

Projet n°3 (commun avec INFO0504)

Les usines du Père Noël

Le but de ce projet est de réaliser une application distribuée constituée d'applications *Java* et *PHP*. Les communications entre les applications *Java* sont chiffrées et chacune d'entre elles doit posséder un propre certificat. Pour cela, une autorité de certification unique est mise en place.

Présentation de l'application

Longtemps, nous avons cru que le Père Noël fabriquait les jouets dans une unique usine de lutins. Mais avec l'accroissement de la population humaine et la diversification des cadeaux, il a fallu créer plusieurs usines réparties dans différentes zones géographiques. Toute la difficulté pour ce personnage mystérieux : contrôler les usines et gérer les commandes des ~~naïfs~~ enfants.



Portrait robot du fameux Père Noël

Architecture de l'application

Le père Noël se connecte via un portail *PHP*. Une fois connecté, il récupère la liste des usines et peut en sélectionner une. Il voit alors la liste des articles créés dans l'usine. Il peut ensuite sélectionner chaque article pour voir le stock actuel, lancer la création de nouveaux articles, etc. Le portail *PHP* n'utilise que deux pages : l'une pour la phase de connexion, l'autre pour tout le reste du site (utilisez *AJaX*).



Les enfants et les chefs d'usine lutins se connectent également au portail *PHP*. Mais nous ne développerons pas cette partie dans ce projet.

Un serveur d'authentification en *Java* gère les clients avec leurs droits : vérification du couple login/mot de passe d'un client, création d'un client, récupération des informations d'un client, etc. Il attend des requêtes en HTTP.

Un serveur nommé *backoffice* fait le lien entre le portail PHP et les usines. Chaque usine, à son démarrage, s'enregistre sur le *backoffice* qui possède ainsi la liste des usines en mémoire. Il sert également de lien lorsque le portail PHP envoie des requêtes (liste de produits, lancement d'une commande, etc.). Il reçoit des requêtes en HTTP.

Les usines reçoivent également des requêtes HTTP : liste des produits, commande d'un produit, etc. Le programme, identique pour chaque usine, récupère les produits, le stock et les commandes depuis des fichiers JSON.

Dans un souci évident de neutralité de par la criticité de ses activités, le Père Noël ne peut faire appel à une autorité de certification tierce. Il possède donc sa propre autorité. Celle-ci utilise le protocole TCP (cf TP 13).

Vous devez également respecter les conditions suivantes :

- Toutes les applications *Java* doivent avoir un certificat.
- Toutes les applications *Java* qui communique avec une autre application *Java* doit récupérer et vérifier le certificat de son interlocuteur.
- Les communications entre le *backoffice* et les usines sont chiffrées en utilisant AES.
- Toutes les communications utilisent le format JSON (sauf pour la phase de connexion entre le navigateur et le portail).
- Toutes les applications *Java* utilisent des fichiers de configuration.
- Toutes les données (fichiers de configuration, articles, commandes, etc.) sont stockées dans des fichiers JSON.

Travail à réaliser (INFO0503)

1. Décrivez le format des fichiers JSON nécessaires pour stocker les données et donnez l'arborescence des fichiers utilisés.
2. Décrivez pour chaque opération le format JSON des messages échangés. N'oubliez pas les cas d'erreur.
3. Décrivez l'ensemble des diagrammes d'échange entre les entités de votre système.
4. Écrivez l'ensemble des applications demandées.



Vous joindrez à votre projet un rapport présentant et expliquant l'ensemble des modélisations demandées. Vous ajouterez également un mode d'emploi : comment et dans quel ordre chaque application doit être exécutée.

Travail à réaliser (INFO0504)

Afin d'expliquer l'architecture de votre système, vous allez proposer un diagramme de classes de ce dernier. Il n'est pas demandé un diagramme détaillé de toutes vos classes (avec les attributs, accesseurs, ...) . L'objectif est de fournir l'architecture globale de votre système indépendamment du (des) langage(s) utilisé(s). Il devra être suffisamment informatif pour qu'un développeur puisse aisément le mettre en œuvre dans les langages de son choix.

En plus du diagramme de classes de votre système, vous allez proposer, pour chaque serveur, un diagramme de cas d'utilisation qui permettra d'identifier les fonctionnalités de celui-ci. Vous ferez notamment attention à bien mettre en évidence les acteurs interagissant avec le serveur.