# Gestion de contenu – pièces jointes - images

- ISM doit permettre de stocker des fichiers de toutes nature sur un « système de stockage » serveur

- Les répertoires du « système de stockage » sont configurables, dans le sens où l’on peut en définir plusieurs

- Un répertoire de stockage possède les caractéristiques suivantes :

* Racine de stockage (ex: D:/ismdb/store, /local/ism/store,…)
* Un quota permettant de limiter la capacité qu’ISM utilisera sur ce filesystem
* Un attribut ‘actif’ pour pouvoir interdire le stockage momentanément

- Le « système de stockage » doit s’assurer que l’on puisse stocker de très nombreux fichiers (> 100 000) sans générer de baisses de performance

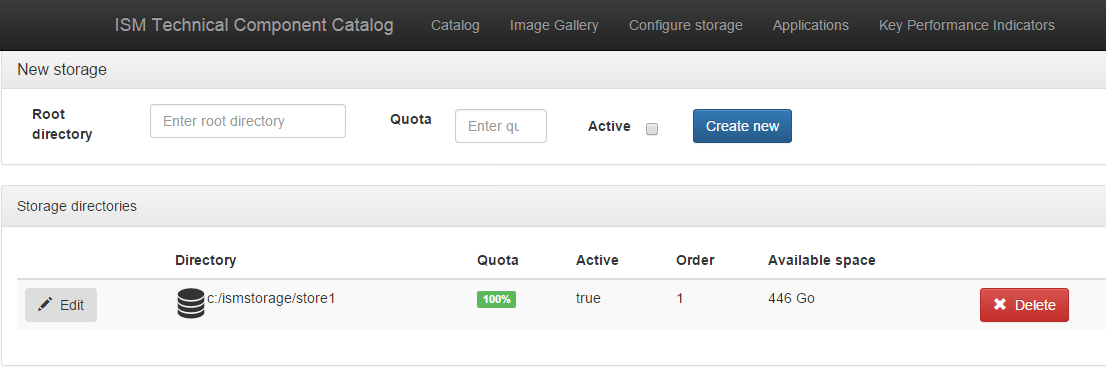
- Dans le cas où plusieurs répertoires de stockages sont définis, le « système de stockage » doit permettre les stratégies suivantes d’utilisation de ces répertoires :

* On remplit le premier jusqu’au quota puis le second et ainsi de suite
* On choisit au hasard les répertoires (en sélectionnant avant les répertoires pouvant stocker la taille du fichier)

- Le « système de stockage » doit permettre de stocker plusieurs versions d’un même fichier

- ISM doit permettre de rendre un fichier accessible même si le répertoire de stockage n’est plus référencé dans les répertoires disponible (on peut supprimer un répertoire de stockage du point de vu ISM sans le détruire physiquement)

## Copies d’écran



# Gestion d’un catalogue de composants techniques

- ISM doit permettre de gérer un catalogue de composants techniques

- Le catalogue doit pouvoir être organisé en catégories (de manière récursive)

- Un composant technique a les caractéristiques suivantes :

* Informations « vendeur »
  + Nom
  + Version
  + Date de mise à disposition
  + Date de dépréciation du composant
  + Date de fin de support
* Informations « locales » (celle spécifiques à une entité)
  + Nom
  + Version
  + Date de mise à disposition
  + Date de dépréciation du composant
  + Date de fin de support
* Un forçage d’obsolescence (cf. exigence sur le calcul de l’obsolescence)
* Une image permettant de « reconnaitre » le composant
  + Une catalogue d’images dédiées aux composants doit être mis en place sur la base du système de stockage précédemment décrit

- ISM doit permettre de calculer l’obsolescence d’un composant en fonction des informations « locales ». L’obsolescence est caractérisée par des « zones temporelles » :

* La zone « available » : le composant est disponible et ne pose pas de « soucis »
* La zone « warning » : on s’approche de la date de fin de support local, il va falloir migrer ce composant dans les applications qui l’utilise
* La zone « hot » : on est très proche de la date de fin de support « local »

- (option) ISM doit permettre de rendre configurable les zones d’obsolescence

- ISM doit permettre de consulter l’ensemble des composants qui sont dans chaque « zone d’obsolescence » à une date quelconque (aujourd’hui, dans 6 mois,…)

- ISM doit permettre d’associer les composants techniques à une version d’application

* 2 natures de liens doivent être gérées :
  + « Est fabriqué à partir de » (ex. jdk1.8, Spring 4.0.1)
  + « Peut tourner sur » (ex. Websphere 7, Tomcat 8, Oracle 11i,…)
* Lors de la copie d’une version d’application, on copie les liens vers les composants techniques

- ISM doit permettre de calculer l’obsolescence d’une version d’application. Celle-ci sera la « moins dissante » des composants techniques associés

## Copies d’écran

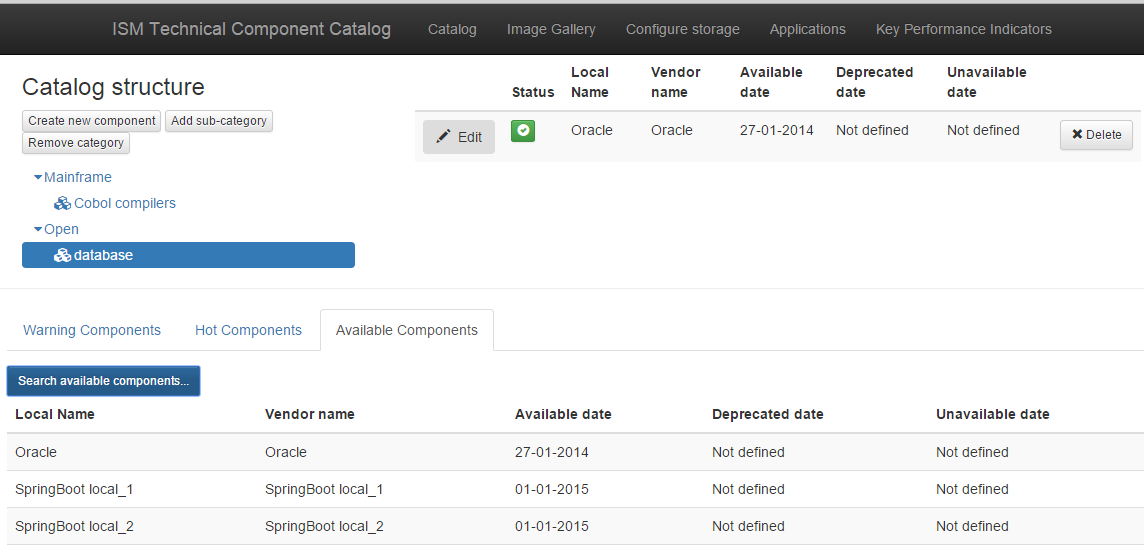


Figure 1 Catalogue des composants (arbre à revoir)

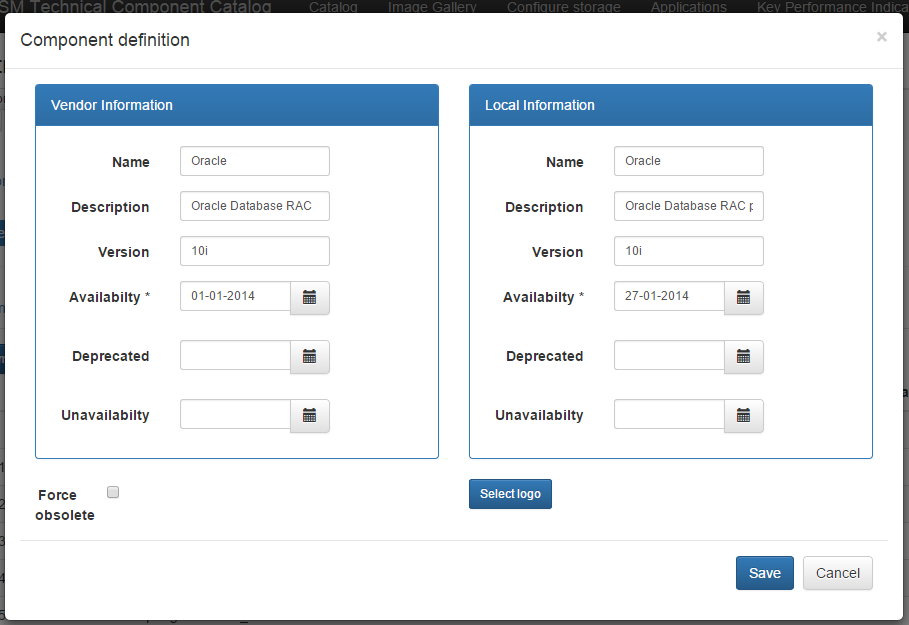


Figure 2 Définition d'un composant

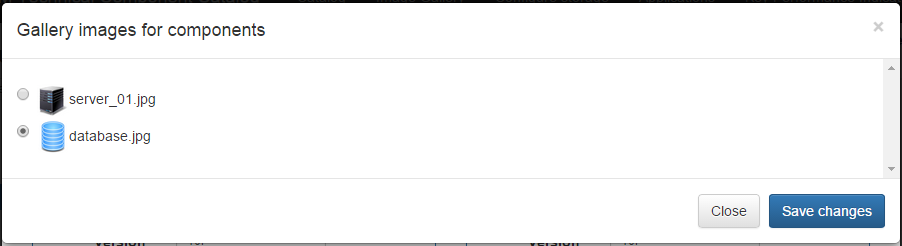


Figure 3 Galerie d'images pour les composants

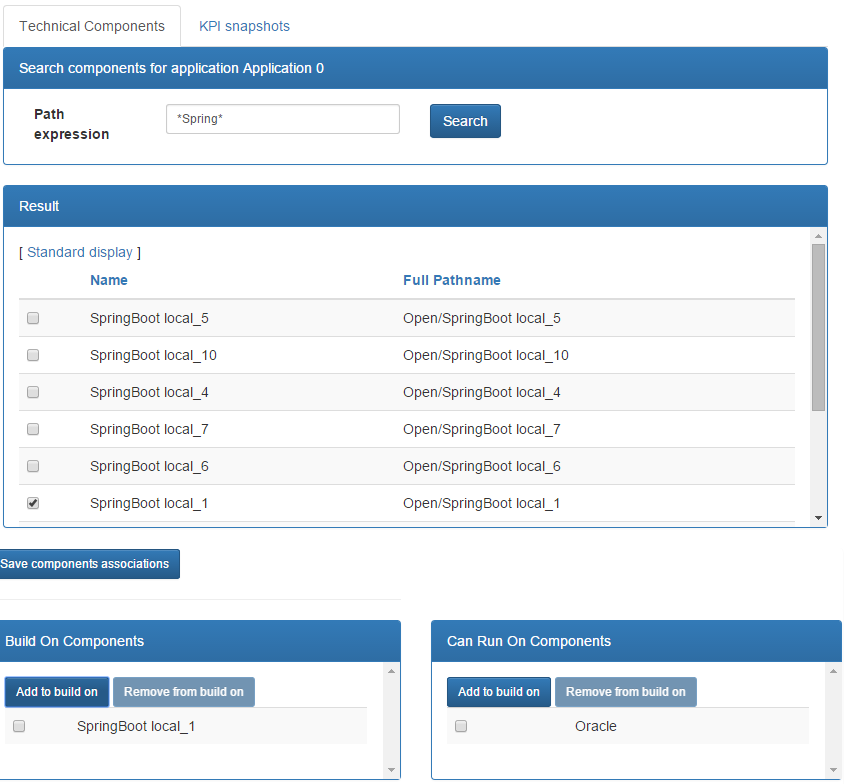


Figure 4Bon ok cette partie n'est pas très belle - à revoir

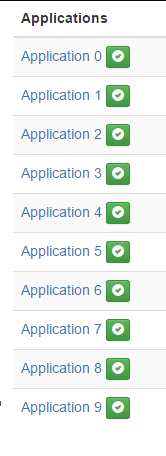


Figure 5 Statuts des applications

# Définition de Key Performance Indicators

- ISM doit permettre de créer de gérer 2 sortes de KPI :

* KPI définis par l’entité et dont les valeurs seront saisies manuellement (KPI « manuelle »)
* KPI « embarquées » dans ISM qu’ISM sait calculer car il a toutes les informations pour le faire (KPI « automatique »)

- Les KPI définies par l’entité peuvent être de 2 natures :

* Valeur numérique avec intervalle [min, max] optionnel. La valeur pouvant être soit entière soit à virgule (choix à faire)
* Valeur numérique mais « discrète » (~énumération)

- Chaque KPI quelle que soit sa nature possède :

* Un nom
* Une description
* Un caractère « Actif » (elle ne peut être utilisée que si elle est « active »)
* Une catégorie (au choix, pas de gestion de liste dans ISM)

- ISM doit permettre, pour chaque version d’application de définir plusieurs « mesures ». Une « mesure » est un ensemble de valeurs pour chaque KPI définie.

* Chaque « mesure » est faite pour une date donnée
* Pour le cas des KPI « manuelles », dans une « mesure », plusieurs personnes peuvent proposer une valeur. Les valeurs sont associées à la personne qui les a proposées.
* ISM doit permettre de visualiser les valeurs d’une même KPI, données par différentes personnes sous la forme d’un « radar »
* Une « mesure » doit pouvoir être « gelée » afin que personne ne puisse plus saisir de valeurs (rôle spécifique à définir pour cela)
* Une KPI « automatique » doit pouvoir dire si elle peut ou non être calculée pour une version d’application donnée (il se peut que les informations pour le calcul ne soient pas disponibles pour au moment du calcul)
* Quand une KPI automatique ne peut pas être calculée, ISM proposera la KPI « manuelle » portant le même nom, si celle-ci existe

NB : A priori, il ne devrait y avoir qu’une mesure par version d’application, car les versions vont évoluer dans le temps. Cependant, pour des applications qui n’évoluent pas mais pour lesquelles, par exemple, la documentation est mise à jour au fil des mois ou années, i peut être intéressant de définir une nouvelle mesure qui montrera que le KPI « Alignement avec les standards documentaires » évolue. D’où plusieurs « mesures » pour une même version d’application.

- ISM doit permettre de monter graphiquement l’évolution des KPI pour chaque application. Il s’agit de représenter ici les différentes valeurs d’une même KPI pour les différentes versions de l’application.

* Pour le cas des KPI « manuelles », et dans le cas où plusieurs personnes ont donnés une valeur, la moyenne des valeurs sera prise. Dans le cas de KPI « discrète », la valeur discrète la plus proche de la moyenne sera prise.

## Copies d’écran

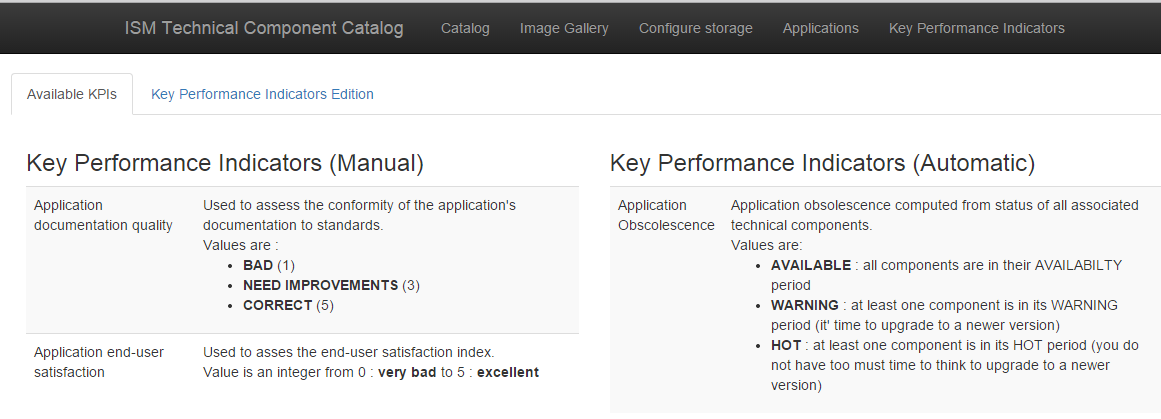


Figure 6 Liste des KPI existantes

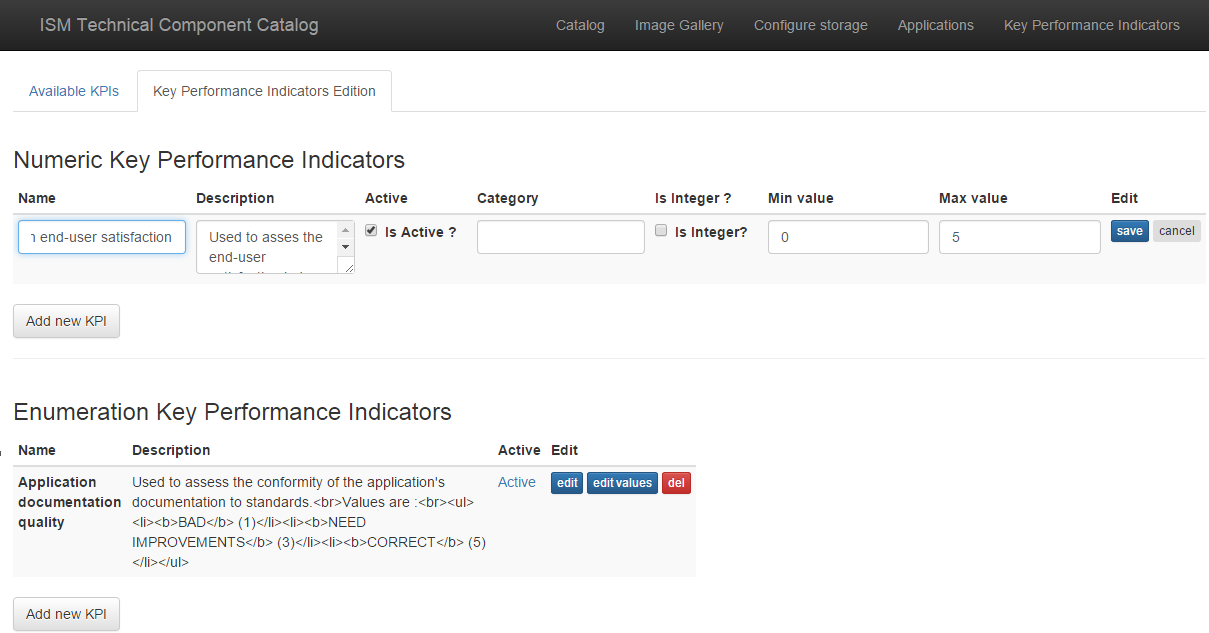


Figure 7 Edition des KPI manuelles

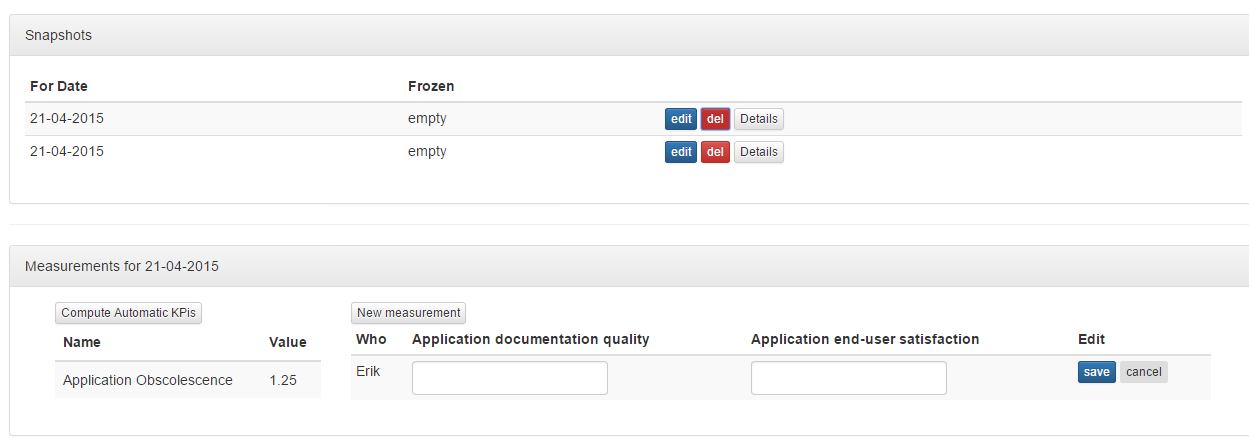


Figure 8 En cours...écran pur définir les "mesures" (snapshot dans la copie)