



HAL
open science

Affichage dynamique sur grands écrans - retour d'expérience avec Xibo

Olivier De-Marchi, Gabriel Pierre André Moreau

► To cite this version:

Olivier De-Marchi, Gabriel Pierre André Moreau. Affichage dynamique sur grands écrans - retour d'expérience avec Xibo. Congrès JRES : Les Journées Réseaux de l'Enseignement et de la Recherche, RENATER, Nov 2017, Nantes, France. hal-01669053

HAL Id: hal-01669053

<https://hal.science/hal-01669053v1>

Submitted on 20 Dec 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC0 - Public Domain Dedication 4.0 International License

Affichage dynamique sur grands écrans - retour d'expérience avec Xibo

Olivier De Marchi

Laboratoire LEGI
1209-1211, rue de la piscine
38400 Saint Martin d'Hères

Gabriel Moreau

Laboratoire LEGI
1209-1211, rue de la piscine
38400 Saint Martin d'Hères

Résumé

Le laboratoire LEGI (UMR-5519) situé sur le campus de Grenoble a mis en place, début 2015, une solution de diffusion d'informations sous forme de contenus multimédias au travers de larges moniteurs suspendus. L'objectif du projet est double :

- *améliorer l'arrivée et l'attente des visiteurs par la diffusion dans le hall d'accueil de contenus présentant les activités de recherche du laboratoire. Les personnes de passage ont alors un aperçu rapide de nos activités ;*
- *enrichir la communication interne via ce nouveau média grâce aux messages diffusés : assemblées générales, projets ANR/ERC/Horizon2020 obtenus, soutenances de thèse et d'HdR, écoles d'été et d'hiver, séminaires, flux RSS des tutelles, etc.*

Nous proposons un retour d'expérience autour de la solution technique retenue : le logiciel libre Xibo. Cette solution présente l'avantage d'une architecture maîtrisée dont le code source est ouvert et porté par une communauté de développeurs. Xibo fonctionne en mode client-serveur. Il est multi-écrans, multi-zones, entièrement programmé pour l'affichage dynamique et contrôlé par une interface web centralisée. Le serveur est une application web basée sur le couple PHP/MySQL pouvant tourner dans une petite machine virtuelle GNU/Linux. Les clients fonctionnent sur Windows (logiciel libre) ou Android (version commerciale). L'architecture de Xibo est robuste aux coupures réseaux de courtes durées grâce au cache local des clients sur l'ensemble des données les concernant.

Mots clefs

affichage dynamique, télévision, broadcast, retour d'expérience, communication, digital signage

1 L'affichage dynamique : définition et intérêts

1.1 Définition de l'affichage dynamique

L'affichage dynamique (en anglais : digital signage), est un outil de communication et de marketing qui permet de diffuser des informations, sous forme de contenus multimédias, dans les lieux publics sur un écran, un mur d'écrans ou par vidéo-projection. Une solution d'affichage dynamique est articulée autour de

trois composants : un écran (souvent un téléviseur LCD), un mini PC et un serveur de contenus multimédia (réseau local ou service en ligne).

1.2 Intérêts

Pour le public, ce canal de communication permet de transmettre une information très attractive grâce aux nombreuses possibilités offertes par cette technologie : images, vidéos, audios, messages texte, flux RSS, widgets, etc.

Pour les gestionnaires, l'interface centralisée a de nombreux avantages. Elle permet :

- de créer ces contenus très variés : images, vidéos et autres objets ;
- d'envoyer et de mettre à jour les contenus instantanément ;
- d'obtenir un affichage uniforme et synchronisé sur plusieurs écrans ;
- de contextualiser et de planifier les affichages grâce aux fonctions de calendrier et d'horaires de diffusion.

1.3 Au LEGI

Au laboratoire LEGI, l'affichage dynamique présente un double intérêt :

- proposer un nouveau média de communication à destination de tous les visiteurs : chercheurs invités, industriels de passage, stagiaires, personnels d'autres laboratoires et de tous les publics qui visitent notre laboratoire (lors de la fête de la science ou des cours de l'Université Inter-Âges du Dauphiné par exemple) ;
- améliorer la communication interne pour diffuser les informations pratiques : assemblée Générale, soutenances de thèse et d'habilitation à diriger des recherches (HDR), séminaires, etc.

2 Choix technique : Xibo

2.1 Le projet Xibo

Xibo est une solution open source d'affichage dynamique. La plate-forme est basée sur un composant web de système de gestion de contenu (serveur) et des applications tournant sur Windows ou Android (clients). Xibo est sous licence GNU Affero General Public. La solution est soutenue par la société **Spring Signage**, le sponsor principal de Xibo.

2.2 Le choix du LEGI : un outil open source

La solution Xibo répond aux principaux critères souhaités par le LEGI :

- le code source est ouvert. Il est développé sous licence **AGPLv3** : <http://xibo.org.uk/legal/> ;
- l'architecture proposée est très « classique » : Xibo fonctionne en mode client-serveur. Le serveur est une application web (système de gestion de contenu ou CMS) basée sur le couple PHP/MySQL. Il peut être facilement installé sur une machine virtuelle GNU/Linux ;
- les clients sont disponibles sur plusieurs plate-formes : Windows et Android (version commerciale) ;
- le projet créé en 2004 est toujours actif. Le développement évolue sans cesse. Un dépôt github est disponible : <https://github.com/xibosignage/xibo/> ;

- les fonctionnalités indispensables pour le LEGI sont présentes : multi-écrans, multi-zones et l'architecture de Xibo est robuste aux coupures réseaux de courtes durées grâce au cache local des clients sur l'ensemble des données les concernant.

2.3 Xibo, une multitude de fonctionnalités

Par ailleurs, la solution Xibo comprend de nombreux autres atouts :

- de nombreuses possibilités de planification grâce aux fonctions de calendrier : événements réguliers, règle de priorité, etc ;
- un réglage fin des permissions pour les utilisateurs ou les groupes d'utilisateurs ;
- des statistiques sur l'utilisation de Xibo. Par exemple, en affichant la bande passante consommée par un client ;
- une documentation complète : <http://xibo.org.uk/manual/en/index.html> ;
- une communauté organisée autour d'un forum très actif : <https://community.xibo.org.uk/> ;
- un blog pour suivre l'actualité du projet : <http://xibo.org.uk/blog/> ;
- une plate-forme de développement accessible à tous : <https://github.com/xibosignage/xibo/> ;
- une conception avancée de la mise en page avec des possibilités de partage de celle-ci avec la communauté d'utilisateurs de Xibo : <http://xibo.org.uk/layouts/> ;
- les dernières versions de Xibo proposent une image docker, pour la partie serveur, prête à l'utilisation (depuis mars 2017) ;
- des possibilités d'hébergement (Cloud) de la partie serveur par **Spring Signage** (payant) ;
- achat de support et de personnalisation de tous les produits par **Spring Signage** ;
- il existe aussi un partenaire industriel qui propose du matériel avec le client Xibo Windows ou Android pré-installé : <http://dsdevices.com/>.

2.4 Autres solutions existantes

La plupart des solutions du marché sont commerciales et payantes. Certaines offres proposent des solutions d'hébergements des contenus en ligne avec une partie cliente gratuite. Voici une liste non exhaustive de solutions d'affichage dynamique :

- <https://www.screenly.io/> ;
- <https://www.telelogos-media4display.com/> ;
- <https://citymeo.fr/> ;
- <https://www.videospot.com/> ;
- <https://www.viewneo.com/> ;
- <http://www.digitalsignage.com>.

3 Mise en place et réalisation

3.1 Maquette de test

En 2015, nous avons accueilli, pour un stage de 8 semaines, un élève en Bac Pro SEN (Systèmes Électroniques et Numériques). Une de ses missions a été de créer un environnement de test pour expérimenter

Xibo. Le stagiaire doté d'une machine de bureau (DELL OptiPlex 960) sous GNU/Linux Debian a réalisé une maquette à base de machines virtuelles VirtualBox. Il a ainsi créé plusieurs machines virtuelles : une machine GNU/Linux Debian Wheezy pour la partie serveur et deux machines clients (Windows 7 64 bits et Ubuntu 14.04 LTS). Les résultats des tests ont permis de valider la solution concernant la partie serveur et la partie cliente Windows. Quant au client Ubuntu, il n'a jamais fonctionné correctement. Ce client a été abandonné en 2014 : <https://blog.xibo.org.uk/xibo-for-ubuntu-alpha-discontinuation-notice/>. Un nouveau client GNU/Linux est en cours de développement depuis septembre 2017 : <https://github.com/xibosignage/xibo-linux>.

3.2 Architecture matérielle au LEGI

Par la suite, la solution a légèrement évolué et a été déployée avec le matériel suivant :

- un DELL R730 pour la partie serveur (machine virtuelle XEN Debian Wheezy) ;
- un DELL OptiPlex 9020 AIO avec Ubuntu 14.04.5 LTS et une machine virtuelle VirtualBox, Windows 7 64 bits ;
- un écran LED de 40 pouces : Samsung ED40C - Full HD : 1920x1080 pixels.

Le client Xibo est installé sur la machine virtuelle Windows. L'OptiPlex est relié à l'écran Samsung ED40C via un câble HDMI. Il y a un double affichage qui permet d'utiliser la partie Ubuntu comme une machine libre-service dans la cafétéria du laboratoire et la partie machine virtuelle Windows déportée vers l'écran Samsung.

La figure 1 représente l'architecture initiale au LEGI en 2015.

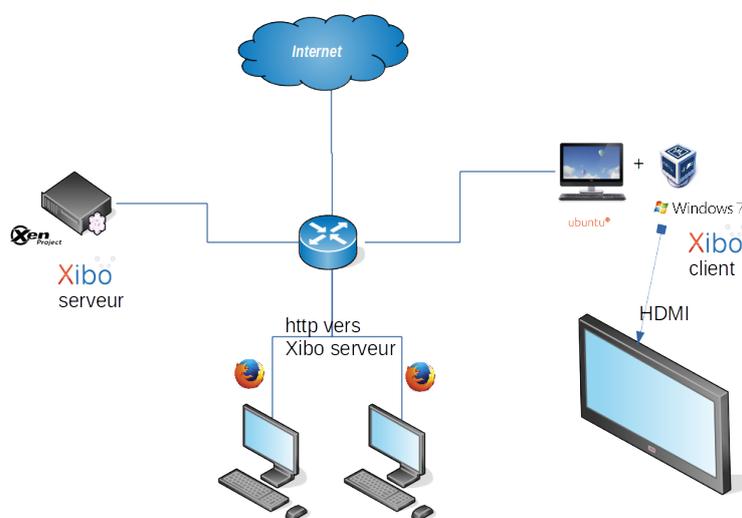


Figure 1 - Architecture matérielle au LEGI en 2015

C'est une solution convenable qui a fonctionné pendant près de 2 ans. Bien entendu, ce n'est pas idéal puisque le poste d'affichage est aussi un poste libre-service. Il est bien plus fiable de dédier le poste client à l'affichage.

Au début de l'année 2017, la cafétéria a été victime d'un cambriolage. Le poste libre-service a été volé. Nous nous sommes donc tournés vers l'architecture recommandée à base de mini PC proche de l'écran

d'affichage. Nous avons opté pour un DELL OptiPlex 7040 format micro (son successeur le DELL 7050 est disponible sur le marché [Matinfo4](#)). Cette solution, qui nécessite la présence d'un câble réseau près du téléviseur, fonctionne très bien depuis plusieurs mois maintenant.

Il existe de nombreuses autres solutions à base de mini PC, voici quelques idées :

- les mini PC <https://www.tinygreenpc.com/> et notamment le [Fitlet T](#) ;
- l'[Intel NUC](#) ;
- l'[Intel Compute Stick](#) (nécessite un réseau Wi-Fi actif) ;
- les mini PC de la marque [Zotac](#) ;
- le [Laboratoire de Mathématiques de Besançon](#) utilise Xibo avec un client Raspberry Pi 3 et Android. Cette solution a été l'objet d'une [présentation](#) lors des journées Journées Mathrice 2017 de Besançon.

3.3 Architecture logicielle au LEGI

La partie serveur a été installée en 2015 sur une machine virtuelle XEN Debian 7 (Wheezy). Xibo-cms était alors en version 1.7.1. En 2017, nous avons migré cette machine virtuelle vers Debian 8 (Jessie) et migré Xibo-cms en version 1.8.2.

Pour la configuration d'Apache, nous avons suivi la recommandation 4 de la [documentation de Xibo](#) qui préconise l'utilisation d'un alias :

```
Alias /xibo "/srv/xibo-cms/web"  
  
<Directory "/srv/xibo-cms/web">  
    AllowOverride All  
    Options Indexes FollowSymLinks MultiViews  
    Order allow,deny  
    Allow from all  
    Require all granted  
</Directory>
```

Il faut aussi activer la réécriture d'URL :

```
a2enmod rewrite
```

La partie cliente a parfaitement fonctionné sur une machine virtuelle VirtualBox Windows 7 64 bits pendant 2 ans. Cette solution présente tous les avantages d'une VM classique (facile à sauvegarder et à restaurer). Mais elle a l'inconvénient de demander plus de ressources à la machine physique.

Depuis la migration vers le DELL OptiPlex 7040, nous avons opté pour une installation du client Xibo directement sur Windows 10. En effet, la configuration est très simple. Elle ne présente aucune complexité qui pourrait nécessiter une sauvegarde régulière.

Le client doit répondre à quelques exigences logicielles :

- l'environnement logiciel .NET version 4.5 ;
- Internet Explorer 8 ou supérieur (version 11 sur Windows 10 par défaut) ;
- le lecteur multimédia Flash Player version 9 ou supérieur (pour les vidéos en Flash) ;
- le lecteur Windows Media version 11 ou supérieur (version 12 par défaut sur Windows 10). Les vidéos utilisent ce lecteur. Il est indispensable d'ajouter les codecs nécessaires au décodage des vidéos que nous utilisons. Pour nous, les codecs [H264](#) et [WebM](#) sont à installer. Nous avons choisis d'installer [Klite-codecs](#) ;

- Microsoft PowerPoint 2003 ou supérieur est nécessaire pour la lecture des fichiers Microsoft PowerPoint. La visionneuse Microsoft PowerPoint n'est pas suffisante. Nous ne diffusons actuellement que des présentations au format PDF compatible avec tous les OS de nos utilisateurs.

Le module facultatif Watchdog est activé par défaut. Il permet de surveiller la bonne exécution de l'application cliente et de la relancer si nécessaire. Ce module est configuré grâce à un fichier XML :

```
C:\Program Files (x86)\Xibo Player\watchdog\x64\XiboClientWatchdog.exe.config
```

La variable Threshold permet de définir en seconde l'attente avant un redémarrage. Par défaut, cette valeur est à 120 secondes.

La machine démarre automatiquement sur un compte utilisateur standard grâce à l'utilitaire Windows cnetplwiz.

Nous avons activé le bureau à distance Windows (RDP) afin de prendre le contrôle de la machine pour les maintenances. Pour optimiser la consommation électrique du client et faciliter l'application des mises à jour, ce dernier est allumé à distance et éteint de façon automatique tous les jours :

- le démarrage automatique est possible grâce au [Wake on Lan \(WoL\)](#). La commande suivante permet d'allumer la machine à distance depuis un serveur du laboratoire ;

```
wakeonlan -i adresse-reseau adresse-mac
```

Nous avons placé cette commande dans un cron pour éviter de démarrer la machine pendant les week-end, les périodes de congés ainsi que les jours de fermeture (lundi de l'ascension par exemple) ;

- l'arrêt automatique de Windows est programmé tous les soirs à 18h grâce à une tâche planifiée Windows à l'aide de l'utilitaire taskschd.msc. Voici un extrait du fichier XML d'export de la tâche planifiée :

```
<Actions Context="Author">
  <Exec>
    <Command>C:\Windows\System32\shutdown.exe</Command>
    <Arguments>/s /hybrid /t 00</Arguments>
  </Exec>
</Actions>
```

3.4 Aspects sécurités

La sécurité physique du matériel dans le Hall d'entrée est assurée par des câbles à clés au niveau du téléviseur et du mini PC. L'accès à la salle serveur est limité à une liste très restreinte de personnels.

La sécurité logicielle du service consiste principalement à configurer Apache avec un certificat TLS et à placer un reverse proxy filtrant entre le client et celui-ci. Le tout est supervisé en temps réel avec Nagios comme l'ensemble des serveurs du laboratoire.

Concernant le client Windows 10 situé dans le Hall d'entrée, nous avons procédé à une configuration standard similaire à un poste « utilisateur » du LEGI (antivirus et déploiement de mise à jour classiques). Nous avons retiré deux fonctionnalités habituelles :

- le poste n'est pas chiffré car le chiffrement ne permet pas un démarrage automatique piloté ;
- le poste n'a pas été intégré dans le domaine afin de limiter l'accès aux seuls comptes administrateurs.

Par ailleurs, Xibo assure la sécurité de son système au travers d'une double protection :

- il est nécessaire de mettre une clé dans la configuration du client. Cette clé est fournie par le serveur. Elle est valable pour tous les clients ;
- il faut autoriser explicitement chaque dispositif d'affichage depuis l'interface d'administration de Xibo.

En cas de vol physique, la perte d'information est faible car le poste client n'a dans son cache que des données peu sensibles dont la plupart sont déjà publiques sur notre site web.

4 Utilisation

4.1 Création d'un design

La figure 2 représente la mise en page que nous avons choisie pour l'affichage dans le hall d'entrée du LEGI.

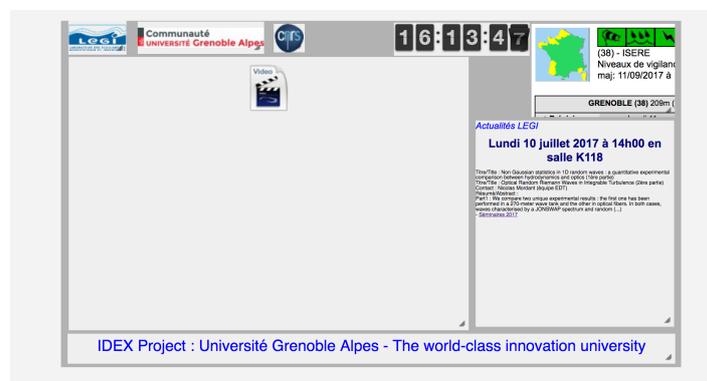


Figure 2 - Mise en page - Hall d'entrée

Cette mise en page comporte 6 zones :

- une zone de logos : avec les logos des tutelles du laboratoire ;
- une zone widget horloge : pour l'affichage de l'heure ;
- une zone widget météo : pour l'affichage des prévisions météorologiques ainsi que les alertes Vigicrues (campus en zone inondable) ;
- une zone flux RSS : pour charger automatiquement du contenu de sites web partenaires ;
- une zone vidéos ;
- une zone bandeau.

4.2 Dépôt et mise à jour du contenu

La mise à jour est automatique pour l'horloge et la météo.

Pour le flux RSS, la figure 3 montre la liste des abonnements aux divers flux RSS de notre mise en page actuelle. La figure 4 détaille le flux RSS du journal du CNRS. Une fois mis en place, plus aucune intervention n'est nécessaire.

Les images, vidéos, diapositives (PowerPoint) et fichiers PDF sont stockés dans une bibliothèque accessible par tous les clients Xibo. De nombreux formats sont acceptés. Depuis l'interface de gestion, il suffit de téléverser ces documents. Ensuite, ils peuvent être associés aux mises en pages disponibles.

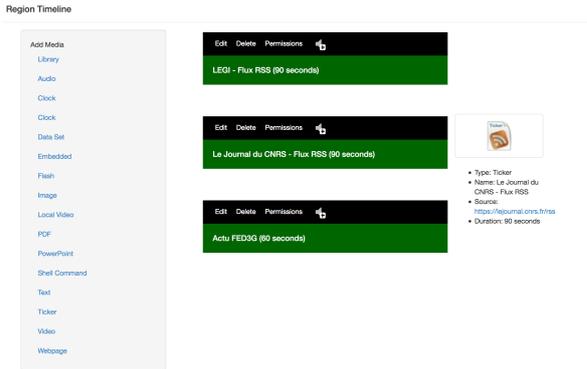


Figure 3 - Liste des flux RSS

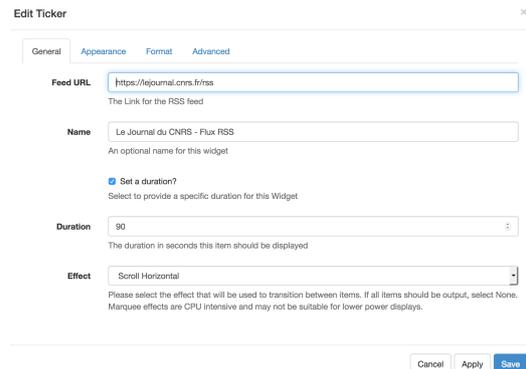


Figure 4 - Flux RSS du journal du CNRS

La zone de bandeau sert à faire défiler des petits messages courts en bas de l'écran. La figure 5 montre un exemple des messages affichés au LEGI. Elle est alimentée via l'onglet DataSets de l'interface d'administration de Xibo. Ces messages peuvent être partagés par plusieurs dispositifs d'affichage.

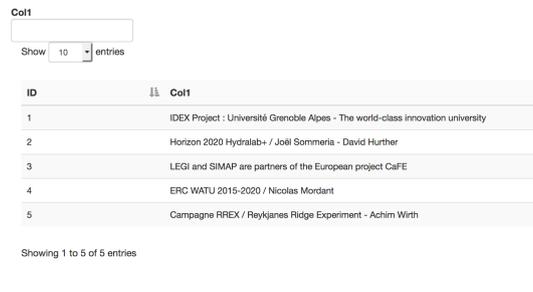


Figure 5 - Messages défilants

4.3 Préparation des fichiers

Depuis la version 1.8 (mars 2017), Xibo supporte l'affichage de PDF grâce à un module : pdfjs. Cependant, la technique présentée ci-dessous peut encore rendre des services !

Dans les versions antérieures, il était nécessaire de transformer les PDF en vidéos. Nous avons mis au point un petit script pour automatiser la transformation. On commence par créer une série d'images. On duplique la première car le codec H264 perd la première avec la version actuelle de ffmpeg. On reclasse alors les images puis on calcule la taille de la future vidéo en fixant la hauteur. La vidéo est alors créée en prenant une image toutes les 5 secondes.

```
export presentation=presentation.pdf
finalheight=1400

rm -f tmp_image-*. *
convert -verbose -density 300 -trim "${presentation}" \
  -quality 100 tmp_image-%03d.jpg
cp tmp_image-000.jpg tmp_image-000.JPG
jhead -n"tmp_image-%04i" tmp_image-*. *
```

```

actualwidth=$(identify tmp_image-0001.jpg | awk '{print $3}' \
| cut -f 1 -d x)
actualheight=$(identify tmp_image-0001.jpg | awk '{print $3}' \
| cut -f 2 -d x)
finalwidth=$(( ( ( ${actualwidth} * \
${finalheight} / ${actualheight} ) / 2 ) * 2 ))
ffmpeg -f image2 -framerate 1/5 -i tmp_image-%04d.jpg -c:v libx264 -r 30 \
-s ${finalwidth}x${finalheight} "$(basename "${presentation}" pdf)mp4"

```

4.4 Widget météo

Dans la zone « météo », nous avons ajouté deux Widget grâce à la fonctionnalité Embedded. Cette fonction permet d'embarquer du code HTML ou JavaScript dans le design de la page. Nous avons ajouté deux codes embarqués : un pour la météo et un pour la prévision des inondations. Voici l'exemple du code embarqué issu du site de Météo France :

```

<div id="vigiwrap">
<iframe id="vigiframe" src="http://www.vigicrues.gouv.fr/niv3grph.php? \
_jpg_csimd=1&CdStationHydro=W131001002 \
&typegraphe=h&AffProfondeur=168&tr0=W141001001&Aff \
Ref=auto&nbrstations=3&ong=1" \
width="649" height="400" frameborder="0">
<p>Votre navigateur est incompatible avec les iframes.</p>
</iframe>
</div>

```

L'ajustement des tailles lorsqu'on embarque du code d'un autre site web est assez complexe. Afin de régler au mieux les dimensions d'affichage des fenêtres, il est possible de jouer sur les paramètres CSS liés à l'attribut id de la balise div englobante.

5 Résultats

5.1 Affichage dans le hall d'accueil



Figure 6 - TV du hall



Figure 7 - Vue du hall d'entrée

5.2 Affichage dans le bureau des informaticiens

Le service informatique s'est lui-même doté d'un affichage sur un écran de PC classique. La mise en page est très simple. Elle nous permet de visualiser en alternance deux serveurs web de supervision du système d'information du LEGI :

- le serveur **Nagios** qui liste les plus de 1000 points de contrôles du système d'information du LEGI ;
- le serveur **OAR / Monika** qui donne l'état de l'ensemble des 528 cœurs des clusters du laboratoire.

Les possibilités sont extrêmement vastes et nous ne sommes qu'au début d'un nouvel usage.



Figure 8 - Nagios

5.3 Une communication améliorée

L'écran situé dans le hall d'entrée du laboratoire est très apprécié par les visiteurs du laboratoire. Il n'est pas rare de voir ces visiteurs observer longuement les vidéos de présentation des travaux de recherche.

La communication à destination des personnels est aussi un réel plus même si elle exige une mise à jour du contenu beaucoup plus soutenue.

6 Conclusion

Xibo est un produit complet qui répond très bien aux exigences du LEGI (solution open source, technologies fiables et éprouvées).

Xibo présente néanmoins quelques inconvénients : l'interface est assez complexe. Elle n'est pas facile à prendre en main. Pour le moment, seuls les informaticiens intègrent des modifications dans l'ordonnancement des contenus, en ajoutent ou en suppriment. Par ailleurs, il n'y a pas de client GNU/Linux (seulement Windows et Android) même s'il y a de nouveau un début de développement depuis septembre 2017.

Le laboratoire est très satisfait de la mise en place de Xibo tant sur l'aspect communication que maintenance.