Univ-RX (pédagogie dans les ENT)

Hervé JAUME (Responsable du projet)
Université Louis Pasteur Strasbourg 1 (service ULPMultimédia)
Herve.jaume@ulpmm.u-strasbg.fr

Marc Chantreux
Université Louis Pasteur Strasbourg 1 (service ULPMultimédia)
Marc.Chantreux@ulpmm.u-strasbg.fr

Matthias Meusburger
Université Louis Pasteur Strasbourg 1 (service ULPMultimédia)
Matthias.Meusburger@ulpmm.u-strasbg.fr

Résumé

Univ-RX est une des briques applicatives pédagogique proposée au sein du projet ENT EPPUN (http://eppun.u-strasbg.fr), déjà déployé dans sa version Windows (http://univ-r.u-strasbg.fr) depuis la rentrée 2002 auprès des trois Universités de Strasbourg, ainsi que l'ENA depuis décembre 2004. Elle sera déployée à la rentrée 2005 dans sa version Linux (Univ-RX) pour les 45000 utilisateurs strasbourgeois. Univ-RX propose aux acteurs pédagogiques d'une université différentes fonctionnalités informatiques au sein d'un même outil communément appelé 'Bureau Virtuel'. Il donne accès à un ensemble de logiciels pédagogiques profilés, des espaces d'échange de documents, des outils de communication (synchrone et asynchrone) ainsi qu'à des outils de création et de diffusion de supports pédagogiques (cours, TP etc.).

Univ-RX est un outil fondé sur les technologies « Client Légers » et est ainsi accessible soit en salle de ressource sur le campus, soit par internet sans pénaliser les machines d'ancienne génération.

Grâce à sa compatibilité CAS, Univ-RX s'intègre dans une mécanique SSO et peut donc être accessible de façon transparente dans un portail ENT d'établissement.

Mots clefs

Pédagogie, Linux, X11, ENT, CAS, JSP, Java, Applet, SSO, LDAP, PAM, NSS, KDE, Firefox.

1 Introduction et contexte

Depuis 1999, l'Université Louis Pasteur a chargé le service ULPMultimédia d'étudier des solutions de rationalisation des ressources informatiques de l'établissement, mais aussi l'optimisation de l'accès aux contenus pédagogiques numériques des composantes de l'établissement. Univ-RX est la dernière évolution du projet BureauNomade lancé à cette période.

2 Contours fonctionnels

2.1 Logithèque profilée

Un ensemble de logiciels adaptés aux différents cursus est accessible via la plateforme à l'ensemble des utilisateurs pour lesquels il a été déployé. Ce profilage permet d'accorder des accès contrôlés aux progiciels nécessaires. Dans le cadre d'Univ-RX, ces logiciels sont tous des logiciels libres sous licence GPL, sauf certains logiciels commerciaux pour lesquels le profilage assure un contrôle des licences.

2.2 Dépôts de documents

Des espaces de dépôt de documents dédiés permettent un travail collaboratif autour de ces fichiers, tels que de la gestion de versions, des commentaires, des forums, etc. De plus, un espace de stockage personnel est proposé à chaque utilisateur. Grâce à la logithèque intégrée, les documents échangés sont de toutes natures. De fait, pour exploiter ces documents, l'utilisateur n'a plus besoin d'installer de logiciels sur son poste.

2.3 Travail collaboratif

Outre les espaces d'échange de documents, des espaces groupe donnent accès à des outils de communication : chat, forum, mail, tableaux d'affichage virtuels. La notion de groupe de travail est directement issue des structures pédagogiques des UFR, et est donc « calquée » sur ces mêmes structures (code étape d'Apogée par exemple).

2.4 Édition et diffusion de cours

Un outil appelé « conducteur de cours » permet aux enseignants de concevoir un squelette de leur cours, par des points de cours essentiels, en y attachant toutes sortes de ressources (n'importe quel fichier pouvant être lu par l'un des logiciels de la logithèque proposée). L'enseignant peut gérer l'accès à tous ou certains points du cours, à l'ensemble des groupes dans lesquels il est inscrit.

3 Technologies mises en œuvre

Univ-RX est un outil fondé sur les technologies « Clients Légers » et est ainsi accessible soit en salle de ressources soit par Internet, sans pénaliser les machines d'ancienne génération.

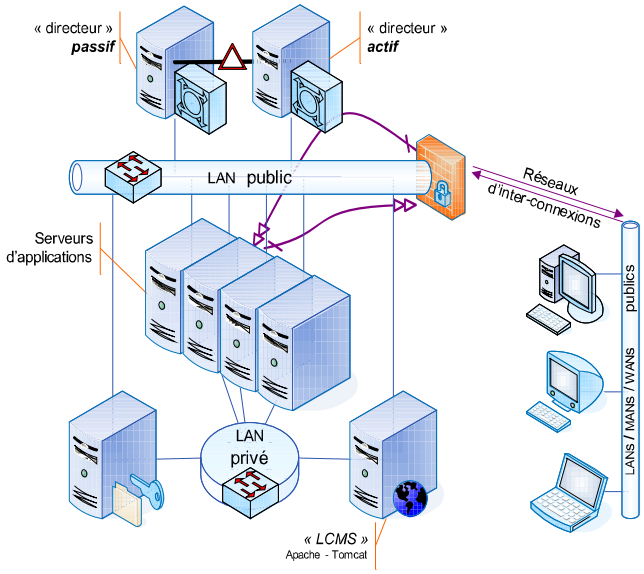

Figure 1 - Infrastructue du projet Univ-RX

3.1 Protocole de connexion (Internet et SRI[1])

La version de l'outil sous Linux est rendue possible par la percée des systèmes de compression du protocole X11 tel que le protocole NX de NoMachine qui a été retenu pour le projet. Il rend possible l'accès distant via Internet à l'ensemble des fonctionnalités d'un bureau virtuel : applications bureautiques (ex : Openoffice), logiciels scientifiques et pédagogiques (ex : Maple), stockage de documents.

Les transferts se font au sein d'un tunnel SSH, augmentant ainsi la confidentialité des données échangées. Les deux composants de compression utilisés sont Nxagent (côté serveur) et Nxproxy (côté client). Ces deux composants sont chargés de la compression / décompression X11 sur des ports TCP établis dynamiquement.

[1] Salle de Ressources Informatiques

Une applet Java réalise la connexion via Internet en utilisant le composant « Jsch » (bibliothèque SSH en Java).

L'agent de compression ainsi que divers outils dépendants du système d'exploitation du client (serveur X, démon de son, etc.) sont téléchargés si nécessaire et exécutés par l'applet (l'applet étant signée, elle est autorisée à lancer des binaires natifs sur le poste client).

La session graphique ainsi établie permet de travailler confortablement à partir d'une connexion ADSL 128 Kb.

Pour plus de détails sur l'applet Java, vous pouvez vous reporter à l'article : *« NOMADISME ET COMPRESSION X11 »* présenté aux JRES2005.

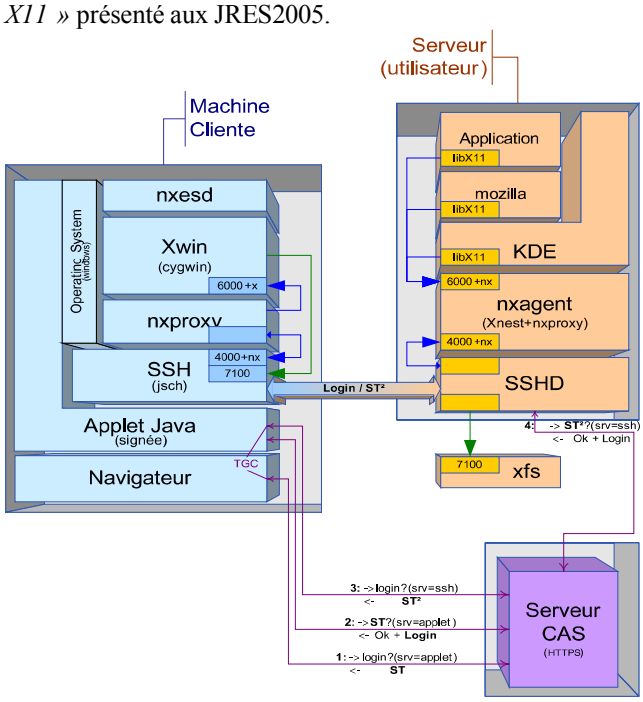

Figure 2 – Schéma de fonctionnement de l'applet avec authentification sur un serveur CAS

En SRI l'accès est confié aux clients natifs de chez NoMachine. À cet effet une version épurée de la distribution Knoppix a été utilisée, en grande partie pour son adaptabilité à l'hétérogénéité du matériel. La machine ainsi configurée offre un accès de type kiosque à la plateforme. De plus, la maintenance de ce parc est facilitée : mise à jour du système client à distance, possibilité de démarrer à partir d'un CD, d'une clé USB ou même directement depuis un système de fichiers Windows existant (le système complet se présente sous la forme d'un fichier amorçable).

3.2 Environnement système des serveurs d'applications

Les serveurs d'applications Univ-RX constituent une ferme dont les noeuds partagent :

- une base et des méthodes communes d'authentification et d'identification des utilisateurs ;

- un serveur commun dédié pour les espaces de stockage personnels ;
- une répartition de la charge.

3.3 Une base d'authentification et d'identification commune

Les informations relatives aux utilisateurs et aux groupes système sont stockées par défaut sous la forme de fichiers à plat présents sur un périphérique de stockage local. Ce qui ne permet pas, à priori, le partage entre les machines d'une base commune d'authentification.

Toutefois, les Unix modernes sont capables de tirer parti de plusieurs sources de résolution de noms (NSS pour Name Service Switch) et de méthodes d'authentification (PAM pour Pluggable Authentification Module).

Ces deux mécanismes, décrits ci-après, sont modulaires. Ainsi, les développeurs de modules ont la possibilité d'interroger n'importe quelle source d'information disponibles localement ou sur le réseau (fichier à plat, base de données, annuaire, etc.).

De nombreux éditeurs (Microsoft, Sun, Novell, etc.) ont fait le choix de confier le stockage des informations à un annuaire LDAP. On peut citer comme raisons de ce choix :

- les faibles temps de réponse sur les interrogations ;
- la hiérarchisation des données ;

D'autre part les recommandations du ministère, par le SDET (Schéma Directeur des Environnement de Travail), nous orientent vers des annuaires LDAP au format SUPPAN pour la gestion des Systèmes d'Information contenant les données personnelles des utilisateurs des ENT.

Nous avons donc choisi d'asseoir la base commune pour l'authentification et l'identification sur un annuaire LDAP. Les modules NSS et PAM des serveurs de la ferme étant des clients LDAP.

De plus, les modules NSS/PAM permettant d'interroger un annuaire LDAP et les schémas LDAP permettant de stocker les informations nécessaires sont existants et maintenus par une large communauté.

NSS : Name Service Switch

Les systèmes d'exploitation de la famille Unix fournissent des appels système pour l'énumération, la recherche, la résolution de noms concernant les utilisateurs, les groupes d'utilisateurs, les machines, les protocoles et services IP, les alias RFC822.
Par exemple, l'appel système getpwuid() permet de résoudre l'identifiant d'un utilisateur à partir de son uid. Cette résolution est effectuée par un service de nommage (« nameservice »).

Par défaut, ce nameservice cherche ces informations dans des fichiers à plat stockés sur le système de fichiers local, mais les limitations de cette première approche (notamment la difficulté de maintenir des noms cohérents entre toutes les machines d'un réseau) ont amené les architectes des systèmes d'exploitation à concevoir un mécanisme modulaire et configurable. Les Unix modernes permettent donc d'utiliser différents services pour la résolution d'un même nommage sur une même machine.

NSS est un mécanisme fourni par la libc permettant :

- à l'administrateur de spécifier les sources d'information à utiliser et la priorité avec laquelle elles seront utilisées ;
- au programmeur de pouvoir développer un nouveau service sans connaître les mécanismes internes de NSS.

```
jd:x:1231:1500:John
Doe,,,jd@ulp.fr:/localhome/jd:/usr/bi
n/zsh
```

Figure 4 : Exemple de source d'information en fichier à plat (/etc/passwd)

Les entrées du fichier de configuration du service de nom (/etc/nsswitch.conf) spécifient pour chaque type de service les différentes sources de données à interroger dans le bon ordre (voir figure 6).

```
passwd:         files ldap
group:          files ldap
shadow:         files ldap
hosts:          files dns
networks:       files
protocols:      db files
services:       db files
ethers:         db files
rpc:            db files
```

Figure 6: Exemple de configuration NSS complète

Dans l'exemple de la figure 6, la résolution de noms de machine se fera dans le fichier local (/etc/hosts) puis elle interrogera le serveur DNS si nécéssaire. De même les noms de protocoles (services:) sont renseignés dans un fichier BerkeleyDB.

Pour l'accès aux informations relatives aux utilisateurs et aux groupes, NSS commence par le fichier à plat puis s'adresse au serveur LDAP.

Le fichier à plat est utilisé de préférence pour tous les comptes système et administrateurs afin de pouvoir intervenir en cas de défaillance (de l'annuaire ou du réseau).

Le serveur LDAP doit respecter le schéma « NIS[2] » notamment les classes PosixAccount et PosixGroup (respectivement utilisateur et groupe).

L'utilisateur « jd » (voir figure 4) devient dans notre annuaire l'entrée de la figure 7.

```
dn:
uid=jd,ou=staff,dc=ulpmm,dc=univ,dc=rx
sn: John
uidNumber: 1231
gidNumber: 1500
```

[2] Network Information Service

```
objectClass: top
objectClass: person
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
cn: John Doe
mail:
gecos: John Doe,,,jd@ulp.fr
homeDirectory: /localhome/jd
loginShell: /usr/bin/zsh
shadowLastChange: 12962
userPassword:: YWZvZX32FVZZE4VcnA=
uid: mc
```

Figure 7: Objet LDAP uid=jd

Cette entrée LDAP contient une version encryptée du mot de passe de l'utilisateur ainsi que diverses informations exploitables par PAM (comme le répertoire personnel de l'utilisateur).

Pour les entrées de groupes, les informations pertinentes sont le nom du groupe, son numéro et les membres (voir figure 9).

```
dn:
cn=perdus,ou=groupes,dc=univ,dc=rx
cn: perdus
objectClass: top
objectClass: posixGroup
gidNumber: 1500
memberUid: jd
memberUid: alain.colas
memberUid: illusions
```

Figure 9: objet LDAP posixGroup

Dans notre optique de centralisation du système d'information, nous configurons le module LDAP, avec les paramètres de notre serveur *central* LDAP (figure 10).

```
host serveur-ldap.ulp.fr
base dc=univ,dc=rx
nss_base_passwd
ou=staff,dc=ulpmm,dc=univ,dc=rx
nss_base_group
ou=staff,dc=ulpmm,dc=univ,dc=rx
ldap_version 3
```

Figure 10: configuration module NSS LDAP

3.4 PAM : Pluggable Authentication Module

PAM permet à l'administrateur de spécifier une pile de modules permettant de gérer :

- l'authentification ;
- l'ouverture de session ;

Nous exploitons trois sources d'authentification (notamment le serveur SSH) :

1. **module pam_cas.so**
   Dans le cadre de l'ouverture d'un bureau Univ-R^x depuis un portail compatible CAS, le mot de passe est remplacé par un ticket de service qu'il faut vérifier auprès du serveur CAS.
   (cf. "Nomadisme et compression X11", JRES 2005).
   En cas de validation réussie, l'option "sufficient" indique sans parcourir le reste de la pile la validité de l'authentification.
2. **module pam_ldap.so**
   Il tente une connexion authentifiée sur l'annuaire *central* LDAP en utilisant le mot de passe et l'entrée correspondant au login. Si la connexion (bind) est acceptée, PAM accepte l'utilisateur ("sufficient" de la même manière).
   Cette authentification est utilisée dans SSH par tous les clients NoMachine.
3. **module pam_unix.so**
   Utilise les fichiers traditionnels à plat. L'option "required" signifie qu"échec de cette méthode provoque un échec de l'authentification car c'est la dernière à être tentée.

```
auth    sufficient      pam_cas.so
-sssh -f/etc/pam_cas.conf
auth    sufficient      pam_ldap.so
use_first_pass debug
auth    required        pam_unix.so
use_first_pass nullok_secure
```

Figure 11: /etc/pam.d/ssh

L'utilisation de l'argument use_first_pass précise au module de tenter d'abord avec le mot de passe déjà saisi avant de redemander un nouvel essai à l'ultilisateur.

3.5 Centralisation des espaces de stockage avec NFS

Les serveurs d'applications de la ferme Univ-Rx disposent d'un espace commun (via NFS) pour le stockage des données des utilisateurs. Les droits sont respectés car les identifiants numériques des utilisateurs (uidNumbers) sont communs sur tous les serveurs de la ferme grâce à l'annuaire *central*.

Les serveurs d'applications accèdent au serveur de stockage par un réseau privé à haut débit physiquement séparé du réseau public.

3.6 Répartition de charge

La répartition de charge est effectuée par IPVS (Linux Virtual Server) appelé aussi commutateur de niveau IV (couche transport) ; cette fonctionnalité est présente dans les noyaux Linux. Tous les serveurs de la ferme sont vus comme une seule et même machine virtuelle (comme une seule adresse IP). Dans ce système des machines appelées « directeur » répartissent les connexions sur les nœuds de la ferme.

Le directeur est la machine qui répond aux requêtes envoyées par les clients à l'adresse IP virtuelle et retransmet la totalité du trafic de la session à un des nœuds (suivant sa disponibilité). L'adresse MAC du nœud de la ferme est utilisée, laissant l'adresse IP virtuelle inchangée. En résumé (figure 12) :

1. Le client établit une connexion avec le directeur ;
2. le directeur transfert la connexion à un des nœuds ;
3. le nœud répond directement au client en retour (sans passer par le directeur) ;
4. La connexion établie (SYN+ACK), le reste de la connexion continue suivant le même schéma, le directeur doit maintenir une table des connexions actives.

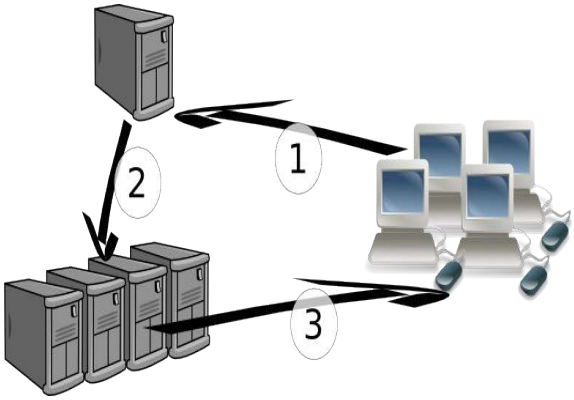

Figure 12: IPVS, fonctionnement général

Considérons la configuration d'un directeur (10.0.10.1) et de deux nœuds (10.0.100.1 et 10.0.100.2). Tous utilisent l'interface `eth0`. L'ensemble constitue la machine virtuelle 10.0.0.1.

Sur les nœuds on configure l'adresse IP réelle sur l'interface `eth0`, puis l'adresse IP virtuelle comme un alias sur l'interface loopback `lo` appelée `lo:ipvs`. On peut faire en sorte que l'interface loopback reste muette au moment des annonces ou requêtes ARP afin de ne pas voler l'adresse IP virtuelle au directeur (ceci est réalisé en modifiant les paramètres du noyau via '/proc').

Au niveau du directeur, l'adresse virtuelle est une adresse secondaire traditionnelle (alias `eth0:ipvs`). Ainsi l'équipement réseau enverra naturellement les trames sur cet alias et donc sur `eth0`.

Keeepalived est une sur-couche à IPVS qui permet :

- la création d'un directeur de secours (*passif*) avec synchronisation des tables de connexions vers les nœuds (les connexions actives seront maintenues en cas de défaillance du directeur *actif*). Le protocole VRRP est utilisé ;
- une répartition efficace des nouvelles connexions par une pondération des nœuds, en tenant compte des nœuds disponibles et de leurs charges.

Voici une configuration commentée d'un directeur.

Première partie : paramètres globaux

```
global_defs {
   # personne a contacter en cas de
problème
   notification_email {
     admin@u-strasbg.fr
   }

   # email utilisé par le client SMTP
pour envoyer des messages
   notification_email_from
lvs_directeur1@u-strasbg.fr

   # paramètres du serveur smtp à
utiliser pour l'envoi de      #courrier
   smtp_server mailhost.u-strasbg.fr
   smtp_connect_timeout 30

      # nom du dircteur
   router_id directeur1
}
```

Deuxième partie : création d'une instance

```
vrrp_instance UnivRX {

      # le présent serveur est le
directeur principal
      # ( BACKUP pour spécifier un
esclave )
    state MASTER

      # keepalived écoute sur eth0
    interface eth0

      # son poids pour ce service par
rapport aux autres directeurs       #
est de 100, ce doit etre un poids plus
fort que les poids
      # de tous les autres esclaves pour
ce service
    virtual_router_id 100
    priority 10
    advert_int 1

    lvs_sync_daemon_interface eth0
      # chaine permettant
l'authentification des directeurs entre
```

```
      # eux. Cette authentification est
importante pour éviter une
      # usurpation de l'identité d'un
des serveurs
    authentication {
        auth_type PASS
        auth_pass AZEZAEZAAZREZREZRE
    }
      # adresse IP virtuelle utilisée
par tous les clients
    virtual_ipaddress {
        10.0.0.1
    }
}
```

Troisième partie : configuration des serveurs de la ferme

```
# définitions des serveurs pour les
connexions ssh sur l'adresse
# virtuelle
virtual_server 10.0.0.1 22 {
    delay_loop 6
      # l'algorithme de répartition est
fonction du nombre de   #connexions mais
aussi du poids du serveur
    lb_algo wlc
    lb_kind DR
    protocol TCP

      # Les serveurs de la ferme
      # leur poids par défaut est 1 pour
le directeur et 0 pour le
      # directeur de secours
      # si le heartbeat du serveur est
perdu,inhibit_on_failure      #permet à
keepalived de pondérer lui-même le
serveur à 0
    real_server  10.0.100.1 22 {
        weight 1
        inhibit_on_failure
        TCP_CHECK {
                connect_timeout 10
        }
    real_server  10.0.100.2 22 {
        weight 1
        inhibit_on_failure
        TCP_CHECK {
                connect_timeout 10
        }

    }
```

3.7 Gestion de l'Interface Utilisateur

La gestion de l'interface utilisateur est fondée sur deux technologies principales : KDE[3] et Firefox. Le premier est un environnement de bureau libre. Le second est un navigateur web moderne, libre, respectueux des standards, configurable et extensible.

KDE prend en charge les éléments systèmes de l'environnement utilisateur : gestion des sessions, gestion des fenêtres, affichage du système de fichiers, etc.

Firefox constitue quant à lui la majeure partie de l'interface visible pour l'utilisateur. En effet, il se substitue au « bureau » KDE (mode kiosque) pour présenter les différentes applications, les données pédagogiques, etc.

3.8 Configuration de KDE

Pour répondre aux besoins d'Univ-RX, KDE est adapté d'une part via son système de configuration « natif » et d'autre part grâce au « Kiosk Framework », qui permet de brider l'environnement de travail.

Les fichiers de configuration de KDE suivent un schéma classique de liste de couples « clé-valeur » organisés en diverses sections.

La souplesse du système de configuration permet d'affecter finement les droits des utilisateurs. En effet, un grand nombre de réglages sont paramétrables : raccourcis clavier, options des menus des applications, possibilité pour l'utilisateur d'accéder à la configuration des applications, etc.

```
[General]
font=Bitstream Vera Sans,10,-
1,5,50,0,0,0,0,0
widgetStyle=plastik
```

Figure 16 : Fichier de configuration de KDE

De plus, l'organisation hiérarchique de la configuration (les préférences peuvent être définies au niveau système, puis pour chaque utilisateur) permet de mettre en place des

[3] KDE Desktop Environment

paramètres par défaut, que les utilisateurs pourront ou non modifier. En effet, les options de configuration peuvent être marquées comme « immuables ». Celles qui le sont au niveau système sont donc verrouillées pour tous les utilisateurs.

```
run_command[$i]=false
custom_config[$i]=false
lock_screen[$i]=false
shell_access[$i]=false
run_desktop_files[$i]=false
```

Figure 17 : options immuables ([$i])

Finalement, il est possible d'utiliser le résultat de commandes dans des options de configuration. Ce mécanisme permet par exemple de pré-configurer automatiquement un client de messagerie en récupérant des informations comme le nom d'utilisateur, l'adresse de courriel ou le nom complet auprès du système d'information.

```
EmailAddress[$e]=$(getent passwd
$USER | cut -d: -f5 | cut -d, -f5)
FullName[$e]=$(getent passwd $USER |
cut -d: -f5 | cut -d, -f1)
Inline Signature[$e]=$(getent passwd
$USER | cut -d: -f5 | cut -d, -f1)
\nULPMultimédia\n
```

Figure 18 : pré-configuration du client de messagerie à l'aide d'appels système ([$e])

La configuration porte sur différents modules de KDE. Ainsi Kwin, le gestionnaire de fenêtres, est configuré pour ne proposer qu'un seul bureau virtuel (au lieu de quatre par défaut). De même, Ksmserver, le gestionnaire de sessions, est configuré pour démarrer avec une session vide (afin de ne pas restaurer les applications lancées lors de la session précédente). D'autre part, le lancement de certains composants de KDE non-utilisés est inhibé, comme Kicker (tableau de bord, barre des tâches).

D'autre part, les fonctions de configuration portant sur les restrictions d'URL nous permettent de limiter le champ d'action de Konqueror, le gestionnaire de fichier utilisé (mais également des autres applications KDE). De cette manière, les utilisateurs ne peuvent pas sortir de leur répertoire personnel pour parcourir l'arborescence du système de fichiers. Ils ne peuvent pas non plus utiliser le gestionnaire de fichiers pour aller sur Internet, ni pour se connecter à des serveurs distants. Il est également possible d'autoriser uniquement certains serveurs pour certains protocoles.

Outre la configuration, certains outils disponibles dans KDE sont sollicités pour accomplir des tâches nécessaires au fonctionnement d'Univ-RX. Ainsi, DCOP[4] (système de communication inter-applications de KDE) permet de provoquer la déconnexion de l'utilisateur à la demande. Kstart, programme permettant de lancer des applications avec des propriétés de fenêtre spéciales, nous permet de lancer Firefox (qui doit afficher l'interface à l'utilisateur) en plein-écran, sans bordure et toujours à l'arrière-plan.

3.9 Configuration de Firefox

Pour correspondre à un fonctionnement en mode « Kiosque », Firefox doit être partiellement bridé. En effet, il ne doit pas être possible de quitter l'interface que celui-ci présente avant la fin de la session. De même, il ne doit pas être possible de « sortir » de l'interface pour surfer sur le web. À l'inverse, il est nécessaire d'étendre les fonctionnalités de base du navigateur pour qu'il puisse communiquer avec le système, pour commander le lancement d'applications par exemple. Pour ce faire, des protocoles spéciaux Univ-RX sont créées, et il est indiqué à Firefox que c'est un script shell qui prendra en charge l'exécution d'URLS de ces protocoles.
Firefox est modifié à différents niveaux. D'une part, des extensions existantes sont utilisées :

- « Autohide » permet d'afficher le contenu du navigateur en plein-écran : les barres d'outils et la barre d'état sont supprimées ;
- « Titlebar tweaks » permet de supprimer le nom du navigateur dans les barres de titre des fenêtres filles de l'interface.

[4] Desktop COmmunication Protocol

```
# Restrictions au niveau des URL
[KDE URL Restrictions][$i]
rule_count=6

# On ne peut ouvrir ni lister aucune ressource d'aucun protocole
rule_1=open,,,,,,,false
rule_2=open,,,,,,,false

# Sauf les fichiers situés dans $HOME...
rule_3=list,,,,file,,$HOME,true
rule_4=open,,,,file,,$HOME,true

# et les partages Samba de l'université
rule_5=open,,,,smb,*.u-strasbg.fr,,true
rule_6=list,,,,smb,*.u-strasbg.fr,,true
```

Figure 19 : restriction d'URL au sein de la configuration de KDE

D'autre part, la configuration de Firefox est modifiée et figée de manière à pouvoir :

- désactiver le cache du navigateur ;
- désactiver la mémorisation des mots de passe ;
- affecter la prise en charge des protocoles spéciaux Univ-RX au script qui doit les gérer ;
- désactiver les avertissements de sécurité sur :
  - l'acceptation des cookies,
  - l'entrée dans une zone sécurisée,
  - l'entrée dans une zone non sécurisée,
  - la sortie d'une zone sécurisée,
  - la soumission de formulaire dans une zone non sécurisée,
  - le mélange d'éléments sécurisés et non sécurisés au sein d'une page ;

Finalement, au plus bas niveau, Firefox est directement modifié pour permettre l'inhibition :

- de tous les raccourcis clavier ;
- du menu contextuel (bouton droit de la souris) ;
- des onglets (bouton du milieu) ;
- du glisser-déposer ;

```
function contentAreaClick(event, fieldNormalClicks) {
[...]
/*
   if (event.button == 1 &&
       !findParentNode (event.originalTarget, "scrollbar") &&
       gPrefService.getBoolPref ("middlemouse.contentLoadURL")) {
     middleMousePaste(event);
   }
   return true;
*/
}
```

Figure 21 : Inhibition du clic avec bouton du milieu dans un fichier XUL (Javascript) de Firefox par mise en commentaire de code.

```
// univ-x
lockPref("network.protocol-handler.external.univ-x", true);
lockPref("network.protocol-handler.app.univ-x", "/usr/local/univ-x/run.sh");
lockPref("network.protocol-handler.warn-external.univ-x", false);
// logout
lockPref("network.protocol-handler.external.logout", true);
lockPref("network.protocol-handler.app.logout", "/usr/local/univ-x/run.sh");
lockPref("network.protocol-handler.warn-external.logout", false);
```

Figure 20 : Définition de nouveaux protocoles dans la configuration de Firefox

L'interface utilisateur de Firefox est réalisée en XUL[5], technologie utilisant notamment XML pour la description de l'organisation de l'interface et Javascript pour la gestion des actions sur l'interface. Ainsi, il n'y a pas eu besoin de recompiler Firefox, la modification de fichiers textes étant suffisante.

3.10 Interface de travail : EEV (Établissement d'enseignement Virtuel)

L' interface de travail (délivrée par le navigateur Firefox) est en fait une application WEB de travail collaboratif J2EE développée en JSP par ULPMultimédia. Elle est hébergée sur un serveur Apache-Tomcat s'appuyant sur un serveur SQL PostgreSQL. Les informations utilisateurs et groupes de travail (calquées sur les versions d'étapes d'Apogee) nécessaires au travail collaboratif sont obtenues par traitement des données administratives des étudiants et personnels de l'université. Ces données sont contenues dans l'annuaire d'établissement respectant les recommandations SUPPAN.

Figure 22 – Exemple d'interface de travail : la plateforme Univ-RX de l'Université Louis Pasteur de Strasbourg, un goupe de travail et ses fonctionnalités et un exemple de logiciels accessibles.

3.11 Accès et sécurité

Grâce à sa compatibilité CAS, Univ-RX s'intègre dans une mécanique SSO et peut donc être accessible de façon transparente dans un portail d'ENT d'établissement.

Les machines clientes se connectant sur Univ-RX sont volontairement séparées en deux catégories, car les contraintes sont différentes.

Les machines sur les campus dans les salles de ressources en accès libres :

Cet accès libre est rendu possible à grande échelle et à moindre coût par la transformation du parc de PC obsolètes de l'université en clients légers (voir 3.1).

Les machines personnelles ou nomades des utilisateurs connectés via Internet :

Cet accès est le plus complexe car les paramètres du poste client ne sont pas maîtrisables. La connexion Web utilisée doit être de type ADSL 128Kb minimum pour donner satisfaction. Elle s'effectue via l'applet Java brièvement décrite au paragraphe 3.1 (voir pour plus d'informations l'article JRES2005 *« NOMADISME ET COMPRESSION X11 »)*.

Le développement de la plateforme Univ-RX est le fruit de 12 mois de développements. Une équipe de 5 personnes a participé au projet. Ce projet est en cours de déploiement sur les établissements d'enseignement supérieur de Strasbourg. Forts de notre expérience d'utilisation de la version Windows (actuellement plus de 20 000 comptes actifs), nous attendons de la version Linux plus de rationalisation, de visibilité et de fiabilité quant à la gestion des ressources informatiques de l'université.

[5] XML User Interface Language