

# Master Info 2<sup>ème</sup> année option CODES

## SECO - Travaux Pratiques

### 2015-2016

Oscar CARRILLO  
oscar.carrillo@femto-st.fr

Pour récupérer les sources du TP, connectez-vous sur moodle et inscrivez-vous au cours Sécurité et Composants (Utilisez la recherche) et entrez la clé d'inscription : seco-m2

## 1 Sujet du TP

Ce TP est structuré comme suit :

- Il y a un répertoire par étape du TP : de exo1 à exo6.
- Le répertoire externals contient tous les jar de l'implantation Julia de Fractal.
- Le répertoire doc contient la javadoc de Fractal.
- Le répertoire initial contient deux exemples d'application : Helloworld et Comanche.

Dans chaque étape, on trouve toujours la même structure :

- **build** contient les fichiers générés lors des étapes de compilation,
- **etc** contient des fichiers de configuration,
- **src** contient les fichiers source.

Un fichier build.xml permet de gérer les aspects compilation et exécution des programmes.

On retrouve les cibles **ant** suivantes :

- **clean** : pour effacer les fichiers générés
- **compile** : pour compiler les sources
- **execute** : pour lancer l'exécution

Pour chaque étape, faites bien attention de travailler dans le bon répertoire.

### 1.1 Etape initiale

Le but de cette étape est de se familiariser avec Fractal. Pour cela, regardez les sources des applications présentes dans le répertoire initial. Deux exemples sont présents :

- Helloworld : Le modèle est entièrement spécifié grâce à l'API Java (Fichier Helloworld.java). Il met en place deux composants **Client** et **Serveur**. Le composant **Client** demande simplement un service au composant **Serveur** qui fait une impression à l'écran.
- Comanche : Cet exemple est un serveur http extrêmement simple. Il est spécifié grâce à Fractal ADL. Pour vous aider à comprendre ce que fait cette architecture, vous pouvez vous rendre à l'adresse <http://fractal.objectweb.org/tutorial/index.html>. Vous

observerez en particulier les fichiers ADL et vérifierez qu'elle est conforme à l'architecture présentée dans la Figure 1.

Faites marcher les deux applications et pour l'exemple comanche, testez le avec un client web en utilisant `http://localhost:8080/index.html`.

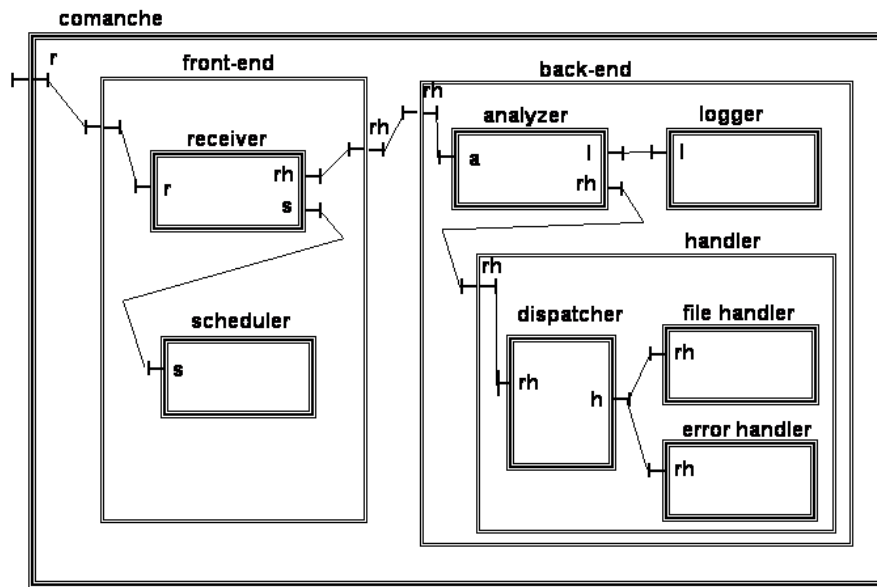


FIGURE 1 – Comanche initial

## 1.2 Exercice 1

Vous devez modifier l'exemple helloworld pour que le serveur renvoie un message d'erreur au client si le message à imprimer est vide.

## 1.3 Exercice 2

Vous devez implanter l'architecture présenté dans la Figure 2 en la programmant grâce à l'API Java et en utilisant le fichier exemple dans src.

## 1.4 Exercice 3

Vous devez implanter la même architecture avec l'ADL Fractal.

## 1.5 Exercice 4

Vous devez implanter un nouvel Handler (`HttpHandler`) qui implante un point de montage vers un autre site web (vous partez de la version initiale du serveur http comanche). Pour cela, vous devez copier et modifier le fichier ADL Fractal de la version initiale. Tester que le nouvel

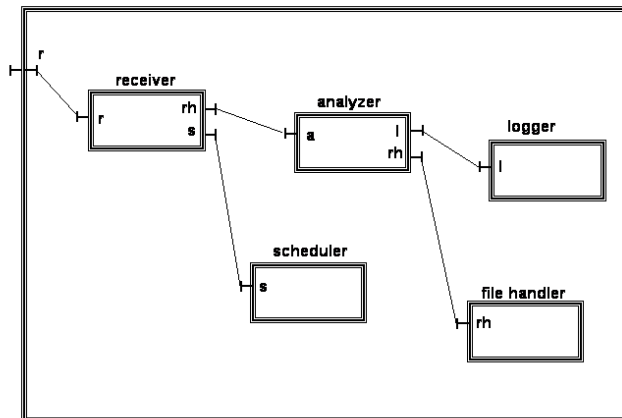


FIGURE 2 – Architecture simplifiée

handler permet de charger une page web d'un autre site web. Pour la redirection vers une page web, vous pouvez utiliser :

```

r.out.print("HTTP/1.0 200 OK\n\n");
r.out.print("<html><meta http-equiv=\"refresh\" ;
            content=\"0 url=http://www.google.fr\"></html>");

```

## 1.6 Exercice 5

Vous devez implanter un nouvel Analyzer qui, lorsqu'il voit passer la requête `http://localhost:8080/reconf`, ajoute le handler précédent (`HttpHandler`) qui n'est initialement pas présent. On implante ainsi une reconfiguration dynamique de l'application Co-manche. Pour cela, copiez les sources et les descriptions ADL d'avant et modifier le code de `RequestAnalyzer`. Vous pouvez vous aider de la partie 5 de l'adresse <http://fractal.objectweb.org/tutorial/index.html>.

## 1.7 Exercice 6

Vous devez proposer une modélisation et une implémentation en Fractal de l'exemple du photocopieur vu en TD. Vous pouvez choisir d'utiliser soit l'API Java, soit Fractal ADL pour spécifier le modèle à composants. Vous êtes libres de mettre en place toutes les fonctionnalités du photocopieur que vous souhaitez.

# 2 Notation

Vous rendrez les différents exercices sous forme d'une archive (nommée `nom_prénom.zip`) contenant un dossier par exercice. Pour les exercices 1 à 5, pensez à commenter vos spécifications Fractal et à expliquer vos choix de modélisation. Pour l'exercice 6, vous devez ajouter, dans le dossier de l'exercice 6, un rapport présentant votre modélisation et votre implantation

du photocopieur. Les différentes interfaces et classes Java que vous avez mises en place doivent être listées et leurs rôles expliqués.

Ce travail devra être rendu pour le dimanche 29 novembre, 23h55. Vous devez déposer l'archive dans le dépôt de fichier moodle (Dépôt du TP).