Proyecto FIBtv. Integración de sistemas para difusión de vídeos públicos y privados

FIBtv Project. Integration of systems to broadcast public or private videos

Iván Fernández, Manel Rodero, Jaume Moral, Albert Obiols

Resumen

FIBtv es un proyecto que mejora la integración entre los diferentes sistemas de información y de audiovisuales de la Facultad de Informática de Barcelona (UPC), implicados en la reserva de salas de presentaciones y en la grabación y emisión de vídeos. También nos permite catalogar estos vídeos y difundirlos públicamente o de manera restringida a un grupo de miembros de la Facultad.

Palabras clave: streaming, WMS, CAS, integración.

Summary

FIBtv is a project that improves integration among information and audiovisual systems at Barcelona School of Informatics (UPC) used for booking lecture halls, video recording and stream broadcasting. It also allows to catalog these videos and to broadcast them to a general public or to a private audience from the Faculty.

Keywords: streaming, WMS, CAS, integration.

FIBtv es un proyecto que mejora la integración entre los diferentes sistemas de información y de audiovisuales de la Facultad de Informática de la UPC

1. Introducción

Dentro de las instalaciones de la Facultad se dispone de una sala de presentaciones, llamada sala de actos, en donde se realizan conferencias, clases y presentaciones. Durante la remodelación de esta sala se decidió equiparla con un equipo de vídeo y sonido para poder grabar los eventos y tenerlos accesibles para aquellas personas que hubieran podido presenciarlas en directo. Así mismo, se decidió que era necesario contar con un servicio capaz de distribuir las grabaciones de eventos anteriores y de poder transmitir el evento en directo a través de la red.

Queremos una administración descentralizada

2. Requisitos

Los requisitos funcionales de nuestro sistema son los siguientes, clasificados por temas.

1. **Catálogos:**
   - Los vídeos deben poder clasificarse por su contenido.
   - Queremos una administración descentralizada, donde los usuarios puedan no sólo visualizar un vídeo, sino también puedan añadir nuevos videos y también puedan realizar cambios en los datos de los vídeos, incluyendo los thumbnails.

2. **Emisiones en directo:**
   - Se debe poder preparar una emisión, proporcionando los datos del evento a emitir.
   - Las emisiones deben tener un grado de privacidad, asignando la emisión a un catálogo, para que únicamente las personas autorizadas puedan visualizar dicha emisión.
   - Se quiere que el sistema vaya grabando la emisión y en el momento de que ésta termine, pase a ser un vídeo del catálogo asociado.
   - El operador de la sala debe tener un sistema simple para arrancar y parar la emisión preparada.

3. **Privacidad:**
    - Necesitamos un sistema de privacidad de catálogos por IP donde únicamente la persona con una IP válida pueda visualizar el catálogo.
    - Queremos también un sistema de privacidad por el tipo de rol que desempeña la persona en la Facultad, para que cuando una persona se autentique, pueda visualizar el catálogo.

4. **Integración:**
    - Aprovechar la infraestructura de sistemas de información de la Facultad como son los siguientes sistemas:
        - CAS (Central Authentication Service)[1]: un servicio de single sign on que nos ayuda a validar e identificar el rol que tiene el usuario dentro de la Facultad.
        - Reservas de salas de presentaciones: un servicio que nos proporciona la información de los eventos que se realizarán en la sala de actos.
        - Repositorio institucional UPCommons[2]: un repositorio de contenidos digitales mantenido por la biblioteca al que interesa enviar los vídeos públicos.

El sistema se definió como una aplicación web

El sistema se definió como una aplicación web para que, a través de Internet, los usuarios puedan tener acceso fácil e inmediato a la información desde cualquier lugar. Además, la tecnología WWW nos ofrece facilidades para la integración con los sistemas de información comentados anteriormente.

Además se decidió utilizar la tecnología de streaming, que tiene la ventaja de permitir visualizar el vídeo desde una página web de una forma simple y transparente mientras éste se está descargando.

3. El Sistema FIBtv

La aplicación en sí se ha planteado como un proyecto final de máster del Máster en Tecnologías de la Información de la Facultad de Informática de Barcelona[3]. Para el desarrollo se ha utilizado el framework CakePHP[4], que utiliza el patrón de diseño MVC (Modelo Vista Controlador), asegurando así una aplicación fácilmente mantenible.

FIGURA 1

La aplicación se ha planteado como un proyecto final del Máster en Tecnologías de la Información de la UPC

Para el servidor de vídeo, se ha escogido Windows Media Services (WMS), ya que nos permitía desarrollar plugins para la autenticación y autorización de los usuarios, necesarios para nuestros objetivos. Estas librerías se han programado en C#.

Otro elemento desarrollado en este proyecto son los scripts de control de la estación codificadora de vídeo, que ejecuta Windows Media Encoder, programados utilizando AutoIt[5]. Tal y como vemos en este esquema, hay una gran interacción entre los diferentes sistemas, para los que se han utilizado protocolos como HTTP, SCP y formatos como ICAL y XML.

4. Vídeos privados

Una de las funcionalidades diferenciadoras de nuestro sistema es la posibilidad de tener catálogos con acceso restringido y vídeos privados. Para implementar esta funcionalidad, lo hemos integrado con el sistema de single sign on CAS.

Funcionamiento básico de CAS

El sistema CAS basa su funcionamiento en la utilización de tickets de un solo uso para servicios. La idea básica es que si un usuario quiere acceder a un servicio protegido, debemos pedir al servidor CAS un ticket para este servicio y usuario. Posteriormente, el servicio protegido puede obtener el usuario que accede validando el ticket en el servidor CAS. Estos tickets se entregan previa validación del usuario, usualmente con username y password.

El sistema CAS basa su funcionamiento en la utilización de tickets de un solo uso para servicios

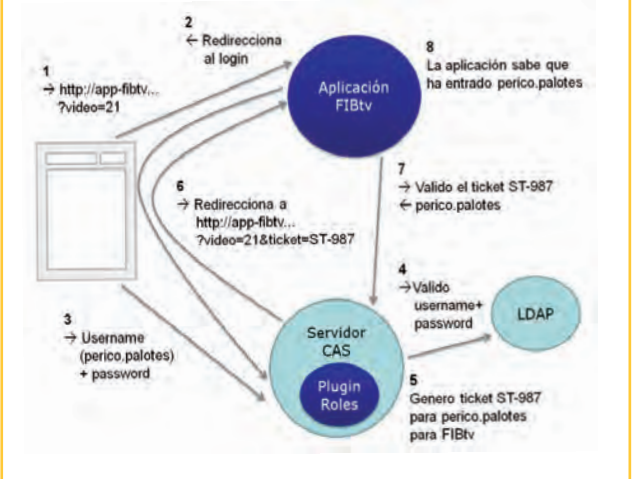
En el esquema que sigue a continuación se muestra la interacción con CAS para hacer el login en la aplicación FIBtv:

1. Intentamos entrar en la aplicación
2. Como no nos hemos autenticado previamente, la aplicación nos redirecciona al servidor CAS para poder hacer el login. Esta redirección incluye el servicio al cual hemos realizado la petición inicial, para que así sepa la URL a la cual tiene que volver
3. Enviamos nuestro username (perico.palotes) y password al servidor CAS
4. El servidor valida nuestras credenciales contra el servicio de directorio de la UPC (LDAP)
5. El servidor genera un ticket de un solo uso para el servicio FIBtv y el usuario perico.palotes
6. El servidor nos redirecciona a la URL inicial de la que proveníamos, añadiendo un parámetro con el ticket. Al mismo tiempo, envía una cookie que nos identifica en el servidor CAS y que evitará que volvamos a tener que identificarnos con username y password cuando accedamos a otra aplicación protegida con CAS.

Con este procedimiento no tenemos que volver a identificarnos con username y password

7. La aplicación FIBtv comprueba que estamos accediendo a la aplicación con un ticket y procede a validarlo en el servidor CAS, obteniendo el username de la persona que está accediendo.

FIGURA 2



8. La aplicación ya sabe quien ha entrado en la aplicación.

Utilización de CAS para proteger el *stream* de vídeo

Para poder proteger videos utilizando CAS, se ha desarrollado un plugin de autenticación y autorización para Windows Media Services que se asocia a un punto de publicación. Este plugin será el que gestione toda la interacción con CAS para saber el usuario que pide los vídeos (la parte de autenticación) y la interacción con el webservice de FIBtv que permite saber si el usuario tiene acceso al vídeo que pide.

Veamos cómo funciona esta interacción mediante un esquema:

1. Nos conectamos a la aplicación para pedir un vídeo concreto (el vídeo 21, que está dentro de un catálogo solo disponible para estudiantes de la asignatura AC). Supondremos que ya hemos hecho login.

2. La aplicación obtiene los roles del usuario. Estos roles se obtienen de un webservice que hemos integrado con CAS y que va a buscar la información a la base de datos de nuestra intranet. En base a estos roles (EST y EST-AC), la aplicación decide que el usuario tiene acceso al vídeo.

3. La aplicación pide un ticket para el usuario perico.palotes para el servicio WMS. Aquí utilizamos el CAS en modo Proxy, que permite que una aplicación intermedia (FIBtv en este caso) pida un ticket en nombre de un usuario (perico.palotes) para acceder a otro servicio protegido.

4. La aplicación genera una página HTML de resultado con el vídeo 21 incrustado. En la URL de este vídeo se le añade el parámetro ticket para que WMS pueda hacer las validaciones pertinentes.

5. Pedimos la URL de streaming, con el parámetro ticket, a WMS.

6. El plugin de autenticación de WMS valida el ticket en CAS y obtiene el usuario.

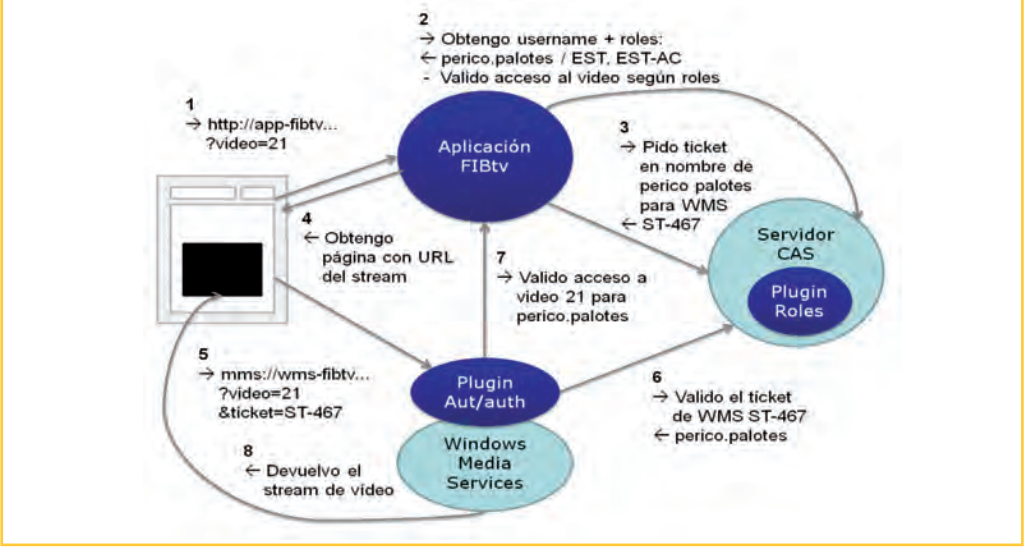
7. El plugin de autorización de WMS valida en un webservice de la aplicación FIBtv que el usuario tenga acceso al vídeo 21.

8. Si todo es correcto, WMS empieza a enviar el video.

Se ha desarrollado un plugin de autenticación y autorización para Windows Media Services

Pedimos la URL de streaming, con el parámetro ticket, a WMS

FIGURA 3



Hay que destacar que este sistema funciona tanto para vídeos grabados como para vídeos en directo, ya que está implementado a nivel de punto de publicación en WMS.

4. Conclusiones

FIBTV es un sistema desarrollado a medida para la Facultad, el cual lleva un control del ciclo de vida de la grabación de un vídeo, ofrece facilidades de uso para el operador y otros usuarios y proporciona seguridad de los vídeos en función de los perfiles del usuario.

FIBTV es un sistema desarrollado a medida para la Facultad

Referencias

[1] *http://www.jasig.org/cas*

[2] *http://upcommons.upc.edu/*

[3] *http://upcommons.upc.edu/pfc/handle/2099.1/7736*

[4] *http://cakephp.org/*

[5] *http://www.autoitscript.com/*

Iván Fernández
(ivan.zwanziger@gmail.com)
Manel Rodero
(manel@fib.upc.edu)
Jaume Moral
(jaumem@fib.upc.edu)
Albert Obiols
(albert@fib.upc.edu)
Facultad de Informática de Barcelona
Universidad Politécnica de Cataluña