

Question 2a : *Les logiciels modernes sont développés selon des architectures logiques comportant plusieurs couches ou niveaux (multi-tier architecture). Indiquez deux avantages importants de la décomposition d'un logiciel en une architecture multi-niveaux.*

Un premier avantage de la décomposition d'un logiciel en une architecture multi-niveaux est la séparation des composantes qui se retrouvent dans le logiciel. Cette séparation permet d'exécuter l'intégralité du logiciel sur plusieurs machines, donc de réduire les erreurs en cas de dysfonctionnement d'une machine s'il y a plusieurs instances d'un même processus. Elle permet aussi l'exportation d'un niveau dans un logiciel différent. Un deuxième avantage de la décomposition en multi-niveaux est la parallélisation du travail qui peut être accordé sur chaque niveau. Étant donné que chaque niveau est indépendant, différentes équipes de développement peuvent travailler en même temps sur chaque niveau sans avoir à attendre qu'une tâche différente de la leur soit atteinte pour tester leur travail.

Question 2b : *En vous basant sur votre diagramme de paquetages, expliquez à quel niveau architectural vous associez la fonctionnalité liée au « Mécanisme d'apprentissage dans l'interface pour proposer des valeurs de défaut "intelligentes" en fonction des choix précédents de l'utilisateur ». Justifiez votre réponse en discutant des possibilités de réutilisation de cette fonctionnalité dans une autre application et de la possibilité que cette fonctionnalité ne soit pas développée par vous, mais plutôt récupérée d'une source externe au projet.*

Le niveau architectural associé au mécanisme d'apprentissage dans l'interface est le niveau de la logique de l'application. Plus précisément, au niveau d'un service du contrôleur. Les informations reliées à un utilisateur seront propres à lui-même. Si plusieurs personnes utilisent le logiciel sur le même ordinateur, les suggestions devraient être appliquées en fonction de l'utilisateur courant et non de l'utilisation du logiciel sur l'ordinateur. En modifiant les champs à superviser pour l'apprentissage, cette fonctionnalité pourrait être utilisée dans une autre application, donc un service semble assez mobile pour permettre l'exportation de cette fonctionnalité. Les informations uniques à chaque utilisateur seraient conservées dans le modèle propre à ce dernier, ce qui lui permettrait de recevoir des propositions adaptées. En récupérant cette fonctionnalité d'une source externe, il faudrait tout de même associer une équipe de développement qui se chargerait d'initialiser le service pour que l'apprentissage du service utilisé par l'utilisateur soit conforme aux objectifs de la tâche.