Generalitat de Catalunya
Centre de Telecomunicacions
i Tecnologies de la Informació

# Framework Corporatiu J2EE

# Applet de signatura digital

**Versió 1.0.3**

Barcelona, 17 / gener / 2007

# Històric de modificacions

| Data | Autor | Comentaris | Versió |
|---|---|---|---|
| 17/01/2007 | Atos Origin, sae | Versió inicial del document | 1.0.3 |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

**Llegenda de Marcadors**

# Índex

# 1. Introducció

## 1.1. Propòsit

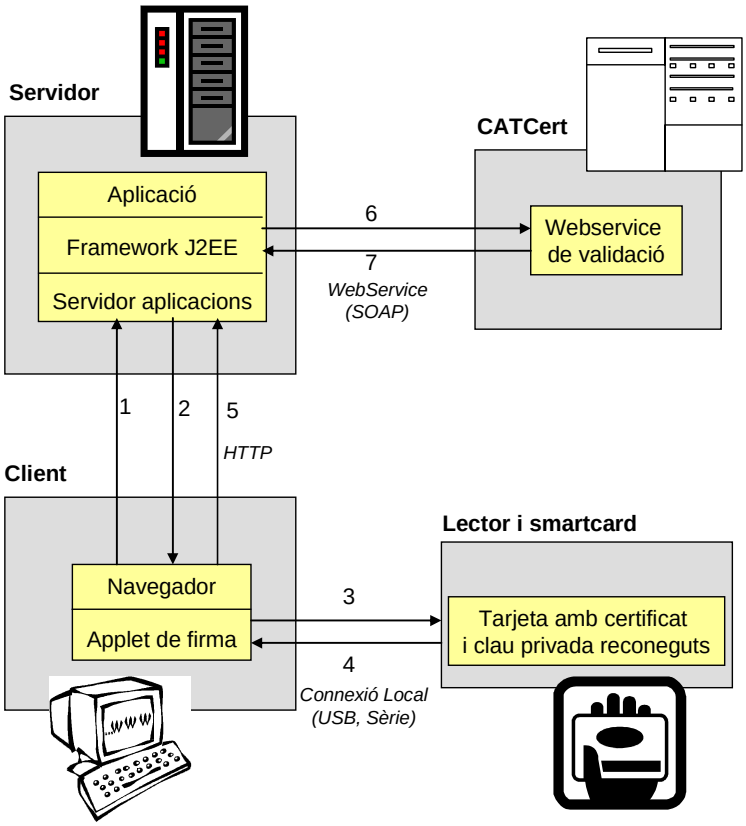
L'applet de signatura digital permet signar documents en certificats en targeta. La base tecnològica està basada en un Applet i la interfície d'accés a targeta PKCS11.

Les funcionalitats de l'Applet són:

1. Signar documents
2. Configurar la targeta a accedir.
3. Configurar servidor de temps.

## 1.2. Context i Escenaris d'Ús

A continuació s'indica el flux de treball per a realitzar la signatura d'un document amb l'applet i la posterior validació de la signatura:




1. El client sol·licita iniciar el procés de firma.

2. L'aplicació retorna la pagina que permet al client carregar el fitxer local que vol firmar i cridar l'applet.
3. L'applet de firma calcula el resum (digest) del fitxer i l'envia a la Smartcard
4. La smartcard fa el xifrat amb la clau privada i ho retorna a l'applet.
5. L'applet construeix l'estructura de firma pkcs7 i l'envia junt amb el fitxer al servidor.
6. L'aplicació del servidor, a través del framework, sol·licita al Catcert que validi la firma (enviant la firma i el fitxer).
7. El Catcert fa la validació i retorna el resultat a l'aplicació.

Per tant l'applet realitza sols el procés de signatura del document. Si es vol validar una signatura ja feta, s'ha d'utilitzar un WebService del CATCert. Actualment existeix un connector dins del framework J2EE corporatiu que ens permet connectar amb aquest servei del CATCert. Veure el document ***openFrame_Connectors_Catcert_Cat.doc*** dins de la documentació de connectors del Framework per saber com utilitzar aquest servei de validació.

> ***Important***
> La targeta ha de contenir un certificat emès per una CA reconeguda pel CATCert.

## 1.3. Versions i Dependències

En el present apartat es mostren quines són les versions i dependències necessàries per fer ús del Servei.

En la versió 1.0.3 s'han fet les següents modificacions:

- Es permet configurar el proveïdor de targeta que es desitgi.
- S'han desactivat les validacions de certificats
- Els certificats que s'obtenen de la targeta no és filtren per a cap identificador.

### *1.3.1. Dependències Bàsiques*

| Nom | Tipus | Versió | Descripció |
|---|---|---|---|
| bouncycastle-bcmail | jar | jdk15-1.3.0 | http://www.bouncycastle.org |
| bouncycastle-bcprov | jar | jdk15-1.3.0 | http://www.bouncycastle.org |
| plugin | jar | | Llibreria del jre de jdk1.5 |

## 1.4. A qui va dirigit

Aquest document va dirigit als següents perfils:

- Programador. Per a conèixer l'ús del servei
- Arquitecte. Per a conèixer quins són els components i la configuració del servei
- Administrador. Per a conèixer com configurar el servei en cadascun dels entorns en cas de necessitat

## 1.5. Documents i Fonts de Referència

[1] BouncyCastle http://www.bouncycastle.org

[2] PKCS11 http://java.sun.com/j2se/1.5.0/docs/guide/security/p11guide.html.

## 1.6. Glossari

**PKCS11**
Interfície que permet l'accés al certificats dins una targeta.

**Servidor de temps**
Servidor que proporciona una data de referència per a facilitar la encriptació

**Applet**
Aplicació gràfica feta en Java que s'executa en el navegador web.

# 2. Descripció Detallada

## 2.1. Arquitectura i Components

### *2.1.1. Applet*


cd SunPKCS11
BCSignDynamic
- PROVEIDORS: String = "proveidorsPkcs...
- keyStoreBuilder: KeyStore.Builder = null
- keyStore: KeyStore = null
- key: Key = null
- providerSunPKCS11: SunPKCS11 = null
- providerBC: BouncyCastleProvider = null
- sunPKCS11CH: PKCS11CallbackHandler = null
- aliasCertificatEscollit: String = null
- signaturaPKCS7DettachedB64: String = null
- idSignant: String = null
- filePath: String = null
- codiErrorSignatura: String = null
- detallErrorSignatura: String = null
- baseURL: URL = null
- myUrlProveidor: URL = null
- myUrlServerTime: URL = null
+ BCSignDynamic()
- getUrlServerTime() : URL
- setUrlServerTime(String) : void
+ setUrlProveidor(String) : void
+ getUrlProveidor() : URL
+ getConfigProveidor() : InputStream
+ inicialitza(URL, String, String) : boolean
+ inicialitza() : boolean
- valida() : boolean
+ finalitza() : boolean
- inicialitzaVariables() : boolean
- finalitzaVariables() : boolean
- finalitzaProveidors() : boolean
- setBaseURL(URL) : void
- getBaseURL() : URL
- setSunPKCS11CH(PKCS11CallbackHandler) : void
- getSunPKCS11CH() : PKCS11CallbackHandler
- getKeyStoreBuilder() : KeyStore.Builder
- setKeyStoreBuilder(KeyStore.Builder) : void
+ getSignaturaPKCS7DettachedB64() : String
- setSignaturaPKCS7DettachedB64(String) : void
- getKeyStoreCustom() : KeyStore
- setKeyStoreCustom(KeyStore) : void
- getKey() : Key
- setKey(Key) : void
- getProviderSunPKCS11() : SunPKCS11
- setProviderSunPKCS11(SunPKCS11) : void
- getProviderBC() : Provider
- setProviderBC(BouncyCastleProvider) : void
+ getCodiErrorSignatura() : String
- setCodiErrorSignatura(String) : void
+ getDetallErrorSignatura() : String
- addDetallErrorSignatura(String) : void
- getAliasCertificatEscollit() : String
- setAliasCertificatEscollit(String) : void
- setIdSignant(String) : void
- getIdSignant() : String
- setFilePath(String) : void
- getFilePath() : String
- getMissatgeUISignatura(String) : String
- comprovaFitxer() : boolean
- inicialitzaProveidors() : boolean
- inicialitzaKeyStoreBuilder() : void
- getSubjectCN(String) : String
- showCertificate(String) : void
+ buscarAliasEscollit(String, Object[], List) : String
- checkKeyUsageAndIdSignant(String, String) : boolean
- escullAliasCertificat(LinkedList) : boolean
- selectSignCert() : boolean
- getFileContent() : byte[]
- showSignatureInfo(CMSSignedData) : void
- getServerTime() : Date
- getSignetAttributesTable() : AttributeTable
- signDataPKCS7Dettached() : boolean
- inicialitzaTargeta() : boolean
+ signFilePKCS7Dettached(String, String) : boolean

| Component | Package | Descripció |
|---|---|---|
| AppletSignaturaDigital | com.dgdej.appletsignaturadigital.SunPKCS11 | Classe que s'encarrega de obtenir els camps del formulari i s'encarrega de delegar la gestió de fer la signatura. |
| BCSignDynamic | com.dgdej.appletsignaturadigital.SunPKCS11 | S'encarrega d'obtenir el certificat del lector PKCS11 i signa el document. |

## 2.2. Instal·lació

### *2.2.1. Instal·lació en client*

A continuació es descriu els elements necessaris que s'han de tenir instal·lats al PC client, per poder utilitzar el mòdul de signatura digital:

- La versió mínima del JRE de Java necessària al client que executa l'applet de signatura és la 1.5. Si no esta instal·lada la 1.5 i executem l'applet, l'applet ens avisarà que no esta instal·lada i ens la farà instal·lar. Aquesta versió la poder-ho troba a: http://java.sun.com/
- Lector de targetes.
- Llibreries del lector de targeta. Permeten accedir al certificat de les targetes.
- Targeta amb el certificat.

### *2.2.2. Instal·lació en servidor*

Els arxius de codi que composen el mòdul de signatura digital han de complir els següents requeriments:

- Per a instal·lar el mòdul de signatura cal un servidor d'aplicacions.

El mòdul de signatura pot ser instal·lat en el mateix servidor d'aplicacions, o bé en un diferent, del de l'aplicació web que el cridarà. En cas de que estigui instal·lat en un servidor d'aplicacions diferent, la crida al mòdul de signatura ha de contenir la URL completa de l'adreça del servidor que conté el mòdul.

El mòdul de signatura ha estat desenvolupat i provat sota:

- Servidor d'aplicacions Tomcat v.5.0
- JDK v.1.4.2.05

Per tal de fer funcionar l'applet es necessita el openFrame-appletSignatura.-1.0.3.jar i les dependències definides en l'apartat '1.3 Versions i Dependències'.

Per a poder veure fàcilment les funcionalitat l'applet de signatura digital, s'ha construït un openFrame-aplicacioSignatura-1.0.3.war que es desplega fàcilment en el servidor d'aplicacions. És molt aconsellable desplegar-lo ja que permet veure totes les opcions de configuració. Es pot trobar en el respository MAVEN del CTITI.

La url és http://213.27.211.100/repository/openFrame/wars

### 2.2.3. Configuració client

El mòdul de signatura realitza una crida a un applet. La funció que realitza aquest applet és la de comprovar que la targeta de certificació de CATCert que disposa l'usuari sigui vàlida. Un cop ha realitzat la validació de la targeta, genera una signatura per a cada document.

En el següent quadre es descriuen els paràmetres d'entrada i sortida de l'applet:

| Paràmetre | Tipus | [E]ntrada / [S]ortida | Descripció | Etiqueta <html> oculta |
|---|---|---|---|---|
| Identificador document | Repetitiu, obligatori | E | Paràmetre que permet Identificar el document a la hora de retornar les dades generades durant la signatura | idDoc + num |
| Ruta completa document | Repetitiu, obligatori | E | Identifica de manera unívoca el fitxer que s'ha de signar mitjançant la ruta completa del fitxer al sistema de fitxers del client que executa l'applet per signar | rutaDoc + num |
| Número de documents | Obligatori | E | Numero total de documents a signar | totalRegistres |
| Crear Fitxer ( S / N ) | Obligatori | E | Crear els fitxer (*.txt) amb les signatures dels documents. Si el paràmetre és S, l'applet crea un fitxer a la mateixa carpeta que el document que es vol signar. Aquest fitxer contindrà la | fitxerSN |

| | | | | |
|---|---|---|---|---|
| | | | signatura i tindrà una extensió ( *.txt). Si el paràmetre és N, no es generarà cap fitxer i el que fa és emmagatzemar les signatures en una cookie, perquè la jsp de sortida pugui accedir a les firmes i les pugui mostrar.<br>El nom del fitxer estarà format pel: Nom del document signat +data (dd/mm/aaaa), hora, minuts i segon. Exemple: aaaa_10012006184612.txt | |
| Servidor de temps | Opcional | E | Url relativa o absoluta per que retorna la data del servidor,. | URL_SERVER_TIME |
| Signatura PKCS#7/CMS dettached | Obligatori | S | Signatura PKCS#7/CMS dettached en format Base64 | sigP7DDocB64 + num |
| Contingut del document signat | Obligatori | S | Contingut del document signat en format Base64 | DocsB64 + num |

- [E]ntrada=el valor del paràmetre ha d'estar informat per l'aplicació Web.
- [S]ortida=el valor del paràmetre l'informa l'applet com a resultat de la signatura

### *2.2.4. Configuració Servidor*

El servidor ha de permetre obtenir els jar que necessita l'applet per executar-se i definir els proveïdors

## Configuració proveïdors

A nivell de servidor es defineixen les targetes a les quals es tindrà accés.

Per a això tenim un fitxer principal, anomenat ***proveidors.conf***

```
CatCert=catcert.conf
#CamerFirmaUrl=http://172.24.43.15:8080/applicex2/camerfirma.conf
CamerFirma=camerfirma.conf
URL_SERVER_TIME=ServerTime.jsp
```

Aquest fitxer està composat per claus i valors. D'una banda les claus són el nom dels diferents proveïdors de certificats i d'altra banda els valors són les URIs dels fitxers de configuració d'aquests proveïdors. Aquestes poden ser relatives o absolutes.

A més també es pot afegir el JSP que permet obtenir el temps del servidor amb la clau URL_SERVER_TIME. Aquesta pot ser també relativa o absoluta.

Cal recordar que l'applet té un paràmetre anomenat URL_SERVER_TIME que també permet especificar la url per a obtenir el temps del servidor. En cas d'informar-se les dues, es prioritza el paràmetre de l'applet.

## Exemples de fitxers de proveïdors:

Els proveïdors que venen ja configurats amb l'applet són CatCert i Camerfirma

1. CatCert

El fitxer de CatCert s'anomena ***catcert.conf***

```
name = spicatoken
library = aetpkss1.dll
slotListIndex = 0
```

2. Camerfirma

El fitxer de camerfirma s'anomena ***camerfirma.conf***

```
name = camerfirma
library = incryptoki2.dll
slotListIndex = 0
```

L'estructura d'aquests fitxers és la que marca l'especificació PCKS11 de la màquina virtual de Java 1.5.
Els exemples que es presenten són bàsics, però si es vol saber totes les opcions possibles es pot consultar la url següent:
http://java.sun.com/j2se/1.5.0/docs/guide/security/p11guide.html.

És important indicar que els fitxers presentats sols són vàlids per a applets que s'executin en sistemes operatius Windows, ja que la llibreria de referència és una dll. Per tal de donar suport també a altres Sistemes Operatius caldria afegir noves entrades amb llibreries pròpies de cada sistema operatiu.

### *2.2.5. Control de finalització*

En la versió 1.0 de l'applet el control de finalització es feia en la funció javascript onValidarSignaturaCatcertJS, que era la que s'invocava per a validar el resultat.

Un cop finalitzada la validació, algunes aplicacions afegien codi en aquesta funció per realitzar tasques al finalitzar la signatura.

En la versió 1.0.3 aquest mètode javascript no es crida ja que les validacions s'han desactivat, però es cridat en un altre mètode propi per a poder realitzar tasques un cop ha finalitzat la signatura del document. Aquest mètode s'anomena onSignaturaFinalitzada.

**Per tant si es migra de l'Applet 1.0 a l' 1.0.3 és molt important tenir en compte aquesta diferència.**

# 3. Exemple

```
<center>
<form name="Form1" action="" method="POST">


   sigsB64Docs<input type="text" name="sigsB64Docs" value="" /><br>
   nomDocsB64<input type="text" name="nomDocsB64" value="" /><br>
   idDocs<input type="text" name="idDocs" value="" /><br>
   rutaDocs<input type="text" name="rutaDocs" value="" /><br>
   fitxerSN<input type="text" name="fitxerSN" value="S" /><br>
        totalRegistres<input type="text" name="totalRegistres" value="2"/><br><br>

   sigP7DDocB640<input type="text" name="sigP7DDocB640" value="" /><br>
   docB640<input type="text" name="docB640" value="" /><br>
   hashDocB640<input type="text" name="hashDocB640" value="" /><br>
   idDoc0<input type="text" size="17" name="idDoc0" value="0900000180042222"
/><br><br>

   sigP7DDocB641<input type="text" name="sigP7DDocB641" value="" /><br>
   docB641<input type="text" name="docB641" value="" /><br>
   hashDocB641<input type="text" name="hashDocB641" value="" /><br>
   idDoc1<input type="text" size="17" name="idDoc1" value="0900000180042223"
/><br><br>


   <table width="650" cellpadding="0" cellspacing="4" border="0">
          <tr>
                 <td>
                 Fitxer:<input type="file" name="file0"/>
                 Fitxer:<input type="file" name="file1"/>
                 </td>
          </tr>
   </table>

</form>

   <applet
code="com.dgdej.appletsignaturadigital.SunPKCS11.AppletSignaturaDigital"
width="100" height="20" mayscript="true" name="signap"
          id="signap">
          <param name="type" value="application/x-java-applet;version=1.5">
          <param name="archive"
                 value="openFrame-appletSignatura-1.0.3.jar,bcprov-jdk15-
1.3.0.jar,bcmail-jdk15-1.3.0.jar">
          <param name="mayscript" value="true">
          <param name="scriptable" value="true">
          <param name="enableButton" value="true">
          <param name="nomIdDoc" value="idDoc">
          <param name="nomRutaDoc" value="file">
          <param name="nomHashDocB64" value="hashDocB64">
          <param name="texteBotoSignatura" value="Signar">
          <param name="nomDocB64" value="docB64">
          <param name="nomSigP7DDocB64" value="sigP7DDocB64">
          <param name="nomTotalRegistres" value="totalRegistres">
          <!-- params personalitzats beg  -->
          <param name="signaResum" value="0">
          <param name="URL_SERVER_TIME" value="ServerTime.jsp">
   </applet>
```

```
</center>
```

Exemple que mostra els paràmetres que se li poden passar a l'applet i els que retorna.

# 4. Limitacions o treballs futurs

En la versió actual de l'applet no suporta tenir connectades diferents lectors de certificats al mateix temps ja que en la configuració de cada proveïdor s'indica el número de slot.

Una altra limitació és que l'applet no permet accedir als certificats que estiguin instal·lats dins el sistema operatiu, com per exemple els del repository de Windows.

El sistema de validacions que tenia l'applet en versió 1.0 s'ha desactivat ja que no era un sistema prou segur de validació. Per tant la funció javascript onValidarSignaturaCatcertJS ja no s'invoca.
La validació s'ha de realitzar a través d'un connector existent en el framework, que permet invocar un WebService del CATCert.

# 5. Annexos

## 5.1. Codis error

En la taula que es presenta a continuació, estan descrits els diferents codis d'error interns que poden aparèixer durant l'ús del mòdul del autoservei.

| Codi error | Descripció error |
|---|---|
| OK | Petició atesa amb èxit |
| ERR_1 | Error en el format de la petició |
| ERR_2 | El certificat no ha estat reconegut com a tal. No es confia en l'entitat emissora o cadena de certificació invàlida. |
| ERR_3 | El certificat no Ús vàlid per a realitzar l'operació de signatura sol·licitada (error en el Key Usage). |
| ERR_4 | El certificat no Ús vàlid per a realitzar l'operació sol·licitada a l'aplicació de referència (polítiques de seguretat). |
| ERR_5 | El certificat ha estat suspès. |
| ERR_6 | El certificat ha caducat. |
| ERR_7 | El certificat ha estat revocat. |
| ERR_8 | La signatura de les dades no Ús vàlida. La integritat de la informació no Ús correcta. |
| ERR_9 | El certificat i la signatura són correctes per= les CRLs no han estat actualitzades correctament. |
| ERR_10 | El certificat i la signatura són correctes, per= l'OID d'aquest certificat no esta autoritzat pel validador. |
| ERR_11 | La data de validesa del certificat Ús posterior a la data actual. |
| ERR_12 | Error del sistema de validació (InternalError, UnrecognizedToken) |
| ERR_13 | Error inesperat |
| ERR_14 | Error de connectivitat amb CatCert. |