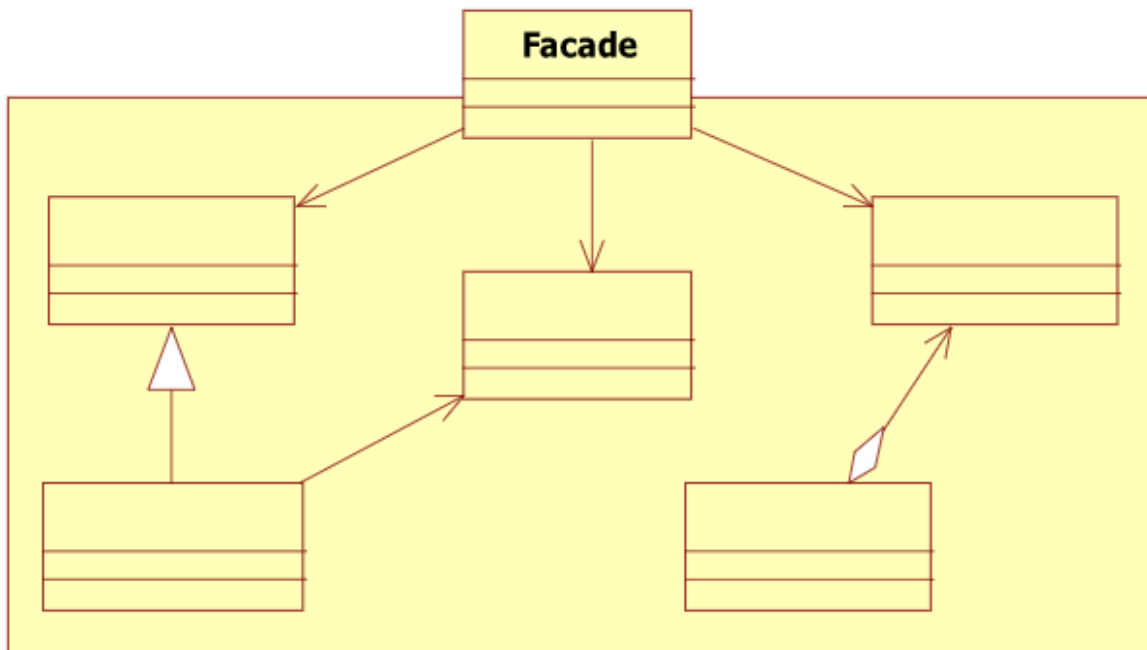


Description de l'architecture

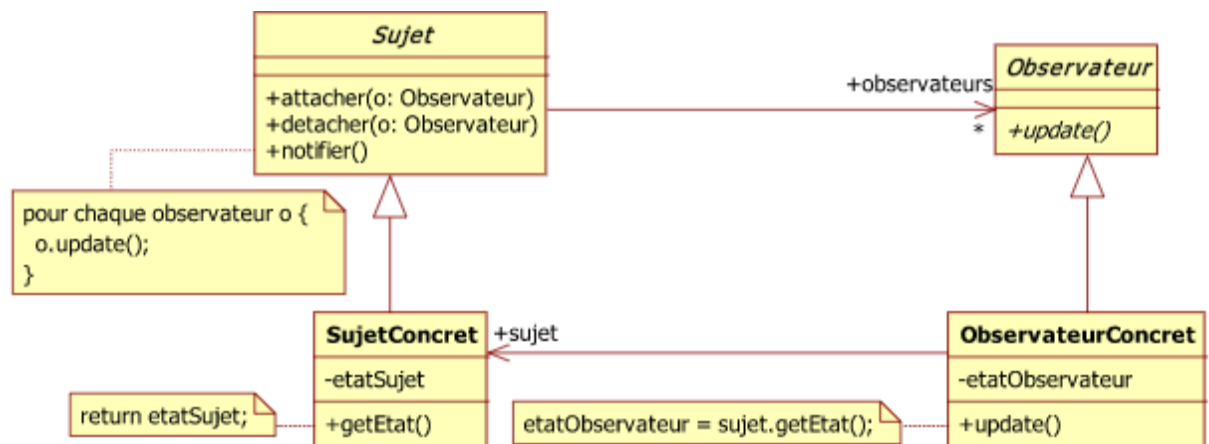
Patrons de conceptions utilisés :

- La Façade :



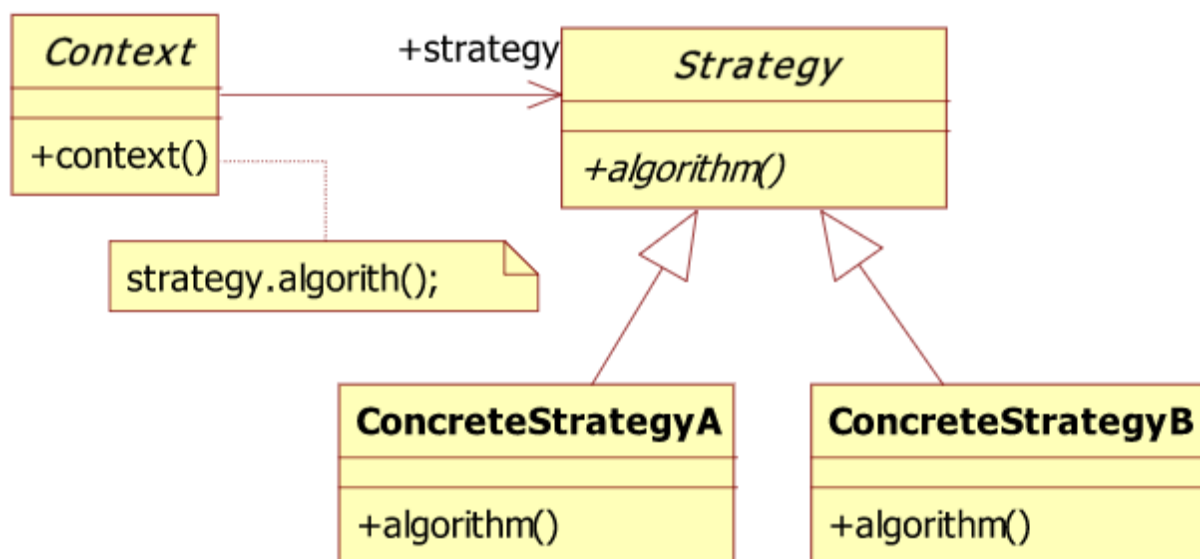
Le patron de conception Façade a pour but de cacher une conception et une interface ou un ensemble d'interfaces complexes difficiles à comprendre. Nous avons utilisé un Manager(ProfilManager) nous permettant de rendre le sous-système sécurisé et plus facile à utiliser. La Façade met à disposition des méthodes faisant fonctionner l'application, cependant les méthodes de la Façade utilisent la logique du sous-système. Cela propose une vue par défaut du sous-système et permet d'être moins dépendant du sous-système.

- L'Observateur



Le rôle de l'observateur est de ne pas faire de l'attente active pour savoir quand il y a eu un changement. Cela permet de notifier le sujet via l'observateur quand l'état souhaité est atteint. Nous utilisons une classe `ObservableObject` qui implémente l'interface `INotifyPropertyChanged` ce qui nous permet de simplement faire hériter une autre classe d'`ObservableObject` pour la rendre Observable. Ensuite les classes ayant besoin de notifier la vue, implémentes `ObservableObject` pour être à leur tour Observable et pour pouvoir notifier la vue en cas de changement. Le fonctionnement est donc que quand il y a un changement dans les données on peut notifier la vue via la méthode `OnPropertyChanged()` pour que celle-ci se mette à jour.

- La stratégie



Le rôle de la stratégie est de définir une famille de classes et de les rendre interchangeable.

Nous avons utilisé une interface `IPersistenceManager` pour pouvoir injecter différents types de persistance juste en changeant le paramètre lors de l'injection. En effet les différentes classes de persistance possèdent le même but mais différent dans leurs comportements. Cela laisse donc le choix dans la persistance utilisée en un minimum d'effort.

Nous avons découpé notre projet en 5 sous projet :

Persistence :

Contient nos différents types de persistance

Tests_fonctionnels :

Contient les tests fonctionnels de notre application

Test_Unitaires :

Contient les tests unitaires de notre application

Class :

Contient les classes C# de notre application

Master_Streaming :

Contient les vues xaml de notre application