

# Formation à esup-helpdesk



**Pascal Aubry**  
IFSIC / Université de Rennes 1

**Alexandre Boisseau**  
Université de Brest



Ce document est la version 1.1.2 du support de la formation intitulée « Formation à esup-helpdesk » dispensée les 13 et 14 juin 2006 au CRI de l'Université de Rennes 1, et organisée par le service de formation continue de l'Université de Rennes 1.

La dernière version est accessible en ligne à l'URL suivante :  
<http://perso.univ-rennes1.fr/pascal.aubry/formation/esup-helpdesk>.

Pour toute remarque, merci de vous adresser à [helpdesk-support@esup-portail.org](mailto:helpdesk-support@esup-portail.org).

## Copyright

Copyright © 2006 Consortium ESUP-Portail – Pascal Aubry & Alexandre Boisseau

Ce document peut être copié et distribué dans son intégralité, sans modification, retrait ou ajout.

L'utilisation de ce document dans un cadre de formation collective est soumise à l'approbation explicite et préalable de ses auteurs.

## Table des matières

<b>1. Description de la formation .....</b>	<b>5</b>
1.1. Informations générales.....	5
1.2. Objectifs .....	5
1.3. Programme .....	5
1.4. Pré requis.....	5
1.5. Pointeurs .....	6
<b>2. Introduction.....</b>	<b>7</b>
2.1. L'enquête du GT-SSD.....	7
<i>Utilisez-vous un SSD ?.....</i>	7
<i>Pourquoi n'utilisez-vous pas de SSD ?.....</i>	7
<i>Êtes-vous globalement satisfait ?.....</i>	8
<i>Conseillerez-vous l'utilisation d'un SSD ?.....</i>	8
<i>Principaux apports des SSD .....</i>	9
<i>Problèmes rencontrés .....</i>	9
2.2. Historique du projet .....	10
2.3. Déploiement actuel .....	11
2.4. Présentation du projet esup-helpdesk.....	11
<i>Mutualisation et support utilisateur.....</i>	11
<i>Objectif principal : la satisfaction des utilisateurs.....</i>	12
<i>Saisie des tickets.....</i>	12
<i>Assignation automatique des tickets.....</i>	12
<i>Gestion des priorités .....</i>	12
<i>Un interlocuteur unique et identifié.....</i>	13
<i>Une IHM à la hauteur... du plus novice .....</i>	13
<i>Visibilité / Confidentialité .....</i>	13
<i>Base de connaissances .....</i>	13
<i>Visibilité du travail effectué.....</i>	14
<i>Intégration dans le S.I. ....</i>	14
<i>Assistance interne et externe.....</i>	14
<i>L'outil esup-helpdesk.....</i>	14
<i>Standards ouverts uniquement .....</i>	14
<i>Documentation .....</i>	15
2.5. Les packages à disposition .....	15
<i>Le package esup-helpdesk-channel .....</i>	15
<i>Le package esup-helpdesk-quick-start .....</i>	16
2.6. Pré requis.....	16
<i>Pour le package esup-helpdesk-channel.....</i>	16
<i>Pour le package esup-helpdesk-quick-start.....</i>	16
2.7. Environnement de la formation .....	16
<i>Système d'exploitation .....</i>	16
<i>Utilisation de EasyPHP .....</i>	17
<i>Initialisation des bases de données .....</i>	17
<b>3. Installation des logiciels .....</b>	<b>18</b>
3.1. Installation de esup-helpdesk-channel .....	18
<i>Instance uPortal pré installée.....</i>	18
<i>Ajout de esup-helpdesk-channel à uPortal .....</i>	20
3.2. Installation de esup-helpdesk-quick-start.....	22
<i>Rapatriement, installation.....</i>	23

<i>Organisation des répertoires</i> .....	23
<i>Configuration des propriétés de base</i> .....	23
<i>Configuration de l'authentification et de la récupération d'attributs</i> .....	24
<i>Déploiement du logiciel</i> .....	24
<i>Initialisation des bases de données</i> .....	25
<i>Contrôle du serveur</i> .....	25
3.3. no user attribute retrieved from uPortal for user 'your-id' .....	25
<b>4. Concepts et opérations de base .....</b>	<b>26</b>
4.1. Les fichiers de configuration .....	26
<i>Fichiers obligatoires</i> .....	26
<i>Fichiers optionnels</i> .....	26
<i>Prise en compte des modifications des fichiers de configuration</i> .....	26
4.2. Les utilisateurs de esup-helpdesk .....	26
<i>Les utilisateurs locaux</i> .....	26
<i>Les utilisateurs externes</i> .....	27
<i>Les administrateurs</i> .....	27
4.3. Configuration du courrier électronique .....	27
4.4. Gestion des logs .....	27
4.5. Recherche dans l'annuaire LDAP .....	28
4.6. Création de services simples .....	28
<i>Les services</i> .....	28
<i>Les gestionnaires des services</i> .....	32
<i>Les catégories</i> .....	32
<i>Les membres de catégories</i> .....	34
4.7. Les tickets .....	34
<i>Création</i> .....	34
<i>Attachement d'un fichier à un ticket</i> .....	34
<i>Cycle de vie</i> .....	35
<i>Propriétés et historique</i> .....	36
4.8. Ajout de modèles de ticket (ou tickets prédéfinis) .....	36
4.9. Les remontées d'alerte .....	37
<i>Pour les gestionnaires</i> .....	37
<i>Pour tous les utilisateurs</i> .....	38
<i>Suivi volontaire d'un ticket</i> .....	38
<i>Invitations à suivre un ticket</i> .....	38
4.10. Les FAQs .....	39
<i>Intérêt</i> .....	39
<i>Structuration</i> .....	39
<i>Qui voit quoi ?</i> .....	39
<i>Qui peut modifier quoi ?</i> .....	39
<i>Exportation des FAQs</i> .....	40
4.11. La base de connaissances .....	40
4.12. Les préférences des utilisateurs .....	41
4.13. Mise à jour d'une version existante .....	41
<i>Éléments de configuration récupérés automatiquement</i> .....	42
<i>Procédure à suivre</i> .....	42
<b>5. Concepts et opérations avancées .....</b>	<b>43</b>
5.1. Utilisation des classes de personnalisation fournies avec la distribution .....	43
5.2. L'assignation automatique des tickets .....	43
5.3. La virtualisation .....	44

5.4.	Mise en œuvre d'une vision adaptative.....	44
	<i>Exemples d'organisations</i> .....	45
	<i>Utilisation de la classe <code>custom.departmentSelection.CXmlConfig</code></i> .....	46
	<i>Combien de temps cela prend-il ?</i> .....	48
5.5.	Ajout automatique d'informations sur les utilisateurs à la création des tickets.....	49
5.6.	Alimentation à partir d'une boîte IMAP .....	50
5.7.	Personnalisation de l'application .....	50
	<i>Look n feel</i> .....	50
	<i>Chaînes de caractères</i> .....	50
	<i>Autres personnalisations</i> .....	51
5.8.	Expiration automatique des tickets non approuvés.....	51
5.9.	Développement de classes de personnalisation .....	51
	<i>L'interface commune <code>ICustom</code></i> .....	52
	<i>L'interface <code>ICustomComputerUrl</code></i> .....	52
	<i>L'interface <code>ICustomNewTicketAssignment</code></i> .....	53
	<i>L'interface <code>ICustomNewTicketProperties</code></i> .....	56
	<i>L'interface <code>ICustomTicketMonitoring</code></i> .....	58
	<i>L'interface <code>ICustomDepartmentSelection</code></i> .....	59
	<i>L'interface <code>ICustomUserUrl</code></i> .....	61
	<i>L'interface <code>ICustomUserInfo</code></i> .....	63
5.10.	Les autres directives de configuration.....	65
<b>6.</b>	<b>Perspectives.....</b>	<b>66</b>
6.1.	Feuille de route (roadmap).....	66
6.2.	Les statistiques .....	66
	<i>Exemples</i> .....	67
	<i>Mode de calcul</i> .....	68
	<i>Portée des statistiques</i> .....	68
	<i>Statistiques calculées</i> .....	68
	<i>Format de sortie</i> .....	68
6.3.	Le passage en portlet/servlet .....	69

# 1. Description de la formation

## 1.1. Informations générales

**Public concerné** : les ingénieurs et techniciens souhaitant/devant mettre en place un Système de Suivi des Demandes (SSD) des utilisateurs.

**Durée** : deux jours (14h).

**Nombre de stagiaires** : 16 maximum.

**Intervenants** :

- Pascal Aubry (IFSIC/Université de Rennes 1, [pascal.aubry@univ-rennes1.fr](mailto:pascal.aubry@univ-rennes1.fr))
- Alexandre Boisseau (Université de Brest, [alexandre.boisseau@univ-brest.fr](mailto:alexandre.boisseau@univ-brest.fr))

## 1.2. Objectifs

Les objectifs sont :

- Dès la fin de la formation, installer, configurer, et mettre en production l'outil esup-helpdesk.
- A travers le logiciel esup-helpdesk, adresser les problèmes humains et techniques rencontrés lors de la mise en place d'un Système de Suivi des Demandes (SSD).

## 1.3. Programme

### 1er jour, 10h-12h30 : contexte

- Problématique du support utilisateur
- Introduction à esup-helpdesk

### 1er jour, 13h30-18h : prise en main

- Installation des logiciels
- Problématique de l'authentification
- Opérations de maintenance
- Découverte des concepts de base
- Configuration et utilisation de toutes les fonctionnalités de base

### 2ème jour, 8h30-12h30 : maîtrise de l'outil

- Création d'une structure de support
- Configuration des services
- Personnalisation du support sans développement

### 2ème jour, 13h30-16h30 : fonctions avancées

- Configuration et utilisation des fonctions avancées
- Création de classes personnalisées
- Conclusion

## 1.4. Pré requis

Seules les fonctions avancées nécessitent une connaissance du développement Java. Les stagiaires ne connaissant pas Java comprendront néanmoins dans cette dernière partie quelles sont les possibilités de personnalisation de l'outil.

## 1.5. *Pointeurs*

- Le projet ESUP-Portail (<http://www.esup-portail.org>).
- Le site de référence du helpdesk ESUP-Portail (<http://esup-helpdesk.sourceforge.net>)
- ESUP-Portail Helpdesk : le suivi des demandes des utilisateurs à l'échelle d'un établissement, JRES2005 (<http://perso.univ-rennes1.fr/pascal.aubry/presentations/helpdesk-jres2005/index.php>)
- GT-SSD : Groupe de Travail Systèmes de Suivi de Demandes (<http://gt-ssd.crrp-bordeaux.cnrs.fr>)

## 2. Introduction

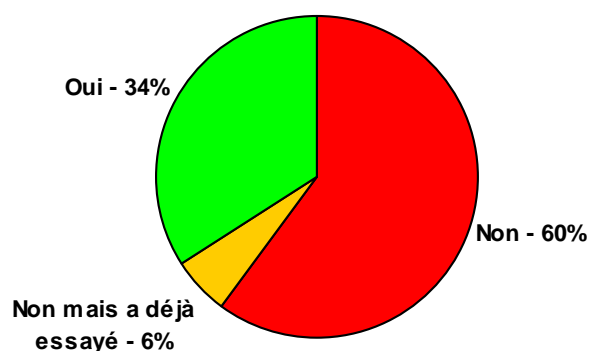
### 2.1. L'enquête du GT-SSD

Groupe de Travail « Système de Suivi de Demandes », soutenu par l'UREC et le CRU.

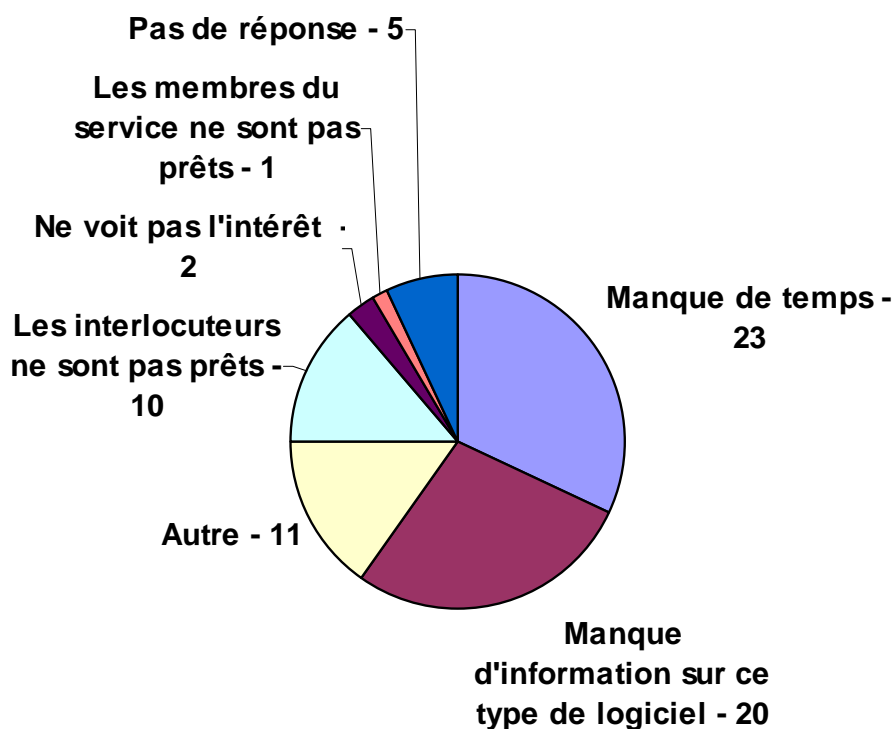
Enquête fin 2004 sur l'utilisation des SSD (120 réponses).

<http://gt-ssd.crpp-bordeaux.cnrs.fr/>

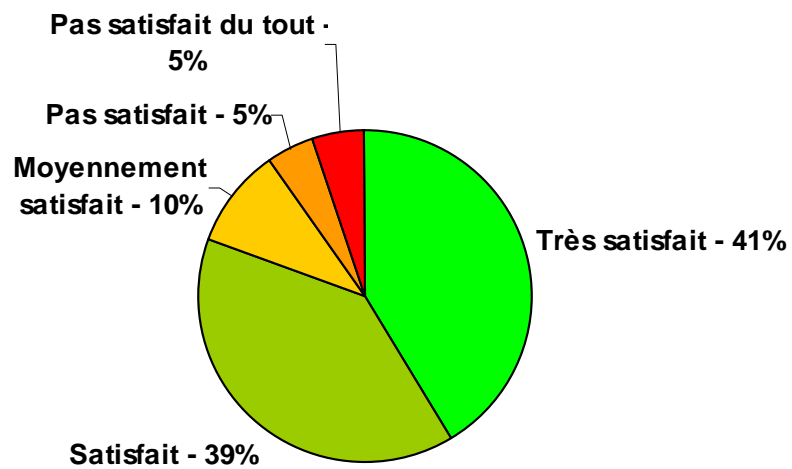
#### Utilisez-vous un SSD ?



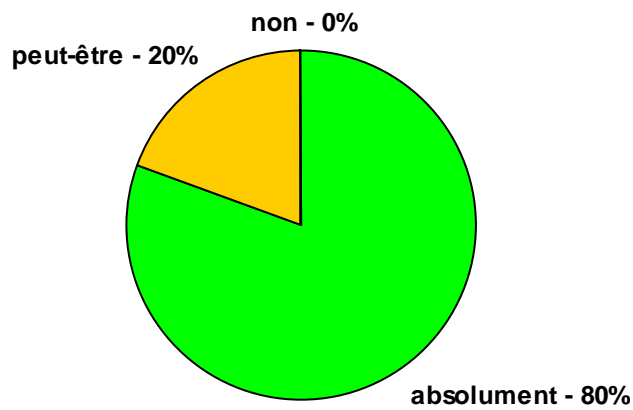
#### Pourquoi n'utilisez-vous pas de SSD ?



## Êtes-vous globalement satisfait ?



## Conseilleriez-vous l'utilisation d'un SSD ?

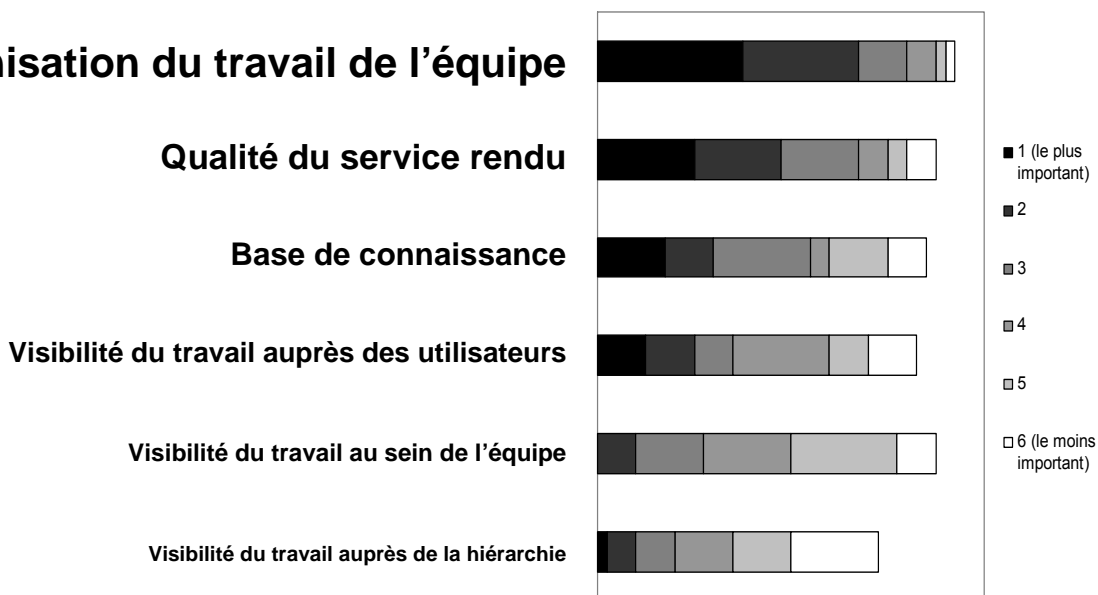


OTRS, Request Tracker, Gédéon, ...

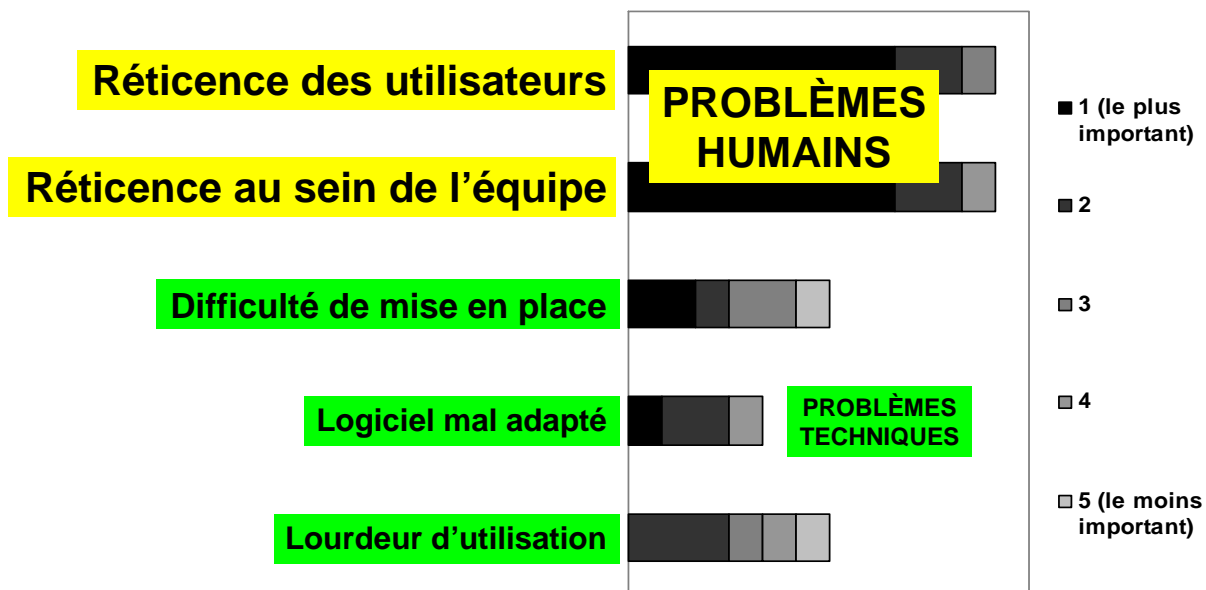


## Principaux apports des SSD

### Organisation du travail de l'équipe

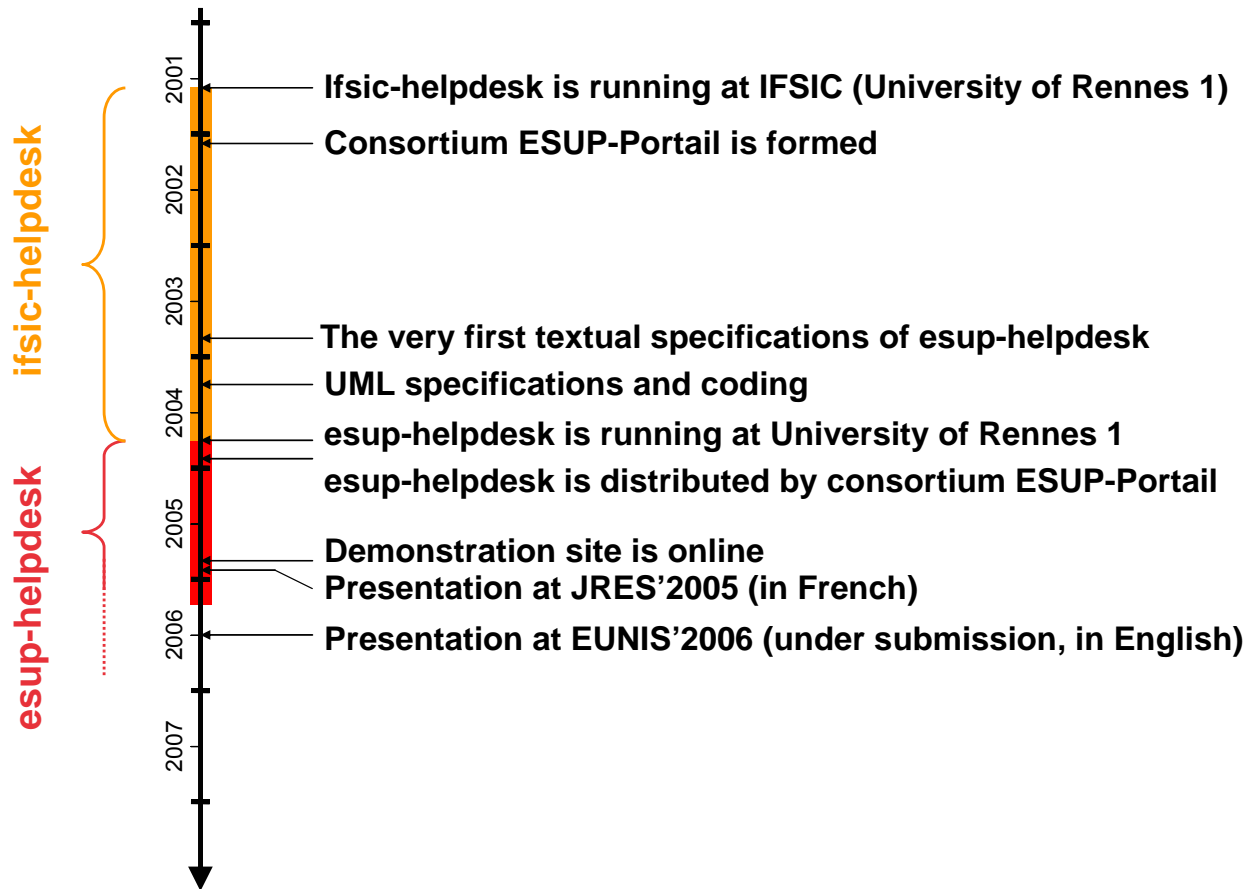


## Problèmes rencontrés

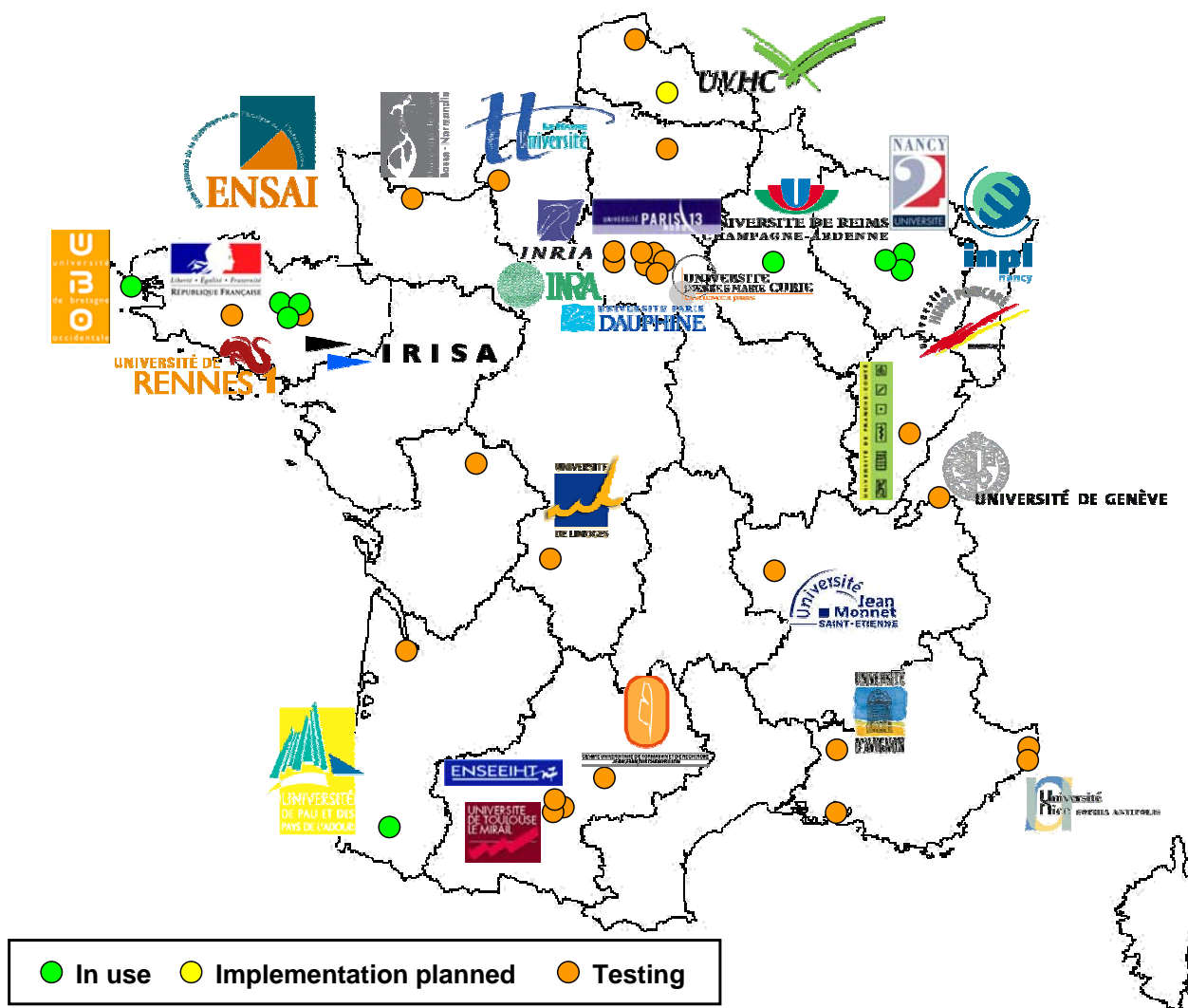


esup-helpdesk : un SSD supplémentaire ? Pas seulement.

## 2.2. Historique du projet



## 2.3. Déploiement actuel



## 2.4. Présentation du projet esup-helpdesk

### Mutualisation et support utilisateur

Partout on mutualise

- Applications centralisées
- Matériels communs
- Services partagés
- Bâtir l'offre applicative autour du Système d'Information

... sauf pour les Systèmes de Suivi des Demandes

- esup-helpdesk se propose de combler ce manque

## Objectif principal : la satisfaction des utilisateurs

- Résolution plus rapide des problèmes
- Meilleure gestion des priorités
- Suivi en temps réel des tickets
- Identification d'un interlocuteur unique et connu
- Interface ergonomique et universelle
- Confidentialité possible des échanges

## Saisie des tickets

Objectif : posséder tous les éléments nécessaires à la résolution dès la saisie du ticket

Étapes :

- Choix du service puis de la catégorie
- Choix d'un problème-type
- Présentation d'un ticket pré rempli

Exemple :

- « UFR Math » → « impression » → « bourrage interne »

## Assignment automatique des tickets

Le temps nécessaire à la prise en charge est souvent très pénalisant → assignation dès la création du ticket

Algorithmes prédéfinis

- au premier gestionnaire disponible,
- de manière cyclique,
- de manière aléatoire,
- au gestionnaire le moins chargé,
- ...

Possibilité de personnalisation

- Développement d'une classe Java
- Exemple : interrogation d'une application de gestion de congés

## Gestion des priorités

Sans SSD, gestion par courrier électronique

- Plusieurs stratégies possibles : FIFO, FILO, SISO, ...SINO
- Selon l'humeur du gestionnaire, le demandeur, le moment de la journée, ...

Il faut déconnecter le suivi du courrier électronique !

Remontées d'alerte (gestionnaires)

Pour les non gestionnaires

- Rediriger vers l'interface (Reply-to: nobody@univ.fr)

Pour les gestionnaires

- Rapidement filtrées vers /dev/null si trop nombreuses
- Hautement configurables

## Un interlocuteur unique et identifié

Interlocuteur unique

- Pas de traitement parallèle des tickets

Interlocuteur identifié

- Toutes les actions sont nominatives et tracées

## Une IHM à la hauteur... du plus novice

Assistance informatique

Accueil Tableau de bord Recherche Journal Services Administration Préférences A propos de

Tableau de bord

ajouter un nouveau ticket Tickets [ 1 -20 ] parmi 189 rafraichir

Service : tous État : ouverts Implication : indifférent Taille de page : 20

n°	Service	Catégorie	Sujet	État	Priorité	Création	Propriétaire
▶ 3164	CRI central	Postes de travail	poste en local : thunderbird dans les c [...]	libre	NORMALE	17:39	Marc Bousse
▶ 3163	CRI central	Postes de travail	portable en local , c'est pas ça!	libre	NORMALE	17:25	Thierry Bedouli
▶ 3162	CRI Centre	Conseil	demande d'hébergement d'un site+forum a	libre	NORMALE	16:28	Yann Molard
▶ 3161	CRI central		Site ifsic-anciens.univ-rennes1.fr	libre	NORMALE	16:26	Jacques-Olivier Houen
▶ 3160	CRI central		Accès au site http://sso-cas-anciens.un [...]	en cours	NORMALE	16:25	Andre Couver
3158	CRI central	Applications	ENT	en cours	NORMALE	15:52	Olivier Ridoux
▶ 3156	CRI Beaulieu	Installation Windows	Salle Info	libre	NORMALE	14:42	Helene Loira
3154	CRI Sante	Dépannage	point de montage apple	libre	NORMALE	14:21	Claudine Lodi
▶ 3153	IFSIC	windows	pb machine e005m07	libre	NORMALE	13:27	Laurent Bironne
3148	IFSIC	gforge	utilisateur gforge	libre	NORMALE	12:23	Martin Costab
▶ 3146	CRI Beaulieu	Messenger	Pb de messengerie salle info chimie Bat	en cours	NORMALE	11:38	Nathalie Troal
▶ 3145	CRI Centre	Installation Windows	PC à enlever du bureau 138	en cours	NORMALE	11:04	Alain Bellido
▶ 3143	CRI Centre	Installation Windows	postes fixes	en cours	NORMALE	10:59	Yves Quete
▶ 3142	CRI Beaulieu	Dépannage	Toujours probleme avec scilab en salle [...]	libre	BASSE	10:52	Jean-Pierre Dus
▶ 3139	CRI Beaulieu	Messenger	Pb Eudora	libre	NORMALE	10:38	Claude Brodri
▶ 3135	CRI Beaulieu	Dépannage	EBIOINFO, Pbs avec Galmat salles TP 907 [...]	en cours	NORMALE	09:29	Frederique Peron-
▶ 3134	CRI Beaulieu	Installation Windows	EBIOINFO, Install ArcGIS/ArcView 8.x sa [...]	en cours	NORMALE		
▶ 3133	CRI central	Réseau	Switch salle formation CRI				
▶ 3132	CRI Sante	Installation Windows	le fichier arrive en csv				
▶ 3128	CRI Sante	Dépannage	virus				

ajouter un nouveau ticket Tickets [ 1 -20 ] parmi 189 rafraichir

Une interface simple et intuitive créée par itération en fonction des retours des utilisateurs

## Visibilité / Confidentialité

Intérêt de la visibilité

- Pour partager les solutions (base de connaissance)
- Pour éviter les tickets multiples pour un même incident

Inconvénient de la visibilité, besoin de confidentialité

- Échange d'informations sensibles
- Obstacle à l'abandon du courrier électronique

Visibilité et confidentialité ne sont pas antinomiques

- Plusieurs niveaux de visibilité
- Granularité très fine (service, ticket, action)

## Base de connaissances

Permet aux utilisateurs de chercher eux-mêmes les solutions à leurs problèmes

- Accélération de la résolution

Indexation par Lucene

Interface de recherche avec critères avancés

## Visibilité du travail effectué

Vis-à-vis des utilisateurs

- Ouverture et transparence perçues très positivement
- Compatible avec la confidentialité

Au sein des équipes de support

- Partage de compétences

Auprès de la hiérarchie

- Est-il besoin d'expliquer ?

## Intégration dans le S.I.

Single Sign-On

- CAS (Central Authentication Service)

Environnement Numérique de Travail

- uPortal by JASIG

Lien étroit avec les données du S.I.

- Interface de recherche dans l'annuaire LDAP
- Récupération des attributs LDAP
- Utilisation des groupes uPortal

## Assistance interne et externe

Permet de gérer l'ensemble du support utilisateur

Authentification des utilisateurs externes

- « à la Sympa » (mot de passe envoyé par mail)
- Note : certains exploitants n'utilisent l'outil qu'avec l'authentification externe

## L'outil esup-helpdesk

Partie du projet ESUP-Portail

Logiciel libre (licence LGPL)

Spécification UML, 100% Java

Développement collaboratif

## Standards ouverts uniquement

Rendu XML par Xalan/Xerces

Indexation par Lucene

Web Services par Axis

Persistance des données par Hibernate

- Abstraction du SGBD / abstraction des données
- Optimisation des accès à la base de données
- Pas de problème de montée en charge

## Documentation

Le site web du projet

- <http://esup-helpdesk.sourceforge.net>

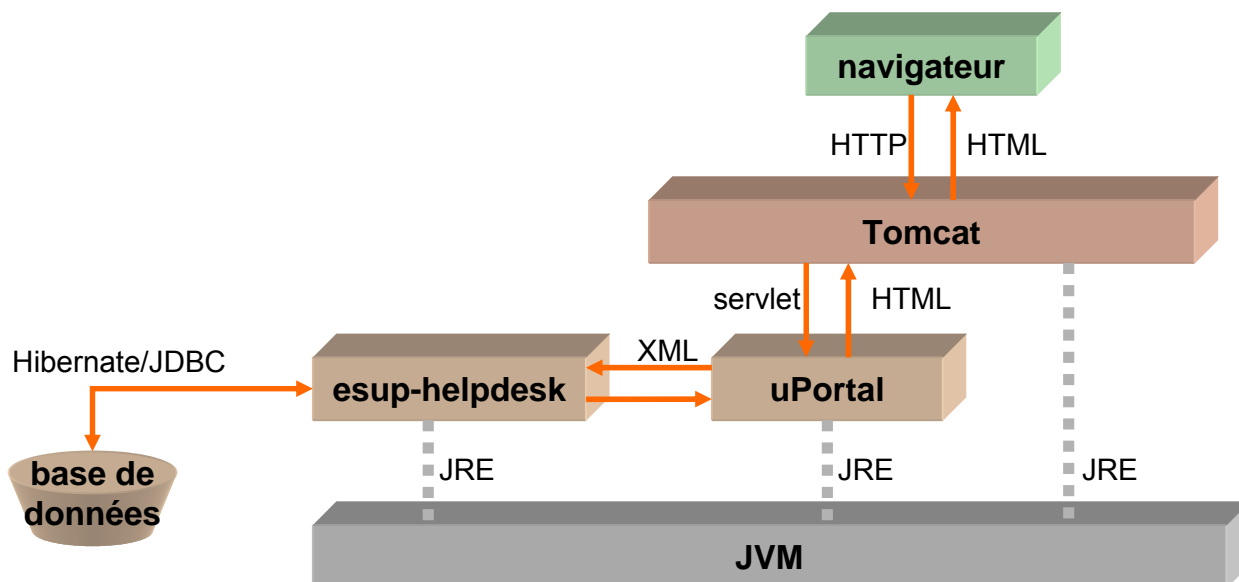
Le support de formation

Les listes de diffusion (@esup-portail.org)

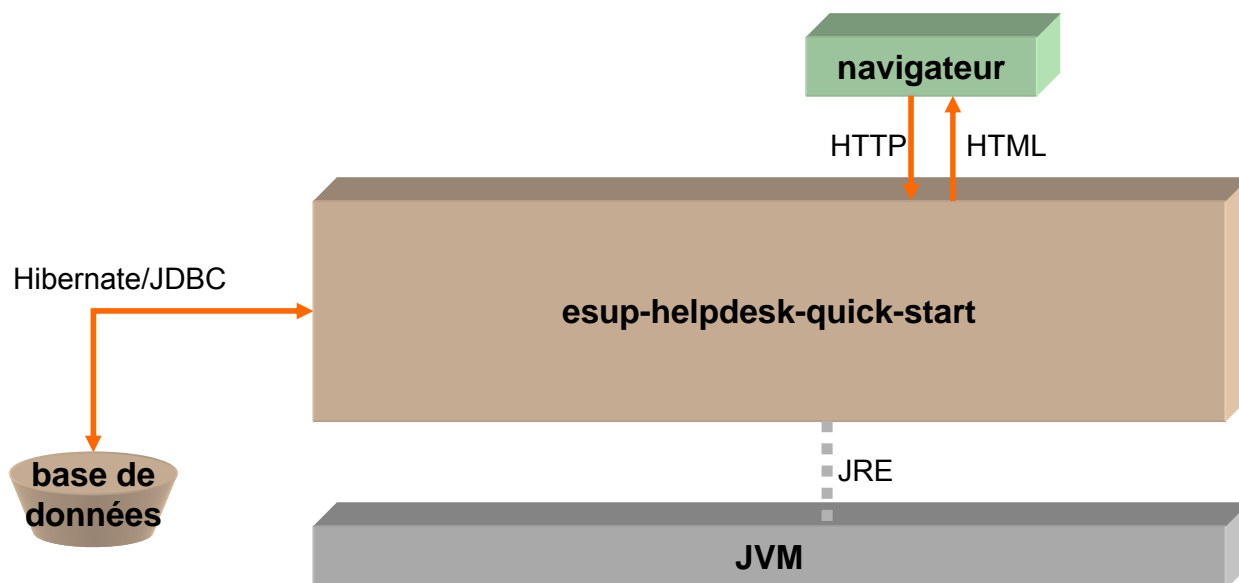
- **helpdesk-utilisateurs** : utilisation et évolutions, en français
- **helpdesk-support** : installation et configuration
- **helpdesk-devel** : développement
- **helpdesk-bugs** : rapports de bugs
- **helpdesk-users** : usage and evolutions, in English

## 2.5. Les packages à disposition

### Le package esup-helpdesk-channel



## Le package esup-helpdesk-quick-start



Environnement non dégradé

### 2.6. Pré requis

Tous les problèmes de compatibilité sont expliqués sur la page <http://esup-helpdesk.sourceforge.net/main/compatibility.html>

#### Pour le package esup-helpdesk-channel

- Une instance uPortal (en ordre de marche), pour esup-helpdesk-channel 2.0, uPortal 2.5.1

#### Pour le package esup-helpdesk-quick-start

- Une Machine Virtuelle Java, pour esup-helpdesk 2.0, JDK 1.5+
- Ant, pour esup-helpdesk 2.0, ant 1.6.5+
- Une base de données MySQL, PostgreSQL ou Oracle.

Pour MySQL, l'activation de InnoDB est indispensable.

### 2.7. Environnement de la formation

#### Systeme d'exploitation

En raison de contraintes liées à la salle de formation, tous les postes de travail sont livrés sous Windows. D'un point de vue pédagogique, cela ne pose de problème ; les commandes et les concepts restent les mêmes (merci Java).

Systeme d'exploitation : Windows XP SP2

Logiciels installés :

- jdk1.5.0\_05
- apache-ant-1.6.5
- EasyPHP 1.8



Les variables d'environnement suivantes ont été positionnées :

- **JAVA\_HOME**=répertoire principal du JDK
- **ANT\_HOME**=répertoire d'installation de ant
- **PATH**=%PATH%:%JAVA\_HOME%\bin:%ANT\_HOME%\bin

Toutes les machines disposent d'un compte dans l'annuaire LDAP de l'Université de Rennes 1, ce compte est administrateur de la machine.

D'autres comptes de tests seront fournis par les formateurs en cours de formation pour que vous disposiez de plusieurs profils.

## Utilisation de EasyPHP

esup-helpdesk et uPortal nécessitent des bases de données pour fonctionner. Nous avons donc installé et paramétré l'environnement EasyPHP sur chacune des machines afin de disposer de bases MySQL.

Comme recommandé dans la documentation d'installation du canal, nous avons forcé MySQL à utiliser des tables InnoDB.

Extrait du fichier my.ini :

```
[mysqld]
[...]
# AB.20060312 force InnoDB
default-table-type=innodb
```

Pour la gestion de la base de données, utilisez phpMyAdmin : faire un clic droit sur l'icône d'EasyPHP dans la barre de menu en bas à droite de l'écran et sélectionner « Administration ».

## Initialisation des bases de données

L'application a besoin de deux bases de données pour fonctionner :

- Une pour le portail, que vous créerez dans EasyPHP et nommerez **uportal**,
- Une pour le helpdesk, que vous créerez également et nommerez **helpdesk**.

Pour plus de facilité, vous utiliserez le même utilisateur pour accéder aux deux bases de données, avec tous les droits sur la base (c'est indispensable).

**Canal uPortal:** Comme le portail uPortal est pré installé, la base de données correspondante ainsi que l'utilisateur ont déjà été créés. Il ne reste donc plus qu'à créer la base de données **helpdesk** et de mettre à jour les droits requis pour l'utilisateur **uportal**.

**Exercice 1:** Se connecter à l'interface phpMyAdmin du serveur MySQL, créer les deux bases de données : **uportal** et **helpdesk**, et créer l'utilisateur **uportal** et lui donner les pleins droits sur les deux bases.

Note : les utilisateurs d'Oracle doivent initialiser la propriété **sql.date.column** à la valeur **dat**, sans quoi le *mapping* des données ne sera pas possible (pour les autres SGBDs, la propriété **sql.date.column** vaut par défaut **date**). Il suffit pour cela de décommenter la ligne correspondante du fichier **build.properties**.

## 3. Installation des logiciels

### 3.1. Installation de esup-helpdesk-channel

#### Instance uPortal pré installée

Dans le cadre de la formation, une instance opérationnelle de uPortal a été installée pour vous. Cette instance est basée sur un uPortal-2.5-esup-2 distribué par ESUP-Portail et est connecté au SI de l'Université de Rennes1 (serveurs CAS et LDAP).

#### Répertoires de travail

Nous avons fixé le répertoire de travail dans : **C:\fh** (Formation Helpdesk)

<b>C:\fh\</b>	Répertoire racine
<b>C:\fh\apache-ant-1.6.5</b>	ANT (ne pas modifier)
<b>C:\fh\files</b>	différents fichiers à disposition
<b>C:\fh\logs</b>	la majeure partie des fichiers de logs se trouvent ici (ne pas modifier)
<b>C:\fh\releases</b>	différents packages à disposition
<b>C:\fh\Tomcat</b>	le serveur Tomcat (ne pas modifier)
<b>C:\fh\uPortal</b>	l'environnement de "compilation" du portail (ne pas modifier)
<b>C:\fh\uPortal-2.5-esup-2</b>	La distribution uPortal-esup
<b>C:\fh\webapps</b>	le répertoire de déploiement de la servlet uPortal (ne pas modifier)

#### Propriétés du package uPortal-esup

Nous avons installé un uPortal-2.5-esup-2 de base, en gardant une grande partie des valeurs par défaut : nous n'avons édité que le fichier esup.properties.

Les propriétés modifiées de ce fichier sont :

```
java_home=C:/Program Files/Java/jdk1.5.0_05
esup.root=C:/fh/uPortal-${esup.distrib.version}
esup.distrib=C:/fh/uPortal
esup.deploy=C:/fh/webapps
server.home=C:/fh/Tomcat
esup.log.file=C:/fh/logs/portal.log
esup.stats.file=C:/fh/logs/stats.log
tomcat.jk=false
esup.public.host=localhost
```

`localhost` est utilisé pour des raisons de généricité pour la formation. Dans le cadre d'un déploiement réel, on mettrait le nom réel de la machine.

```
esup.public.port=:8080
esup.real.port.https=8443
esup.ldap.host=ldapglobal.univ-rennes1.fr
esup.ldap.port=389
esup.ldap.baseDN=ou=People,dc=univ-rennes1,dc=fr
esup.cas.host=sso-cas.univ-rennes1.fr
esup.cas.uri=
esup.db.username=fh_uportal
esup.db.password=fh_uportal
esup.db.url=jdbc:mysql://localhost/ uportal
esup.db.jdbcDriverJar=mysql-connector-java-3.0.15-ga-bin.jar
esup.db.className=com.mysql.jdbc.Driver
esup.db.db-version=4.1.9-max
esup.session.lifetime=120
```

Par ailleurs, pour un affichage légèrement plus attractif, nous avons modifié le fichier `esup-fragments.xml` afin d'afficher le canal d'accueil d'ESUP-Portail.

Pour plus d'information sur ces propriétés, référez vous à la documentation d'installation de uPortal sur le site du projet ESUP-Portail (<http://www.esup-portail.org>).

### Contrôle du serveur

Démarrage du serveur : à partir du répertoire `uPortal-2.5-esup-2` :

```
ant esup.start
```

Arrêt du serveur : à partir du répertoire `uPortal-2.5-esup-2` :

```
ant esup.stop
```

Accès au portail : <http://localhost:8080/>

Accès à l'espace d'administration du portail : <http://localhost.univ-brest.fr:8080/private/index.jsp>  
- login/mot de passe : admin/admin

**Exercice 2 :** Tester votre portail.

### Commandes intéressantes

Réinitialiser un compte : à partir du répertoire `uPortal-2.5-esup-2` :

```
ant esup.stop
ant uportal.deluser -Duser=username
ant esup.start
```

### Personnalisation de l'installation

Si vous souhaitez utiliser les serveurs CAS et LDAP de votre établissement, vous devez mettre à jour l'installation. La procédure à suivre est la suivante.

1. Editer le fichier `uPortal-2.5-esup-2\esup.properties` et mettre à jour les propriétés :

```
esup.ldap.host
esup.ldap.port
esup.ldap.baseDN
esup.cas.host
esup.cas.uri
```

2. A partir du répertoire uPortal-2.5-esup-2 :

```
ant esup.init
ant esup.deploy1
```

**Exercice 3 :** Personnaliser votre installation si souhaité.

## Ajout de esup-helpdesk-channel à uPortal

### Rapatriement, installation

Télécharger `esup-helpdesk-channel-2.0.x.zip` depuis <http://esup-helpdesk.sourceforge.net>

Décompresser dans `c:\fh`.

Se placer dans le répertoire créé (`esup-helpdesk-channel-2.0.x`).

### Organisation des répertoires

L'application `esup-helpdesk-channel` utilise du code commun à d'autres applications, qui concerne notamment l'accès aux données par Hibernate. Pour cette raison, un premier niveau de répertoires distingue le code commun à d'autres applications (`esup-hibernate-channel`) du code propre à l'application (`esup-helpdesk-channel`).

- `/esup-hibernate-channel/properties` : des fichier de configurations (récupération des attributs des utilisateurs, recherche dans l'annuaire LDAP).
- `/esup-hibernate-channel/lib` : des librairies.
- `/esup-hibernate-channel/source` : du code Java.
- `/esup-hibernate-channel/webpages/stylesheets` : du code XSLT.
- `/esup-helpdesk-channel/properties` : les autres fichiers de configuration.
- `/esup-helpdesk-channel/lib` : d'autres librairies.
- `/esup-helpdesk-channel/source` : encore du code Java.
- `/esup-helpdesk-channel/webpages/stylesheets` : encore du code XSLT.
- `/esup-helpdesk-channel/webpages/media` : les images et le code javascript de l'application.
- `/esup-helpdesk-channel/build` : un répertoire temporaire servant à la compilation.
- `/esup-helpdesk-channel/docs` : toutes les documentations, identiques à celles mises en ligne sur le site du projet.
- `/esup-helpdesk-channel/custom` : le répertoire de personnalisation du canal.
- `/esup-helpdesk-channel/custom-examples` : un répertoire regroupant plusieurs exemples de personnalisation du canal.

Toutes les commandes `ant` s'exécutent depuis le répertoire `/esup-helpdesk-channel`, dans lequel se trouve le fichier `build.xml`.

---

<sup>1</sup> La commande `uportal.deploy` a été renommée en `esup.deploy` en version 2.5-esup-2.

## Création des fichiers de configuration

Les fichiers de configuration suivants sont nécessaires à l'installation :

- `esup-hibernate-channel/CEsupConfig.xml`
- `esup-helpdesk-channel/build.properties`
- `esup-helpdesk-channel/CHelpdeskConfig.xml`
- `esup-helpdesk-channel/helpdeskChannelHibernate.cfg.xml`

**Exercice 4 :** Créer les fichiers de configuration par copie des fichiers d'exemple correspondants.

Note : les fichiers de configuration seront étudiés plus en détail dans la partie 4.1.

**Exercice 5 :** Editer si nécessaire le fichier `esup-helpdesk-channel/helpdeskChannelHibernate.cfg.xml` pour refléter votre configuration de base de données.

## Configuration des répertoires

Editer `esup-helpdesk-channel/properties/build.properties` et modifier les propriétés suivantes (se référer aux commentaires du fichier `build-example.properties`) :

```
esup.dir=C:/fh
uportal.home=${esup.dir}/uPortal
deploy.home=${esup.dir}/webapps/uPortal
tomcat.home=${esup.dir}/Tomcat
log.path=${esup.dir}/logs
helpdesk.log.level=INFO
helpdesk.log.path=${log.path}/esup-helpdesk.log
hibernate.log.level=INFO
hibernate.log.path=${log.path}/hibernate.log
```

Attention : dans les fichiers de configuration, on utilise le slash (/) à la place de l'anti-slash (\). L'utilitaire `ant` remplace ensuite ces caractères si besoin.

## Déploiement du canal

Exécuter `ant deploy`.

Cette commande lance :

- La compilation des sources Java
- Puis le déploiement dans la servlet du portail.

## Déploiement des librairies

Le déploiement des librairies nécessaires au fonctionnement du canal (Lucene, Hibernate, ...) peut être fait automatiquement grâce à la commande `ant deploy-libs`.

### Note sur les conflits de librairies :

Dans le cas d'une installation réelle, il convient de faire attention en déployant ces librairies, à ne pas écraser des versions déjà installées (plus récentes ou non). Il est de la responsabilité de l'administrateur de gérer les librairies installées.

Par exemple, vous pouvez avoir un canal fonctionnant avec Hibernate2 et un autre avec Hibernate3. Il faut donc vérifier que les deux versions soient compatibles, ou que la plus récente est compatible avec la plus ancienne, ...

### Publication du canal

Pour compléter l'installation, il faut publier le canal esup-helpdesk-channel dans votre instance d'uPortal.

Néanmoins, le but n'étant pas de nous attarder sur l'administration uPortal, nous avons donc préparé le fichier permettant d'effectuer la publication du canal. Vous n'avez donc plus qu'à suivre les instructions pour compléter la publication.

Publication du canal esup-helpdesk-channel :

1) Copier le fichier `files/esup-helpdesk-chanpub.xml` dans `C:/fh/uPortal/properties/chanpub`.

2) Depuis le répertoire uPortal-2.5-esup-2, exécuter :

```
ant uportal.pubchan -Dchannel=esup-helpdesk-chanpub.xml
```

### Poussée des fragments

Il ne nous reste plus qu'à afficher le canal dans le portail. Néanmoins, comme expliqué lors de la publication du canal, nous ne souhaitons pas nous attarder sur ce point. Nous vous conseillons donc de suivre les instructions suivantes.

Poussée du fragment helpdesk :

1) Copier `files/esup-helpdesk-pushfragments.xml` dans `c:/fh/uPortal/properties/al`.

2) Depuis le répertoire uPortal-2.5-esup-2, exécuter :

```
ant uportal.pubfragments \  
-DfragmentFile=/properties/al/esup-helpdesk-pushfragments.xml
```

**Exercice 6 :** Installer le canal, le déployer, publier le canal, pousser un fragment pour le rendre accessible, redémarrer le portail et tester.

Pour créer la base de données, exécuter `ant initdb` depuis le répertoire du canal. Attention : cette commande détruit toutes les données du helpdesk !!!

Il est normal de tomber sur l'erreur « `no user attribute retrieved from uPortal for user 'your-id', can not create it in the database` ». Cette erreur est traitée au point 3.3.

A partir de ce point, l'installation, la configuration et la gestion des packages esup-helpdesk-channel et esup-helpdesk-quick-start sont semblables. Les différences seront notées de la manière suivante :

**Canal uPortal :** Instructions spécifiques pour les utilisateurs du canal uPortal (esup-helpdesk-channel).

**Quick Start :** Instructions spécifiques pour les utilisateurs du Quick Start (esup-helpdesk-quick-start).

## 3.2. Installation de esup-helpdesk-quick-start

Afin d'homogénéiser les postes de travail de la formation, les logiciels seront tous installés sur `C:\fh`.

## Rapatriement, installation

Télécharger `esup-helpdesk-quick-start-2.0.x.zip` depuis <http://esup-helpdesk.sourceforge.net>

Décompresser dans `C:\fh\`.

Se placer dans le répertoire créé (`esup-helpdesk-quick-start-2.0.x`)

## Organisation des répertoires

L'application `esup-helpdesk-quick-start` est composée :

- Du conteneur de servlet Tomcat,
- De l'environnement ESUP-Portail, basé sur uPortal,
- Du canal uPortal `esup-helpdesk-channel`

Les répertoires sont organisés de la manière suivante :

- `/properties` : les fichiers de configuration de l'application.
- `/packages` : les binaires nécessaires à l'installation.
- `/esup-patch` : des fichiers modifiant l'environnement ESUP-Portail.
- `/esup-custom` : le répertoire de personnalisation de l'environnement ESUP-Portail.
- `/esup-custom-examples` : des exemples de personnalisation de l'environnement ESUP-Portail.
- `/channel-custom` : le répertoire de personnalisation du canal `esup-helpdesk-channel`.
- `/channel-custom-examples` : des exemples de personnalisation du canal `esup-helpdesk-channel`.

D'autres répertoires sont créés ultérieurement par l'installation.

## Configuration des propriétés de base

Les fichiers de configuration suivants sont nécessaires à l'installation (dans le répertoire `properties`) :

- `CEsupConfig.xml`
- `build.properties`
- `CHelpdeskConfig.xml`
- `helpdeskChannelHibernate.cfg.xml`

Note : les fichiers de configuration seront étudiés plus en détail dans la partie 4.1.

**Exercice 7 :** Créer les fichiers de configuration par copie des fichiers d'exemple correspondants.

**Exercice 8 :** Editer si nécessaire le fichier `esup-helpdesk-channel/helpdeskChannelHibernate.cfg.xml` pour refléter votre configuration de base de données.

Selon le type d'installation désiré, choisir un fichier d'exemple parmi `properties/build-example-*.properties` et le copier dans `properties/build.properties`.

Editer `properties/build.properties` et modifier les propriétés suivantes (se référer aux commentaires du fichier `build-example-*.properties`) :

```
esup-helpdesk-quick-start.root=C:/fh/esup-helpdesk-quick-start-${app.version}
esup-helpdesk-quick-start.server-name=localhost
esup-helpdesk-quick-start.http-port=8080
esup-helpdesk-quick-start.shutdown-port=8005
esup-helpdesk-quick-start.jk2-port=8009
esup-helpdesk-quick-start.uri=
esup-helpdesk-quick-start.uportal.db.username=uportal
esup-helpdesk-quick-start.uportal.db.password=uportal
esup-helpdesk-quick-start.uportal.db.url=jdbc:mysql://localhost/uportal
esup-helpdesk-quick-start.uportal.db.driver=\
    mysql-connector-java-3.0.15-ga-bin.jar
esup-helpdesk-quick-start.uportal.db.class=com.mysql.jdbc.Driver
esup-helpdesk-quick-start.uportal.title=ESUP-Portail Helpdesk Quick Start
esup-helpdesk-quick-start.session.lifetime=120
esup-helpdesk-quick-start.uportal.log.level=INFO
esup-helpdesk-quick-start.channel.log.level=INFO
esup-helpdesk-quick-start.hibernate.log.level=INFO
```

## Configuration de l'authentification et de la récupération d'attributs

### Avec CAS/LDAP

Si vous utilisez une authentification CAS couplée à une récupération des attributs LDAP, alors ajoutez les propriétés suivantes :

```
esup-helpdesk-quick-start.use-cas=true
esup-helpdesk-quick-start.cas.server-url=https://sso-cas.univ-rennes1.fr

esup-helpdesk-quick-start.use-ldap=true
esup-helpdesk-quick-start.ldap.server=ldapglobal.univ-rennes1.fr
esup-helpdesk-quick-start.ldap.port=389
esup-helpdesk-quick-start.ldap.baseDN=ou=People,dc=univ-rennes1,dc=fr
```

Note : par défaut, le package `esup-helpdesk-quick-start` embarque l'autorité racine de l'IGC du CRU (<http://igc.cru.fr>). Si vous utilisez une autorité racine différente de celle du CRU, alors modifiez également la propriété `esup-helpdesk-quick-start.keystore.filename`.

### Sans CAS/LDAP

Si vous souhaitez seulement utiliser l'authentification par mot de passe envoyé par courrier électronique, alors ajoutez les propriétés suivantes :

```
esup-helpdesk-quick-start.use-cas=false
esup-helpdesk-quick-start.use-ldap=false
```

## Déploiement du logiciel

Exécuter `ant install`.



Cette commande installe :

- Une instance de Tomcat, le conteneur de servlets
- L'environnement ESUP-Portail, basé sur uPortal
- Le canal esup-helpdesk-channel dans uPortal

## Initialisation des bases de données

L'initialisation de la base du portail se fait à l'aide de la commande `ant esup.initdb`. Cette commande peut être faite autant de fois que nécessaire.

L'initialisation de la base du helpdesk se fait à l'aide de la commande `ant helpdesk.initdb`. Attention : l'exécution de cette commande efface irrémédiablement toutes les données du helpdesk, elle ne doit être exécutée qu'une seule fois !

## Contrôle du serveur

### Démarrage du serveur

```
ant start
```

### Arrêt du serveur

A partir du répertoire uPortal-2.5-esup-2 :

```
ant stop
```

### Accès à l'application

- <http://localhost:8080/>

**Exercice 9 :** Installer l'application et tester.

Il est normal de tomber sur l'erreur « `no user attribute retrieved from uPortal from user 'your-id', can not create it in the database` ». Cette erreur est traitée au point 3.3.

A partir de ce point, l'installation, la configuration et la gestion des packages esup-helpdesk-channel et esup-helpdesk-quick-start sont semblables. Les différences seront notées de la manière suivante :

**Canal uPortal :** Instructions spécifiques pour les utilisateurs du canal uPortal (esup-helpdesk-channel).

**Quick Start :** Instructions spécifiques pour les utilisateurs du Quick Start (esup-helpdesk-quick-start).

### 3.3. `no user attribute retrieved from uPortal for user 'your-id'`

En démarrant pour la première fois l'application, cette erreur est montrée :

```
no user attribute retrieved from uPortal from user 'your-id', can not create it in the database
```

Le message d'erreur indique que l'utilisateur « *your-id* » n'existe pas. Le nom de cet utilisateur est spécifié dans le fichier `CHelpdeskConfig.xml`, et c'est l'identifiant de l'utilisateur qui sera déclaré administrateur de la base de données en premier.

La résolution de cette erreur est donnée plus loin dans ce document (cf partie 4.2).

## 4. Concepts et opérations de base

### 4.1. Les fichiers de configuration

**Canal uPortal:** Les fichiers de configuration sont situés dans un des deux répertoires `esup-helpdesk-channel/properties` ou `esup-hibernate-channel/properties`.

**Quick Start:** Les fichiers de configuration sont tous situés dans le répertoire `properties`.

#### Fichiers obligatoires

- `version.properties` : donne le numéro de version aux opérations de maintenance.
- `build.properties` : définit la manière dont le canal est déployé.
- `CEsupConfig.xml`, `CHelpdeskConfig.xml` : définissent le comportement de l'application. Le fichier `CHelpdeskConfig.xml` est le fichier de configuration le plus important, il est très rare d'avoir à modifier le fichier `CEsupConfig.xml`.
- `helpdeskChannelHibernate.cfg.xml` : définit l'accès à la base de données de l'application.
- `batchLogger.properties` : définit la manière dont sont traités les logs en mode batch (exécution de tâches en ligne de commande).

#### Fichiers optionnels

D'autres fichiers de configuration (`LdapSearchConfig.xml`, `departmentSelection.xml`, `userInfo.xml`, `esup-helpdesk_*-custom.xml`, `faqExport.xml`) sont utilisés de manière optionnelle, ils seront vus dans la suite de ce document.

#### Prise en compte des modifications des fichiers de configuration

**Canal uPortal:** Pour que les modifications d'un fichier de configuration soient prises en compte, il faut exécuter `ant deploy` et redémarrer le portail. Note : dans le cas de `CEsupConfig.xml` et `CHelpdeskConfig.xml`, les fichiers sont relus à chaque nouvelle instantiation du canal, le redémarrage du portail n'est donc pas nécessaire.

**Quick Start:** Pour que les modifications d'un fichier de configuration soient prises en compte, il faut exécuter `ant install` et redémarrer l'application.

### 4.2. Les utilisateurs de esup-helpdesk

#### Les utilisateurs locaux

Ils sont identifiés par leur UID dans le Système d'Information (par exemple `paubry`).

Ils sont authentifiés par le portail (uPortal), éventuellement par CAS.

Ils sont caractérisés par leurs attributs dans le système d'information (définis dans le fichier de configuration de uPortal `PersonDirectory.xml`<sup>2</sup>).

---

<sup>2</sup> `PersonDirs.xml` avant uPortal 2.5.1.

## Les utilisateurs externes

Ils sont identifiés par leur adresse électronique (par exemple `alexandre.boisseau@univ-brest.fr`).

Ils sont authentifiés par le canal (`esup-helpdesk-channel`) grâce à un mot de passe propre à l'application. Le mot de passe est transmis par email sur demande.

**Canal uPortal:** Le portail fonctionne dans ce cas en mode *guest*.

Aucun attribut n'est alors disponible pour cet utilisateur (pas de lien avec le Système d'Information).

## Les administrateurs

Les administrateurs du portail sont de « vrais » utilisateurs.

Le premier administrateur est créé lorsqu'il n'en existe pas, à chaque instanciation du canal. Pour cela, le canal utilise le tag `/config/first-administrator-id` du fichier `CHelpdeskConfig.xml`.

Les administrateurs sont en suite co-optés, à l'aide de la rubrique « Administration » de l'interface web. Note : cette interface web n'est visible que par les administrateurs.

**Exercice 10 :** Déclarer le premier administrateur dans la configuration et observer la création automatique de l'utilisateur correspondant (dans la base de données).

**Exercice 11 :** Essayer de se connecter comme utilisateur externe (avec une adresse électronique).

Constater que le mot de passe n'est jamais transmis par courrier électronique et visualiser l'erreur dans les logs.

**Exercice 12 :** Donner les privilèges d'administrateur à un autre utilisateur.

## 4.3. Configuration du courrier électronique

Le serveur sortant à utiliser lors de la formation est `smtp.univ-rennes1.fr`.

**Exercice 13 :** Configurer le courrier électronique dans `CHelpdeskConfig.xml` en utilisant le tag de configuration `/config/email/intercept` (avec votre propre adresse électronique) et re-tester.

## 4.4. Gestion des logs

**Canal uPortal:** Par défaut, les logs du canal sont versés dans ceux de uPortal. Afin de les distinguer simplement, on peut exécuter `ant deploy-log`, qui modifie le fichier `log4j.properties` du portail et paramètre les logs générés par Hibernate et ceux générés par le canal Helpdesk vers des fichiers spécifiques. La target `deploy-log` prend en compte les propriétés `helpdesk.log.*` et `hibernate.log.*` du fichier `build.properties`.

**Quick Start :** Les logs du portail et du canal sont séparés, respectivement dans les fichiers `portal.log` et `esup-helpdesk.log`. La target `install` prend en compte les propriétés

`esup-helpdesk-quick-start.channel.log.level` et `esup-helpdesk-quick-start.hibernate.log.level` du fichier `build.properties`.

Tout changement du fichier `log4j.properties` doit être suivi d'un redémarrage. Le fichier `log4j.properties` peut également être édité manuellement par les administrateurs connaissant `log4j`.

**Exercice 14 :** Modifier le niveau de log du canal en DEBUG et tester (se reconnecter en utilisateur externe). Repasser ensuite en mode INFO.

## 4.5. Recherche dans l'annuaire LDAP

Il est plus aisé pour trouver des utilisateurs de les rechercher dans un annuaire LDAP, lorsqu'un tel annuaire est disponible.

Pour cela, il faut :

- Activer la recherche LDAP en éditant le tag `/config/use-ldap-search` du fichier `CHelpdeskConfig.xml`.
- Renseigner le fichier `LdapSearchConfig.xml`.

**Canal uPortal :** Pour mémoire, le fichier `LdapSearchConfig.xml` se trouve dans le répertoire `esup-hibernate-channel/properties`.

**Exercice 15 :** Utiliser la recherche dans l'annuaire LDAP de l'Université de Rennes 1 pour ajouter de nouveaux administrateurs.

La recherche LDAP est utilisée à de nombreux autres endroits de l'application, il est indispensable de la mettre en place si vous disposez d'un tel annuaire.

Note : la recherche LDAP se configure bien à deux endroits, pour deux fonctionnalités différentes :

- Au niveau du portail (uPortal) pour la remontée des attributs des utilisateurs ;
- Au niveau du canal (esup-helpdesk) pour la recherche des utilisateurs dans l'annuaire LDAP.

## 4.6. Création de services simples

### Les services

#### Le découpage en services

Assez étanche pour assurer l'indépendance des services

Assez communiquant pour assurer le support au niveau de l'établissement

Assez configurable pour s'adapter :

- à un support déjà existant
- aux différences de fonctionnement des services

#### Appréhender la complexité

S'adapter à la structure complexe des établissements

Découpage quelconque (par composante administrative, par campus, ...)

## Vision adaptative du support

- Masquage de la complexité aux utilisateurs
- En fonction de leur profil et leur localisation

[Accueil](#)
[FAQ](#)
[Tableau de bord](#)
[Recherche](#)
[Journal](#)
[Services](#)
[Administration](#)
[Préférences](#)
[A propos de](#)

## Service IFSIC

[retour à la liste des services](#)

---

Gestionnaire(s) <input type="button" value="ajouter"/>	Catégorie(s) <input type="button" value="ajouter"/>
Stephane Ayello (ayello)	1 homedir (accès au répertoire personnel) ▼
Francois Dagorn (dagorn)	2 email (courrier électronique) ▼▲
Ambroise Diascorn (diascorn)	3 news (forums de discussion) ▼▲
Katy Santerre (krouilla)	4 impression (service d'impression) ▼▲
Raymond Lindor (lindor)	5 imprimantes (imprimantes) ▼▲
Pascal Aubry (paubry)	6 linux (stations de travail Linux) ▼▲
	7 solaris (stations de travail Solaris) ▼▲
	8 windows (stations de travail Windows) ▼▲
	9 serv-linux (serveurs Linux) ▼▲
	10 serv-solaris (serveurs Solaris) ▼▲
	11 helpdesk (logiciel de suivi des demandes) ▼▲
	12 rendu-tp (rendu des TPs) ▼▲
	13 hard (problèmes matériels) ▼▲
	14 planning (emploi du temps) ▼▲
	15 gforge (hébergement des projets) ▼▲
	16 web (mise en ligne de couments) ▼▲
	17 Logiciels (CRI) (Logiciels diffusés par le CRI) ▲ > CRI central - Logiciels

Cliquer sur un gestionnaire pour voir ses détails ou éditer ses propriétés.

Cliquer sur une catégorie pour voir ses détails.

---

**Propriétés du service :**

**Label long :** Institut de Formation Supérieure en Informatique

**Visibilité :** tous les utilisateurs

**URL :** pas d'URL

**Expiration automatique :** 30 jour(s).

**Visibilité par défaut des tickets :** public (Par défaut, les tickets seront visibles par tous les utilisateurs)

**Filtre :** pas de filtre

**Catégorie réelle :** aucune

**Classes de personnalisation:**

**Liens hypertextes des machines :** aucune URL

**Modèle défaut des nouveaux tickets :** récupération des modèles de ticket dans la base de données

**Suivi des tickets par courrier électronique :** aucun courrier spécifique ne sera envoyé

**Assignment automatique des tickets :** aucune assignment automatique

Les propriétés d'un service sont éditables en cliquant sur le bouton « Éditer » :

## Edition du service 'IFSIC'

### Label court

Le label court est utilisé pour l'affichage de tableaux et de listes déroulantes. Il doit être aussi court que possible. Entrez le label court (20 caractères max, ex. "CRI") :

### Label long

Le label long est utilisé pour décrire le service. Entrez le label long (255 caractères max, ex. "Centre de Ressources Informatiques") :

### URL

L'URL est utilisée pour lier le service avec un site web. Entrez l'URL représentant le service, (ou laisser blanc) :

### Filtre

Le filtre est utilisé librement pour les classes de personnalisation, par exemple par org.esupportail.portal.channels.CHelpdesk.custom.enterDepartment.CXmlConfig. Entrez un filtre (ignorez si vous ne pensez pas utiliser la personnalisation departmentSelection) :

### Définissez le temps d'auto expiration des tickets

Lorsqu'un gestionnaire clos un ticket, l'utilisateur doit approuver ou refuser la clôture de ce ticket, s'il estime son problème résolu ou non. Cependant, si l'utilisateur omet de répondre à cette clôture, le ticket peut être automatiquement approuvé après un délai prédéfini. Entrez 0 pour pas d'expiration automatique, vide pour prendre en compte la valeur par défaut ( 10 jours).

### Classes de personnalisation :

#### Hyperlien pour ordinateurs

Les machines peuvent être affichés avec une URL. Ceci donne la possibilité de mettre en place un lien vers une application externe, par exemple un inventaire. Choisissez la manière de construire l'hyperlien associé à une machine :

#### Modèles de ticket

Afin d'aider l'utilisateur à mieux préciser son problème, des modèles prédéfinis de tickets peuvent être créés. Ces modèles peuvent être personnalisés en choisissant une des classes de personnalisation suivante :

Le modèle de ticket par défaut est celui utilisé quand l'utilisateur ne sait pas quelle catégorie choisir parmi celles qui lui sont proposées. Il est possible de définir un modèle de ticket par service ou d'utiliser le modèle de ticket par défaut de l'application

 Utiliser le modèle de ticket par défaut de l'application

**Visibilité par défaut des tickets**

- public (Par défaut, les tickets seront visibles par tous les utilisateurs)
- sujet seulement (Par défaut, les tickets seront visibles par le propriétaire du ticket et les gestionnaires du service, mais le sujet du ticket sera visible par tous)
- privé (Par défaut, les tickets seront visibles par le propriétaire du ticket et les gestionnaires du service)

**Visibilité du service**

- Mettre le service visible par tous les utilisateurs.

**Temps passé**

- Obliger les gestionnaires à remplir le temps passé à la clôture d'un ticket.

**Virtualisation**

Les tickets de ce département sans catégorie seront automatiquement redirigés vers une autre catégorie, réelle.

Les ticket de ce département seront redirigés vers la catégorie suivante :

Aucune, ce département est réel.

**Assignment d'un ticket**

Lors de la création d'un ticket, l'utilisateur est invité à choisir une catégorie dans lequel il pense qu'elle correspond à son problème. La manière d'assigner les tickets est donc définie au niveau des catégories. Cependant, il existe deux cas n'entrant pas dans ce cadre. Le cas où il n'existe pas de catégorie de défini au sein d'un service ; ou le cas où l'utilisateur n'a pu définir la catégorie de son problème. On définit alors une politique par défaut d'assignation des tickets au niveau du service. Choisissez l'assignation par défaut à adopter pour un nouveau ticket :

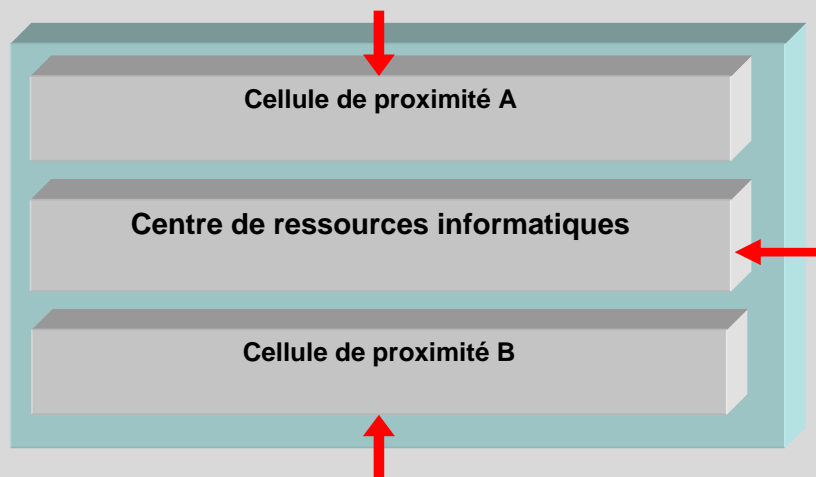
aucune assignation automatique

**Suivi de tickets**

Le suivi des tickets est personnalisé à chaque utilisateur (cf **Préférences**). Il est cependant possible de définir un suivi vers d'autres destinations (adresse électronique externe par exemple). Choisissez la manière de suivre les tickets (vers une adresse électronique externe par ex.).

par défaut [aucun courrier spécifique ne sera envoyé]

**Exercice 16 :** Reproduire la structure des services ci-après :



Les services sont créés par les administrateurs, gérés ensuite par leurs gestionnaires (voir ci-après).

Attention : lorsqu'un service est créé, il n'est pas visible. Il doit être rendu visible manuellement.

## Les gestionnaires des services

Les utilisateurs peuvent être déclarés gestionnaires d'un service.

Le premier gestionnaire d'un service est déclaré par un administrateur, les suivants par un gestionnaire.

Les propriétés des gestionnaires sont les suivants :

### Privilèges

- peut éditer les gestionnaires
- peut éditer les catégories
- peut gérer les FAQs
- peut refuser les tickets
- peut prendre en charge un ticket déjà assigné à un autre gestionnaire
- peut prendre en charge les tickets libres
- peut assigner des tickets à d'autres gestionnaires
- peut ré-ouvrir tous les tickets fermés (même si non gérés)
- peut changer le service des tickets
- peut se rendre disponible/indisponible

### Propriétés

Entrez le taux d'activité du gestionnaire (0 à 100) :

Cochez cette case pour rendre le gestionnaire disponible.

Il est ainsi possible de donner les privilèges de gestionnaire sans pour autant déléguer toute la gestion d'un service.

**Exercice 17 :** Donner les privilèges de gestionnaire à quelques utilisateurs, à l'aide de la recherche LDAP.

## Les catégories

Destiné à l'origine aux Universités :

- Plusieurs dizaines de milliers d'utilisateurs
- Plusieurs dizaines de gestionnaires dans certains services

Découpage quelconque



- Par thématique (courrier électronique, développement, ...)
- Par bâtiment, par système d'exploitation, ...

Accueil FAQ Tableau de bord Recherche Journal Services Administration Préférences A propos de

## Editer la catégorie 'Sauvegardes'

N'oubliez pas de valider vos changement en cliquant sur le bouton "Mettre à jour" avant de quitter cette page !

### Label court

Le label court est utilisé pour l'affichage dans les tableaux et les listes.

Entrez le label court (20 caractères max, ex. web):

### Label long

Le label long est utilisé pour décrire la catégorie. Entrez le label long (255 caractères max, ex. Sites et applications Web):

### URL

L'URL est utilisée pour lier la catégorie à une page Web. Entrez l'URL décrivant cette catégorie (facultatif) :

### Modèle de ticket par défaut

Le modèle de ticket par défaut est celui utilisé quand l'utilisateur ne sait pas quel modèle de ticket choisir parmi ceux qui lui sont proposés. Il est possible de définir un modèle de ticket par catégorie ou d'utiliser le modèle de ticket par défaut du service.

Utiliser le modèle de ticket par défaut du service

### Liste des modèles de ticket :

1 Restauration de fichiers	<input type="button" value="éditer"/>	<input type="button" value="supprimer"/>
----------------------------	---------------------------------------	--

### Liste des membres :

1 Jacques De Poulpique (depoulp)	▼	<input type="button" value="supprimer"/>
----------------------------------	---	--

2 Jean-Francois Radenac (jradenac)	▼ ▲	<input type="button" value="supprimer"/>
------------------------------------	-----	--

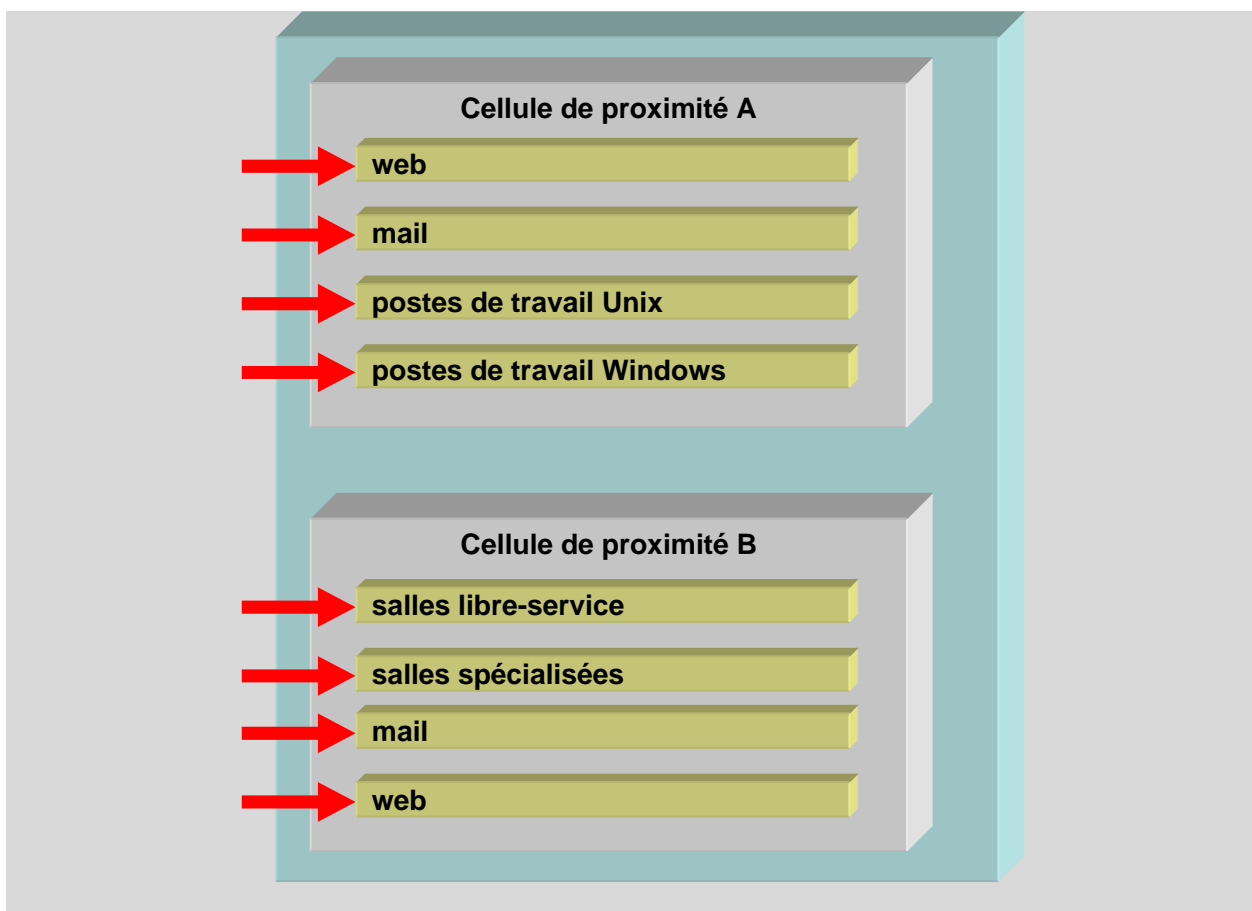
3 Gerard Delpuch (delpuch)	▼ ▲	<input type="button" value="supprimer"/>
----------------------------	-----	--

4 Jean-Pierre Boulard (boulard)	▼ ▲	<input type="button" value="supprimer"/>
---------------------------------	-----	--

5 Ahmed Khoudir (akhoudir)	▲	<input type="button" value="supprimer"/>
----------------------------	---	--




**Exercice 18 :** Créer les catégories suivantes :



## Les membres de catégories

Les gestionnaires d'un service peuvent être désignés comme membre d'une (ou plusieurs) catégorie(s) du service. Cela permet :

- Aux gestionnaires de filtrer les tickets du tableau de bord en fonction de la catégorie des tickets ;
- D'assigner automatiquement les tickets aux membres de la catégorie des tickets (concept vu plus loin dans la formation).

**Exercice 19 :** Ajouter des membres aux catégories existantes.

## 4.7. Les tickets

### Création

**Exercice 20 :** Créez des tickets dans les services et catégories existantes.

Noter qu'une fois créés, les tickets sont libres (non suivis par un gestionnaire).

### Attachement d'un fichier à un ticket

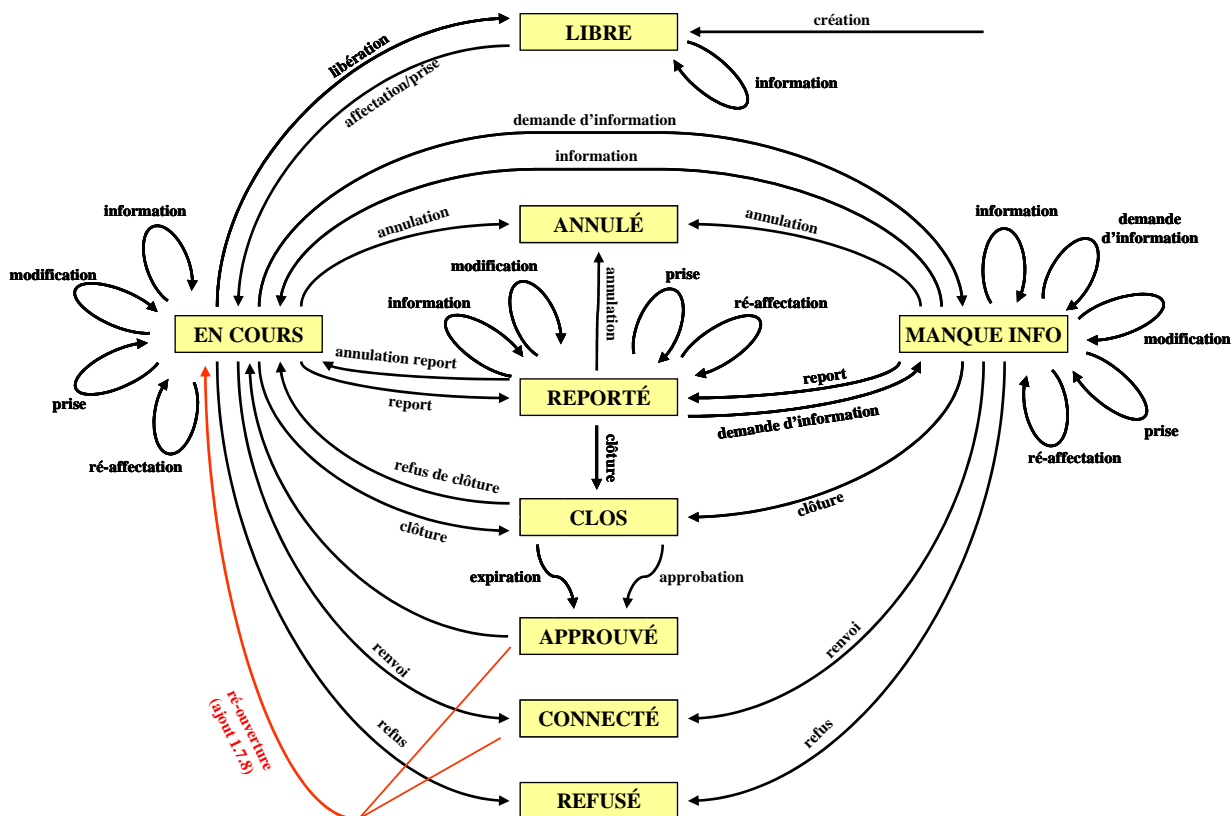
Il est possible, à la création d'un ticket ou ultérieurement, d'attacher au ticket un fichier en le téléchargeant dans l'application. Les fichiers sont stockés sur le système de fichiers du serveur, à l'endroit indiqué par le tag `/config/upload/storage-path` du fichier `CHelpdeskConfig.xml`.

Note : ce répertoire doit être partagé entre tous les serveurs d'un cluster, puisqu'un fichier peut être uploadé sur un serveur et downloadé depuis un autre.

**Exercice 21 :** Tester l'upload et le download d'un fichier attaché à un ticket.

## Cycle de vie



La figure ci-dessous montre le cycle de vie d'un ticket dans esup-helpdesk.



**Exercice 22 :** Tester les changements d'état d'un ticket.

## Propriétés et historique

À partir du tableau de bord de l'application, un clic sur un ticket affiche toutes ses propriétés et son historique.


Propriétés du ticket	Historique du ticket
<p>État : <b>clos/expiré</b></p> <p>Service : <b>CRI Beaulieu</b></p> <p>Catégorie : <b>Dépannage</b></p> <p>Propriétaire : <b>Anne-Gaëlle Carmillat (26000887)</b></p> <p>Gestionnaire : <b>Dominique Erelie (erelie)</b></p> <p>Visibilité : <b>par défaut</b> </p> <p>Machine : <b>me070121.etu.univ-rennes1.fr</b></p> <p>Priorité : <b>NORMALE</b></p> <p>Temps passé : <i>5 minute(s)</i></p> <p>Création : <b>07-02-2006</b></p> <p>Dernière Action : <b>27-02-2006</b></p> <hr/> <p><b>Suivi du ticket</b></p> <p><input type="checkbox"/> Suivre ce ticket</p> <p>Utilisateurs suivant ce ticket :</p> <ul style="list-style-type: none"> <li>• Anne-Gaëlle Carmillat (26000887)</li> <li>• Francois Albert (albert)</li> <li>• Dominique Erelie (erelie)</li> <li>• Michele Guezou (guezou)</li> <li>• Irene Pennaneac'H (ibodin)</li> <li>• Loic Perdoux (lperdoux)</li> <li>• Rodolphe Marsolier (rmarsoli)</li> <li>• Gaston Stenger (stenger)</li> </ul>	<p><b>27-02-2006 : The ticket expired.</b></p> <p><b>16-02-2006 : Dominique Erelie (erelie) closed the ticket (waiting for acceptance or refusal).</b> Pas de nouvelles, bonnes nouvelles.</p> <p><b>09-02-2006 : Dominique Erelie (erelie) added information.</b> Bonjour, Je pense qu'il y a un pb de mot de passe. Pourriez-vous le changer (à partir de sesame) et refaire un essai de connexion. Si le pb persiste, repassez à la cellule informatique, bât 1. Merci de nous tenir au courant Cordialement</p> <p><b>09-02-2006 : Dominique Erelie (erelie) changed the owner to 'Anne-Gaëlle Carmillat (26000887)' (was previously 'Claude Boisseau (boisseau)').</b></p> <p><b>07-02-2006 : Dominique Erelie (erelie) took charge of the ticket.</b></p> <p><b>07-02-2006 : The application changed the department (was previously 'SVE').</b></p> <p> <b>07-02-2006 : The application added information.</b> uid = boisseau cn = Boisseau Claude departmentNumber = UR1, 912, 12NA telephoneNumber = +33 2 23 23 63 75 mail = claude.boisseau@univ-rennes1.fr mailAlternateAddress = boisseau@univ-rennes1.fr, claude.boisseau</p> <p><b>07-02-2006 : Claude Boisseau (boisseau) created the ticket.</b> 26000887, qui est passée cette AM, ne peut toujours pas se connecter. Pa s de pb, en revanche, pour son binôme !!!</p>


### 4.8. Ajout de modèles de ticket (ou tickets prédéfinis)

Les tickets prédéfinis ont été ajoutés afin de mieux cerner les problèmes des utilisateurs, et ainsi espérer avoir toutes les informations nécessaires à la résolution des tickets dès leur saisie.



## Edition du modèle de ticket

**Attention :** les données concernant les modèles de ticket ne sont prises en compte que par certaines classes de personnalisation, comme la classe `org.esupportail.portal.channels.CHelpdesk.custom.defaults.CCustomNewTicketProperties`. Pour savoir si les classes de personnalisation que vous avez paramétrées utilisent ces données, veuillez contacter l'administrateur de votre application.


Label court  :


Label long  :

---

Priorité  :  

Utiliser la priorité de la catégorie

Label du modèle de ticket  :

Texte  : 

Pour toute demande de restauration, veuillez préciser :  
 nom du serveur :  
 chemin du répertoire ou fichier à restaurer :  
 restauration dans le répertoire ou fichier existant avec le même nom (préciser oui ou non) :  
 date et heure approximative de la perte des fichiers :

Utiliser le message de la catégorie

**Exercice 23 :** Ajouter la catégorie « Impression » à une cellule de proximité, puis les tickets prédéfinis « bourrage d'imprimante », « plus de papier ».
















### 4.9. Les remontées d'alerte

Afin d'éviter les effets de SPAM, elles sont hautement configurables, par les utilisateurs eux-mêmes.

#### Pour les gestionnaires

Exemple (dans les préférences des gestionnaires) :

##### Remontées d'alertes pour le service IFSIC

	création	clôture	autres
n'importe quel ticket	JAMAIS 	JAMAIS 	JAMAIS 
ticket que je gère	≥ TRÈS BASSE 	≥ TRÈS BASSE 	≥ TRÈS BASSE 
ticket dans une catégorie dont je suis membre	≥ NORMALE 	≥ NORMALE 	≥ NORMALE 
ticket libre	JAMAIS 	JAMAIS 	≥ TRÈS ÉLEVÉE 
ticket sans catégorie	JAMAIS 	JAMAIS 	JAMAIS 

Note : il est possible de configurer les préférences utilisées par défaut pour les nouveaux gestionnaires en éditant les tags `/config/ticket-monitoring/managers-default-preferences/*` du fichier `CHelpdeskConfig.xml`.

**Exercice 24 :** Tester les remontées d'alerte des gestionnaires.

## Pour tous les utilisateurs

Les utilisateurs reçoivent aussi des alertes pour les actions sur les tickets en tant que simple utilisateur (non gestionnaire), et ces alertes sont également configurables.

- Prévenez moi lorsqu'un évènement se produit sur un ticket que j'ai créé.
- Prévenez moi lorsqu'un évènement se produit sur un ticket auquel je suis lié.

Choisissez le format de vos courriers électroniques à adopter :

Les deux (votre lecteur de courrier électronique devrait choisir pour vous) ▼

**Exercice 25 :** Tester les modes d'envoi des remontées d'alerte (HTML, texte ou les deux).

## Suivi volontaire d'un ticket

Il est possible pour un utilisateur d'indiquer qu'il souhaite suivre un ticket particulier, dont ses préférences ne lui permettraient normalement pas de le suivre, en cochant sur la page d'un ticket la case suivante :

Suivre ce ticket

Décocher la case supprime ensuite le suivi volontaire.

**Exercice 26 :** Tester le suivi volontaire d'un ticket.

## Invitations à suivre un ticket

Un utilisateur peut inviter un autre à suivre un ticket en cliquant sur le bouton:

inviter un utilisateur

**Exercice 27 :** Tester l'invitation à suivre un ticket.

## 4.10. Les FAQs

### Intérêt

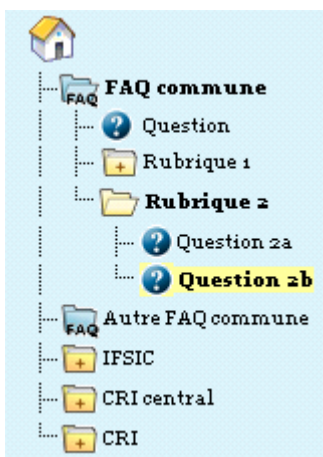
L'intérêt de disposer des FAQs dans l'outil (par opposition à l'utilisation d'un outil externe) est triple :

- Etablir des liens entre les FAQs et les tickets (renvoyer un ticket sur une FAQ par exemple).
- Avoir des niveaux de visibilité basés sur les rôles des utilisateurs dans l'application.
- Indexer les FAQs et les tickets dans une même base de connaissance.

### Structuration

Les FAQs sont structurées de la manière suivante :

- Une entrée de FAQ est un couple question/réponse.
- Les entrées de FAQ sont dans des FAQs, éventuellement organisées en parties.
- Les FAQs sont soit rattachées à un service, soit sans service.



### Qui voit quoi ?

Chaque élément des FAQs (FAQ, partie de FAQ ou entrée de FAQ) a un niveau de visibilité qui peut être :

- **DEFAULT** : en fonction du nœud supérieur.
- **ALL** : tous les utilisateurs peuvent voir cet élément.
- **AUTHENTICATED** : seuls les utilisateurs authentifiés voient cet élément.
- **DEPARTMENT** : seuls les utilisateurs qui voient le service auquel est rattaché cet élément<sup>3</sup> voient l'élément.
- **MANAGER** : seuls les gestionnaires du service auquel est rattaché cet élément<sup>4</sup> voient l'élément.

### Qui peut modifier quoi ?

Un élément de FAQ peut être modifié par les gestionnaires du service auquel l'élément est rattaché et qui possèdent les privilèges suffisants.

<sup>3</sup> Dans le cas d'un élément non rattaché à un service, les utilisateurs authentifiés voient cet élément.

<sup>4</sup> Dans le cas d'un élément non rattaché à un service, les administrateurs voient cet élément.

Lorsqu'un élément n'est rattaché à aucun service, alors seuls les administrateurs peuvent le modifier.

**Exercice 28 :** Créer une arborescence de FAQs.

## Exportation des FAQs

Les FAQs peuvent être exportées en mode « ligne de commande » aux formats XML et/ou PDF. Il faut pour cela exécuter `ant export-faqs`, qui utilise le fichier de configuration `faqExport.xml`.

Voici un exemple de configuration pour l'exportation des FAQs :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE exports SYSTEM "faqExport.dtd">

<exports output-dir="c:/esup-helpdesk/faq-export"
  default-output-format="both">
  <!-- export all the FAQs to PDF for outside users -->
  <export output-basename="outside" output-format="pdf"
    role="unauthenticated"/>
  <!-- export the FAQs to PDF for authenticated users -->
  <export output-basename="local" output-format="pdf"
    role="authenticated"/>
  <!-- export the FAQs of department MATH for the non managers -->
  <export output-basename="math"
    role="department-member" department-label="MATH"/>
  <!-- export a specific FAQ to XML for managers -->
  <export output-basename="1"
    role="department-manager" faq-id="1"/>
  <!-- export the root FAQs for unauthenticated users -->
  <export output-basename="root"
    role="unauthenticated" root-faqs="true"/>
</exports>
```

**Exercice 29 :** Créer le fichier `faqExport.xml` à partir du fichier d'exemple `faqExport-example.xml` et tester l'exportation des FAQs.

Note : le choix des services peut également être fait en utilisant la notion de filtre de service, notion qui sera développée à la partie 5.4.

## 4.11. La base de connaissances

Permet aux utilisateurs de chercher eux-mêmes la solution à leurs problèmes (accélération de la résolution des problèmes, parfois sans poster de ticket).

Les tickets et les FAQs sont indexés par Lucene, dans la même base. Les fichiers de Lucene sont stockés dans le répertoire spécifié par le tag de configuration `/config/indexing/storage-path` du fichier `CHelpdeskConfig.xml`.

L'interface de recherche en mode avancé propose différents critères.

**Exercice 30 :** Mettre en place l'indexation automatique à l'aide d'une tâche avancée (appeler automatiquement `ant update-index`) et tester l'interface de recherche avancée.

**Exercice 31 :** tester également la tâche `create-index`.



## 4.12. Les préférences des utilisateurs

Accueil FAQ Tableau de bord Recherche Journal Services Administration **Préférences** A propos de

### Préférences de l'utilisateur

#### Identité

Vous êtes authentifié comme un utilisateur uPortal.

- Identifiant : **paubry**
- Courrier électronique : **pascal.aubry@univ-rennes1.fr**
- Identité : **Pascal Aubry**

#### Langue

Choisissez votre langue :

Comportement de l'application

Remontées d'alertes

Accueil FAQ Tableau de bord Recherche Journal Services Administration **Préférences** A propos de

### Préférences de l'utilisateur (comportement de l'application)

#### Suivi automatique de tickets

Le suivi de ticket est une fonctionnalité permettant aux utilisateurs d'être prévenu par courrier électronique lorsqu'un évènement se produit sur un ticket. Les utilisateurs peuvent choisir un suivi personnalisé à chaque ticket. Il peuvent aussi définir un suivi par défaut automatique sur les tickets.

Choisissez le type de tickets dont vous souhaitez automatiquement avoir un suivi :

- Prévenez moi lorsqu'un évènement se produit sur un ticket que j'ai créé.
- Prévenez moi lorsqu'un évènement se produit sur un ticket auquel je suis lié.

Choisissez le format de vos courriers électroniques à adopter :

#### Rafraîchissement automatique des pages

Le tableau de bord et le journal peuvent être configurés pour être périodiquement rafraîchis. Ainsi, les nouveaux tickets et les nouvelles actions apparaissent automatiquement.

Rafraîchir mon tableau de bord toutes les  minute(s).

Entrez 0 pour aucun rafraîchissement, ou laissez blanc pour prendre la valeur par défaut (5 minute(s)).

Rafraîchir mon journal toutes les  minute(s).

Entrez 0 pour aucun rafraîchissement, ou laissez blanc pour prendre la valeur par défaut (5 minute(s)).

#### Page de démarrage

Cochez si vous souhaitez démarrer sur le tableau de bord à la place de la page de bienvenue.

#### Affichage des tickets modifiés

Cochez si vous souhaitez afficher en gras les tickets modifiés depuis votre dernière visite du tableau de bord.

#### Page d'accueil des nouveaux tickets

Cochez si vous souhaitez afficher la page d'explication lors de l'ajout d'un ticket.

#### Message d'avertissement à la clôture des tickets

Cochez si vous souhaitez afficher un message d'avertissement à la clôture/annulation des tickets.

## 4.13. Mise à jour d'une version existante

Comme on le constate facilement, le nombre de fichiers de configuration est assez important lorsque l'on utilise toutes les fonctionnalités de l'application. C'est pourquoi la procédure de mise à jour permet de récupérer automatiquement tous les fichiers de configuration et les personnalisations de la version précédente.

## Éléments de configuration récupérés automatiquement

Plus précisément, les fichiers de configuration suivants sont récupérés :

- `CEsupConfig.xml`,
- `build.properties`,
- `CHelpdeskConfig.xml`,
- `helpdeskChannelHibernate.cfg.xml`,
- `batchLogger.properties`,
- `departmentSelection.xml`,
- `LdapSearchConfig.xml`,
- `esup-helpdesk_*-custom.properties`

**Canal uPortal :** La procédure récupère également le répertoire de personnalisation `/custom`.

**Quick Start :** La procédure récupère également les répertoires de personnalisation `/esup-custom` et `/channel-custom`.

## Procédure à suivre

Il faut tout d'abord s'assurer que la propriété `app.version.previous` correspond bien à votre version précédente, si besoin la modifier pour refléter votre installation.

**Canal uPortal :** La propriété `app.version.previous` est définie dans le fichier `version.properties`.

**Quick Start :** La propriété `app.version.previous` est définie dans le fichier `esup-helpdesk-channel-version.properties`.

Il suffit ensuite d'exécuter `ant recover-config`.

**Exercice 32 :** Mettre à jour votre installation avec les packages fournis par les formateurs.

## 5. Concepts et opérations avancées

### 5.1. Utilisation des classes de personnalisation fournies avec la distribution

La configurabilité de esup-helpdesk est atteinte en particulier grâce à l'utilisation de classes de personnalisation, sur lesquelles on se base pour tous les besoins non génériques. Il s'agit de classes Java, qui doivent implémenter des interfaces données pour être reconnues comme des classes de personnalisation.

Cette partie de la formation illustre comment utiliser des classes de personnalisation déjà existantes (l'ajout de nouvelles classes de personnalisation sera vu en fin de formation).

**Canal uPortail :** Des classes d'exemple sont fournies dans la distribution, dans le répertoire `/custom-examples/esup-portal.org/source`. Pour les utiliser, il faut copier ce répertoire `source` avec toute son arborescence dans le répertoire `/custom`.

**Quick Start :** Des classes d'exemple sont fournies dans la distribution, dans le répertoire `/channel-custom-examples/esup-portal.org/source`. Pour les utiliser, il faut copier ce répertoire `source` avec toute son arborescence dans le répertoire `/channel-custom`.

Il faut également indiquer à l'application dans quel package se trouvent les classes de personnalisation. Ajouter pour cela dans `CHelpdeskConfig.xml` le tag `/config/custom/packages/package-name` avec la valeur `org.esupportail.portal.channels.CHelpdesk.custom.examples`.

Remarque : on peut utiliser plusieurs tags `package-name` et ainsi utiliser des classes de personnalisation issues de plusieurs sources.

Une fois le canal redéployé et redémarré, les classes d'exemple sont disponibles à l'utilisation dans l'application.

### 5.2. L'assignation automatique des tickets

Afin de diminuer le temps de prise en charge des tickets, il est possible de les assigner automatiquement aux gestionnaires. Cela se fait en éditant les services et les catégories (en utilisant des classes de personnalisation).

Les algorithmes prédéfinis sont :

- au premier gestionnaire disponible,
- de manière cyclique,
- de manière aléatoire,
- au gestionnaire le moins chargé,

**Exercice 33 :** Editer les propriétés d'une catégorie en indiquant une assignation automatique des tickets pour la catégorie, créer un ticket dans la catégorie et observer l'historique du nouveau ticket.

**Exercice 34 :** Editer les propriétés d'un service en indiquant une assignation automatique des tickets pour le service, créer un ticket sans catégorie dans le service et observer l'historique du nouveau ticket.

En développant une classe de personnalisation, on peut implémenter n'importe quel algorithme, par exemple en se liant à une application externe telle un gestionnaire des congés des gestionnaires.

### 5.3. La virtualisation

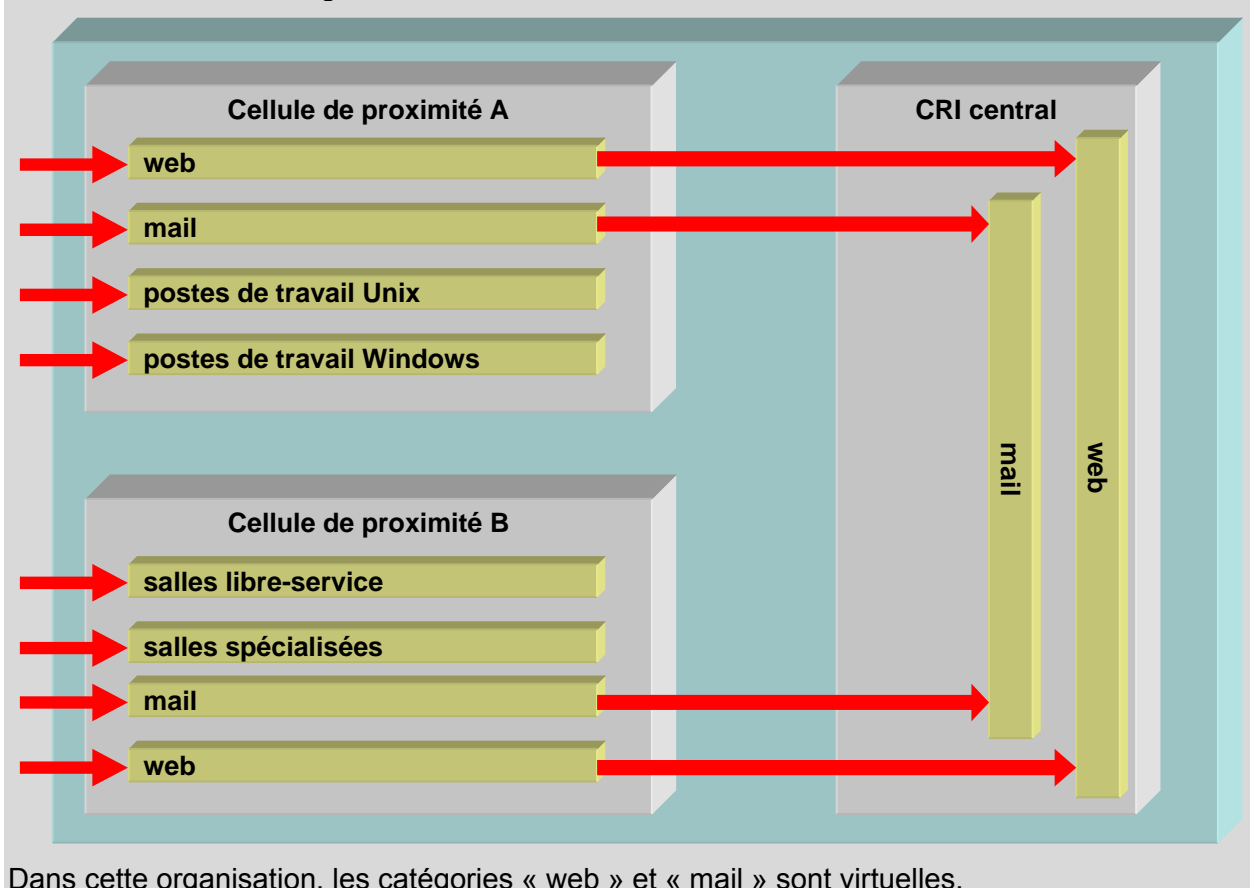
La virtualisation est la possibilité pour l'application de rediriger automatiquement les tickets créés.

**Exercice 35 :** Editer les propriétés d'une catégorie en la virtualisant vers une autre catégorie du même service, créer un ticket dans la catégorie et observer l'historique du nouveau ticket.

**Exercice 36 :** Recommencer en virtualisant la catégorie vers une autre catégorie d'un autre service.

**Exercice 37 :** Virtualiser un service (applicable aux tickets sans catégorie du service).

**Exercice 38 :** Créer l'organisation suivante :



### 5.4. Mise en œuvre d'une vision adaptative

Afin de limiter le nombre de services, on peut utiliser une classe de personnalisation qui donne pour chaque utilisateur, les services qu'il verra dans le helpdesk.

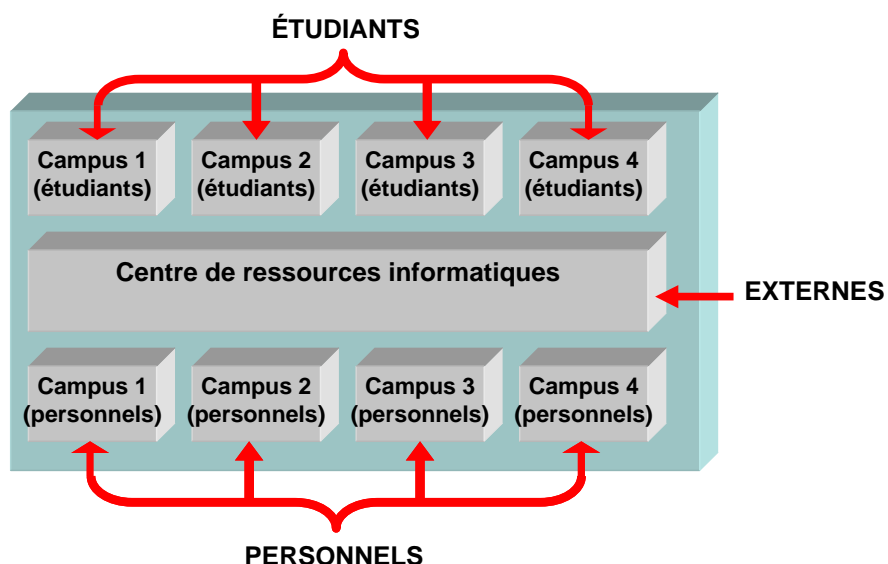
La vision du helpdesk par les utilisateurs peut être fonction :

- **Du rôle des utilisateurs dans l'application.** On peut par exemple faire en sorte que tous les gestionnaires (ou tous les gestionnaires d'un service) aient accès à certains services.
- **Du profil des utilisateurs.** On peut par exemple faire en sorte que les utilisateurs externes (identifiés par leur adresse électronique) aient une vision particulière, ou encore que les utilisateurs locaux ayant un attribut positionné à une valeur donnée en aient une autre vision.
- **De la localisation physique des utilisateurs.** En analysant l'adresse (IP ou FQDN) du client des utilisateurs, on pourra par exemple faire apparaître des services seulement depuis une plage réseau donnée.

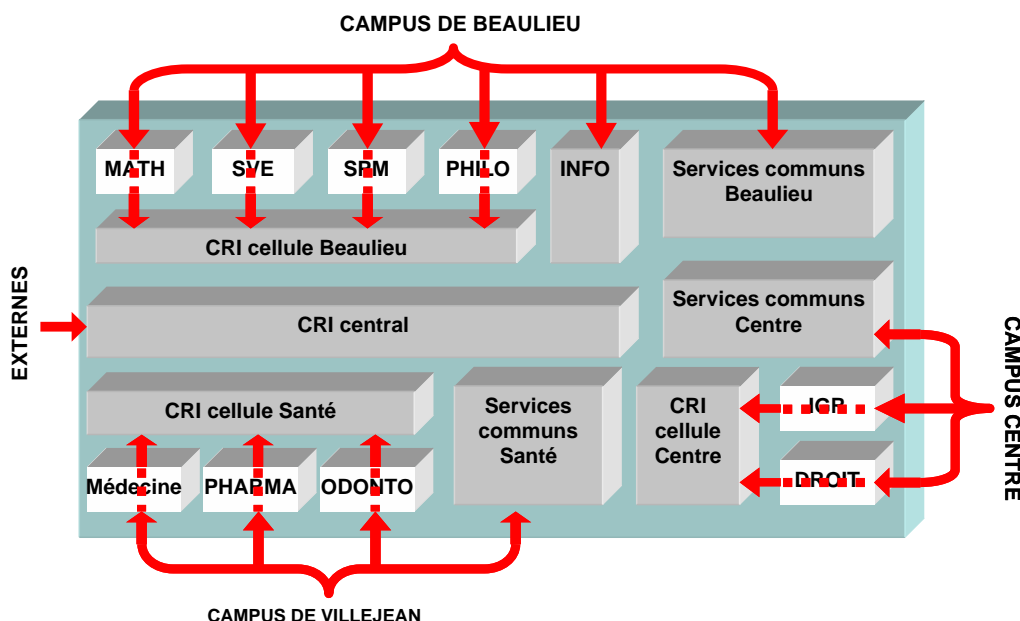
## Exemples d'organisations

Les exemples présentés ci-dessous ont volontairement été simplifiés.

### Université de Nancy 2



## Université de Rennes 1



### Utilisation de la classe `custom.departmentSelection.CXmlConfig`

Afin de pouvoir utiliser cette fonctionnalité sans développement, une classe de personnalisation générique a été développée, qui s'appuie sur un fichier de configuration au format XML : `departmentSelection.xml`.

Pour utiliser cette classe, il faut l'indiquer à l'application en donnant au tag `/config/custom/department-selection-class` du fichier `CHelpdeskConfig.xml` la valeur :

```
org.esupportail.portal.channels.CHelpdesk.custom.departmentSelection.CXmlConfig
```

La description complète des possibilités de personnalisation est à l'URL :

[http://esup-helpdesk.sourceforge.net/channel/custom\\_departmentSelection.html](http://esup-helpdesk.sourceforge.net/channel/custom_departmentSelection.html)

Pour tester la classe de personnalisation :

- Créer `departmentSelection.xml`.
- Déployer et exécuter `ant test-department-selection`, qui utilise les cas de test spécifiés en fin du fichier `departmentSelection.xml`.

Nous allons illustrer l'utilisation de la classe par un exemple concret. Supposons que l'on veuille que la visibilité des services soit le résultat des contraintes suivantes :

- Les utilisateurs externes ne voient que le CRI.
- Tous les gestionnaires voient la cellule de proximité A et le CRI central.
- Les utilisateurs possédant un attribut `composante` égal à 913 voient la cellule de proximité B.
- Les utilisateurs possédant un attribut `eduPersonAffiliation` égal à `member` voient le CRI central.

Une implémentation possible des contraintes ci-dessus est la suivante :

```

<custom>
  <description>University of Rennes 1</description>
  <define-conditions>
    <define-condition name="staff">
      <attribute-eq name="eduPersonAffiliation" value="member"/>
    </define-condition>
  </define-conditions>
  <list>
    <list-rule>
      <condition>
        <external-user/>
      </condition>
      <actions>
        <add-by-label label="CRI"/>
        <stop/>
      </actions>
    </list-rule>
    <list-rule>
      <condition>
        <department-manager/>
      </condition>
      <actions>
        <add-by-label label="PROX_A"/>
        <add-by-label label="CRI"/>
      </actions>
    </list-rule>
    <list-rule>
      <condition>
        <attribute-eq name="composante" value="913"/>
      </condition>
      <actions>
        <add-by-label label="PROX_B"/>
      </actions>
    </list-rule>
    <list-rule>
      <condition>
        <named-condition name="staff"/>
      </condition>
      <actions>
        <add-by-label label="CRI"/>
      </actions>
    </list-rule>
    <when-empty>
      <add-by-label label="CRI"/>
    </when-empty>
  </list>
  <test>
    <case uid="paubry"
      client="148.60.4.1"/>
    <case uid="alexandre.boisseau@univ-brest.fr"
      client="www.univ-rennes1.fr"/>
    <case uid="56913021"
      client="www.esup-portail.org"/>
  </test>
</custom>

```

**Exercice 39 :** Mettre en œuvre une vision adaptative du helpdesk avec les contraintes données ci-dessus.

Dans l'exemple précédent, les services sont sélectionnés par leur label. Les services peuvent également être spécifiés par leur filtre, ce qui permet d'avoir dans le helpdesk deux services qui ont le même label.

Le filtre d'un service est une chaîne de caractères, comme le label, éditable par l'interface web (sur la même page que les autres propriétés des services). La sélection des services correspondant à un filtre est effectuée à l'aide d'un tag :

```
<add-by-filter filter="xxx"/>.
```

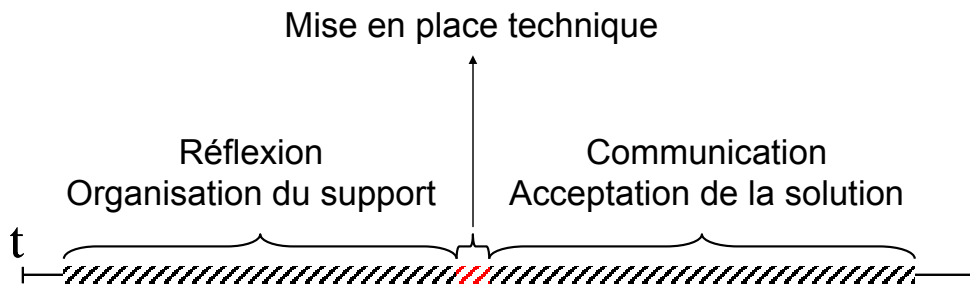
On peut également combiner filtre et label en utilisant :

```
<add-by-label label="xxx" filter="yyy"/>.
```

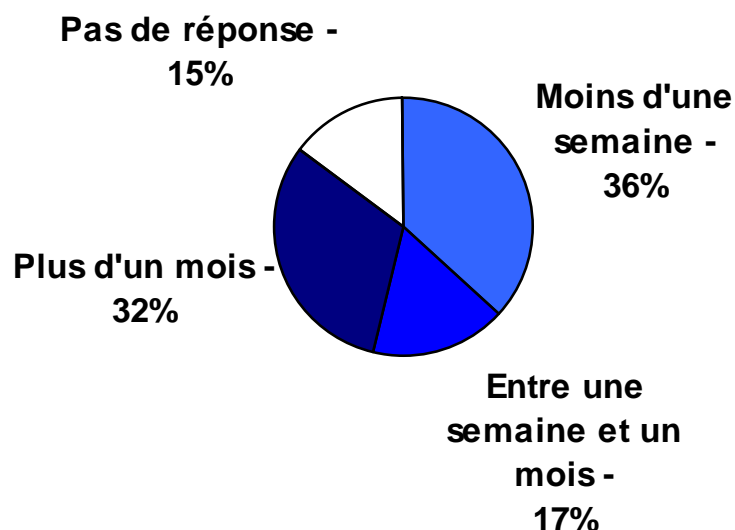
**Exercice 40 :** Tester la sélection des services par filtre.

### Combien de temps cela prend-il ?

Pas mal, mais pas toujours là où l'on pense :



La mise en place d'un helpdesk est un projet en soi (cf GT-SSD) :





## 5.5. Ajout automatique d'informations sur les utilisateurs à la création des tickets

Afin de limiter la recherche par les gestionnaires d'information sur les utilisateurs, il est possible d'ajouter, à chaque création de ticket, des informations sur le propriétaire du ticket.

Afin de pouvoir utiliser cette fonctionnalité sans développement, une classe de personnalisation générique a été développée, qui s'appuie sur un fichier de configuration au format XML : `userInfo.xml`.

Pour utiliser cette classe, il faut l'indiquer à l'application en donnant au tag `/config/custom/user-info-class` du fichier `CHelpdeskConfig.xml` la valeur :

`org.esupportail.portal.channels.CHelpdesk.custom.userInfo.CXmlConfig`

La description complète des possibilités de personnalisation est à l'URL :

[http://esup-helpdesk.sourceforge.net/channel/custom\\_userInfo.html](http://esup-helpdesk.sourceforge.net/channel/custom_userInfo.html)

Pour tester la classe de personnalisation :

- Créer `userInfo.xml`.
- Déployer et exécuter `ant test-userInfo`, qui utilise les cas de test spécifiés en fin du fichier `userInfo.xml`.

Nous allons illustrer l'utilisation de la classe par un exemple concret.

Supposons que l'on veuille que l'application ajoute les attributs `uPortal` suivants à chaque création de ticket :

- `cn`
- `departmentNumber`
- `telephoneNumber`
- `email`
- `mailAlternateAddress`
- `homedir`

Supposons enfin que nous voulions tester cette configuration pour les utilisateurs `paubry` et `56913021`.

Le contenu du fichier `userInfo.xml` devra alors être :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE custom SYSTEM "userInfo.dtd" >
<custom>
  <description>This is my customization class</description>
  <attributes>
    <attribute>cn</attribute>
    <attribute>departmentNumber</attribute>
    <attribute>telephoneNumber</attribute>
    <attribute>email</attribute>
    <attribute>mailAlternateAddress</attribute>
    <attribute>homedir</attribute>
  </attributes>
  <test>
    <case uid="paubry"/>
    <case uid="56913021"/>
  </test>
</custom>
```

**Exercice 41 :** Mettre en œuvre l'ajout automatique d'informations avec les contraintes données ci-dessus.

## 5.6. Alimentation à partir d'une boîte IMAP

Le répondeur mail est un procédé permettant de transformer automatiquement un courrier électronique en ticket. Cette fonctionnalité a été développée pour faciliter le passage du courrier électronique au helpdesk.

Exemple : vous avez une adresse email : `sos@univ.fr`, sur laquelle les utilisateurs envoient leurs problèmes sur cette boîte.

A intervalle régulier, un *cron* (sous Unix, ou une tâche programmée sous Windows) vient scanner cette boîte et transformer les courriers électroniques en tickets, à l'aide de la commande :

```
ant feed-with-email
```

Une analyse est effectuée par l'outil pour déterminer le propriétaire du ticket (l'expéditeur du courrier), le sujet du ticket (celui du courrier) et le contenu du ticket (le contenu du courrier). Les pièces jointes sont transformées en fichiers attachés.

**Exercice 42 :** Configurer un répondeur en utilisant les informations données par les formateurs.

Notes :

- on peut créer plusieurs répondeurs.
- On peut transférer les demandes d'interventions reçues par courrier électronique au répondeur, celui-ci les transformera automatiquement en tickets.

## 5.7. Personnalisation de l'application

### Look n feel

**Canal uPortal :** La personnalisation du look du canal est normalement faite par le portail.

**Quick Start :** La personnalisation du look de l'application est faite en remplaçant des fichiers de l'installation existante. Pour cela, les fichiers doivent être placés dans le répertoire `/esup-custom`, qui est automatiquement utilisé lors de l'exécution de `ant install`. Des exemples de personnalisation du look peuvent être trouvés dans le répertoire `/esup-custom-examples`.

**Exercice 43 :** Ajouter l'icône de votre institution dans l'application.

### Chaînes de caractères

L'application est complètement internationalisée (livrée aujourd'hui en français et anglais). Toutes les chaînes de caractères utilisées sont trouvées dans des fichiers de propriétés appelés *bundles*.

Pour modifier les chaînes utilisées, la bonne méthode est d'utiliser les fichiers `esup-helpdesk_*-custom.properties`.

**Exercice 44 :** Modifier le titre de la page d'accueil en le remplaçant par « Bienvenue sur le helpdesk de formation » (en anglais et en français).

## Autres personnalisations

**Canal uPortal:** Pour *patcher* n'importe quel élément de l'application, il suffit d'utiliser le répertoire de personnalisation `/custom`, qui est pris en compte lorsque l'on exécute `ant deploy`.

**Quick Start:** Pour *patcher* n'importe quel élément de l'application, il suffit d'utiliser le répertoire de personnalisation `/channel-custom`, qui est pris en compte lorsque l'on exécute `ant install`.

L'intérêt d'utiliser les répertoires de personnalisation et ne pas toucher « sauvagement » aux fichiers déployés de l'application est que la procédure de mise à jour permet de récupérer automatiquement toutes les personnalisations.

### 5.8. Expiration automatique des tickets non approuvés

Cette fonctionnalité permet d'expirer les tickets qui ont été clos par les gestionnaires mais dont la clôture n'a pas été approuvée par les propriétaires des tickets.

La fonctionnalité s'active simplement en exécutant `ant expire-old-tickets`, et est généralement appelée depuis une *crontab* (sous Unix, ou une tâche programmée sous Windows).

Note : le délai d'expiration est paramétrable, par service.

### 5.9. Développement de classes de personnalisation

Avertissement : le développement de classes de personnalisation nécessite des compétences basiques de développement Java. Il est simple pour les administrateurs ayant cette compétence, notamment parce que des classes d'exemple sont fournies dans la distribution. Il sera en revanche quasi inaccessible pour ceux qui n'ont jamais développé en Java.

Dans les deux cas, l'objectif de la formation n'est pas de développer des classes, mais de bien comprendre les personnalisations possibles par ce biais.

**Canal uPortal:** Les contributions sont situées dans le répertoire d'exemple `custom-examples`.

**Quick Start:** Les contributions sont situées dans le répertoire d'exemple `channel-custom-examples`.

Chaque classe de personnalisation doit implémenter deux interfaces :

- `ICustom`, commune à toutes les classes de personnalisation
- Une interface particulière correspondant à la personnalisation souhaitée.

## L'interface commune `ICustom`

L'interface `ICustom` fournit simplement une méthode permettant d'associer une chaîne de caractères à la classe de personnalisation. Cette chaîne est notamment utilisée pour l'identification des classes dans les boîtes de sélection déroulante :

```
/**
 * Basic features common to all customization classes.
 */
public interface ICustom {
    /**
     * Return the description of the class. This description is used to
     * ease the choice of users when choosing a customization class.
     * @param language the language
     * @return a (free-style) string.
     */
    String getDescription(String language);
}
```

## L'interface `ICustomComputerUrl`

### Utilisation

L'implémentation de cette interface permet d'ajouter automatiquement sur les noms de machines une URL, par exemple pour se lier à une application d'inventaire existante.

Les classes implémentant l'interface `ICustomComputerUrl` sont utilisées dans les propriétés des services et la classe utilisée par défaut par les services est définie par le tag `/custom/default-classes/computer-url`.

### Définition

La définition de l'interface est la suivante :

```
/**
 * Features needed to customize the URLs of computers.
 */
public interface ICustomComputerUrl {
    /**
     * Return the URL corresponding to a computer.
     * @param session a Hibernate session
     * @param computer the computer
     * @return a URL, or null.
     * @throws Exception
     */
    String getComputerUrl(Session session, String computer)
    throws Exception;
}
```

### Classe incluse dans la distribution

La classe `custom.defaults.CCustomComputerUrl` n'ajoute pas de lien hypertexte aux machines (la méthode `getComputerUrl()` renvoie `null`).

### Contribution

La classe `custom.examples.CCustomComputerUrlExternal` renvoie un lien vers une application externe en passant en paramètre le nom de la machine.

## Exemple

Le code de la classe `custom.examples.CCustomComputerUrlExternal` est le suivant :

```
/**
 * An example of ICustomComputerUrl implementation.
 */
public final class CCustomComputerUrlExternal
implements ICustom, ICustomComputerUrl {

    /**
     * @see ICustom#getDescription(java.lang.String)
     */
    public String getDescription(final String language) {
        if (language.equals("fr")) {
            return "lien vers une application d'inventaire";
        }
        return "link to an external inventory application";
    }

    /**
     * @see ICustomComputerUrl#getComputerUrl(
     *      org.hibernate.Session, java.lang.String)
     */
    public String getComputerUrl(
        final Session unusedSession,
        final String computer) {
        return new StringBuffer()
            .append("http://inventory.domain.edu/show.cgi?computer=")
            .append(computer).append("&orig=esup-helpdesk")
            .toString();
    }
}
```

Pour une machine `xxx`, l'URL associée sera :

```
http://inventory.domain.edu/show.cgi?computer=xxx&orig=esup-helpdesk
```

## L'interface `ICustomNewTicketAssignment`

### Utilisation

L'implémentation de cette interface permet d'assigner automatiquement les nouveaux tickets à des gestionnaires.

Les classes implémentant l'interface `ICustomNewTicketAssignment` sont utilisées dans les propriétés des services et des catégories. La classe utilisée par défaut (par les services) est définie par le tag `/custom/default-classes/new-ticket-assignment`.

## Définition

La définition de l'interface est la suivante :

```
/**
 * Features needed to customize the assignment of new tickets.
 */
public interface ICustomNewTicketAssignment {
    /**
     * Return the assignment of a new ticket
     * (manager and state of the algorithm).
     * @param session a Hibernate session
     * @param ticket the new ticket
     * @return a CNewTicketAssignment instance.
     * @throws Exception
     */
    CNewTicketAssignment getAssignment(Session session, CTicket ticket)
    throws Exception;
}
```

## Classe incluse dans la distribution

La classe `custom.defaults.CCustomNewTicketAssignment` n'assigne pas les nouveaux tickets (ils sont laissés libres).

## Contributions

La classe `custom.examples.CCustomNewTicketAssignmentFirst` assigne les nouveaux tickets au premier gestionnaire de la catégorie du ticket (du service du ticket si le ticket n'est pas placé dans une catégorie).

La classe `custom.examples.CCustomNewTicketAssignmentCyclical` assigne les nouveaux tickets aux gestionnaires de la catégorie du ticket (ou du service du ticket si le ticket n'est pas placé dans une catégorie) de manière cyclique.

La classe `custom.examples.CCustomNewTicketAssignmentRandom` assigne les nouveaux tickets aux gestionnaires de la catégorie du ticket (ou du service du ticket si le ticket n'est pas placé dans une catégorie) de manière aléatoire.

## Exemple

Le code de la classe `custom.examples.CCustomNewTicketAssignmentFirst` est le suivant :

```
/**
 * An example of ICustomNewTicketAssignment implementation, assigning
 * tickets to the first managers available of the category, or of the
 * department if no category given.
 */
public final class CCustomNewTicketAssignmentFirst
implements ICustom, ICustomNewTicketAssignment {
    /* Constructor. */
    public CCustomNewTicketAssignmentFirst() {
        super();
    }
    /**
     * @see ICustom#getDescription(java.lang.String)
     */
    public String getDescription(final String language) {
        if (language.equals("fr")) {
            return "assignation au premier gestionnaire disponible";
        }
        return "assignment to the first available manager";
    }
    /**
     * @see ICustomNewTicketAssignment#getAssignment(
     * org.hibernate.Session,
     * org.esupportail.portal.channels.CHelpdesk.CTicket)
     */
    public CNewTicketAssignment getAssignment(
        final Session session,
        final CTicket ticket)
        throws CDepartmentManagerNotFoundException {
        CNewTicketAssignment newta = new CNewTicketAssignment();

        if (ticket.getCategory() != null) {
            // no category, choose the first manager of the department
            List groupMembers =
                ticket.getCategory().getAvailableMembers(session);
            if (!groupMembers.isEmpty()) {
                CCategoryMember choosedMember =
                    (CCategoryMember) groupMembers.iterator().next();
                nta.setAssignedUser(choosedMember.getUser());
            }
        } else {
            // choose the first manager of the category
            List managers =
                ticket.getDepartment().getAvailableManagers(session);
            if (!managers.isEmpty()) {
                CDepartmentManager choosedManager =
                    (CDepartmentManager) managers.iterator().next();
                nta.setAssignedUser(choosedManager.getUser());
            }
        }
        return nta;
    }
}
```

## L'interface `ICustomNewTicketProperties`

### Utilisation

L'implémentation de cette interface permet de pré remplir les nouveaux tickets (sujet, priorité et message) en fonction de :

- L'utilisateur qui crée le ticket ;
- Le service dans lequel va être créé le ticket ;
- La catégorie (éventuellement vide) dans laquelle va être créé le ticket ;
- Le modèle de ticket utilisé ;

Les classes implémentant l'interface `ICustomNewTicketProperties` sont utilisées dans les propriétés des services, et la classe utilisée par défaut est définie par le tag `/custom/default-classes/new-ticket-properties`.

### Définition

La définition de l'interface est la suivante :

```
/**
 * Features needed to customize the properties of new tickets.
 */
public interface ICustomNewTicketProperties {
    /**
     * Return the initial properties of a new ticket.
     * @param session a Hibernate session
     * @param language the language
     * @param user the user creating the new ticket
     * @param department the department
     * @param category the category
     * @param ticketTemplate the template to use
     * @return a CNewTicketProperties.
     * @throws Exception
     */
    CNewTicketProperties getNewTicketProperties(
        final Session session,
        final CUser user,
        final String language,
        final CDepartment department ,
        final CCategory category ,
        final CTicketTemplate ticketTemplate) throws Exception;
}
```

### Classe incluse dans la distribution

La classe `custom.defaults.CCustomNewTicketProperties` récupère le sujet, la priorité par défaut et le message du ticket dans la base de données.



## Contributions

La classe `custom.examples.CCustomNewTicketPropertiesSimple` pré remplit le ticket de la manière suivante :

- Le sujet est toujours le même ;
- Le niveau de priorité est celui par défaut de l'application ;
- Le message rappelle l'identifiant, l'adresse électronique et l'identité de l'utilisateur qui crée le ticket.

## Exemple

Le code de la classe `custom.examples.CCustomNewTicketPropertiesSimple` est le suivant :

```
/**
 * An example of ICustomNewTicketProperties implementation.
 */
public final class CCustomNewTicketPropertiesSimple
implements ICustom, ICustomNewTicketProperties {
    /** constructor. */
    public CCustomNewTicketPropertiesSimple() {
        super();
    }
    /**
     * @see ICustom#getDescription(java.lang.String)
     */
    public String getDescription(final String language) {
        if (language.equals("fr")) {
            return " identifiant, email et identit\u00E9";
        }
        return "print id, email and display name";
    }
    /**
     * @see ICustomNewTicketProperties#getNewTicketProperties(
     * org.hibernate.Session,
     * CHelpdesk.CUser,
     * java.lang.String,
     * CHelpdesk.CDepartment,
     * CHelpdesk.CCategory,
     * CHelpdesk.CTicketTemplate)
     */
    public CNewTicketProperties getNewTicketProperties(
        final Session unusedSession,
        final CUser user,
        final String unusedLanguage,
        final CDepartment unusedDepartment,
        final CCategory unusedGroup,
        final CTicketTemplate unusedTicketTemplate) {
        return new CNewTicketProperties(
            "please change this title",
            "id: " + user.getId()
                + "\nemail: " + user.getEmail()
                + "\nidentity: " + user.getDisplayName(),
            CHelpdeskConfig.getDefaultPriorityLevel());
    }
}
```

## L'interface `ICustomTicketMonitoring`

### Utilisation

L'implémentation de cette interface permet d'envoyer des courriers électroniques à des utilisateurs (de l'application) ou à des adresses électroniques données quand il se passe une action sur un ticket.

Notons que ces éventuels envois s'ajoutent à ceux effectués normalement par les remontées d'alerte de l'application (suivi des tickets par les utilisateurs).

Les classes implémentant l'interface `ICustomTicketMonitoring` sont utilisées dans les propriétés des services. La classe utilisée par défaut est définie par le tag `/custom/default-classes/ticket-monitoring`.

### Définition

La définition de l'interface est la suivante :

```
/**
 * Features needed to customize ticket monitoring.
 */
public interface ICustomTicketMonitoring {
    /**
     * Return the users and emails to warn.
     * @param session a Hibernate session
     * @param ticket the new ticket
     * @return a list of emails.
     * @throws Exception
     */
    CTicketMonitoringResults getResults(Session session, CTicket ticket)
    throws Exception;
}
```

### Classe incluse dans la distribution

La classe `custom.defaults.CCustomTicketMonitoring` n'envoie aucun courrier supplémentaire.

### Contributions

La classe `custom.examples.CCustomTicketMonitoringFixed` envoie des courriers électroniques à une adresse externe.

## Exemple

Le code de la classe `custom.examples.CCustomTicketMonitoringFixed` est le suivant :

```
/**
 * An example of ICustomTicketMonitoring implementation that
 * sends emails to an external email adress.
 */
public final class CCustomTicketMonitoringFixed
implements ICustom, ICustomTicketMonitoring {
    /** * constructor. */
    public CCustomTicketMonitoringFixed() {
        super();
    }
    /**
     * An external email address.
     */
    private String email = "sysadmin@domain.edu";

    /**
     * @see ICustom#getDescription(java.lang.String)
     */
    public String getDescription(final String language) {
        return new StringBuffer("send to ")
            .append(email).append(" ").toString();
    }
    /**
     * @see ICustomTicketMonitoring#getResults(
     * org.hibernate.Session,
     * org.esupportail.portal.channels.CHelpdesk.CTicket)
     */
    public CTicketMonitoringResults getResults(
        final Session unusedSession,
        final CTicket unusedTicket) {
        CTicketMonitoringResults results = new CTicketMonitoringResults();
        results.addEmail(this.email, false);
        return results;
    }
}
}
```

## L'interface ICustomDepartmentSelection

### Utilisation

L'implémentation de cette interface permet d'indiquer quels sont les services visibles par les utilisateurs en fonction de :

- L'identifiant de l'utilisateur ;
- L'adresse (IP ou FQDN) du client sur lequel il se trouve.

Les classes implémentant l'interface `ICustomDepartmentSelection` sont utilisées dans la configuration de l'application. La classe utilisée par l'application est spécifiée par le tag `/custom/department-selection-class`.

## Définition

La définition de l'interface est la suivante :

```
/**
 * Features needed to customize the choice of the departments for a user.
 */
public interface ICustomDepartmentSelection {
    /**
     * Return a list of accessible departments corresponding to a user.
     * @param session a Hibernate session
     * @param user the user
     * @param client the client
     * @return a list of departments, or null for all the departments.
     * @throws Exception
     */
    List getDepartments(
        Session session, CUser user, InetAddress client)
        throws Exception;
}
```

## Classes incluse dans la distribution

La classe `custom.defaults.CCustomDepartmentSelection` montre tous les services à tous les utilisateurs.

La classe `custom.departmentSelection` (vue précédemment) s'appuie sur un fichier de configuration au format XML.

## Exemple

Le code de la classe `custom.default.CDepartmentSelection` est le suivant :

```
/**
 * The default ICustomComputerUrl implementation.
 */
public final class CCustomDepartmentSelection
implements ICustom, ICustomDepartmentSelection {
    /** constructor. */
    public CCustomDepartmentSelection() {
        super();
    }
    /**
     * @see ICustom#getDescription(java.lang.String)
     */
    public String getDescription(final String language) {
        if (language.equals("fr")) {
            return "tous les services ";
        }
        return "all the departments";
    }
    /**
     * @see ICustomDepartmentSelection#getDepartments(
     * org.hibernate.Session, CHelpdesk.CUser, java.net.InetAddress)
     */
    public List getDepartments(
        final Session session,
        final CUser unusedUser,
        final InetAddress unusedClient) {
        return CDepartment.getVisibleDepartments(session);
    }
}
```

## L'interface ICustomUserUrl

### Utilisation

L'implémentation de cette interface permet d'ajouter automatiquement sur les noms d'utilisateurs des liens hypertextes, par exemple pour se lier à une application d'annuaire existante.

Les classes implémentant l'interface `ICustomUserUrl` sont utilisées dans la configuration de l'application. La classe utilisée par l'application est spécifiée par le tag `/custom/user-url`.

## Définition

La définition de l'interface est la suivante :

```
/**
 * Features needed to customize the URLs of users.
 */
public interface ICustomUserUrl {
    /**
     * Return the URL corresponding to a user.
     * @param session a Hibernate session
     * @param user the user
     * @return a URL, or null.
     * @throws Exception
     */
    String getUserUrl(Session session, CUser user) throws Exception;
}
```

## Classe incluse dans la distribution

La classe `custom.defaults.CCustomUserUrl` n'ajoute pas de lien hypertexte aux utilisateurs (la méthode `getUserUrl()` renvoie `null`).

## Contributions

La classe `custom.examples.CCustomUserUrlMailto` ajoute un lien hypertexte de la forme « `mailto:xxx` ».

La classe `custom.examples.CCustomUserUrlExternal` ajoute un lien hypertexte vers une application (sur le même modèle que la classe `custom.examples.CCustomComputerUrlExternal`).

## Exemple

Le code de la classe `custom.examples.CCustomUserUrlMailto` est le suivant :

```
/**
 * An example of ICustomUserUrl implementation.
 */
public final class CCustomUserUrlMailto implements ICustom, ICustomUserUrl {
    /** constructor. */
    public CCustomUserUrlMailto() {
        super();
    }
    /**
     * @see ICustom#getDescription(java.lang.String)
     */
    public String getDescription(final String language) {
        if (language.equals("fr")) {
            return "lien 'mailto'";
        }
        return "'mailto' link";
    }
    /**
     * @see ICustomUserUrl#getUserUrl(
     * org.hibernate.Session,
     * org.esupportail.portal.channels.CHelpdesk.CUser)
     */
    public String getUserUrl(
        final Session unusedSession,
        final CUser user) {
        String email = user.getEmail();
        if (email == null) {
            return null;
        }
        return new StringBuffer("mailto:").append(email).toString();
    }
}
```

## L'interface ICustomUserInfo

### Utilisation

L'implémentation de cette interface permet d'ajouter automatiquement, à la création d'un ticket, des informations sur le propriétaire du ticket.

Les classes implémentant l'interface `ICustomUserInfo` sont utilisées dans la configuration de l'application. La classe utilisée par l'application est spécifiée par le tag `/custom/user-info`.

## Définition

La définition de l'interface est la suivante :

```
/**
 * Features needed to customize the information added on users
 * when they enter tickets.
 */
public interface ICustomUserInfo {
    /**
     * Return the information corresponding to a user.
     * @param session a Hibernate session
     * @param user the user
     * @return a URL, or null.
     * @throws Exception
     */
    String getUserInfo(Session session, CUser user) throws Exception;
}
```

## Classe incluse dans la distribution

La classe `custom.defaults.CCustomUserInfo` n'ajoute aucune information sur l'utilisateur (la méthode `getUserInfo ()` renvoie `null`).

La classe `custom.userInfo.CXmlConfig` (vue précédemment) indique à l'application d'ajouter automatiquement une action à chaque nouveau ticket, le message de l'action donnant les valeurs d'attributs du propriétaire du ticket. Le nom des attributs est défini dans un fichier de configuration au format XML.

## Exemple

Le code de la classe `custom.defaults.CCustomUserInfo` est le suivant :

```
/**
 * The default ICustomUserInfo implementation.
 */
public final class CCustomUserInfo implements ICustom, ICustomUserInfo {
    /** constructor. */
    public CCustomUserInfo() {
        super();
    }
    /**
     * @see ICustom#getDescription(java.lang.String)
     */
    public String getDescription(final String language) {
        if (language.equals("fr")) {
            return "aucune information";
        }
        return "no information";
    }
    /**
     * @see ICustomUserInfo#getUserInfo(org.hibernate.Session,
     * org.esupportail.portal.channels.CHelpdesk.CUser)
     */
    public String getUserInfo(
        final Session unusedSession,
        final CUser unusedUser) {
        return null;
    }
}
```

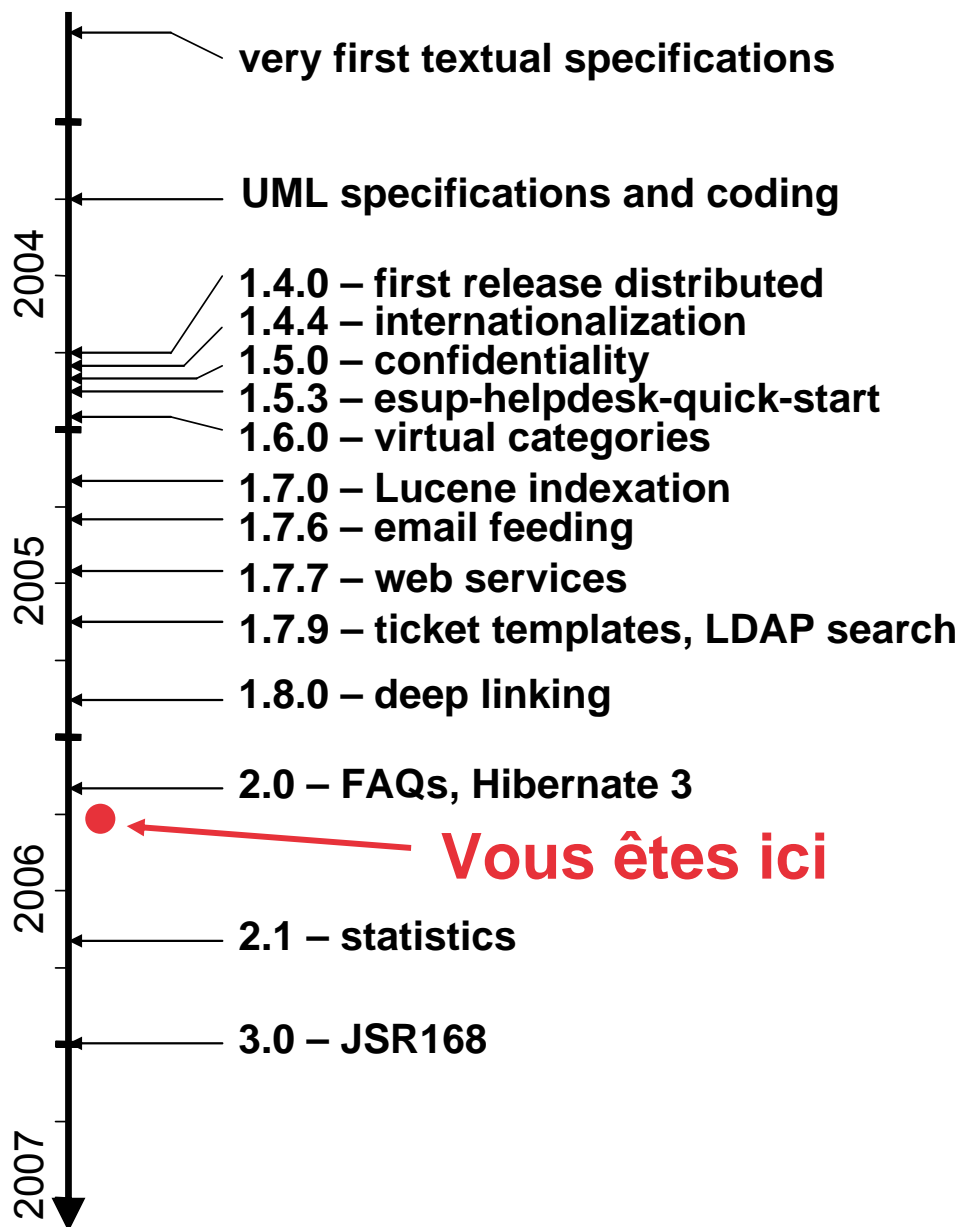


## **5.10. Les autres directives de configuration**

Les autres paramètres de configuration de `CEsupHelpdesk.xml` et `CHelpdeskConfig.xml` non vus jusque là son parcourus de manière exhaustive avec les formateurs.

## 6. Perspectives

### 6.1. Feuille de route (roadmap)



À venir :

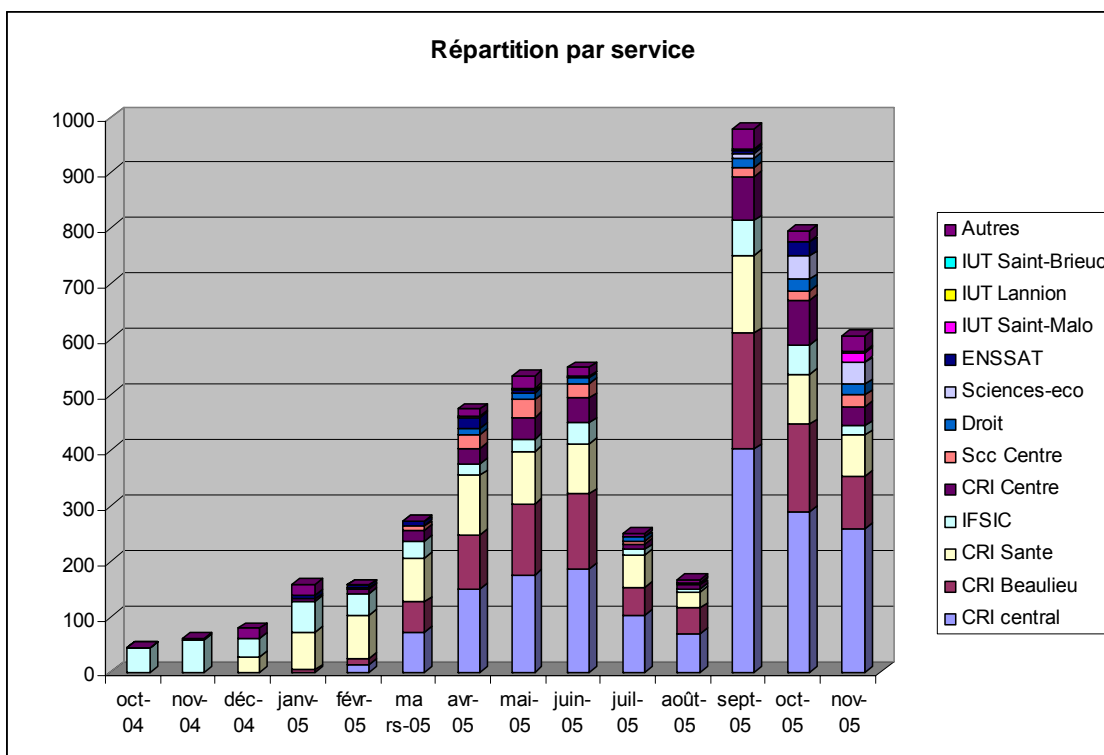
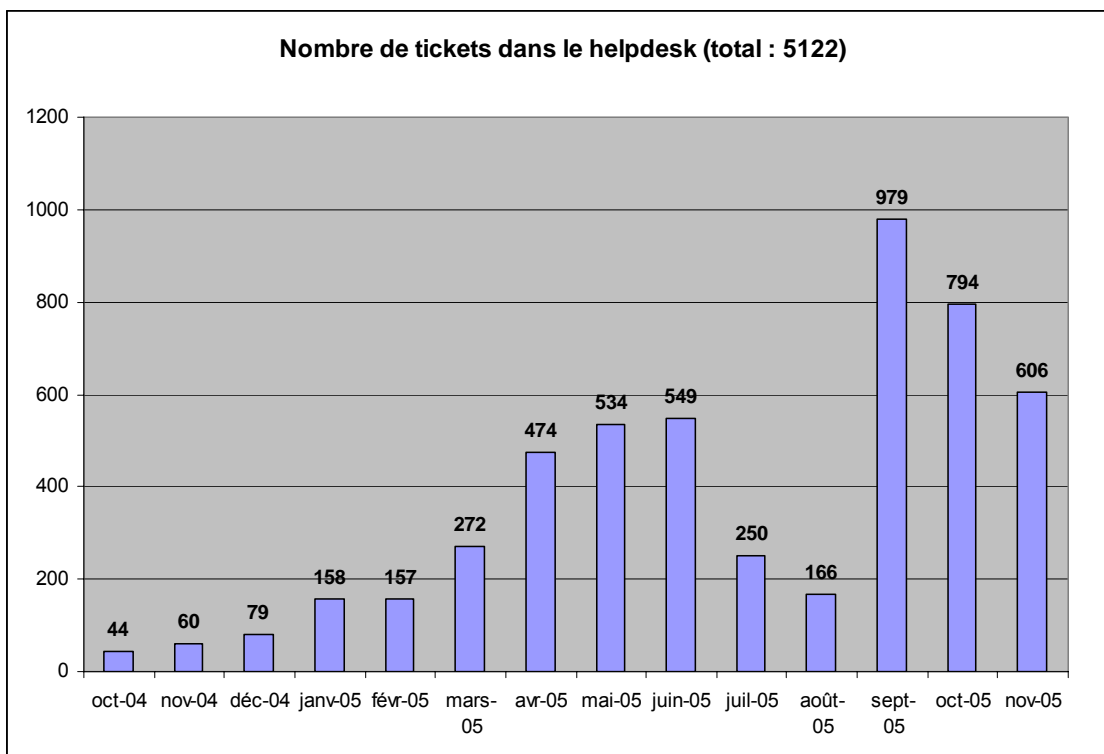
- Ajout d'un générateur de statistiques (été 2006)
- Portage de l'application en portlet (JSR-168)
- Distribution à l'étranger

### 6.2. Les statistiques

Le générateur de statistiques sera ajouté dans l'application au cours de l'été 2006.

## Exemples

Voici ci-dessous quelques statistiques issues de esup-helpdesk à l'Université de Rennes 1.



Les fonctionnalités des statistiques à venir, discutées sur la liste [helpdesk-utilisateurs](#), sont données ci-dessous.

## Mode de calcul

Le calcul de certaines statistiques étant parfois gourmand en CPU, elles ne sont pas compatibles avec un fonctionnement web (calcul sur une requête HTTP et attente du résultat). En conséquence, les statistiques seront déclenchées de manière asynchrone (par cron sous Unix ou tâche programmée sous Windows), comme on le fait déjà par exemple pour l'expiration automatique des tickets. Typiquement, la génération des statistiques serait faite la nuit, à une période de moindre activité du helpdesk.

## Portée des statistiques

La portée des statistiques sera globale (tout le helpdesk) ou par service, éventuellement avec un détail par catégorie.

Les statistiques pourront être calculées par jour, par mois, par an ou depuis le démarrage de l'application.

## Statistiques calculées

Les statistiques suivantes seront générées :

- Nombre de tickets créés
- Nombre de tickets fermés
- Nombre de tickets ouverts
- Durée d'ouverture moyenne (difficile à calculer)
- Nombre de tickets par auteur au-delà d'un certains seuil
- Temps moyen de traitement
- Temps moyen de prise en charge
- Top10 des créateurs de tickets
- Répartition par priorité
- Répartition par origine
- Répartition par jour de la semaine
- Répartition par horaire de la journée

## Format de sortie

Le format de sortie des statistiques n'a pas été tranché.

On sent bien dans ce domaine un besoin double :

- Avoir les statistiques au sein de l'outil, pour ne pas disperser les informations. Nous retenons pour le moment le premier besoin, car c'est le seul qui permettra de remplir un besoin fonctionnel important : présenter des niveaux de visibilité différents selon les publics sur les statistiques produites. On pourrait par exemple désirer que certaines statistiques ne soient accessibles que par les gestionnaires du service duquel elles sont extraites.
- Sortir les données pour pouvoir les retravailler dans un autre outil dédié (typiquement un tableur). Pour ce deuxième besoin, deux possibilités sont offertes :
  - a. générer dans la base de données des tables nouvelles correspondant aux statistiques, tables qui seraient remplies de manière asynchrone. Ces tables seraient alors accédées par les outils externes comme une source de données. Si la solution paraît simple, Il n'est certainement pas souhaitable d'aller chatouiller directement la base de données.
  - b. produire les statistiques sous forme de fichiers, qui pourront ensuite être utilisés librement par les outils externes. Cette solution offre l'avantage de ne pas solliciter directement la base de données (seulement lors de la production des fichiers et non à la demande des outils externes). Il faut cependant se mettre

d'accord sur le format de sortie. Si cette voie était retenue, sans doute faudra-t-il aller chercher un standard de sortie du côté XML, s'il en existe un.

Comme indiqué lors de la présentation de JRES, il n'y aura pas (au moins tant que Alexandre et moi nous occuperons de ce projet) de statistiques individuelles sur les gestionnaires, pour des raisons évoquées également dans la présentation.

Enfin, il est envisageable d'introduire un nouveau privilège concernant les statistiques, qui donnerait à un gestionnaire le droit de lire les statistiques du service pour lequel il est gestionnaire.

### **6.3. *Le passage en portlet/servlet***

- Elimination des conflits de bibliothèques avec d'autres canaux
- Portage possible sur d'autres portails
- Simplification du quick-start (déploiement d'une servlet dans tomcat).

## Historique de ce document

### 20060314 : version 1.0

Version initiale pour la formation des 14 et 15 mars 2006 organisée par L'Université de Rennes 1 dans le cadre des formations du Grand-Ouest.

### 20060316 : version 1.1.0

Modifications suite à la formation des 14 et 15 mars 2006.

- Ajout de sites de déploiement
- Ajout d'exemples et de précisions pour les classes de personnalisation
- Intégration des remarques des participants à la formation
- Déplacement dans les opérations de base de la mise à jour
- Ajout de l'utilisation de la classe `custom.userInfo.userInfo`

### 20060321 : version 1.1.1

Mise à jour pour la version 2.1 de esup-helpdesk :

- Ajout de la propriété `sql.date.column`
- Ajout d'une note sur le changement de `PersonDirs.xml` en `personDirectory.xml`

### 20060612 : version 1.1.2

- Ajustements pour la formation des 13 et 14 juin 2006, organisée par le service de formation continue de l'Université de Rennes 1.

# Notes

# Notes