



De la pureté dans les dépendances

Le concept de **pureté** qui existe pour les **fonctions** peut être étendu aux **dépendances** dans un logiciel, notamment quand ces dépendances sont organisées en **couches** associées à des **domaines de responsabilités**, comme c'est le cas, par exemple, pour l'architecture 3 tiers.

En effet, certains de ces domaines de responsabilités peuvent impliquer de nécessaires entrées et sorties. Depuis et vers des interfaces d'utilisation concernant des couches de présentation. Depuis et vers des supports de stockage concernant des couches de persistance.

Or, qui dit opérations d'entrées et de sorties dit impossibilité de n'avoir que du code pur dans les composants logiciels qui les auront en charge. Donc, de la même façon que l'on peut qualifier d'impures les fonctions qui effectuent des opérations d'entrées et de sorties, on peut qualifier d'impures les couches logicielles qui ont vocation à gérer de telles opérations d'entrées et de sorties.

Si on suppose que les couches qui ne sont pas ainsi directement qualifiable d'impures ont la potentialité d'être pures, alors il découle qu'on peut noter le fait qu'une architecture logicielle impose ou non des dépendances qui induiront des contaminations d'impureté.

C'est par exemple le cas dans l'architecture 3 tiers avec la dépendance qui va de la couche logique métier vers la couche persistance. Sans cela, la couche logique métier aurait le potentiel d'être composé de code pur. Mais comme elle devra utiliser le code nécessairement impur de la couche persistance, elle se retrouvera elle-même impure par contamination.

La présence ou non de telles dépendances contaminantes peut être un critère que l'on se donnera, parmi d'autres, pour évaluer les architectures que l'on voudra utiliser pour le développement de nos logiciels.