

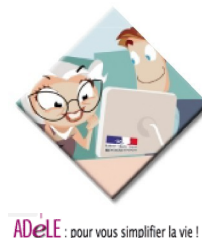
## Rapport d'analyse sur le contournement des Applets de CybEO

### Objet :

Ce document a pour but de dresser l'état des lieux actuel sur les applets utilisées dans CybEO et de présenter les méthodes envisagées pour le contournement.



# CybEO



## 1 - Gestion du document

	Nom	Société	Date	Signature
Rédigé par		Process-one		
Vérifié par				
Validé par				
Approuvé par				

### 1Statut

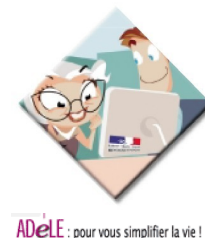
Statut du document	En rédaction
Limitation diffusion	

### 2Diffusion

Diffusé à Société / Entité	Noms	Noms
ADAE	Patrice Posez	



# CybEO



## 3Gestion de version

Version	Date	Description	Editeurs

## 4Documents de référence

N° Référence	Document	Référence

## 5Liens

Liens amont	
Liens aval	

## 6Bibliographie

Document	Auteur

## 7Abréviations

Terme	Description

## 8Glossaire

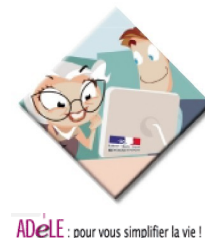
Terme	Description

## 9Annexes séparées

Repère	Référence et titres	Version	Date



# CybEO



## 2 - Sommaire

<b>1 - GESTION DU DOCUMENT.....</b>	<b>2</b>
<b>2 - SOMMAIRE.....</b>	<b>4</b>
<b>3 - INTRODUCTION.....</b>	<b>5</b>
<b>4 - RECENSEMENT DES APPLETS DE CYBEO.....</b>	<b>6</b>
<b>4.1 - Recensement des Applets.....</b>	<b>6</b>
4.1.1 - Les fichiers contenant des Applets.....	6
<b>4.2 - Applets utilisés et analyse de leur usage.....</b>	<b>7</b>
<b>5 - MÉTHODOLOGIE D'IMPLÉMENTATION DU CONTOURNEMENT DES APPLETS.....</b>	<b>8</b>
<b>5.1 - Solution de contournement du chat.....</b>	<b>8</b>
5.1.1 - Lace Chat.....	8
5.1.2 - XHTML Live Chat.....	8
5.1.3 - Ajchat.....	8
5.1.4 - Chategory.....	8
5.1.5 - ARSC.....	9
5.1.6 - CGI:IRC.....	9
5.1.7 - Jabber.....	9
5.1.8 - Conclusion et sélection de la solution de contournement du Chat.....	11
5.1.9 - Mise en oeuvre de la solution de contournement du chat.....	12
<b>5.2 - Méthode d'implémentation des éléments « multimédia ».....</b>	<b>12</b>
5.2.1 - Pourquoi utiliser des bibliothèques existantes ?.....	12
5.2.2 - Remplacement de l'applet AppletCybEO.....	13
<b>5.3 - Méthode d'implémentation des éléments « métiers » : l'API SCORM.....</b>	<b>13</b>
5.3.1 - Claroline.....	13
5.3.2 - eSCORTE for Javascript.....	13
5.3.3 - Conclusion et choix de la méthode de contournement.....	13

## 3 - Introduction

---

L'application CybEO utilise les applets afin de permettre l'intégration facile d'une petite application Java au sein d'un navigateur web (par exemple le chat dans le cadre de notre application).

Cependant, les applets affichent parfois certaines incompatibilités avec certains navigateurs car ces derniers ne sont pas nécessairement compatibles Java. La présence d'un JRE, éventuellement sous forme de plugin, est nécessaire pour utiliser ces applets. Par ailleurs, celles-ci sont plutôt reconnues pour être lentes à l'exécution.

Cette limitation est pénalisante pour le projet CybEO et ses utilisateurs car :

- elle restreint les contextes d'utilisation de cette plate-forme libre, qui a vocation à une large diffusion, dans l'administration ou dans le secteur privé, à certains navigateurs ;
- malgré des performances raisonnables, elle ne contribue pas à l'optimisation en terme de rapidité d'exécution de l'application CybEO ;
- la politique de sécurité mise en place dans certains établissements ne permet pas l'utilisation d'applets Java, soit parce que le firewall les filtre, soit parce que l'installation du plugin n'est pas autorisée sur les postes de travail.

Ce document recense l'ensemble des applets présentes dans l'application CybEO, étudie les outils et *frameworks* permettant de simplifier et de pérenniser leur contournement et propose une méthodologie d'implémentation de ces contournements.

## 4 - Recensement des Applets de CybEO

### 4.1 - Recensement des Applets

#### 4.1.1 - Les fichiers contenant des Applets

- dev/dev-  
apiscorm/business/src/fr/gouv/adae/cybeo/scorm/api/applet/  
APIApplet.java

API SCORM permettant d'effectuer les opérations relatives au statut des composants de la LMS.

##### Appelants :

Cette applet est appelée par les vues

dev-web/presentation/web/jsp/default/pages/apiscorm.jsp et  
/dev-web-editor/presentation/web/editors/htmleditor1/asset/asset\_roman\_photo/loadcourse.jsp du côté serveur.

Du côté client par

/lib/tomcat-4.1.31/webapps/cybeo/jsp/default/pages/apiscorm.jsp et  
/lib/tomcat-  
4.1.31/webapps/cybeo/editors/htmleditor1/asset/asset\_roman\_photo/loadcourse.jsp.

- dev/dev-chat12/business/src/fr/gouv/adae/cybeo1/applet/AppletCybeo.java

AppletCybeo a pour fonctions le chargement initial d'image et l'affichage de boîte de dialogue.

La classe VirtualClass hérite de AppletCybeo.

##### Appelants :

Elle est appelée par /lib/tomcat-4.1.31/webapps/cybeo/jsp/test/chat/diapo.jsp côté client.

- dev/dev-chat12/business/src/fr/gouv/adae/cybeo1/applet/chat/ChatApplet.java

Par héritage, respectivement ChatAppletCommon.java et ChatAppletAdmin.java.

Cette applet remplit les fonctions relatives au chat.

##### Appelants :

Elle est appelée par le fichier

dev/dev-chat12/presentation/web/jsp/test/chat/chat.jsp côté serveur, et  
/lib/tomcat-4.1.31/webapps/cybeo/jsp/test/chat/chat.jsp côté client.

- dev/dev-  
tracking/business/src/fr/gouv/adae/cybeo/tracking/applets/TrackerApplet.java

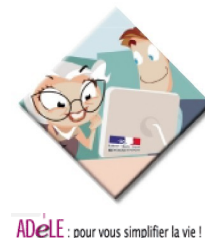
Applet dont le rôle est de calculer le temps passé sur une présentation SCO (*Sharable Courseware Object*).

##### Appelants :

Pas d'appel référencé.



# CybEO



- `dev/dev-web/business/src/fr/gouv/adae/cybeo/web/applets/DetectPluginApplet.java`

Ce fichier est une applet chargée de détecter la version du JRE à partir de la propriété système « `java.version` ».

## Appelants :

Appelé par `/dev-web/presentation/web/jsp/default/pages/_login.jsp`

et `/dev-web/presentation/web/jsp/default/pages/updatebrowser.jsp` côté serveur.

Coté client par

`/lib/tomcat-4.1.31/webapps/cybeo/jsp/default/pages/updatebrowser.jsp`

et `/lib/tomcat-4.1.31/webapps/cybeo/jsp/default/pages/_login.jsp`.

## 4.2 - Applets utilisés et analyse de leur usage

Nous pouvons décomposer les différentes applets contenues dans CybEO en trois catégories :

- Les applets inutilisées : c'est le cas ici de `TrackerApplet.java`.
- les applets « métier » :
  - `APIApplet.java` fournit une API SCORM 1.2 sur le navigateur. Elle communique avec la servlet `APIScormGateway` côté serveur . Sa fonctionnalité doit être gardée à l'identique, c'est-à-dire que l'API actuelle doit être conservée. Les données envoyées au serveur peuvent éventuellement être différentes sous réserve d'une modification côté serveur.
  - `DetectPluginApplet.java` permet de détecter la version de Java sur le navigateur. Comme les applets Java seront remplacées, cette applet deviendra inutile, elle peut donc être enlevée.
- les applets « graphiques » :
  - `ChatApplet.java` est utilisé pour le module de chat. Ce module permet aux différents intervenants d'accéder à des salons de discussion et de discuter entre eux. Il est également possible de voir le profil CybEO de chaque utilisateur (ensemble des propriétés associées à un utilisateur), d'envoyer un fichier à un autre utilisateur et d'utiliser un tableau blanc pour communiquer. Le chat en lui même peut être remplacé par un autre composant n'utilisant pas Java (HTML pur ou HTML+Javascript). Les fonctions spécifiques (vue du profil, tableau blanc et transfert de fichier) ne pourront peut-être pas être reprises telles quelles.
  - `AppletCybeo.java` n'est utilisé que pour afficher les vignettes des utilisateurs dans la *frame* supérieure de la fenêtre de chat. Comme le chat sera entièrement remplacé par un autre outil, cette applet peut être enlevée.

Pour résumer, nous avons donc deux fonctions pour lesquelles nous devons trouver une solution de contournement : le chat et l'API SCORM.

## 5 - Méthodologie d'implémentation du contournement des Applets

---

### 5.1 - Solution de contournement du chat

La solution consistera à substituer l'applet utilisée par un outil existant. Cet outil devra être *Open Source* pour se conformer à l'esprit et à la licence de CybEO. Il sera également conforme aux exigences de compatibilité de navigateurs.

Nous préconiserons la solution la plus facile à intégrer et la plus évolutive.

#### 5.1.1 - Lace Chat

Lace Chat (GNU General Public License, <http://socket7.net/lace/>) est un logiciel de communication Web écrit en PHP, Javascript, XHTML et CSS.

Il est compatible Firefox, IE 5.5 +, Safari et Opéra 8. Lace a été construit sur Apache 1.3.27 et PHP 4.3.10, il n'y a aucune garantie qu'il fonctionne sur une autre version de ces plate-formes.

Il ne gère pas nativement plusieurs salons de discussion.

#### 5.1.2 - XHTML Live Chat

XHTML Live Chat (CC by-nc-sa 2.0, [http://www.plasticshore.com/logfile/entry/legacy\\_483/](http://www.plasticshore.com/logfile/entry/legacy_483/)) est un chat implémenté avec PHP, MySQL, XHTML et AJAX.

Sa licence le rend incompatible avec la diffusion de CybEO (pas d'exploitation commerciale).

#### 5.1.3 - Ajchat

Ajchat (<http://www.ajchat.com/chat/ajax>) est un système de messagerie instantanée en AJAX souple et portable. Il ne s'agit pas d'un logiciel *Open Source* mais d'un service librement utilisable en ligne.

Il est compatible avec Firefox, Flock, Safari, IE 5.5+. La solution est facilement intégrable mais repose sur la disponibilité d'un service tiers, dont la pérennité n'est pas garantie. De plus, un accès à Internet est requis pour pouvoir utiliser le service, ce qui n'est pas toujours le cas dans le cadre d'un Intranet sécurisé.

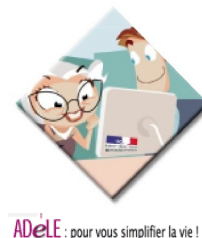
#### 5.1.4 - Chategory

Chategory (Apache License V2.0, <http://chategory.sourceforge.net/>) est un système de chat écrit en PHP. Il rafraîchit ses contenus sans recharger sa page grâce à AJAX. Il est prévu pour être intégré à un autre site web, mais ne gère pas nativement plusieurs salons de discussion.





# CybEO



## 5.1.5 - ARSC

ARSC (GNU *General Public License*, <http://manuel.kiessling.net/projects/software/arsc/>) est un système de chat qui utilise PHP et MySQL. Il gère nativement plusieurs langues et tous les navigateurs courants. Il est possible de mettre en place plusieurs salons de discussion, privés, modérés ou publics. Il est utilisé sur de nombreux sites.

## 5.1.6 - CGI:IRC

CGI:IRC (GNU *General Public License*, <http://cgiirc.sourceforge.net>) est un client IRC Web qui utilise une interface CGI. Il faut donc installer un serveur IRC pour le mettre en oeuvre, et effectuer une intégration entre CybEO et le serveur IRC pour la gestion des utilisateurs et des droits d'accès, ce qui semble assez coûteux. De plus, l'utilisation d'IRC et de CGI rend l'utilisation de l'interface peu intuitive.

## 5.1.7 - Jabber

Jabber (<http://jabber.org>) est un système de messagerie instantanée. Il fonctionne en mode client/serveur : des clients Jabber se connectent à un serveur Jabber via le protocole XMPP standardisé par l'IETF. Il existe des clients Jabber Web qui peuvent être intégrés à CybEO. L'utilisation de Jabber présente plusieurs avantages dans le cadre de cette intégration :

- pérennité : XMPP est le protocole standard pour le chat sur Internet. Il est actuellement mis en oeuvre sur de nombreux sites de production. Il a notamment été choisi par Google pour son service « goole talk » ;
- interopérabilité : la standardisation du protocole garantit une interopérabilité maximale qui facilitera les évolutions futures de la fonctionnalité. Ainsi, l'utilisation de clients lourds, plus ergonomique que des interfaces Web, est tout à fait envisageable, pour les tuteurs par exemple ;
- performances : les serveurs Jabber sont capables de supporter une montée en charge importante pour gérer de nombreux utilisateurs simultanément ;
- fonctionnalités : il existe de nombreuses fonctionnalités prévues par le protocole. Ainsi, il est possible de mettre en place des discussions entre deux personnes ou des discussions de groupe via des conférences (« MUC » pour « *Multi User Chat* »), de transférer des fichiers, d'utiliser des tableaux blancs, etc.

### 5.1.7.1.Serveurs Jabber

Les principaux serveurs Jabber *Open Source* utilisés sont ejabberd, Jabberd (1.4 et 2) et Wildfire.

#### 5.1.7.1.1.ejabberd

ejabberd (<http://www.process-one.net/en/projects/ejabberd/>) est un serveur Jabber écrit en Erlang. C'est le seul serveur Jabber *Open Source* à revendiquer

la compatibilité complète avec le standard XMPP défini par l'IETF (XMPP Core and XMPP IM). Ce serveur possède certaines caractéristiques intéressantes :

- le clustering (permet donc une montée en charge importante) ;
- le support intégré des connexions HTTP (*HTTP Polling*) ;
- la compatibilité XMPP ;
- l'architecture modulaire ;
- la localisation dans 11 langues ;
- l'intégration d'un composant de publication / abonnement (Publish-Subscribe) ;
- la gestion des statistiques (JEP-0039: Statistics Gathering) ;
- le support de l'authentification SASL anonyme (dans la version 1.0.1).

Il est publié sous *GNU General Public License*.

#### 5.1.7.1.2.Jabberd 1.4

Jabberd est l'un des serveurs Jabber *Open Source* les plus utilisés. Ses fonctionnalités sont toutefois moins étendues qu'ejabberd (5.1.7.1.1 ejabberd page 9 supra). Il n'implémente pas la totalité du protocole XMPP, il ne supporte le *HTTP polling* que via un module externe et ne permet pas l'authentification SASL anonyme.

Des problèmes de montée en charge de ce serveur ont conduit à sa réécriture totale (cf 5.1.7.1.3 Jabberd 2 page 10 infra).

Il est publié sous *GNU General Public License*.

#### 5.1.7.1.3.Jabberd 2

Jabberd 2 (<http://jabberd.jabberstudio.org/2/>) est une réécriture totale de la première version du serveur Jabberd (5.1.7.1.2 Jabberd 1.4 page 10 supra). La compatibilité avec l'ensemble du standard XMPP n'est pas encore parfaite ; il lui manque notamment la possibilité d'utiliser les connexions HTTP (*HTTP polling*). La version 1.4 reste pour cela encore très utilisée.

Il est publié sous *GNU General Public License*.

#### 5.1.7.1.4.Wildfire

Wildfire (<http://www.jivesoftware.org/wildfire/>) est un serveur Jabber écrit en Java. Il dispose d'une bonne couverture fonctionnelle, notamment l'authentification SASL anonyme. Il ne gère toutefois pas les connexions HTTP (*HTTP polling*).

Il présente l'avantage d'être écrit en Java qui est également la technologie déjà utilisée dans CybEO. Il peut cependant présenter de légers problèmes de stabilité ou de montée en charge.

Il est publié sous double licence : propriétaire et *GNU General Public License*.

### 5.1.7.2.Clients Web Jabber

#### 5.1.7.2.1.JWChat

JWChat (<http://jwchat.sourceforge.net>) est un client Jabber *Open Source* basé sur HTTP. Il est écrit en Javascript, ce qui lui confère une très bonne ergonomie et ne nécessite pas de composant serveur autre que le serveur Jabber. Il est également capable de gérer les conférences (MUC). Chaque utilisateur doit posséder un compte Jabber sur le serveur pour pouvoir utiliser le service, ce qui implique de gérer l'authentification entre les profils Jabber et CybEO.

Il fonctionne sur les navigateurs suivants :

MS-Windows :

- MS Internet Explorer version 5.0 et plus
- Netscape versions 6 & 7
- Mozilla (stable) version 1.0.2 et plus, ainsi que les navigateurs basés sur Gecko (Firefox inclus)

Mac OS X :

- Mozilla (stable) version 1.0.2 et plus
- Camino 0.7
- safari n'est pas encore supporté

Linux:

- Mozilla (stable) version 1.0.2 et plus, ainsi que les navigateurs basés sur Gecko (Firefox inclus)

Il est publié sous *GNU General Public License*.

## 5.1.7.2.2.Laffer

Laffer (<http://labs.linux-bg.org/cgi-bin/laffer/index.pl>) est un client Jabber Web, similaire à JWChat dans son utilisation, bien qu'un peu moins ergonomique. Il utilise PHP et MySQL côté serveur, ce qui rend son installation et sa maintenance plus lourde. Il est publié sous *GNU General Public License*.

## 5.1.7.2.3.MUCKI

MUCKI (<http://flossoft.biz:8080/jwchat/MUCKI/>) est une application Web de conférence (MUC) utilisant Jabber. C'est une version simplifiée de JWChat (5.1.7.2.1 JWChat page 10 supra), mais spécialisée dans les salons de discussion. Il est utilisable de façon anonyme : des comptes temporaires Jabber sont créés à la connexion au service et détruits ensuite. Un travail d'intégration pour connecter automatiquement un profil CybEO et pour créer les salons de discussions visibles par un utilisateur est nécessaire pour l'utiliser avec CybEO. Il est publié sous *GNU General Public License*.

## 5.1.8 - Conclusion et sélection de la solution de contournement du Chat.

Les solutions basées sur MySQL (ou toute autre base autre que PostgreSQL) ne sont pas retenues puisque CybEO utilise déjà une base PostgreSQL ; la présence de deux systèmes de gestion de bases de données différents nuirait grandement à l'installation, à l'administration et à la maintenance du produit.

De même, les solutions ne gérant pas nativement plusieurs salons de discussion et les logiciels non libres sont éliminées.

Il ne reste donc que les solutions basées sur IRC et Jabber.

IRC est un protocole très répandu sur Internet et le protocole XMPP utilisé par Jabber est une norme officielle (standard IETF). La pérennité des deux solutions est donc acquise. IRC est un protocole ancien et son utilisation souffre d'une ergonomie peu conviviale, malgré les efforts effectués au niveau des clients récents. L'ergonomie des clients Jabber est souvent bien meilleure. En terme d'évolutivité, XMPP est un protocole bien plus ouvert que IRC ; il a été conçu pour pouvoir évoluer facilement sans compromettre sa compatibilité ascendante. Il est de plus basé sur XML, ce qui facilite l'interopérabilité entre différents environnements. IRC quand à lui est un protocole beaucoup plus ancien et plus « brut ». Son évolution est stoppée et il n'a pas été conçu dans une optique d'interopérabilité et de diversification de ses fonctionnalités.

Il apparaît donc qu'une solution reposant sur Jabber présente les meilleurs avantages en terme de pérennité, d'interopérabilité, d'évolution et de fonctionnalités.

## 5.1.8.1. Serveur

Le tableau 1 (Comparaison des serveurs Jabber, infra) recense les différentes caractéristiques des serveurs Jabber. L'accent a été porté les fonctionnalités qui sont susceptibles d'être importantes dans le cadre de l'intégration avec CybEO :

- Le *Multi-User Chat* (MUC) permet de mettre en place des salons de discussion adaptés à chaque formation ;
- les clients Web utilisent la plupart du temps la connexion HTTP (http polling) pour communiquer avec le serveur Jabber ;
- l'authentification SASL anonyme permet de se connecter sans création préalable de compte Jabber, ce qui peut faciliter son intégration.

Serveur	performances	MUC	HTTP Polling	SASL anonyme	Avantages
<i>ejabberd</i>	Très bonne, notamment grâce à la clusterisation.	Natif	Natif	Oui (version 1.0.1)	Conforme XMPP
<i>Jabberd 1.4</i>	Problèmes de montée en charge	Possible via un plugin	Natif	Non	Très répandu
<i>Jabberd 2</i>	Bonne	Pas en natif, mais des contournements existent	Non	Oui	Successeur de Jabberd 1.x
<i>Wildfire</i>	Problèmes de montée en charge	Natif	Non	Oui	En Java, comme CybEO

Tableau 1: Comparaison des serveurs Jabber

Nous recommandons l'utilisation de ejabberd.

## 5.1.8.2.Client

La solution reposant sur un client Web Jabber peut être mise en place de deux manières :

### 5.1.8.2.1.Authentification classique

C'est l'utilisation classique de Jabber, où chaque utilisateur est authentifié sur le serveur. Cette solution est plus coûteuse en terme d'intégration, puisqu'il faut créer sur le serveur des comptes qui correspondent aux profils CybEO. L'utilisation de JWChat (5.1.7.2.1 JWChat page 10 supra) est recommandée pour mettre en oeuvre cette solution, il est plus simple d'installation que Laffer (5.1.7.2.2 Laffer page 11 supra), puisqu'il fonctionne uniquement sur le navigateur.

### 5.1.8.2.2.Comptes temporaires

L'utilisation de comptes Jabber temporaires, créés à la volée et détruits à la fin de la session, permet une intégration de Jabber moins coûteuse. Cette technique est utilisée par MUCKI (5.1.7.2.3 MUCKI page 11 supra). MUCKI permet l'accès anonyme, or il importe tout de même de restreindre l'accès au chat aux seuls utilisateurs authentifiés sur CybEO. Un travail d'intégration au niveau de la sécurité (SSO CybEO/Jabber) reste donc à effectuer.

Cette solution est à privilégier pour sa simplicité d'utilisation.

## 5.1.9 - Mise en oeuvre de la solution de contournement du chat.

L'équivalence des fonctionnalités sera évaluée en utilisant le cahier de test réalisé pour l'ADAE dans le marché « CybEO : Soutien à la communauté ». Les éléments pertinents du cahier de tests concernent essentiellement :

- les incorporations de chat dans des sessions de formation ;
- le fonctionnement du chat face aux différentes connexions des utilisateurs ;
- l'affichage des boîtes de dialogue et le chargement des images.

## 5.2 - Méthode d'implémentation des éléments « métiers » : l'API SCORM

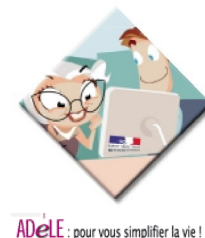
Le remplacement de l'applet `APIApplet`, qui gère l'API SCORM, peut se faire avec du code Javascript. Plutôt que réécrire un script fournissant toute l'API, ce qui sera coûteux en terme de développement et de stabilisation du code, il serait judicieux de se baser sur les implémentations *Open Source* qui ont déjà été réalisées en Javascript.

### 5.2.1 - Claroline

Le logiciel libre Claroline ([http://www.claroline.net/doc/en/index.php/Main\\_Page](http://www.claroline.net/doc/en/index.php/Main_Page)) possède un script Javascript fournissant l'API SCORM 2004. Le script en question



# CybEO



(`scormAPI.inc.php`) est généré par du PHP, mais il doit être possible de le convertir en JSP pour une intégration dans CybEO. Le mécanisme utilisé pour envoyer les données au serveur se base sur une *frame* cachée qui contient un formulaire qui est validé par le script. Les données sont envoyées via la méthode HTTP POST. Le code est disponible sous la GNU *General Public License*.

## 5.2.2 - eSCORTE for Javascript

Il existe également une API SCORM 2004 écrite entièrement en Javascript : eSCORTE for Javascript (<http://sourceforge.net/projects/escorte4js>). Le script côté client (sur le navigateur) utilise `XMLHttpRequest` pour communiquer en SOAP avec un Webservice sur le serveur. Le fichier WSDL qui décrit les méthodes que doit fournir le Webservice est déjà écrit, mais la réalisation du Webservice dans CybEO reste à effectuer. Le code Javascript semble prévu pour fonctionner avec Microsoft Internet Explorer, Mozilla Firefox et Safari. Le code est disponible sous la *Mozilla Public License*.

## 5.2.3 - Conclusion et choix de la méthode de contournement

La méthode utilisée par Claroline demande le plus gros travail d'intégration, puisqu'il faut ajouter dans les pages HTML de CybEO des *frames* cachées contenant des formulaires. Le programme Javascript est généré par du code PHP. Cela implique de convertir ce générateur en JSP. Il faut également développer les servlets ou les JSP côté serveur sur lesquelles les formulaires vont effectuer le HTTP POST. L'intégration de eSCORTE for Javascript ne demande quant à lui que le chargement d'un seul fichier Javascript pour fonctionner côté client. Il faut toutefois coder les Webservices qui seront utilisés par le code Javascript via SOAP, afin de remplacer la servlet `APIScormGateway`. Cette méthode de communication entre le client et le serveur est de plus plus élégante, standard, robuste et maintenable que les *frames* cachées utilisées par Claroline. Le code de eSCORTE for Javascript semble également plus complet que celui de Claroline. Cependant, le code de eSCORTE for Javascript n'est pas aussi répandu et testé que celui de Claroline ; la dernière version disponible est d'ailleurs une *release candidate 1*. Il faudra donc prévoir une phase de test de ce code, ainsi que d'éventuelles corrections, ajouts et adaptations du code à CybEO. Cette solution reste néanmoins moins coûteuse que le développement *from scratch* de cette API.

**Fin de document**