Description du diagramme de cas de la persistance :

Toutes les classes de persistances possèdent les mêmes méthodes de chargements et de sauvegarde. Seul diffère l’algorithme derrière. Il est donc intéressant de mettre en place le patron de conception stratégie, avec l’interface IPersistanceManager. De plus cela permet plus facilement de prendre plus facilement une source de persistance plus facilement.

L’interface IPersistance, outre le fait qu’elle est importée par Manager, elle précise aussi les méthodes à avoir, SauvegarderDonnées prendra en paramètres toutes les listes de données (la liste des ContenusVideoludiques, et des Utilisateurs, afin de les enregistrer en XML.

ChargeDonnées renverra des IEnumerables de ContenusVideoludique et Utilisateurs, pour charger les données.

De plus, Stub et DataContract2XML exploite la stratégie du singleton, car il est inutile d’avoir plusieurs instances de ces classes, une est amplement suffisante. Ainsi, chacune de ces classes possèdes un attribut privé singleton, possédant la référence de l’instance de la classe (ou bien null si aucune instance n’existe). Le constructeur est dès lors privé, et c’est dans Getnstance que l’on appelle le constructeur si jamais le singleton est null, sinon il renvoie l’instance déjà existante.

La sauvegarde de données sur le stub ne sauvegardera rien, étant donné que le stub est un chargement de données en dur, statique.  
DataContract2XML sauvegardera dans des fichiers XML, et chargera les données depuis les fichiers crées.