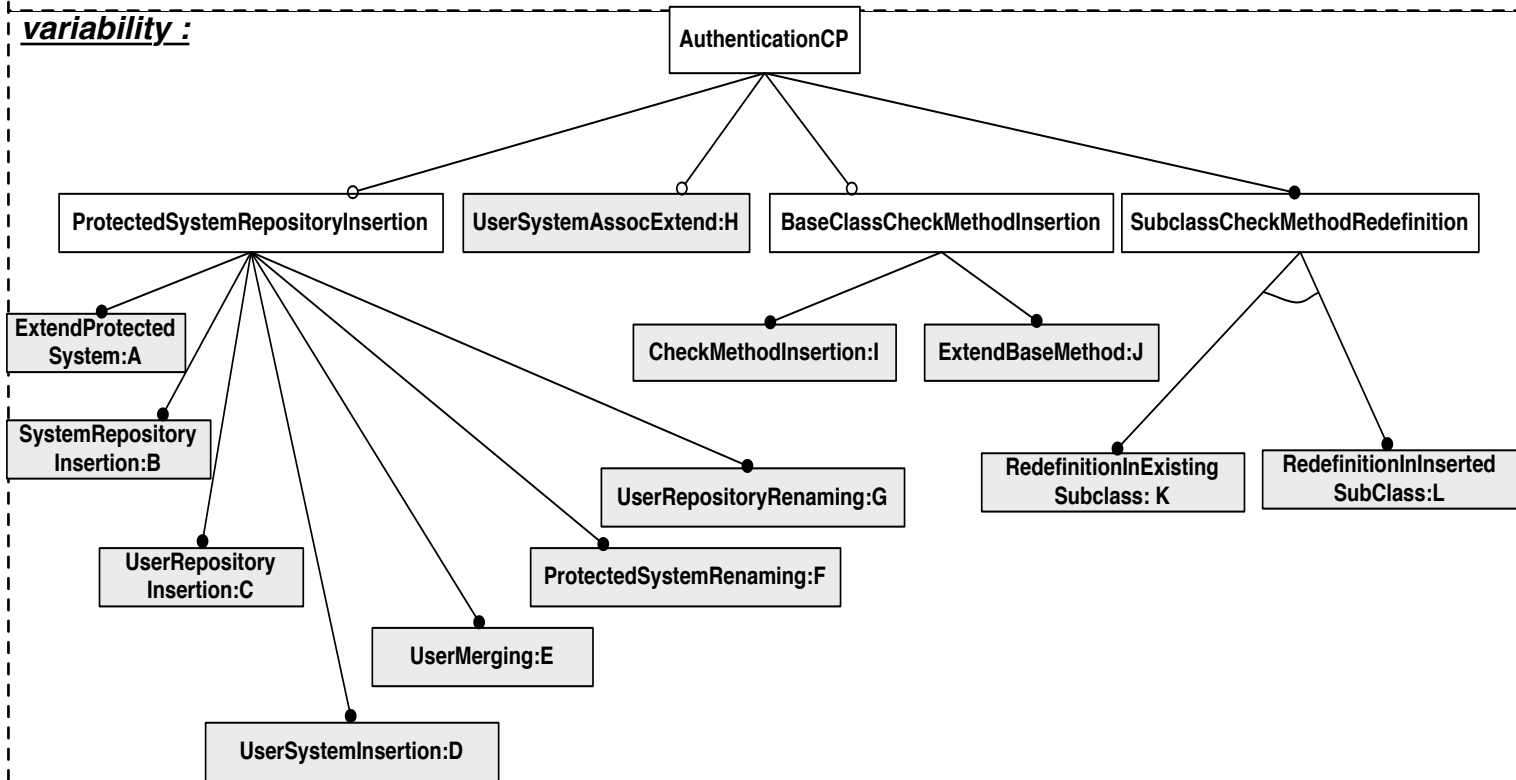


AuthenticationAdapter <<DerivableAdapter>>

pointcut model :

AuthenticationPointcutModel

variability :



constraints :

- RedefinitionInInsertedSubclass => ProtectedSystemRepositoryInsertion \wedge BaseClassCheckMethodInsertion
- RedefinitionInExistingSubclass => **bounded** ProtectedSystemTargetClass
- UserSystemAssocExtend => **bounded** GenUserTargetClass

adaptations :

A	extend class ToProtectTargetClass with descendant ProtectedSystem
B	insert association systemRepository
C	insert association userRepository
D	insert association userSystem
E	merge classifier UserToAuthenticate in UserTargetClass
F	modify name ProtectedSystem with "Protected" + ToProtectTargetClass.getName() + "4" + UserTargetClass.getName() alias NewProtectedSystem
G	modify name UserRepository with UserTargetClass.getName()+"Repository"
H	extend association userSystem with targetUserSystem
I	create method (methodToProtect.getName() + "Check") in ProtectedSystemTargetClass alias checkMethod with body { return false; }
J	extend method (methodToProtect) in ToProtectTargetClass with before { if (call(checkMethod) == false) then logError("Unauthorized Access"); }
K	create method (methodToProtect.getName() + "Check") in ProtectedSystemTargetClass with body { return authenticated; }
L	create method (methodToProtect.getName() + "Check") in NewProtectedSystem with body { return authenticated; }