

Description de l'architecture :

Description des dépendances (diagramme de paquetage) :

Les vues de notre application, contenues dans WpfApp1, dépendent de la Persistence, du Business, du Modèle et de la StubLib (temporairement). La Persistence contient toutes les classes utiles à cette dernière (**DataContractPersistence**, *IPersistenceManager*, **DataToPersistReq**, **DataToPersistUsers**) et dépend du modèle puisqu'elle contient des listes des classes du Modèle. Le Business contient les managers (**Manager** et **ManagerUsers**) et dépend de la Persistence, puisqu'un manager contient une instance de **IPersistenceManager**, et pareil pour le **Stub** même s'il n'en a eu besoin que temporairement. Le Modèle contient toutes les classes propres à notre application et ne dépend de rien, puisqu'il est à la base du code. La StubLib contenant le **Stub** dépend également du Modèle, puisqu'il contient des listes des classes de notre application. Enfin, les UnitTests dépendent du Modèle, puisqu'ils permettent de tester les classes de l'application.

Patrons de conception utilisés :

Pour notre application, nous avons d'abord utilisé le patron de conception de Stratégie, qui permet à un objet d'effectuer plusieurs traitements différents en fonction d'une variable, comme c'est le cas pour notre persistance, dans laquelle nous avons une interface qui permet plusieurs instances de la persistance (le stub et le data contract en xml) en fonction de la variable strat de notre application, qui choisit l'implémentation d'*IPersistenceManager* à utiliser. Ensuite, nous avons utilisé le patron de conception de l'observateur, notamment avec nos collections de **Requin** *ObservableCollection*, qui nous permet de notifier un changement d'état et d'agir en conséquence.