



**Barcelona  
Supercomputing  
Center**  
*Centro Nacional de Supercomputación*

A central image showing a classical building with a pediment and columns, possibly a university or government building, set against a light sky. The image is framed by a dotted border and is the focal point of the slide.

## **Autenticación centralizada mediante CAS y federación de servicios**

Córdoba, 18 de noviembre de 2010

Juan C. Sánchez – Jordi Valls  
Departamento de Operaciones  
BSC



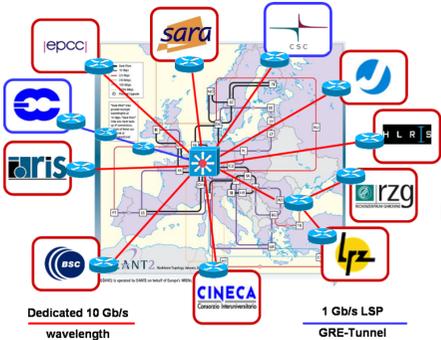
## Contenido

- Motivación y objetivos.
- Selección de la plataforma.
- Especificación de la SAA.
- Visión global de la SAA.
- Despliegue del servidor CAS.
- Casificación de servicios.
- Caso de uso: casificación de Symfony.
- Casificación de servicios mediante proxy.
- Caso de uso: casificación de Webmail.
- Federación.



# Motivación y objetivos [ 1 / 2 ]

- Crecimiento exponencial de usuarios desde 2005.  
( 20 → 350 usuarios activos)
- Red Española de Supercomputación.  
( 80 → 750 usuarios activos)
- Colaboración con otros centros de investigación.





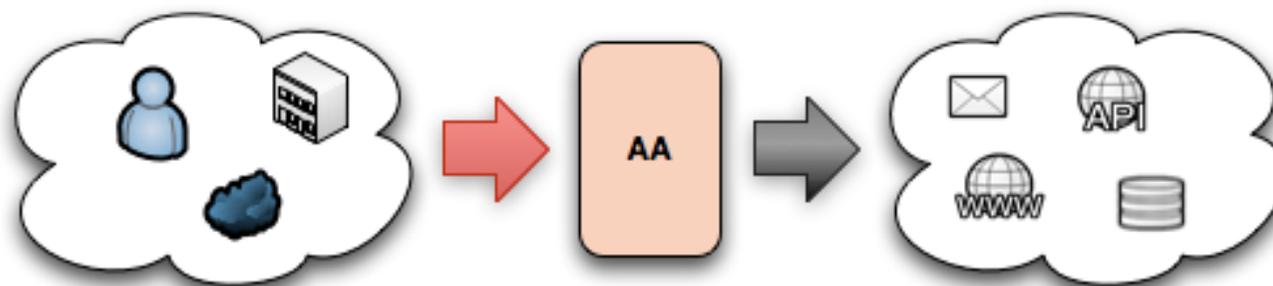
## Motivación y objetivos [ 2 / 2 ]

- Ofrecer un acceso único a servicios de Autenticación y Autorización.
- Ofrecer servicios a otras entidades con las que se colabora (RES, DEISA, PRACE, HPC-e, sector privado).
- Proveer de un marco de seguridad común para todos los servicios.



## Motivación y objetivos [ 2 / 2 ]

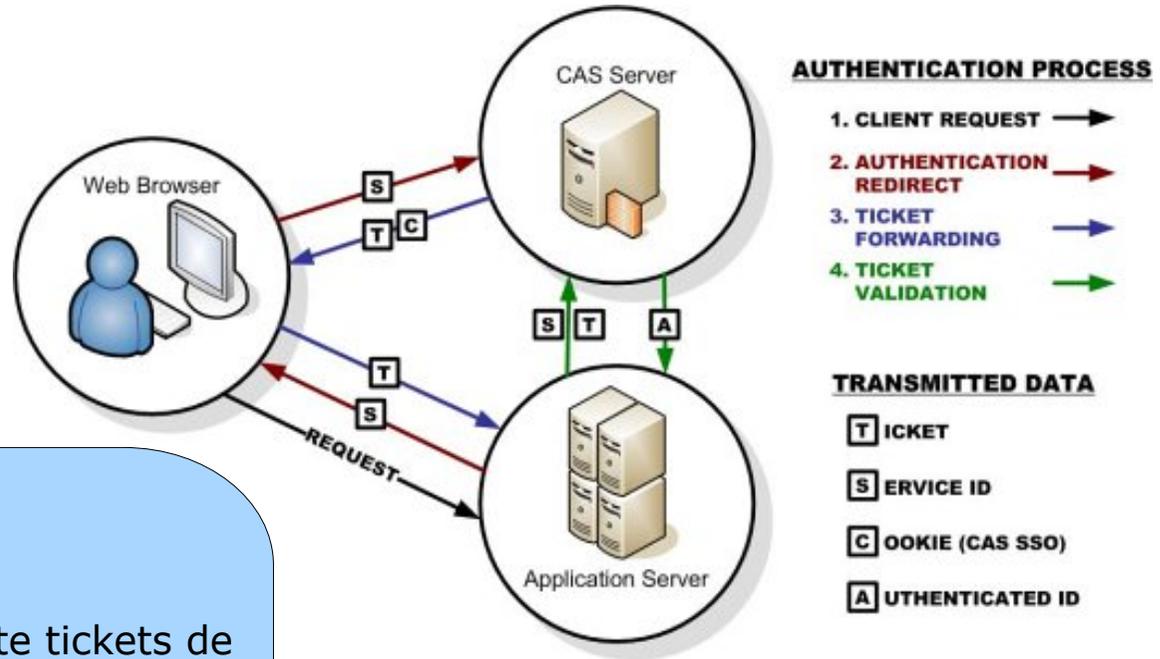
- Ofrecer un acceso único a servicios de Autenticación y Autorización.
  - Ofrecer servicios a otras entidades con las que se colabora (RES, DEISA, PRACE, HPC-e, sector privado).
  - Proveer de un marco de seguridad común para todos los servicios.
- ➔
- Integrar los mecanismos AA en una única arquitectura.
  - Crear una capa de servicios AA (SAA).
  - Proveer servicios que se integren en la SAA y facilitar el know-how.
  - Federar la SAA para integrar los servicios con la comunidad.



## Selección de la plataforma [ 1 / 2 ]

### ¿ Por qué CAS ?

- Control de sesiones mediante tickets de uso único.
- Single Sign-On.
- Permite autenticar contra múltiples back-ends.
- ...





## Selección de la plataforma [ 2 / 2 ]

Plataforma seleccionada: **Ja-Sig CAS**  
[ <http://www.jasig.org/cas> ]



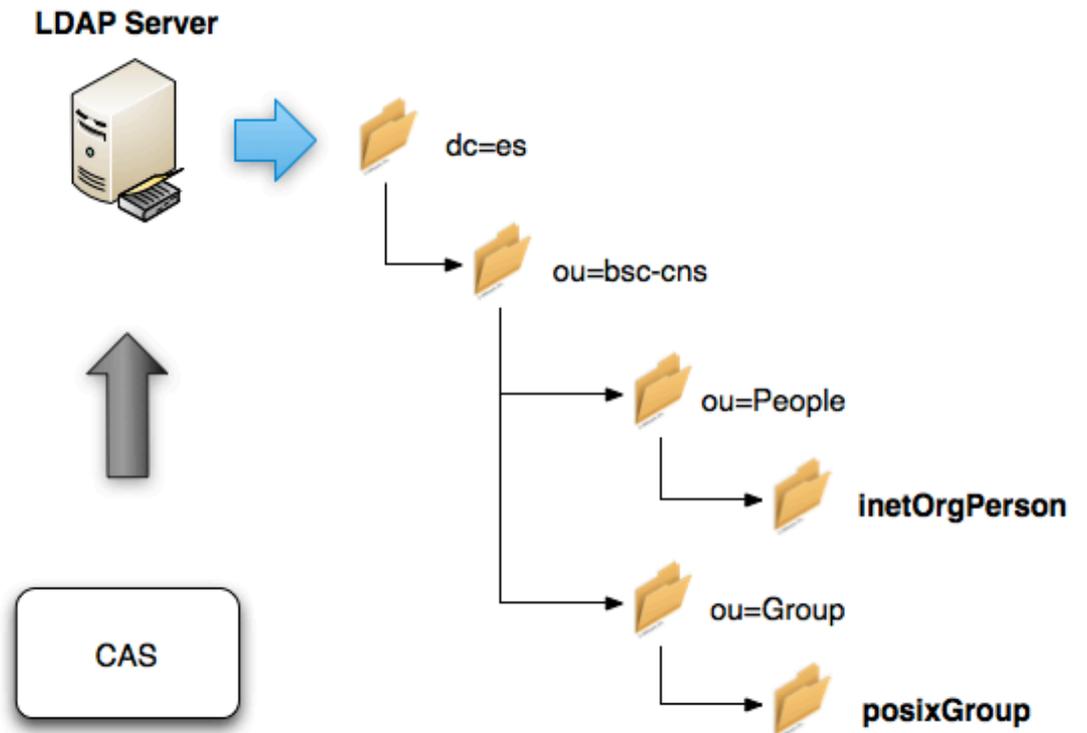
### ¿ Por qué Ja-Sig CAS ?

- Plataforma abierta y moderna.
- Flexible, escalable y seguro.
- Compatible con los más conocidos protocolos de federación.
- Comunidad de desarrollo muy activa.

# Especificación de la SAA [ 1 / 3 ]

## Escenario 1: Usuarios locales

- CAS se provee de la información contenida en un directorio LDAP centralizado.
- La autenticación se realiza interactivamente mediante una aplicación web (login y password).
- CAS también obtiene todas las credenciales necesarias (grupos, roles, etc.) del directorio LDAP.

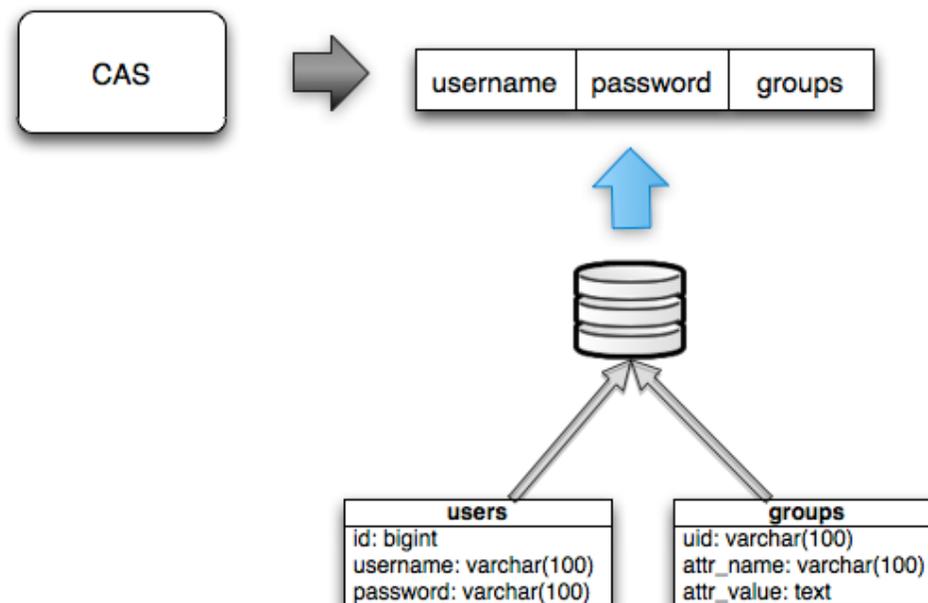




## Especificación de la SAA [ 2 / 3 ]

### Escenario 2: Usuarios invitados

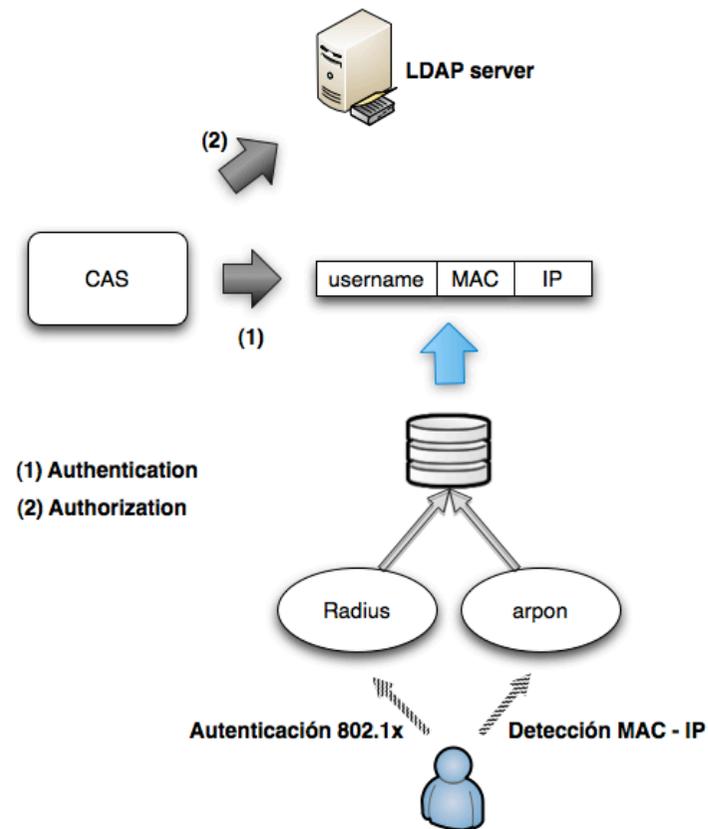
- Escenario introducido por la necesidad de incorporar usuarios externos a la organización (similar a AESIR).
- CAS se alimenta de una base de datos independiente para autenticar a los usuarios.
- La autenticación se realiza mediante el mismo mecanismo que en el *Escenario 1* (login y password).



## Especificación de la SAA [ 3 / 3 ]

### Escenario 3: Red 802.1x

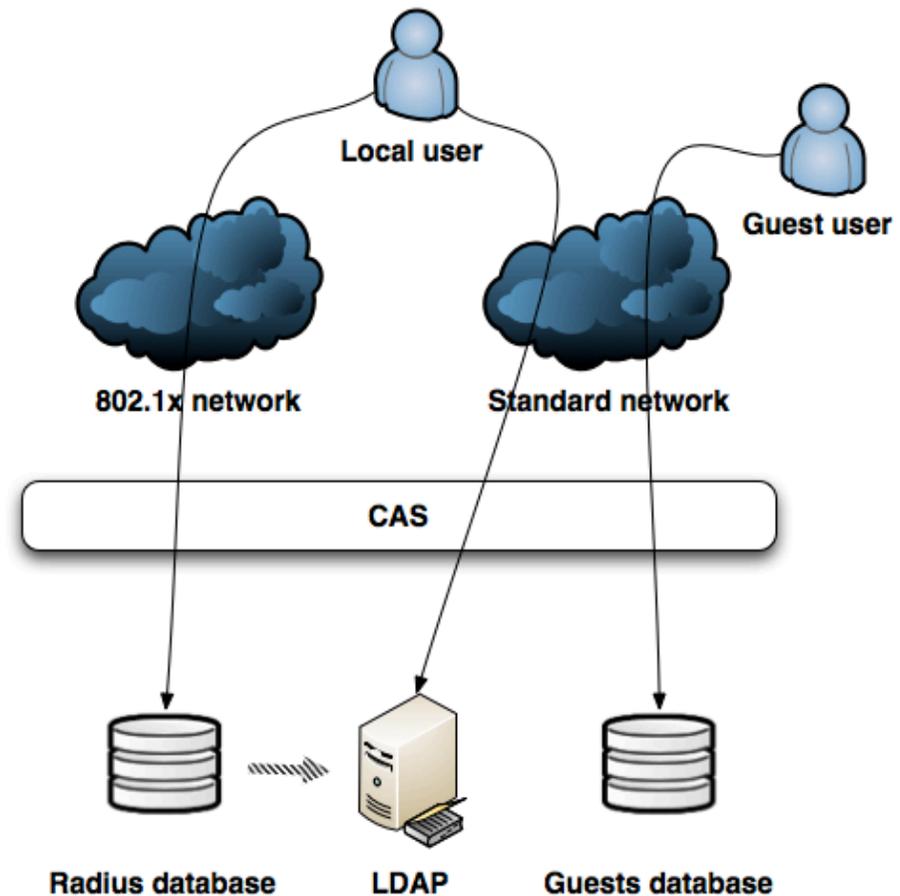
- CAS se alimenta de la información proporcionada por la autenticación 802.1x.
- 802.1x nos proporciona la relación entre usuario y MAC de acceso.
- Mediante el uso de 'arpon' se correla la dirección IP con estos datos.
- En conjunto, en el momento en que el usuario accede a CAS desde una IP podemos saber si está autenticado. Sino se pasa al *Escenario 1*.



### En producción: red Wi-Fi

## Visión global de la SAA

- Radius proporciona autenticación de usuarios en redes 802.1x.
- LDAP proporciona autorización de usuarios en redes 802.1x y autenticación y autorización de usuarios en redes estándar.
- La base de datos CAS permite autenticar usuarios externos a la organización.





## Contenido

- Motivación y objetivos.
- Selección de la plataforma.
- Especificación de la SAA.
- Visión global de la SAA.
- Despliegue del servidor CAS.
- Casificación de servicios.
- Caso de uso: casificación de Symfony.
- Casificación de servicios mediante proxy.
- Caso de uso: casificación de Webmail.
- Federación.



## Despliegue del servidor CAS

- Instalación software (Tomcat 5)
- Configuración (mysql, certificados, firewall).
- Ampliaciones
  - Configuración de los 3 back-ends.
  - Personalización de la ventana de login.
  - Integración con 802.1x.

The screenshot shows the CAS project page on the jasig website. The page has a green header with the jasig logo and navigation links: Home, Projects, Community, Contribute, About. A left sidebar menu lists various categories like uPortal, CAS, Community, Download, etc. The main content area is titled 'Barcelona Supercomputing Center' and contains the following information:

- Institution:** Barcelona Supercomputing Center, Centro Nacional de Supercomputación.
- Deployment URL:** <https://www.bsc.es/cas>
- Number of Users:** 300
- Implementation:**
  - In Production:** Yes
  - Version:** 3.3.5
  - Last Updated:** 26 May 2010
  - Authentication:** OpenLDAP
  - App server:** Tomcat 5
  - DB server:** MySQL

## Casificación de servicios [ 1 / 2 ]

¿ Cómo integrar un servicio con CAS ?



- La validación interactiva debe ser delegada a CAS.
- La validación de un usuario se realiza a través del ticket.
- ¿ Cómo casificar servicios no basados en web ?



**Actualizar los módulos de autenticación para soportar el protocolo CAS en base a los clientes de la comunidad Ja-Sig**



## Casificación de servicios [ 2 / 2 ]

Dashboard › CAS Clients › Home



Added by [Scott Battaglia](#), last edited by [Scott Battaglia](#) on Aug 03, 2009 ([view change](#))

### Table of Contents

- ▣ [Official Clients](#)
  - ▣ [Acegi as CAS Client](#)
  - ▣ [CAS Client for Java 3.0](#)
  - ▣ [CAS Client for Java 3.1](#)
  - ▣ [mod\\_auth\\_cas](#)
  - ▣ [phpCAS](#)
- ▣ [Legacy Clients](#)
  - ▣ [CAS and JSR-168](#)
  - ▣ [ISAPI Filter](#)
  - ▣ [JSP Client](#)
  - ▣ [MOD\\_CAS \(Deprecated\)](#)
  - ▣ [PAM Module](#)
  - ▣ [Yale CAS client distribution](#)
  - ▣ [Yale Java Client](#)
- ▣ [Incubating Clients](#)
- ▣ [Unofficial CAS Clients](#)
  - ▣ [.Net Http module](#)
  - ▣ [ASP.NET Forms Authentication](#)

As of May 15th, we've reorganized the clients into various groups. Those groups are:

### Official Clients

Official Clients are generally being actively maintained by the client and can assist you with...

### Legacy Clients

Legacy Clients are those that are not actively being developed but are still supported by the Jasig CAS Client team. You can contact us with these...

### Incubating Clients

Incubating Clients are those that are...

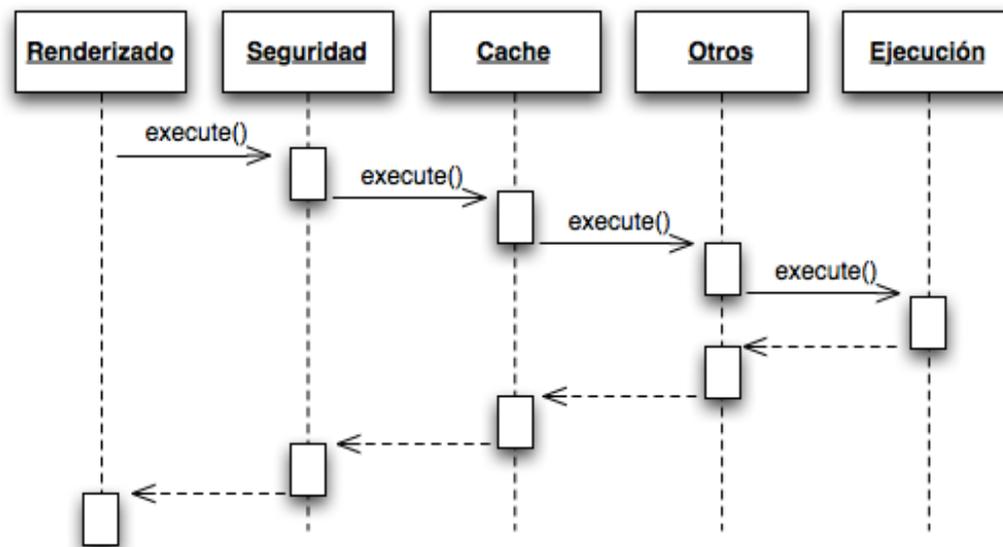
- Servicios web
  - `mod_auth_cas`
  - `phpCas`
  
- Servicios no web
  - `pam_cas`



## Caso de uso: Symfony [ 1 / 2 ]

### Symfony es un framework para el desarrollo de aplicaciones PHP

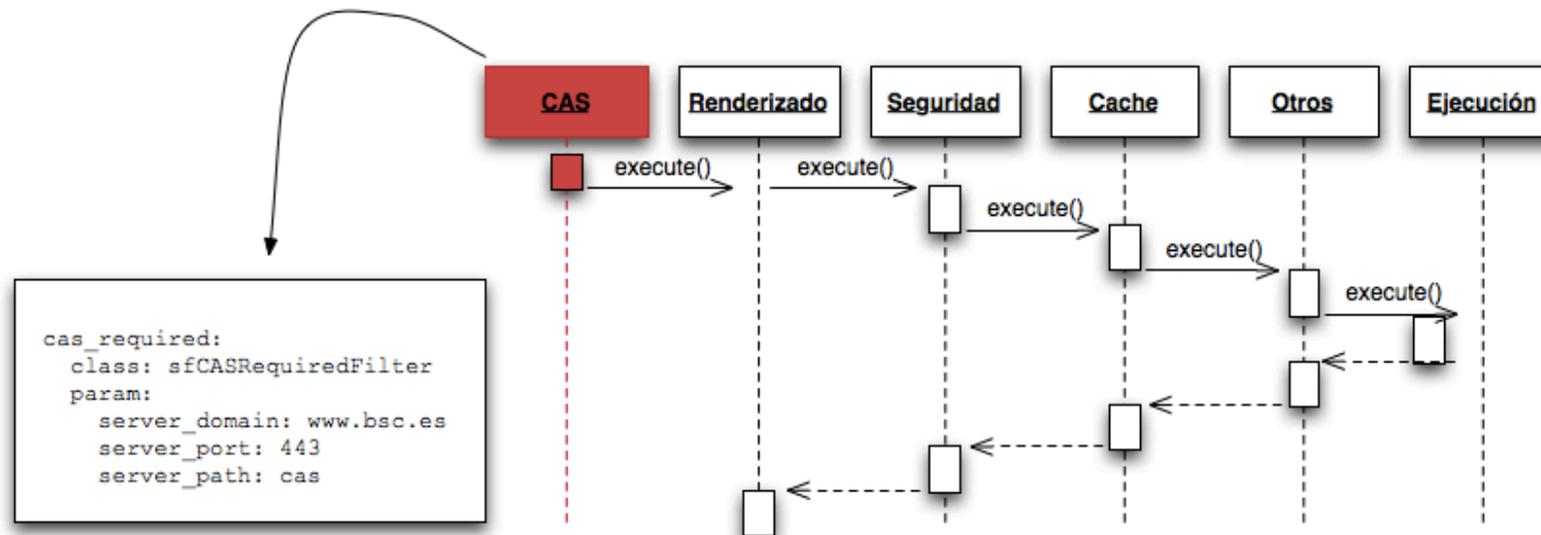
- Implementa el paradigma Modelo-Vista-Controlador.
- Permite el desarrollo de plugins.
- Incorpora un sistema de filtros encadenados que se ejecutan en un orden determinado al recibir una petición por parte de un cliente.



## Caso de uso: Symfony [ 1 / 2 ]

### Symfony es un framework para el desarrollo de aplicaciones PHP

- Implementa el paradigma Modelo-Vista-Controlador.
- Permite el desarrollo de plugins.
- Incorpora un sistema de filtros encadenados que se ejecutan en un orden determinado al recibir una petición por parte de un cliente.



## Caso de uso: Symfony [ 2 / 2 ]

```
<?php
class sfCASRequiredFilter extends sfBasicSecurityFilter
{
    public function execute ($filterChain)
    {
        if ($this->isFirstCall() && ! $this->getContext()->getUser()->isAuthenticated()) {
            require_once('CAS.php');

            // Get parameters and create phpCAS instance
            $domain = $this->getParameter('server_domain');
            $port = $this->getParameter('server_port');
            $path = $this->getParameter('server_path')
            phpCAS::client(SAML_VERSION_1_1, $domain, $port, $path);

            // Authenticate user
            phpCAS::forceAuthentication();

            // User is valid
            $this->getContext()->getUser()->setAuthenticated(true);
            $this->getContext()->getUser()->setAttribute('username', phpCAS::getUser(), 'cas');
            $this->getContext()->getUser()->addCredential(phpCAS::getUser());

            // [ BSC ] Add group credentials
            // Last attribute contains user groups
            $groups = split(",", end( phpCAS::getAttributes() ));
            foreach ($groups as $value) {
                $this->getContext()->getUser()->addCredential($value);
            }
        }

        ...

        // Execute next filter in the chain
        $filterChain->execute();
    }
}
?>
```

## Caso de uso: Symfony [ 2 / 2 ]

```
<?php
class sfCASRequiredFilter extends sfBasicSecurityFilter
{
    public function execute ($filterChain)
    {
        if ($this->isFirstCall() && ! $this->getContext()->getUser()->isAuthenticated()) {
            require_once('CAS.php');

            // Get parameters and create phpCAS instance
            $domain = $this->getParameter('server_domain');
            $port = $this->getParameter('server_port');
            $path = $this->getParameter('server_path')
            phpCAS::client(SAML_VERSION_1_1, $domain, $port, $path);

            // Authenticate user
            phpCAS::forceAuthentication();

            // User is valid
            $this->getContext()->getUser()->setAuthenticated(true);
            $this->getContext()->getUser()->setAttribute('username', phpCAS::getUser(), 'cas');
            $this->getContext()->getUser()->addCredential(phpCAS::getUser());

            // [ BSC ] Add group credentials
            // Last attribute contains user groups
            $groups = split(",", end( phpCAS::getAttributes() ));
            foreach ($groups as $value) {
                $this->getContext()->getUser()->addCredential($value);
            }
        }

        ...

        // Execute next filter in the chain
        $filterChain->execute();
    }
}
?>
```

## Caso de uso: Symfony [ 2 / 2 ]

```
<?php
class sfCASRequiredFilter extends sfBasicSecurityFilter
{
    public function execute ($filterChain)
    {
        if ($this->isFirstCall() && ! $this->getContext()->getUser()->isAuthenticated()) {
            require_once('CAS.php');

            // Get parameters and create phpCAS instance
            $domain = $this->getParameter('server_domain');
            $port = $this->getParameter('server_port');
            $path = $this->getParameter('server_path');
            phpCAS::client(SAML_VERSION_1_1, $domain, $port, $path);

            // Authenticate user
            phpCAS::forceAuthentication();

            // User is valid
            $this->getContext()->getUser()->setAuthenticated(true);
            $this->getContext()->getUser()->setAttribute('username', phpCAS::getUser(), 'cas');
            $this->getContext()->getUser()->addCredential(phpCAS::getUser());

            // [ BSC ] Add group credentials
            // Last attribute contains user groups
            $groups = split(",", end( phpCAS::getAttributes() ));
            foreach ($groups as $value) {
                $this->getContext()->getUser()->addCredential($value);
            }
        }

        ...

        // Execute next filter in the chain
        $filterChain->execute();
    }
}
?>
```

## Caso de uso: Symfony [ 2 / 2 ]

```
<?php
class sfCASRequiredFilter extends sfBasicSecurityFilter
{
    public function execute ($filterChain)
    {
        if ($this->isFirstCall() && ! $this->getContext()->getUser()->isAuthenticated()) {
            require_once('CAS.php');

            // Get parameters and create phpCAS instance
            $domain = $this->getParameter('server_domain');
            $port = $this->getParameter('server_port');
            $path = $this->getParameter('server_path')
            phpCAS::client(SAML_VERSION_1_1, $domain, $port, $path);

            // Authenticate user
            phpCAS::forceAuthentication();

            // User is valid
            $this->getContext()->getUser()->setAuthenticated(true);
            $this->getContext()->getUser()->setAttribute('username', phpCAS::getUser(), 'cas');
            $this->getContext()->getUser()->addCredential(phpCAS::getUser());

            // [ BSC ] Add group credentials
            // Last attribute contains user groups
            $groups = split(",", end( phpCAS::getAttributes() ));
            foreach ($groups as $value) {
                $this->getContext()->getUser()->addCredential($value);
            }
        }

        ...

        // Execute next filter in the chain
        $filterChain->execute();
    }
}
?>
```

## Caso de uso: Symfony [ 2 / 2 ]

```
<?php
class sfCASRequiredFilter extends sfBasicSecurityFilter
{
    public function execute ($filterChain)
    {
        if ($this->isFirstCall() && ! $this->getContext()->getUser()->isAuthenticated()) {
            require_once('CAS.php');

            // Get parameters and create phpCAS instance
            $domain = $this->getParameter('server_domain');
            $port = $this->getParameter('server_port');
            $path = $this->getParameter('server_path')
            phpCAS::client(SAML_VERSION_1_1, $domain, $port, $path);

            // Authenticate user
            phpCAS::forceAuthentication();

            // User is valid
            $this->getContext()->getUser()->setAuthenticated(true);
            $this->getContext()->getUser()->setAttribute('username', phpCAS::getUser(), 'cas');
            $this->getContext()->getUser()->addCredential(phpCAS::getUser());

            // [ BSC ] Add group credentials
            // Last attribute contains user groups
            $groups = split(",", end( phpCAS::getAttributes() ));
            foreach ($groups as $value) {
                $this->getContext()->getUser()->addCredential($value);
            }
        }

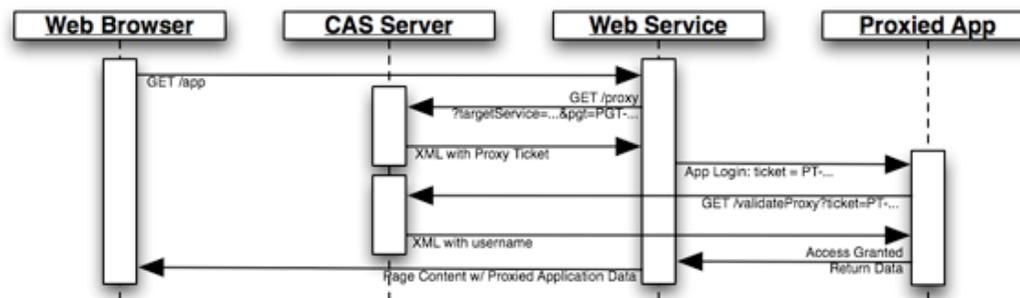
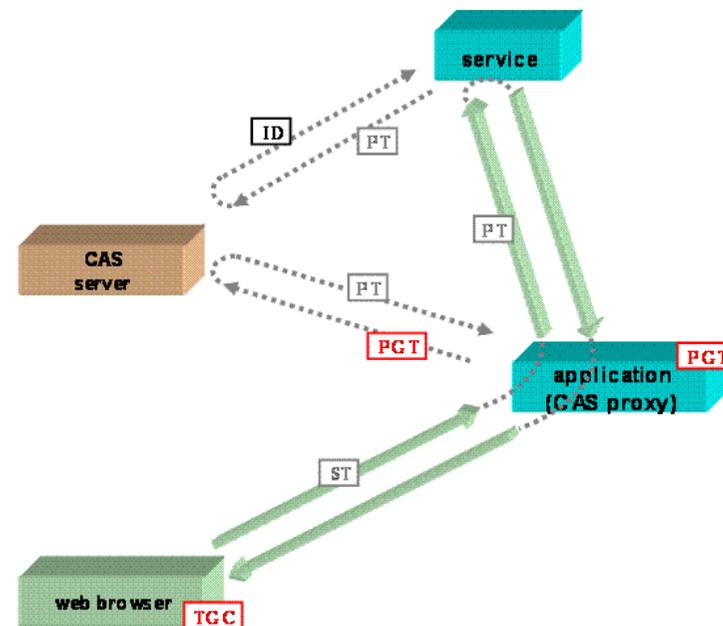
        ...

        // Execute next filter in the chain
        $filterChain->execute();
    }
}
?>
```



## Casificación de servicios mediante proxy

- Las aplicaciones N-Tier no están contempladas en la definición básica de CAS.
- Desde la versión 2.0, CAS soporta autenticación proxy, mediante la definición de Proxy Tickets (PT) y Proxy Granted Tickets (PGT).
- En combinación con el módulo pam\_cas, esto nos permite casificar cualquier servicio que tenga un interfaz web (Navegadores de ficheros, Webmail, etc.).





## Caso de uso: Webmail (Dovecot + RoundCube)

- El servidor IMAP debe soportar autenticación via PAM. En caso de no ser así, debe recompilarse con soporte para PAM.
- Se instala `pam_cas` y se configura el servicio IMAP a tal efecto.

```
/etc/dovecot.conf  
passdb pam {  
    args = dovecot  
}
```

```
/etc/pam.d/dovecot  
auth required pam_cas.so -simaps://mail.bsc.es -f/etc/pam_cas.conf  
account sufficient pam_permit.so
```

```
/etc/pam_cas.conf  
host www.bsc.es  
port 443  
uriValidate /cas/proxyValidate  
ssl on
```

- El cliente webmail también debe estar casificado. En este caso existe plugin `rc_cas` a tal efecto.
- [ <http://code.google.com/p/rc-cas-plugin/> ]

## Federación [ 1 / 2 ]

- Gracias a la utilización de CAS, BSC-CNS se ha convertido en proveedor de identidad para la red federada SIR.

English | Español

Red IRIS

### Servicio de Identidad digital - RedIRIS

**Seleccione su proveedor de identidad**

Seleccione uno de los siguientes proveedores de identidad digital:

- AESIR
- B.C.B.L.
- B.O.E.
- BSC-CNS**
- Biblioteca de Catalunya
- C.E.S.G.A.
- C.I.C.A.
- C.N.I.O.
- CIEMAT

**BSC-CNS**

Ha seleccionado el proveedor de identidad de la institución **BSC-CNS** en el cual debe tener una cuenta de usuario válida.

Una vez se haya identificado correctamente en dicho proveedor, se comprobarán los atributos asociados en su cuenta que decidirán si tiene derecho a acceder al recurso.

Si se encuentra en un ordenador privado, puede marcar la siguiente casilla para que se elija automáticamente esta institución en futuros accesos a recursos protegidos por el Servicio de identidad digital de RedIRIS.

Elegir siempre este proveedor

CANCELAR      ACEPTAR

Done

# Federación [ 2 / 2 ]

- El siguiente paso consiste en federar diferentes aplicaciones desarrolladas dentro del BSC-CNS para que puedan ser usadas por el resto de la comunidad.

## Área de gestión interna

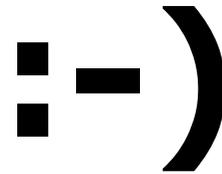
The screenshot shows the 'BSC-CNS - SupPerDB WEB' interface. At the top, there is a navigation bar with tabs for 'People', 'Assignments', 'Software', 'Accounting', 'Mail', and 'Info/Week'. The 'Accounting' tab is active. Below the navigation bar, there are several input fields and dropdown menus for configuring an analysis. The 'Machines' dropdown is set to 'MareNostrum'. The 'Time interval' is set to 'Weekly (4 weeks)'. The 'Interval Resolution' is set to 'Monthly (4 weeks)'. Below these fields is a 'Generate Analysis' button. The main content area displays an 'Accounting Report for bsc99 from 2010-11-05 to 2010-11-12'. The report includes a 'General Stats' section and an 'Accounting Summary' section. The summary shows: 'MareNostrum: Total hours consumed: 2.19 KWh, Current Projects usage: 3307.01 GB, Current Events usage: 3750.31 GB'. On the left side, there is a sidebar with details for the 'UnixGroup: bsc99', including 'Activity Type: Internal', 'Area: OP', 'Title: Grupo de operaciones', 'Leader: Sergi Girona', and 'Leader Mail: sergi.girona@bsc.es'.

## Intranet de usuarios

The screenshot shows the 'BSC Administration Area' user intranet. At the top, there is a welcome message: 'Welcome BSC Support Department to the RES-Red Española de Supercomputación BSC Administration Area'. Below this, there are buttons for 'Change Password' and 'Logout'. A search bar is present with the text 'Enter Unix Group or ActivityId:' and a 'SEARCH' button. Below the search bar, there is a section titled 'USAGE OF COLORS ON THIS PAGE ?' with a link to 'HPC-E users'. The main content area is titled 'Current Period Applications and Activities: 2010-3 (2010, November 1st - 2011, February 28th)'. It lists several applications with their details, including 'Application', 'Users (Researchers: X)', 'Info', 'Reports', 'CPU Usage', and 'Dissemination'. The applications listed are: AECT-2010-3-0001: BRIDGE Sustainable urban planning Decision support accounting for urban metabolism (Unix Group: upm39); AECT-2010-3-0002: Gala: Simulation of Telemetry Stream (Unix Group: ub32); AECT-2010-3-0003: The MareNostrum Numerical Cosmology Project: Grand Challenge simulations of structure formation in the Universe (Unix Group: uam37); AECT-2010-3-0008: IBERREF: A 3D seismic reference model for the Iberian lithosphere (Unix Group: ugr33); AECT-2010-3-0009: Magneto-convection and wave simulations of solar and stellar atmospheres (Unix Group: iac44); AECT-2010-3-0010: Coalescence of Black Hole Binary systems (Unix Group: ub68); AECT-2010-3-0011: Eruptive phenomena in the atmosphere of the Sun and cool stars. (Unix Group: iac04); AECT-2010-3-0012: On the stellar wind-jet interaction in high-mass microquasars (Unix Group: uv85); AECT-2010-3-0013: Multidimensional simulation of stellar explosions: neon-rich and primordial novae, and type I X-ray bursts (Unix Group: upc28); AECT-2010-3-0014: Assessment of the limit of initial-condition useful skill in Interannual climate prediction (Unix Group: ecm86).



¡ GRACIAS !



**Barcelona  
Supercomputing  
Center**

*Centro Nacional de Supercomputación*