

Projet AffDist

Accès aux applications distantes depuis un ENT

Philippe Alves

ENT-DSI, Université Pierre et Marie Curie, Paris 6
Case 1906, 4 place Jussieu, 75252 Paris Cedex 05
philippe.alves@upmc.fr

Christian Cousquer

CPM-SGTICE, Université Pierre et Marie Curie, Paris 6
Case 1205, 4 place Jussieu, 75252 Paris Cedex 05
christian.cousquer@upmc.fr

Thierry Koscielniak

ENT-DSI, Université Pierre et Marie Curie, Paris 6
Case 1906, 4 place Jussieu, 75252 Paris Cedex 05
thierry.koscielniak@upmc.fr

Jean-François Lévêque

CPM-SGTICE, Université Pierre et Marie Curie, Paris 6
Case 1205, 4 place Jussieu, 75252 Paris Cedex 05
jean-francois.leveque@upmc.fr

Roland Mergoil

L'UTES-SGTICE, Université Pierre et Marie Curie, Paris 6
Case 1205, 4 place Jussieu, 75252 Paris Cedex 05
roland.mergoil@upmc.fr

Bernard Somphone

ENT-DSI, Université Pierre et Marie Curie, Paris 6
Case 1906, 4 place Jussieu, 75252 Paris Cedex 05
bernard.somphone@upmc.fr

Jean-Marie Thia

SIG-DSI, Université Pierre et Marie Curie, Paris 6
Case 1906, 4 place Jussieu, 75252 Paris Cedex 05
jean-marie.thia@upmc.fr

Résumé

Le projet AffDist de l'UPMC met à disposition des utilisateurs de l'ENT un service innovant d'accès à des applications en déport d'affichage. Le service gère la manière dont les applications sont publiées, les autorisations d'accès pour les utilisateurs, la répartition de charge par serveur applicatif.

Un utilisateur se voit proposer des applications en fonction de son profil et de son appartenance à des groupes. A l'aide d'une interface développée avec un souci de rendre l'utilisation du service la plus ergonomique possible, il peut choisir son application et lire une courte description. À l'aide d'une signalétique simple, il peut voir si l'application est disponible, s'il reste des licences disponibles ou bien si l'utilisation de cette application sera aisée (en fonction de la charge des serveurs qui l'hébergent).

Le service AffDist propose différentes interfaces en fonction des rôles des utilisateurs, et permet un découpage des tâches / responsabilités dans la publication des applications. La personne qui publie une application, ou la réserve pour une durée limitée, ne sera pas obligatoirement la même qui administre le système.

Mots clefs

Portail, Déport d'affichage, NoMachine, NX, RDP, Portlet, JSR168, ESUP, ENT, CEVIF, uPortal, SSO, CAS, SPRING, Hibernate, JAVA, J2EE, UPMC, Université Pierre et Marie Curie, Paris 6

1 Introduction

Le projet AffDist (*Affichage Distant*) a pour objectif de proposer dans un espace numérique de travail de type portail uPortal / ESUP / CEVIF, l'accès à des ressources logicielles en déport d'affichage.

Après s'être authentifié sur le portail, l'utilisateur se voit proposer un certain nombre de vignettes représentant les applications qu'il peut utiliser en fonction de son profil ou de son appartenance à des groupes. Ces vignettes présentent l'application avec une courte description, montrant à l'utilisateur si l'application est utilisable et son niveau de disponibilité.

Il peut alors lancer l'application qui s'ouvre suite à une validation par le SSO et accéder à son espace de stockage pour sauvegarder, modifier des documents.

Ces applications peuvent être de divers types, applications bureautiques ou pédagogiques, ou progiciels spécifiques à certains métiers qui peuvent être hébergés sur des serveurs de tout type (Windows ou UNIX / Linux).

2 Expression des besoins

Deux principaux profils ayant des besoins particuliers ont été identifiés, les utilisateurs « classiques » et les administrateurs / gestionnaires du service.

2.1 Besoins utilisateur

En tant qu'utilisateur final du service, qu'il soit étudiant ou personnel, les principaux besoins exprimés sont :

- Accéder à des applications courantes ou spécifiques à des enseignements.
- Accéder à des applications s'exécutant dans des environnements différents (Windows, Linux, Unix)
- Editer / modifier des documents se trouvant dans son espace de stockage personnel.
Espace qu'il retrouvera au travers du portail, qui sera "monté" lors de l'utilisation d'une application.
- Accéder aux applications depuis le portail.
- Avoir une vision claire et simple des applications disponibles mises à sa disposition.
- Accéder aux applications suite à une validation par le SSO (donc sans réauthentification).

2.2 Besoins administrateurs / gestionnaires

Les administrateurs / gestionnaires ont la charge d'administrer les serveurs, les applications et de gérer l'affectation de droits sur les applications. Leurs besoins sont :

- gérer les applications (mises à disposition, contrôle d'accès, serveurs, publication ...)
- mettre en place des interfaces simples en fonction des rôles ;
- générer des tableaux de bord.

3 Mise en œuvre

La solution développée regroupe différentes parties qui forment l'architecture du service AffDist.

Elle est composée d'une importante partie système qui traite des problématiques d'accès aux applications suite à une validation par le SSO, des techniques de déport d'affichage utilisées, de l'accès au profil utilisateur ou homedir, de l'accès à l'espace de stockage personnel, de la sécurité et de la maintenance du service.

Une deuxième grande partie est matérialisée par les bases de données et les différentes interfaces qui permettent leur accès et qui assurent le cœur de métier du service AffDist.

Enfin la partie la plus visible pour l'utilisateur, l'intégration dans le portail, est une partie importante car c'est elle qui, par le biais d'interfaces ergonomiques, renforce la sensation qu'à l'utilisateur : « Accéder à une application en cliquant sur l'icône, c'est tout ! »

4 Technologies utilisées

Pour l'intégration dans le portail, nous avons choisi de développer des portlets utilisant des frameworks issus des recommandations préconisées par ESUP dans esup-commons [1].

Nous utilisons Spring¹ pour la simplicité de découpage en couche qu'il apporte et pour la facilité d'utiliser des interfaces et implémentations configurées dans des fichiers extérieurs au code.

Nous utilisons Hibernate pour l'accès aux données, JSF et Ajax pour les interfaces utilisateur.

La partie système fait intervenir les technologies NoMachine pour le déport d'affichage, CAS [2] pour le SSO, AXIS pour les WebServices (ws.apache.org/axis), NFS, Samba pour l'accès à l'espace de stockage de

¹ www.spring framework.org

l'utilisateur (enregistrement de ses documents à partir des applications déportées).

Le choix de la technologie NoMachine pour le déport d'affichage a été fait après une étude menée à l'UPMC et décrite ci-après.

5 Déport d'affichage

Nous souhaitons utiliser un système de déport d'affichage offrant une entrée unique vers des applications Windows ou Linux.

Les caractéristiques attendues étaient :

- un affichage fluide quel que soit le débit du réseau ;
- un accès à des ressources locales (imprimantes, dossiers partagés) depuis l'application s'exécutant sur le serveur ;
- un accès aux applications depuis les systèmes des utilisateurs Windows, linux, Mac ;
- une sécurisation des sessions.

Nous avons retenu le protocole NX mis à disposition par NoMachine² parce qu'il semblait être le plus intéressant et le plus performant :

- Son mode de fonctionnement utilisant un proxy permet l'accès à partir d'un point unique aux environnements Linux / Windows.
- Des connexions SSH sont utilisées pour le trafic entre le client et le proxy.
- La relativement faible occupation de la bande passante ainsi que la possibilité de régler le taux de compression (influence la qualité et le dynamisme de l'affichage) utilisé par le protocole permettent une configuration plus fine du contexte de déport d'affichage.
- Le déport des espaces partagés et des imprimantes locales sur le serveur pour un utilisateur donné lui permet d'utiliser une application distante et de sauvegarder ou imprimer sur ses propres ressources (attention aux virus et autres désagréments ...).
- La mise à disposition d'une applet simplifie l'installation du client NoMachine disponible pour les environnements Windows, Linux / Unix, Mac, Solaris.
- La possibilité est donnée de déporter uniquement l'affichage de l'application demandée et non un environnement entier.

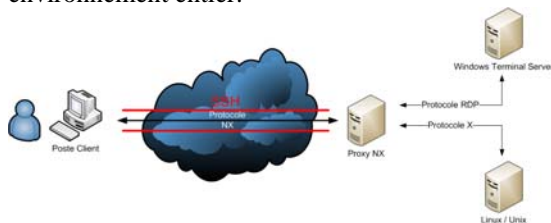


Figure 1 – Architecture NoMachine

La figure 1 nous montre que l'utilisateur utilise une liaison SSH pour se connecter sur le proxy NX. Après connexion, il utilise le protocole NX pour échanger des données avec le proxy NX qui, en fonction de l'application utilisée, retranscrit en protocole RDP ou en protocole X11 vers le serveur hébergeant l'application.

L'utilisateur doit pour cela installer sur son poste un client dit « lourd » puisqu'il s'agit d'une application qui lui permettra d'initialiser la connexion SSH et ensuite le déport d'affichage.

Lorsque l'utilisateur souhaite utiliser une application, une applet est lancée. Elle vérifie que l'application cliente NoMachine est accessible. Si elle ne l'est pas, l'applet propose à l'utilisateur de la télécharger et de l'installer depuis le serveur. Le client NoMachine a un poids raisonnable (environ 5 Mo). Après quoi l'applet lance le client NoMachine avec les paramètres de l'application souhaitée.

L'utilisateur peut alors utiliser l'application déportée.

6 Métier du service AffDist

L'objet du service AffDist est :

- de proposer des applications de façon claire avec des informations en plus des descriptions textuelles qui se font par le biais de signalétiques simples informant l'utilisateur final de la qualité du service ;
- d'apporter du confort à l'utilisateur en lui proposant d'accéder aux applications via le déport d'affichage suite à une validation par le SSO ;
- de gérer les applications mises à disposition et les partager avec leurs licences entre différents services ;
- de réserver des serveurs ou des applications pour des usages spécifiques ;
- de générer des tableaux de bords pour les administrateurs, afin d'établir des statistiques d'usage pour améliorer le service ;
- de proposer une architecture logicielle capable de prendre en compte la répartition de charge par serveur et par application pour faire le meilleur choix de performance pour l'application souhaitée.

² www.nomachine.com

7 Gestion du service AffDist

L'accès aux applications est d'abord obtenu par le biais de la portlet qui expose les vignettes des applications que l'utilisateur peut exécuter. L'accès à l'ensemble des applications est restreint par des filtres, qui contrôlent les jetons de l'utilisateur et son appartenance à des groupes accrédités pour exécuter l'application.

La gestion de ces accréditations et le fonctionnement du service AffDist sont assurés par des administrateurs ayant différents rôles.

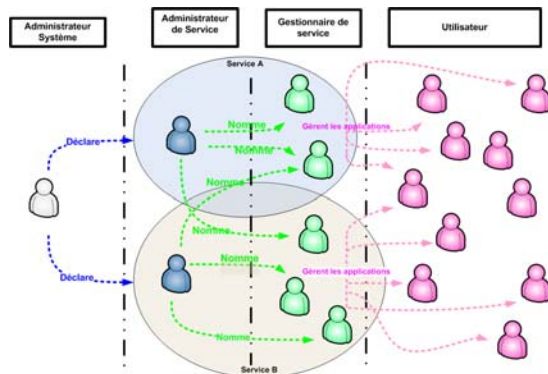


Figure 2 – Organisation des différents rôles

7.1 Administrateur système

- Il est en charge de l'infrastructure permettant le déport d'affichage
- Il crée les services et il inscrit les administrateurs de Service³
- Il assure la sécurité et la maintenance de l'architecture.

7.2 Administrateur de Service

- Il est l'administrateur système des serveurs mis à disposition par son Service.
- Il met à disposition des applications pour son Service (il installe les applications sur les serveurs et il les indexe dans l'application AffDist).
- Il publie les applications de son Service en effectuant un contrôle d'accès au niveau population / groupe.
- Il est responsable des licences logicielles.
- Il inscrit les gestionnaires de son Service.
- Il assure la sécurité et la maintenance de ses plates-formes.

7.3 Gestionnaire de Service

- Il est responsable de la publication (contrôle d'accès au niveau temporel) des applications de son Service.

- Il peut « réserver » des licences / jetons pour des applications mises à disposition par son Service.

Une attention très particulière a été portée sur l'ergonomie des interfaces, ainsi l'utilisateur classique peut rapidement voir si une application est disponible selon différents critères :

- une licence ou un jeton encore disponible ;
- la charge des serveurs l'hébergeant en fonction notamment du nombre d'utilisateurs connectés.

8 Conclusion

L'intégration d'AffDist dans un portail permet aux utilisateurs d'accéder aux ressources logicielles de leur établissement avec la possibilité de n'exécuter que l'application souhaitée, intégrée de façon transparente dans l'ENT. Ces ressources sont partagées et contrôlées par différents administrateurs ayant des rôles particuliers.

Une proof of concept a été réalisée début 2006 et présentée à la communauté⁴ [3]. Une maquette fonctionne actuellement au sein de l'UPMC [4].

Bibliographie

- [1] Pascal Aubry, esup-commons : framework de développement ESUP. ESUP Days 3, Paris, 26 janvier 2007.
- [2] Vincent Mathieu, Pascal Aubry, Julien Marchal, Single sign-on opensource avec CAS (Central authentication service). Dans *Actes du congrès JRES2003*, pages 747-760, Lille, Novembre 2003.
- [3] Thierry Koscielniak, Les services pédagogiques dans l'ENT de l'UPMC - Paris 6 : ENF (CEVIF). *Communication orale* : ESUP-Days N°2, Paris, Juin 2006.
- [4] Philippe Alves, Exemples de nouveaux services dans l'ENT de l'Université Pierre et Marie Curie - Affichage d'application à distance, *Communication orale* : ESUP-Days N°4, Paris, Juillet 2007.

⁴ www.esup-portail.org