

## 1.2 Méthodologie de développement



A compléter

Schémas et fonctionnement en modules

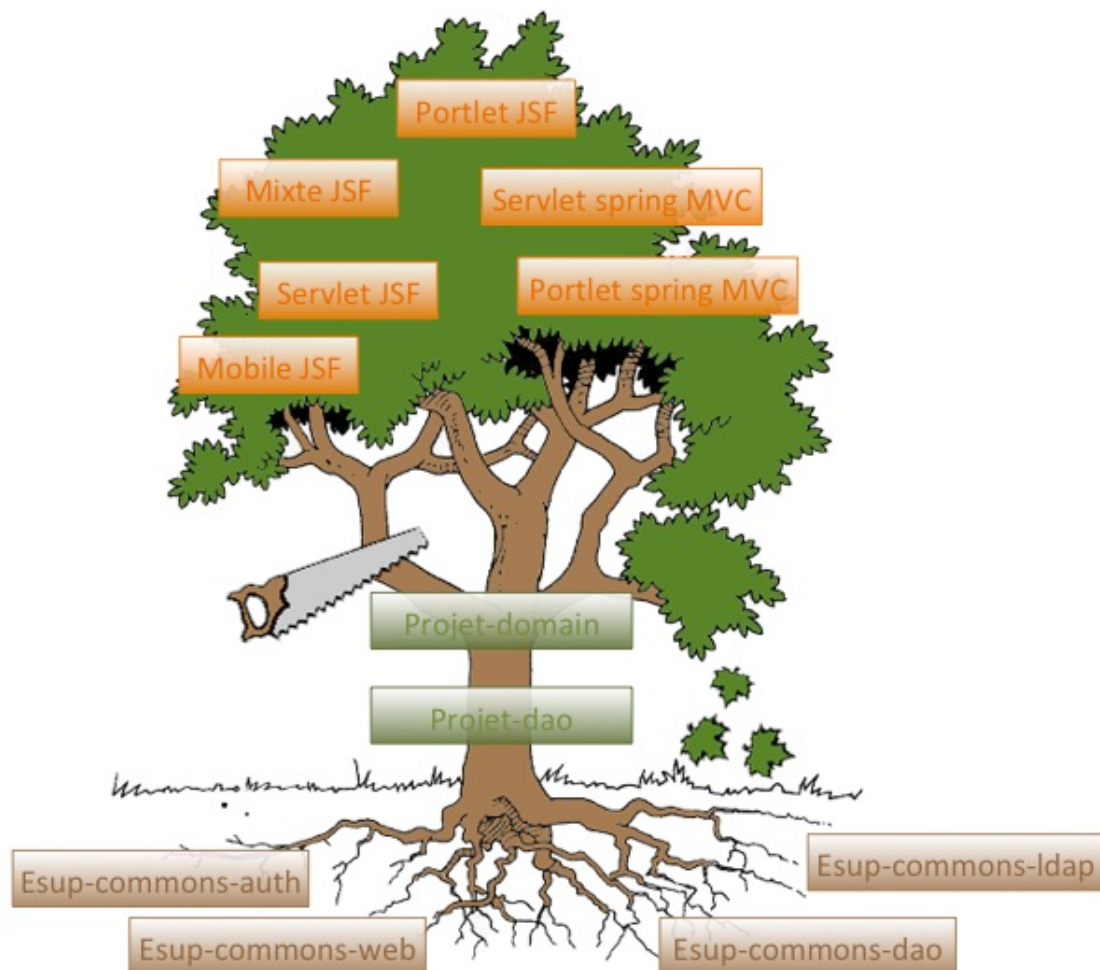
### Sommaire :

- [Ce que permet \(ou permettra\) esup-commons](#)
- [Les notions de blank et exemple](#)
- [Le fonctionnement en modules](#)

### Ce que permet (ou permettra) *esup-commons*

... des applications web :

- portlet
- servlet
- mixte (servlet et portlet)



### Les notions de *blank* et *exemple*

Dans sa deuxième version, *esup-commons* adopte une méthodologie complètement différente de la première. Auparavant, le développement d'un projet passait par le checkout SVN d'un projet "blank" qui devait dépendre du projet *esup-commons*, lui-même récupéré par un checkout SVN. Voir ([03 Méthodologie de développement](#))

Désormais, un projet *esup-commons* se base sur Maven et la notion d'archetype.

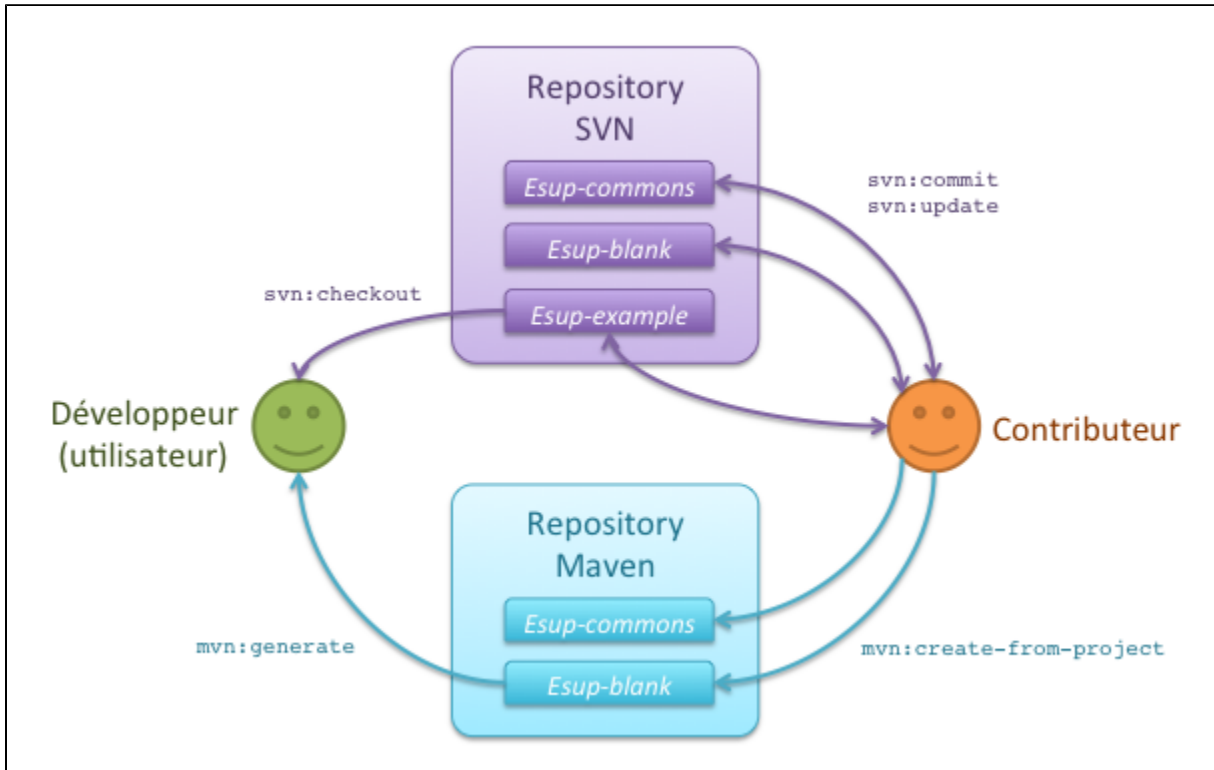
Pour démarrer un nouveau projet, on partira sur la base d'un archétype *blank* qui, grâce au mécanisme de dépendance Maven, intégrera *esup-commons* sous forme de jar.



Mettre un écran structure vide

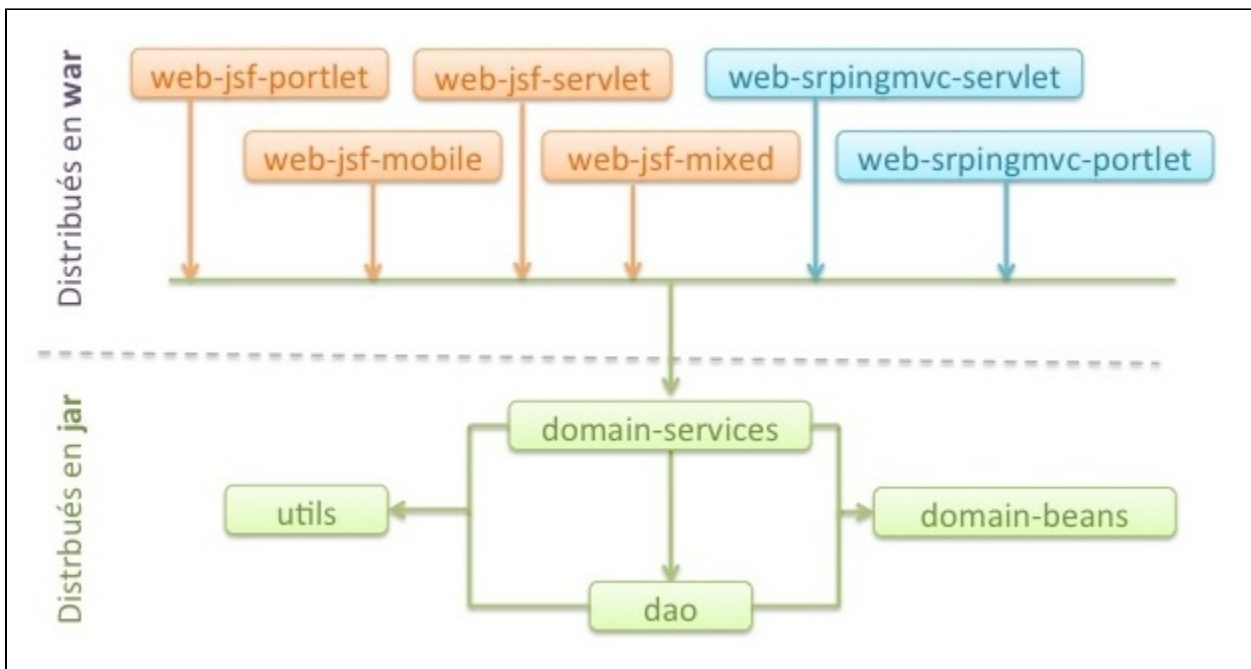
La notion de projet d'exemple (*example*) existe toujours. Cette application a pour but de présenter une implémentation simple d'un projet *esup-commons* et quelques fonctionnalités incontournables d'une application. Cette dernière sera récupérée par un checkout SVN.

**Attention :** les projets *esup-commons* et *esup-blank* disponibles sur le SVN ne sont utilisés que par les contributeurs du projet *esup-commons*



## Le fonctionnement en *modules*

Un projet *esup-common* se décompose en sous-projets à part entière appelés "modules" au sens Maven et dépendants les uns des autres .



Au packaging, chaque module sera proposé sous forme soit d'un fichier jar pour les couches basses, soit d'un fichier war pour la présentation.

- WEB-JSF-SERVLET **war**
- WEB-JSF-PORTLET **war**
- WEB-JSF-MOBILE **war**
- WEB-JSF-SHARED **jar** => Controllers
- DOMAIN-SERVICES **jar** => Les services métiers (authenticator, accès au WS, exposition de WS)
- DOMAIN-BEANS **jar** => Les objets métiers
- DAO **jar** => Accès aux données
- UTILS **jar** => Utilitaires transverses à toute l'application