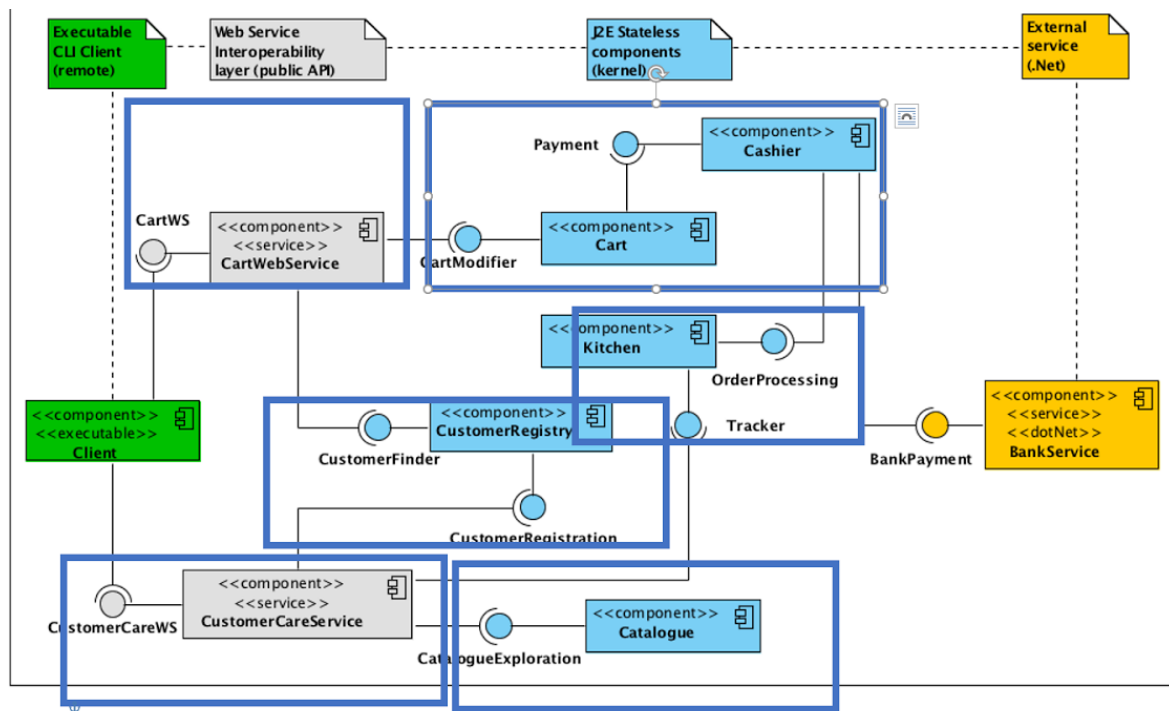


DevOps & Continuous Testing

Robin ALONZO – Khadim GNING – Thomas JALABERT – Rémy KALOUSTIAN

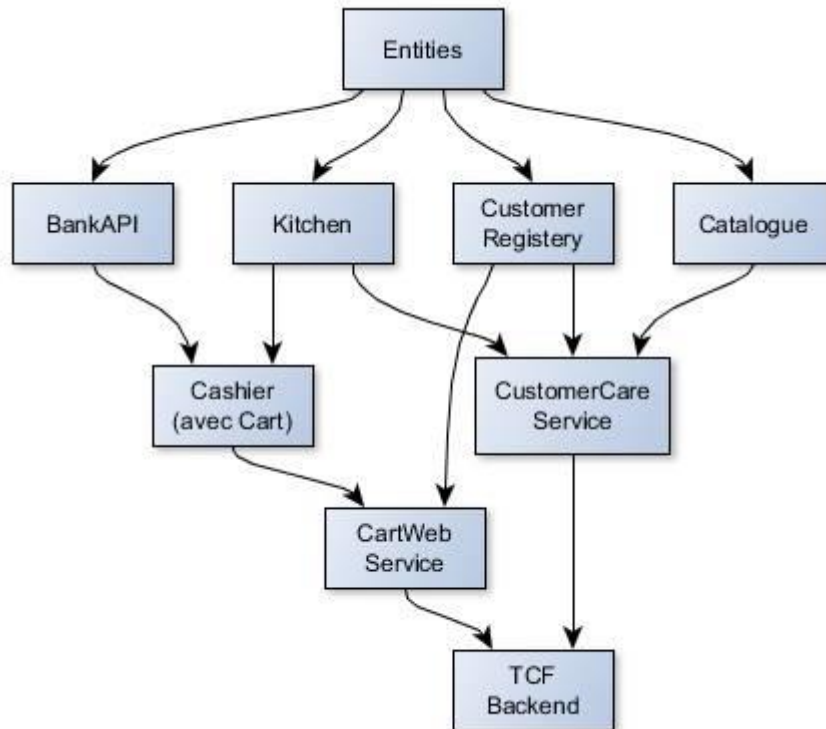
Nous avons souhaité modulariser le projet Cookie Factory, afin de pouvoir utiliser un serveur de build. Cette manière de procéder permet de proposer une compilation par étape, ainsi en cas de modification d'un module en bout de chaîne de compilation nous n'avons pas à tout recompiler. De plus, la modularisation permet d'identifier les sources d'erreur.



Nous avons choisi de découper les projets en différents modules correspondant aux fonctionnalités. Nous avons opté pour ce découpage tout d'abord parce qu'il nous semblait pratique de regrouper les composants qui s'occupent d'une même partie ensemble. En effet, supposons que l'on est dans un cadre professionnel avec plusieurs équipes de développement. Si chacune des équipes traite un module, il est beaucoup plus pratique qu'une équipe travaille sur une même fonctionnalité et sur les mêmes composants, cela nous évite de perdre du temps de communication et de synchronisation du travail.

Dans cette optique, nous pouvons compiler les modules indépendamment. Par ailleurs, les modules BankAPI, Kitchen, Catalogue, et CustomerRegistry peuvent être construits en parallèle.

Ci-dessous, les dépendances de nos modules :



Ensuite, cette approche nous permet de réaliser un découpage qui limite les dépendances. Nous organisons les modules en fonction des interfaces qui leur serviront, ce qui explique que la plupart de nos modules ne contiennent qu'un composant. Nous avons volontairement laissé Cart et Cashier ensemble car les deux servent à la même chose (payer).

Enfin, nous avons séparé les Services, là aussi, pour séparer les comportements différents. En effet, les Services ne sont pas des objets métiers qui vont contenir une information cruciale, ils agissent plus comme intermédiaires entre le côté client et les données. Si on les compare au pattern MVC, ils agissent plus en tant que "Controller", alors que les composants bleus seraient du "Model".

Un inconvénient que nous avons pu identifier est que nous devons à présent gérer les dépendances de plusieurs modules, contrairement à un module unique comme au départ. Imaginons qu'on veuille ajouter un élément dont tous les modules ont besoin. Dans ce cas-là, en suivant notre logique, nous devrions faire de cet élément un module, ce qui peut devenir assez lourd.

PRISE DE REcul :

Ce projet nous a permis de mieux comprendre l'organisation de cookie Factory, les dépendances liant nos différentes classes et la manière d'utiliser les EJB.