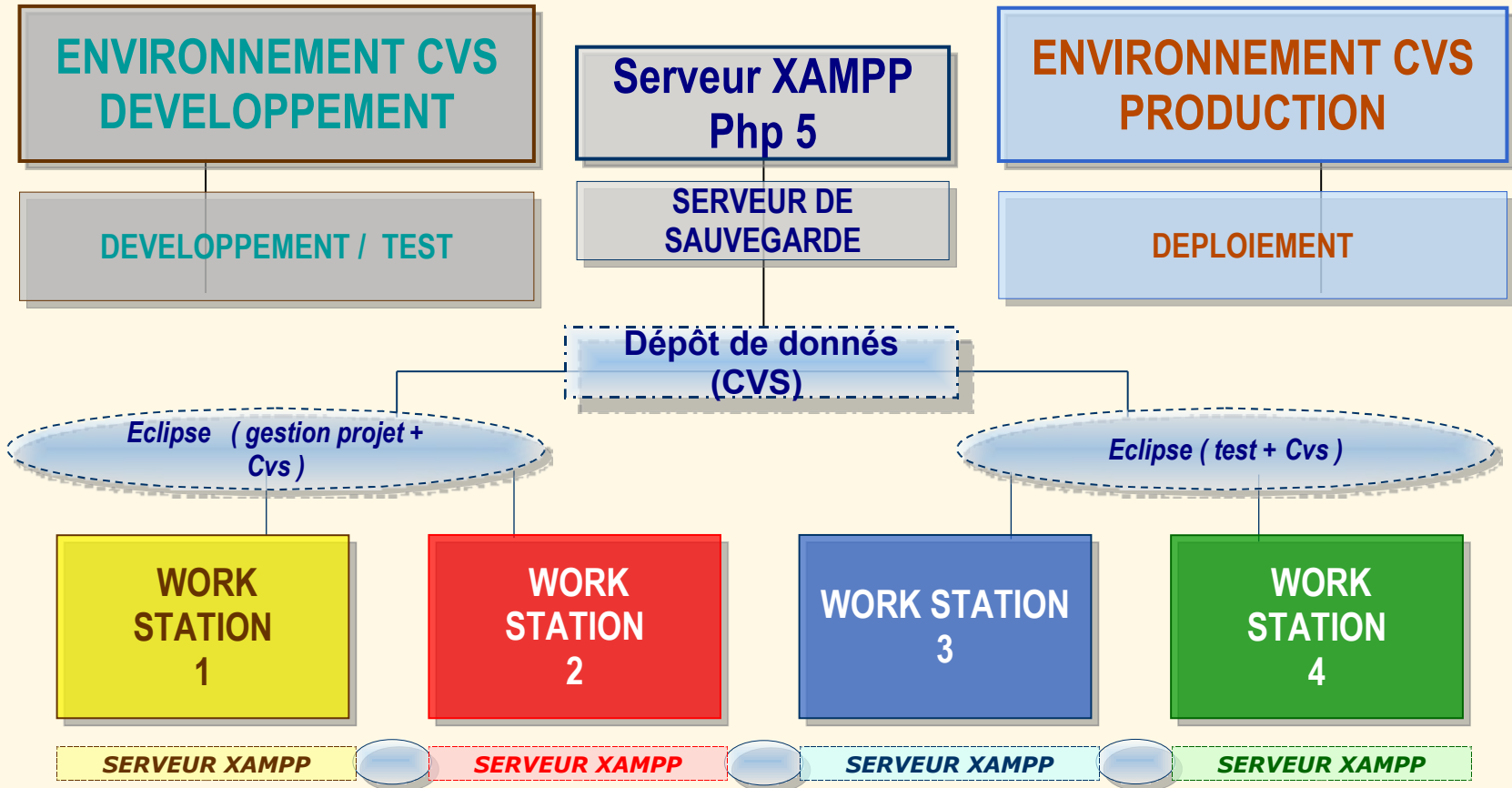
Ensuite, pour ce qui concerne le protocole de communication entre le serveur et les machines clientes, le progiciel permet d'utiliser le **protocole SSL**.

Enfin, les identifiants et la vérification des utilisateurs permettent une **sécurité avancée**. Les identifiants et mot de passe sont cryptés (128 bits). Lors de l'accès à une page, l'utilisateur est vérifié ainsi que son adresse IP afin de limiter les falsifications d'identité.



## ► 5. EXPLOITATION PI

### ► 5.3 Versionnages, Sauvegarde et Archivage





## ▶ 6. DOCUMENTATION PI

### ▶ 6.1 Génération de documentation

Un générateur de documentation pour PHP (**phpdocumentor**) permet de **générer une documentation** selon la norme **PEAR** du code source de l'application à partir des commentaires inclus directement dans le code.

- Génération documentation
- ▶ lisibilité
    - ▶ interfaces
    - ▶ structures
  - ▶ meilleure diffusion des sources
  - ▶ maintenabilité d'un projet



## ▶ 6. DOCUMENTATION PI

### ▶ 6.2 Documents de gestion

Identification de la documentation par un **mécanisme normalisé d'identification** des documents (homogénéité).

Chaque document reçoit une référence unique au sein du projet constituée de plusieurs champs et référencé sur chaque page de garde d'un document :

- ▶ Nom Projet : **nom\_projet / nature\_document / identification\_document**
- ▶ Nature Document :
  - ▶ document de gestion : **GE**
  - ▶ document de développement :
    - étape d'étude préalable : **EP**
    - étape d'étude détaillée : **ED**
    - étape d'étude technique : **ET**
    - étape de réalisation : **RE**
    - étape de réception : **RC**
  - ▶ document de mise en oeuvre : **MO**
    - documentation lié à la documentation utilisateur : **DO**
    - documentation lié à la formation des utilisateurs : **FO**
    - documentation lié au laboratoire d'évaluation: **LE**
  - ▶ document qualité : **QU**



## ▶ 6. DOCUMENTATION PI

### ▶ 6.3 Documents de référence

#### **DOCUMENTS GENERES :**

- ▶ **Installation.pdf**
- ▶ **Configuration.pdf**
- ▶ **Authentification.pdf**
- ▶ **Gestion Squelette.pdf**
- ▶ **Gestion Equipe.pdf**
- ▶ **Gestion Membre.pdf**
- ▶ **Gestion Responsable.pdf**
- ▶ **Gestion Service.pdf**
- ▶ **DataObject function.pdf**
- ▶ **Tools class.pdf**
- ▶ **PEAR Xipe template.pdf**

#### **DOCUMENTS ANNEXES :**

- ▶ **presentation PI.odp**
- ▶ **installation framework PI.odt**
- ▶ **utilisation administration PI.odt**
- ▶ **utilisation service module.odt**
- ▶ **dossier exploitation.odt**
- ▶ **conduite de projet.pdf**
- ▶ **gestion doc des projets inf.pdf**
- ▶ **livre blanc.pdf**
- ▶ **divers dossier de formation ...**



## ***CONCLUSION***





# PI-GROUPE