

# DevOps

## The Cookie Factory : Rendu 3

### Groupe 6

Alexandre Bar

Gaspard Lacroix

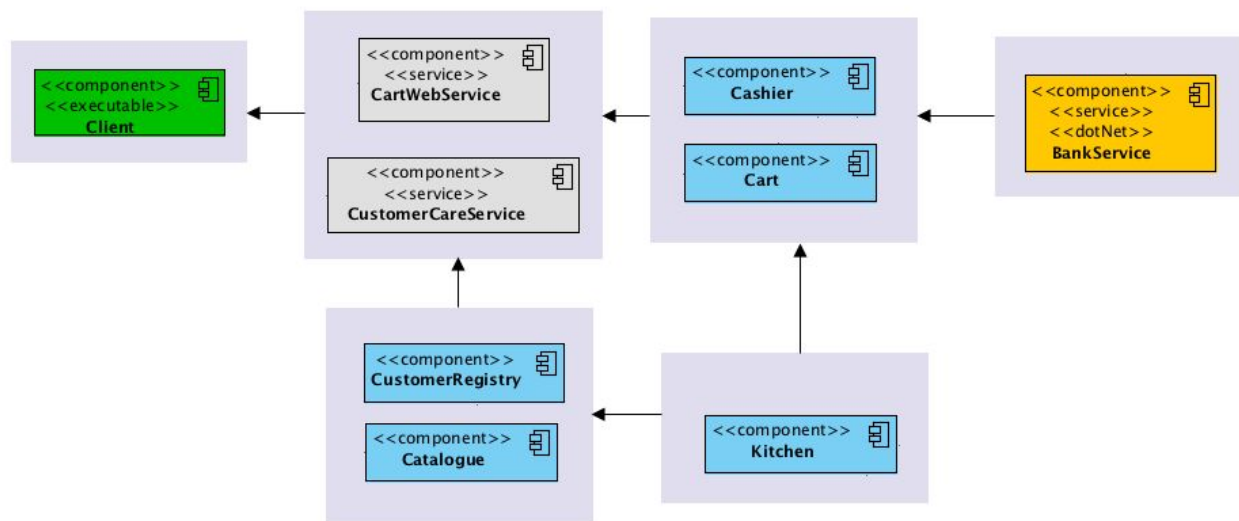
Alexis Metge

Liavona Zheltanosava

Dépôt git : [https://github.com/gaspardL/4A\\_ISA\\_TheCookieFactory](https://github.com/gaspardL/4A_ISA_TheCookieFactory)

## 1. La décomposition

Nous avons dû légèrement revoir notre découpage précédent car notre vision du projet n'était pas assez approfondie lors des premières séances. Ainsi, les *Webservices* ne dépendent plus directement de *Kitchen* mais indirectement au travers de *Cashier-Cart* et *CustomerRegistry-Catalogue*.



## 2. Travail réalisé

Après avoir découpé le projet de Cookie Factory en plusieurs sous-modules, nous avons mis en place un dépôt d'artefacts: Artifactory. Grâce à Maven, nous avons connecté chaque sous-module à ce dépôt afin d'y déposer les builds réussis. Enfin, nous avons configuré Maven afin de récupérer les dépendances directement dans Artifactory. Seul *Kitchen* et *Client* n'ont pas cette configuration spécifique puisqu'ils ne dépendent d'aucun autre module.

Après cette configuration initiale nous avons installé notre serveur d'intégration continue, Jenkins. Nous avons créé un plan de build par module.

Tous +		
S	M	Nom du projet ↓
		<a href="#">cashier_cart</a>
		<a href="#">client</a>
		<a href="#">customer-registry_catalogue</a>
		<a href="#">dotNet</a>
		<a href="#">kitchen</a>
		<a href="#">webservices</a>

La plupart de nos projets utilisant Maven, nous avons utilisé le plugin Maven de Jenkins pour faciliter l'importation de ces modules et la gestion de leurs dépendances. Jenkins fait ainsi en sorte de relancer les plans de build des modules dépendants de ceux qui sont modifiés.

Exemple si on relance le build plan de kitchen

État du lanceur de compilations		
1	Au repos	
2	<a href="#">kitchen</a>	#9 

Les modules dépendants de kitchen se recompilent ensuite

État du lanceur de compilations		
1	<a href="#">customer-registry_catalogue</a>	#8 
2	<a href="#">cashier_cart</a>	#7 

Suivi du module dépendant de ceux-ci, etc ..

État du lanceur de compilations		
1	<a href="#">webservices</a>	#7 
2	Au repos	

Grâce aux plugins Git, Jenkins récupère les sources directement de notre dépôt Git, les compile et les déploie sur Artifactory. Malheureusement, n'ayant pas de serveur externe nous n'avons pas pu, sur le réseau de l'université, lancer le build d'un module lorsqu'un commit modifie son code. Cependant, nous avons programmé une relance des build plan principaux (kitchen et client) toute les heures afin de toucher toute la chaîne.

État du lanceur de compilations		
1	<a href="#">client</a>	#4 
2	<a href="#">kitchen</a>	#7 

### 3. Conclusion

D'après cet exercice, nous avons approfondi les connaissances sur Maven et ses fonctionnalités, notamment sa gestion de dépôts d'artefacts. De plus, nous avons pu découvrir les principes d'intégration continue au travers d'Artifactory et Jenkins. De plus, nous avons pu développer nos compétences sur ces deux outils en les mettant en oeuvre et en les connectant afin d'automatiser au maximum chaque étape du développement.