

# Retour d'expérience

Erlang Factory – 30 nov. 2012

## ■ Fonctionnalités Centre de Contact

- **Front:** Serveurs vocaux interactifs / mail /chat / Réseaux sociaux
- **Back :** Stratégies de mise en relation du client et de distribution
- Contextualisation du parcours client
- Reporting

## ■ Exigences Centre de Contact

- Temps réel (latence inférieure à 300ms)
- Continuité de service

## ■ Agilité et processus industriels

- Tests et analyses
- Déploiement

- Technologie actuelle autour des langages objets
  - Serveurs d'applications
  - Orienté objet
  - Nombreux frameworks
  
- Caractéristiques Erlang attendues
  - Chargement du code à chaud
  - Robustesse et continuité de service
  - Parallélisation des traitements
  - Agilité du développement

## ■ Constitution de l'équipe

- 2 internes experts en développement



- 2 intervenants experts sur la technologie Erlang



## ■ Formation

- Formation théorique sur le langage Erlang
  - Egalement OTP / Yaws / Mnesia
- Accompagnement des experts Erlang durant tout le projet
  - Fonctionnement en binômes / relectures

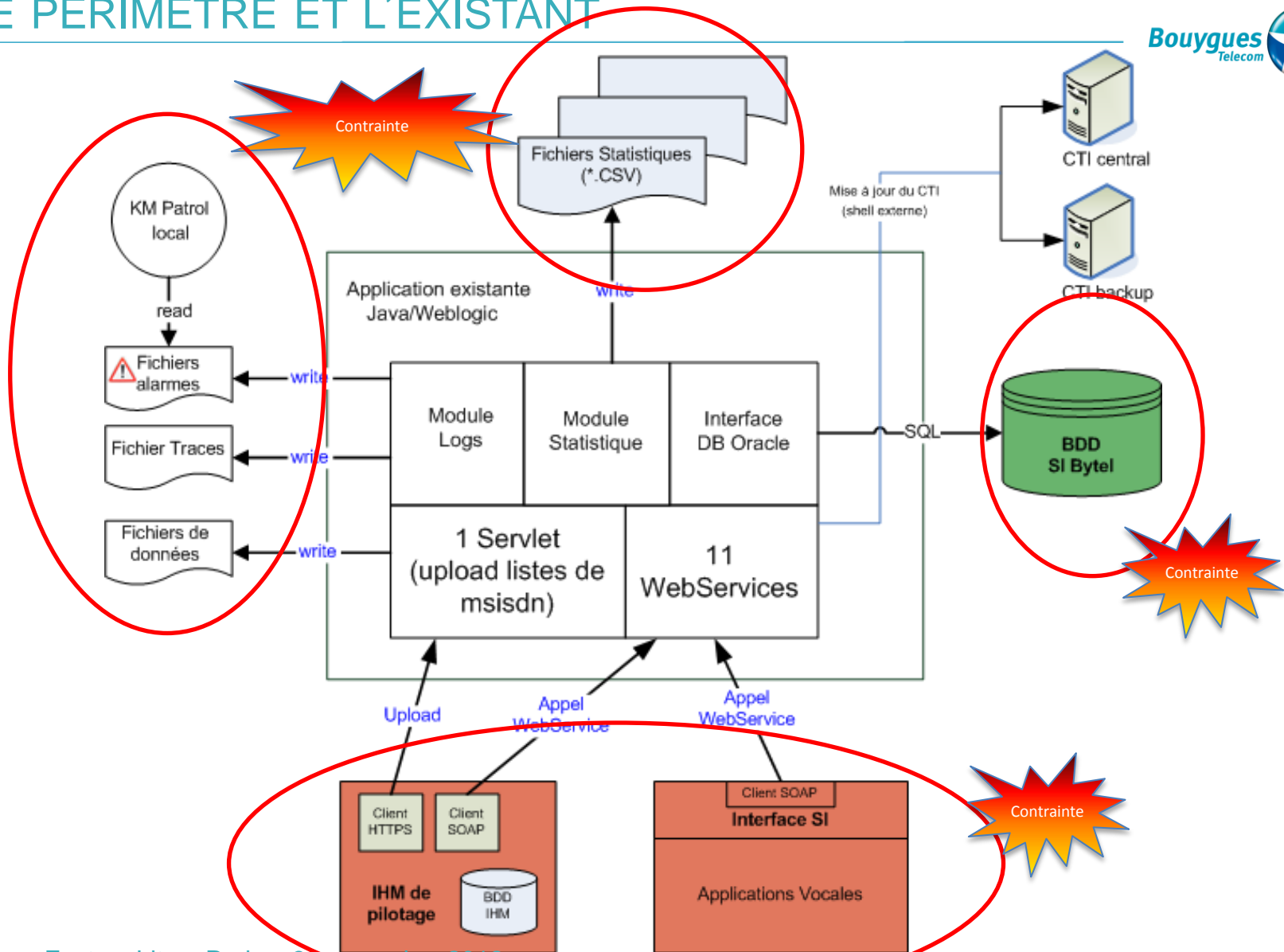
## ■ Objectif du projet

- Réduire la complexité et les temps de MEP (update à chaud)
- Assurer la robustesse et la performance du système
- Expérimenter l'agilité et la rapidité de développement du langage

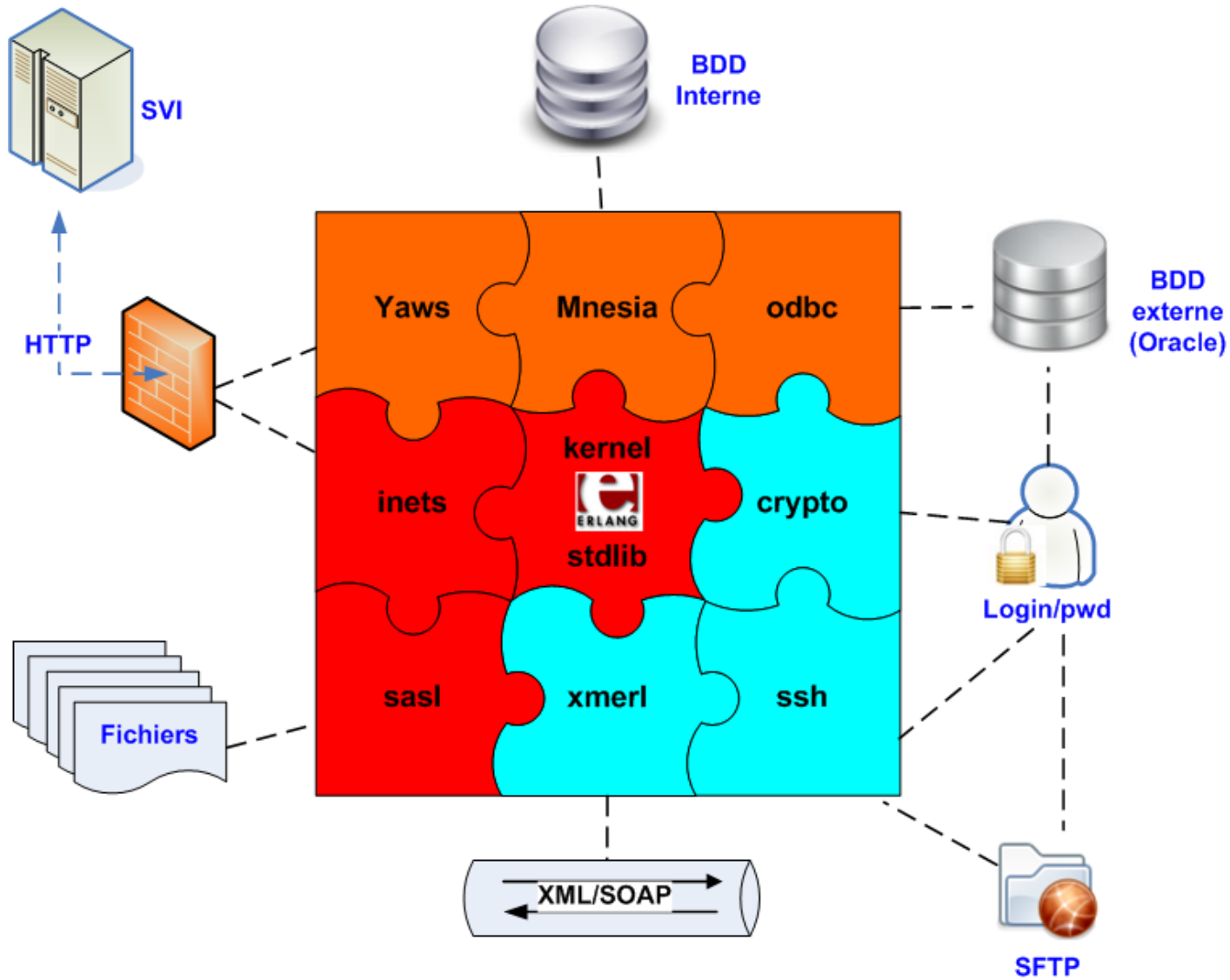
## ■ Sélection du périmètre

- Correspondance à la roadmap
- Application se prêtant à Erlang
- Risques modérés : Périmètre identifié en obsolescence
- Nouvelles fonctionnalités

# LE PÉRIMÈTRE ET L'EXISTANT



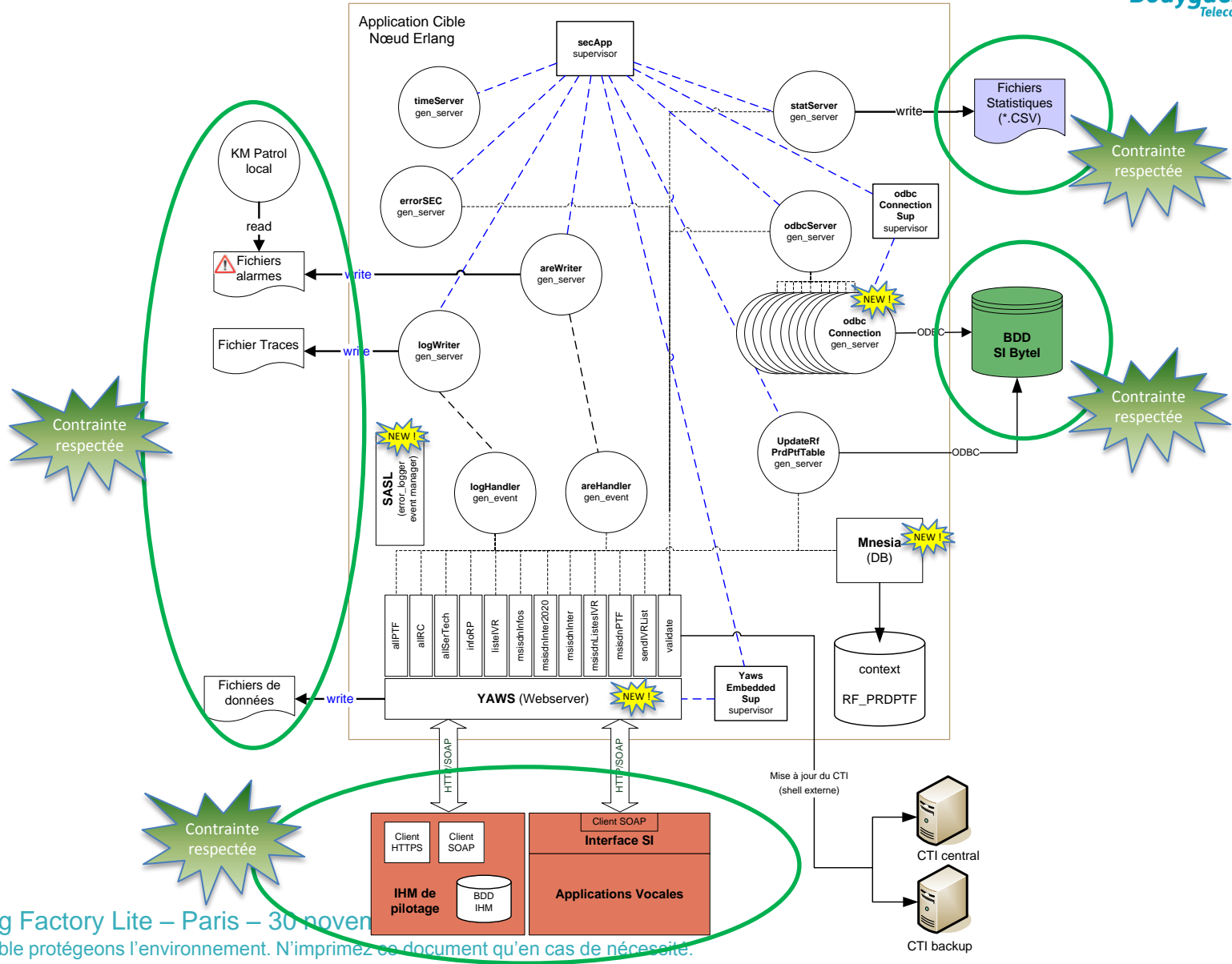
# COMPOSANTS ERLANG



Erlang Factory Lite – Paris – 30 novembre 2012

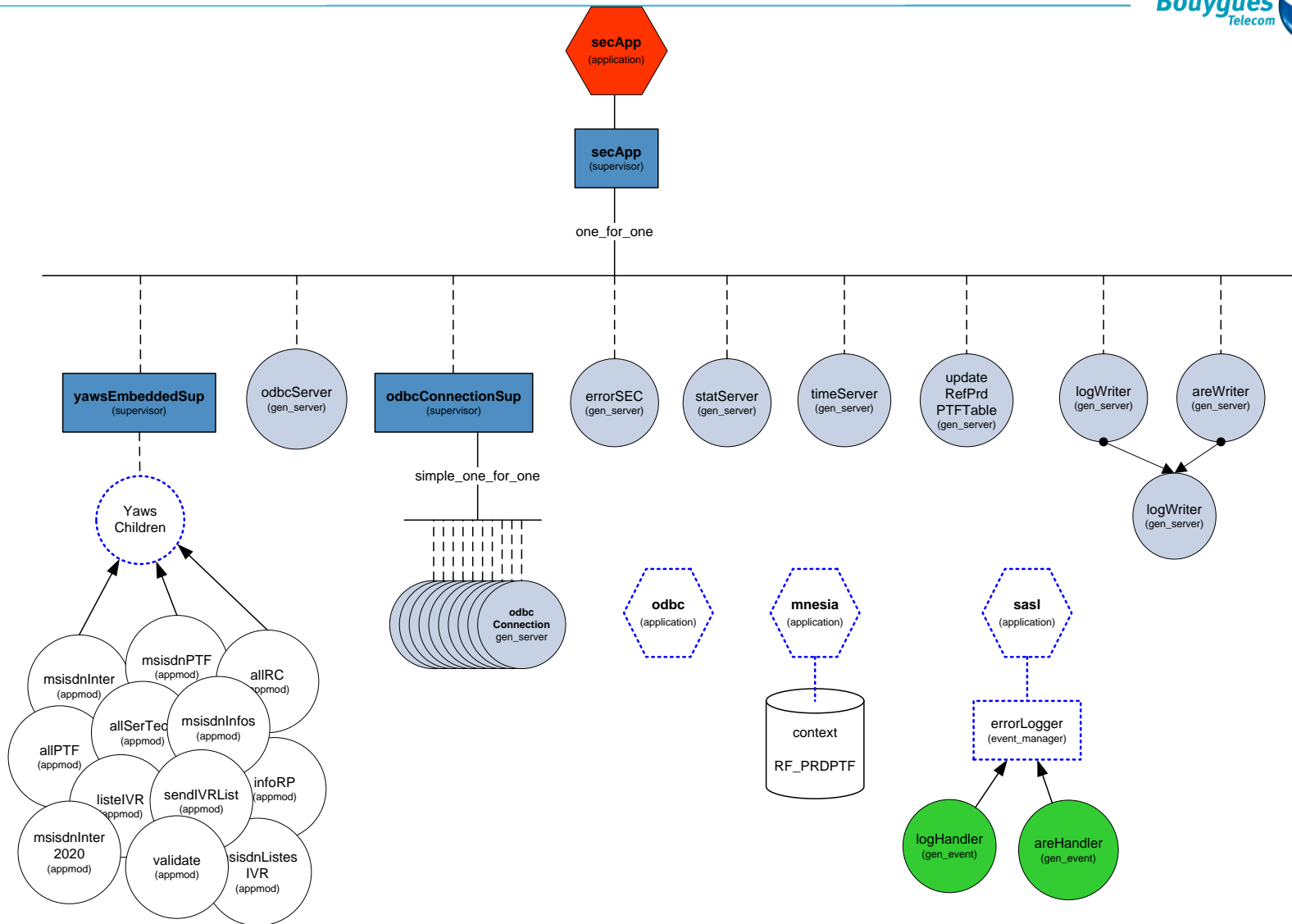
Ensemble protégeons l'environnement. N'imprimez ce document qu'en cas de nécessité.

# L'ARCHITECTURE OBTENUE



Erlang Factory Lite – Paris – 30 novem  
 Ensemble protégeons l'environnement. N'imprimez ce document qu'en cas de nécessité.





- Permet l'installation et le retour arrière d'évolutions sans arrêter l'application
  - Une simple ligne de commande shell
- Toutefois, l'intelligence se trouve en amont
- Importance de suivre les standards OTP
- Nécessite une conception de l'architecture dynamique de l'application tenant compte :
  - de l'arbre de supervision
  - des relations entre les processus serveurs
  - des états internes des processus serveurs
- Mais aussi une conception de la mise en œuvre
  - Développement d'un script d'upgrade dédié inspiré de Rebar



# DÉVELOPPEMENT & DÉPLOIEMENT



Windows XP

- 1 Packaging
- 2 Upgrade



1  
Release  
OTP



2



Unix (Solaris)

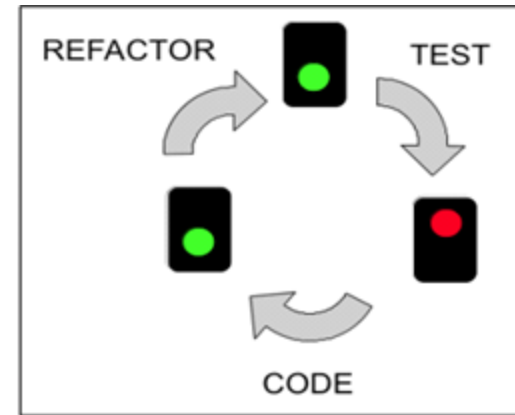


- > install
- > start
- > stop
- > info
- > upgrade

**Tests unitaires**  
-  
**Intégration continue**

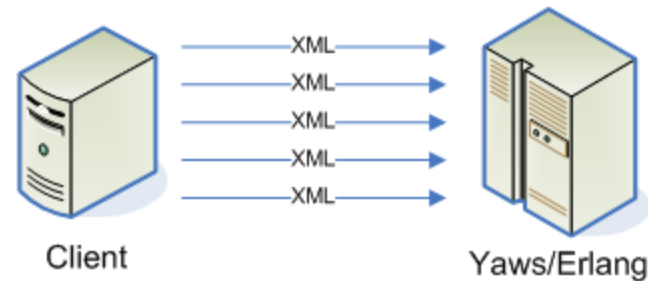


Active Ring™

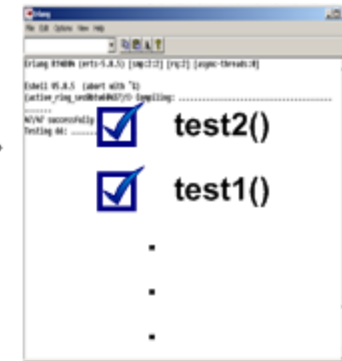
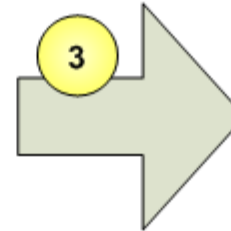
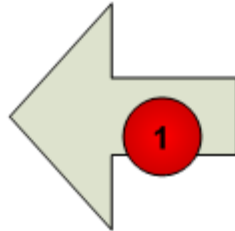


+

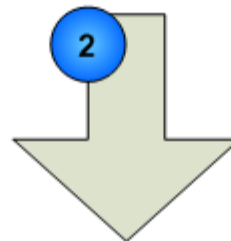
**Tests d'appel webservice**  
-  
**Tests de charge**



# TDD OU « TESTS DRIVEN DEVELOPMENT »



**47 modules de test  
totalisant  
66 tests unitaires**



Détecte les modifications



Recompile les sources



Lance les tests unitaires

- 2 Mises à jour à chaud réussies et rapides
- Facilité d'exploitation : une seule ligne de commande pour
  - Installer
  - Démarrer/arrêter
  - Obtenir des informations
  - Mettre à jour « à chaud »
- Élimination de la brique « serveur d'application » (Weblogic)
- Approche « agile », incrémentale & déclarative de la programmation
  - Possibilité de coder directement dans la console (boucle d'interaction)
- Adapté pour les traitements parallèles et distribués
- Technologie pensée pour une application industrielle



- Gestion « manuelle » des WSDL
- Goulot d'étranglement lié au connecteur ODBC
- Communauté active mais relativement restreinte
- Compétences Erlang et fonctionnelles relativement rares
- Documentation hétérogène



