POLYTECH NICE-SOPHIA

Rapport de DevOps

Découpage en composants

DALIE Basil, LAFAURIE Paul, LEQUIENT Steven, MERINO Mathieu

I. Introduction

Dans le contexte de ce second travail pour le module DevOps, nous avons eu à récupérer le code source du projet Cookie Factory afin de proposer un nouveau découpage en regroupant les composants en fonction de leurs caractéristiques communes et de la manière dont ils interagissent entre eux.

II. Description du découpage

D'après la lecture du diagramme de composant de Cookie Factory, nous sommes parvenu au découpage suivant :

- 4 artifacts "de base" que sont utils, interfaces, exceptions et entities.
- 3 packages pour les components: catalogue, customerRegistry et cashierCartKitchen.
- 2 packages pour les webservices: cartWebService et customerCareService

Les 4 artifacts "de bases" regroupent un ensemble de classes importées par plusieurs composants à la fois, nécessaires au fonctionnement du programme. Il s'agit des classes communes. Par exemple, les exceptions sont les mêmes du début à la fin de l'architecture j2e (puisque les exceptions peuvent remonter de méthodes en méthodes à travers les composants). Les entités également sont utilisées de partout.

Concernant les composants relatifs au modèle de l'application (Cart, Cashier, Kitchen, CustomerRegistry et Catalog) nous avions initialement décidé de les regrouper en un composant. Mais finalement, nous avons décidé d'isoler Catalog et CustomerRegistry dans leur artifact car, d'un point de vue fonctionnel, ils sont sans liens avec le reste du code,

à part lorsqu'ils sont utilisés par un des deux WebService. En revanche, nous avons regroupé les composants Cashier, Cart et Kitchen car comme on peut le voir sur le diagramme d'architecture de Cookie Factory, ils sont assez liés (Cart et Kitchen utilisent Cashier) dans le code métier. Ainsi on obtient un unique composant CashierCartKitchen, qui expose deux interfaces (CartModifier et Tracker) aux web services.

Enfin nous avons isolés les deux web services dans leur packages respectifs car chacun représente un bout du programme assez indépendant et cela permettra par exemple de déployer les deux services sur deux machines différentes.

III. Conclusion

Le découpage effectué permet ainsi à des équipes différentes de travailler indépendamment, sans avoir de conflits entre les parties du projet.