Desarrollo de Aplicaciones en Entorno de Clave Pública

10 de marzo de 2009

JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

INDICE

1. Conceptos básicos.
2. Formatos de firma.
3. Arquitectura @firma.
4. El cliente de firma.
5. Integración con @firma v5.
6. Preparación del entorno de pruebas.
7. Supuestos prácticos.
8. Conclusiones. Ruegos y preguntas.

JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

1. Conceptos básicos

a. Conceptos preliminares.
b. Funciones resumen (hash). Colisiones.
c. Cifrado simétrico y asimétrico.
d. Certificados digitales.
e. Infraestructura de clave pública.

1.a. Conceptos preliminares

¿Qué es @firma?

@FIRMA: La plataforma corporativa de autenticación y firma

Última actualización: 25/02/2009

@FIRMA es la plataforma corporativa de la Junta de Andalucía para autenticación y firma electrónica. Gracias a @firma, las aplicaciones de la Junta de Andalucía pueden incorporar procesos de autenticación y firmado digital mediante el uso de certificados digitales, independientemente del entorno de desarrollo en que hayan sido programadas.

JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

1.a. Conceptos preliminares (I)

Autenticación

- Proceso que permite identificar a las entidades implicadas en una transacción y garantiza que estas entidades son quienes dicen ser.
- Utiliza certificados digitales para su objetivo.
- Aporta mayor información y contenido al proceso de logado a una aplicación.

Firma electrónica

- Una firma electrónica es el resultado de aplicar una serie de operaciones criptográficas a una entidad de información utilizando para ello un certificado electrónico, para obtener dos cosas: la integridad del documento firmado y el no repudio de la firma realizada.

1.b. Funciones resumen (hash). Colisiones

- Una función hash permite obtener a partir de un bloque de información un identificador o resumen imagen menor.
- Una función hash cumple las siguientes características:
  - Todos los resúmenes obtenidos a partir del mismo método tienen la misma longitud.
  - Es rápido calcular el resumen.
  - A partir del resumen no es posible obtener el contenido original (computacionalmente impracticable).
  - No existen dos bloques de información distintos cuyo resumen coincida.
- Algoritmos de hash habituales: MD5 (128 bits), **SHA1** (160 bits), SHA-256 (256 bits), etcétera.
- Una colisión se produce cuando dos entradas de datos distintas producen el mismo resumen.
- Se ha demostrado la existencia de colisiones en MD5, por lo que se desaconseja su uso.

1.c. Cifrado simétrico y asimétrico

- El cifrado es el proceso mediante el cual una información se ofusca por medio de un algoritmo determinado haciendo uso de una clave.
- A la información original se le denomina texto en claro. A la representación obtenida texto cifrado.

Cifrado simétrico

- Utiliza la misma clave para cifrar y descifrar.
- Requiere el conocimiento de la clave por el remitente y el destinatario de la información cifrada.
- El principal inconveniente reside en el intercambio de claves.
- Los algoritmos son eficientes.
- DES, Triple DES, **AES**, …

Cifrado asimétrico

- Utiliza una clave para el proceso de cifrado y otra para el descifrado.
- Lo cifrado con una clave sólo puede descifrarse con su complementaria.
- Las claves se denominan pública y privada.

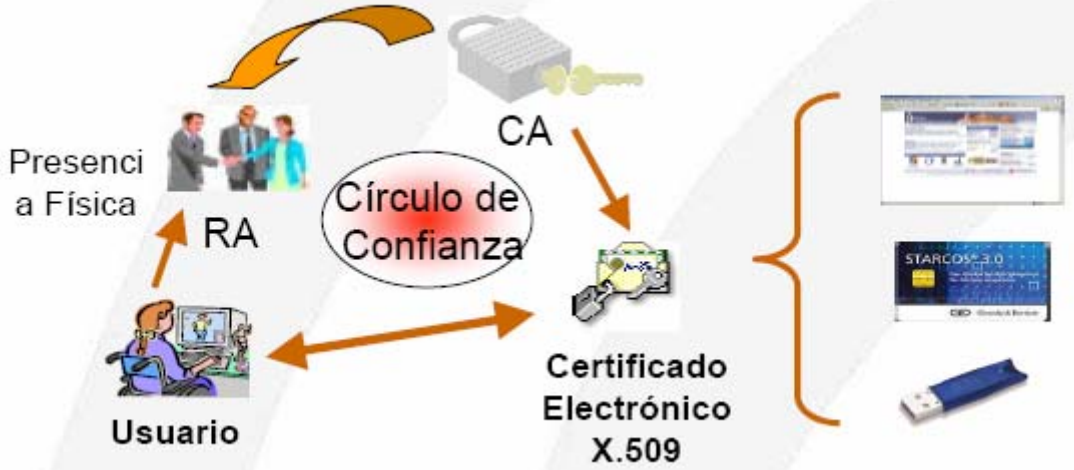
1.c. Cifrado simétrico y asimétrico (I)

Cifrado asimétrico

- La clave pública puede difundirse. La clave privada debe ser custodiada por su propietario.
- Longitudes de clave a partir de 512 bits.
- Son computacionalmente costosos.
- DSA, RSA.

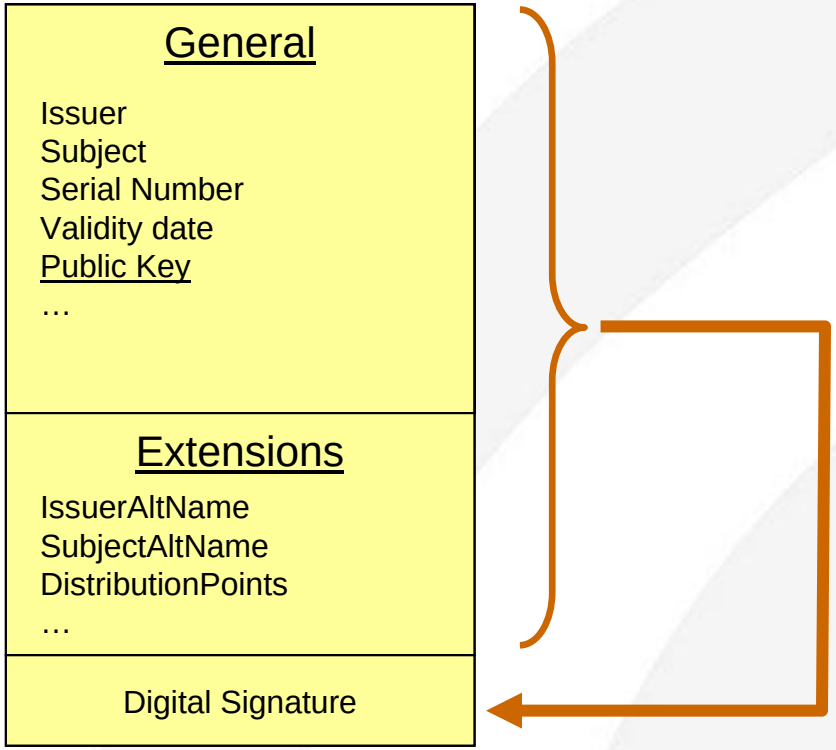
1.d. Certificados digitales

- Un certificado digital es un documento electrónico que relaciona una clave pública a un conjunto de información que identifica a una entidad (una persona, por ejemplo) que está en posesión de la correspondiente clave privada.
- Son emitidos por una Entidad de Certificación (CA) que garantiza la equivalencia entre el certificado y la entidad.
- Se almacenan en navegadores, ficheros (formato PKCS#12) y tokens criptográficos PKCS#15 (Tarjetas inteligentes, tokens USB, etcétera).



1.d. Certificados digitales (I)

Estructura de un certificado X.509v3



$C(Hash(\text{Info Cert.}))_{Prkissuer}$

JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

1.d. Certificados digitales (II)

Estado de certificado

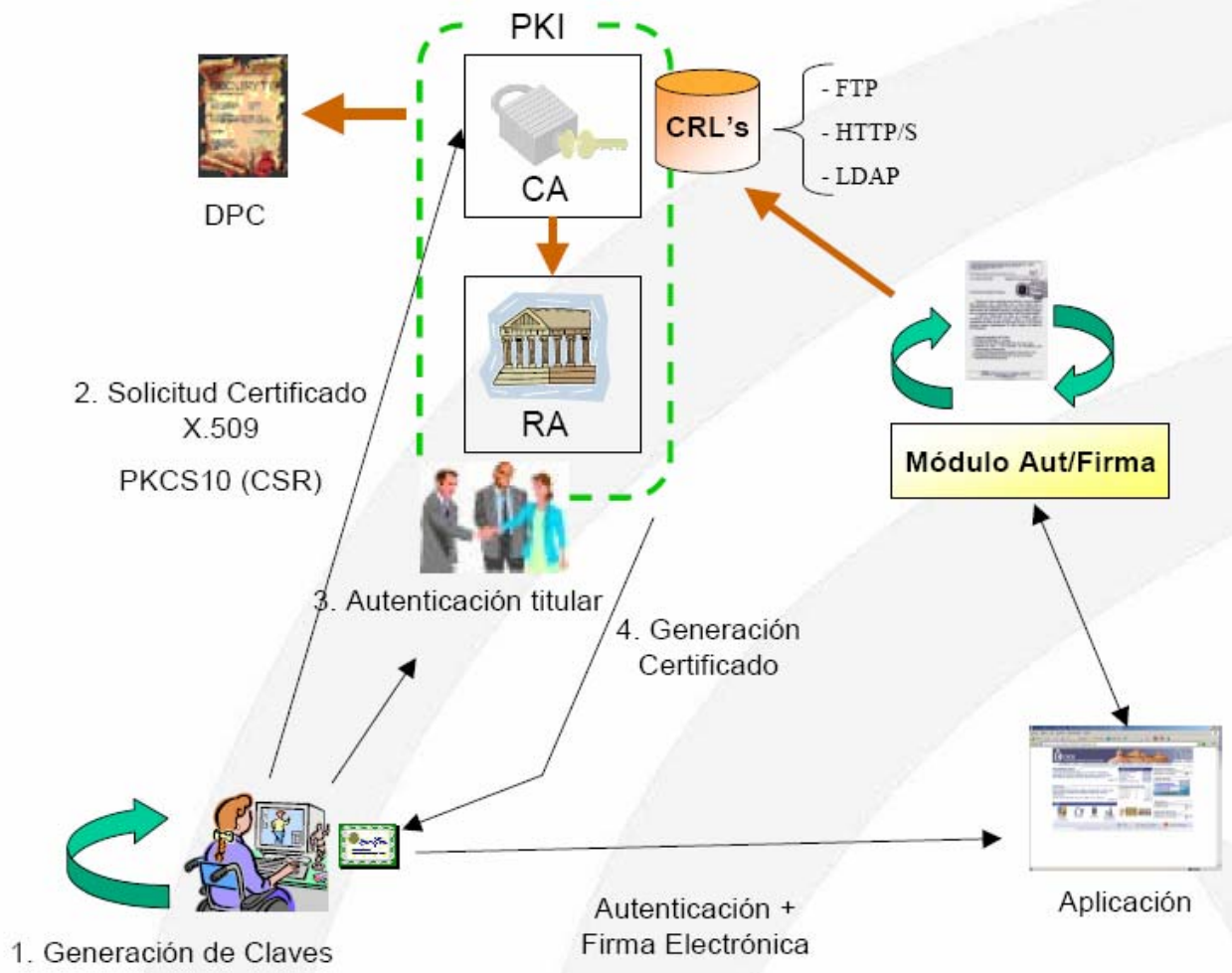
- Caducado: se ha superado la fecha de vigencia del certificado.
- Revocado: cuando ha sido rechazado, o bien por la Autoridad Certificadora que lo emite o bien el propio titular. Posibles motivos de revocación pueden ser el extravío, robo, copiado por terceros, etcétera.
- Suspendido: el certificado se ve afectado por una investigación o procedimiento judicial o administrativo, por lo que se cancela la validez del certificado durante un cierto período de tiempo, pudiendo volverse a levantar la suspensión dentro del periodo de validez del certificado.
- Válido: el certificado no se encuentra en ninguno de los estados anteriores.
- Un certificado caducado, revocado o suspendido no tiene validez, por lo que las firmas realizadas dejan de tener valor desde el momento de su revocación.

1.d. Certificados digitales (III)

Tipos de certificados electrónicos

- No existe un criterio estándar que nos permita clasificar de manera única todos los certificados disponibles en el mercado. La tipología de los certificados electrónicos varía en función de los PSCs que los emiten y gestionan. No obstante, se puede tomar la siguiente clasificación General:
  - **Certificados de persona física:** Certificado DNIe o PF de FNMT, por ejemplo.
  - **Certificados de persona jurídica o de representación:** Certificado de pertenencia a organización, por ejemplo.
  - **Certificados de Entidad:** Certificado emitido a nombre de Consejería determinada.
  - **Certificados de componentes:** Certificado SSL o de firma de código, por ejemplo.

1.e. Infraestructura de clave pública



JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

1.e. Infraestructura de clave pública (I)

Prestadores de servicios de certificación (PSC o CA)

- Son entidades que despliegan y mantienen entornos de confianza en ámbitos bien definidos.
- Ponen a disposición de sus usuarios herramientas para solicitar la obtención de certificados digitales de forma telemática.
- Gestionan el ciclo de vida de los certificados emitidos.
- Dan servicios de consulta de estado de certificados, etcétera.
- Pueden poseer infraestructuras de Autoridad de Registro (RA), o delegar este servicio.
- Definen jerarquías de certificación que permiten dar servicio a sus usuarios.
- Definen sus políticas de funcionamiento en Documentos de Prácticas de Certificación (DPC).
- Ejemplos: FNMT, DGP (DNIe), Firma Profesional, etcétera.

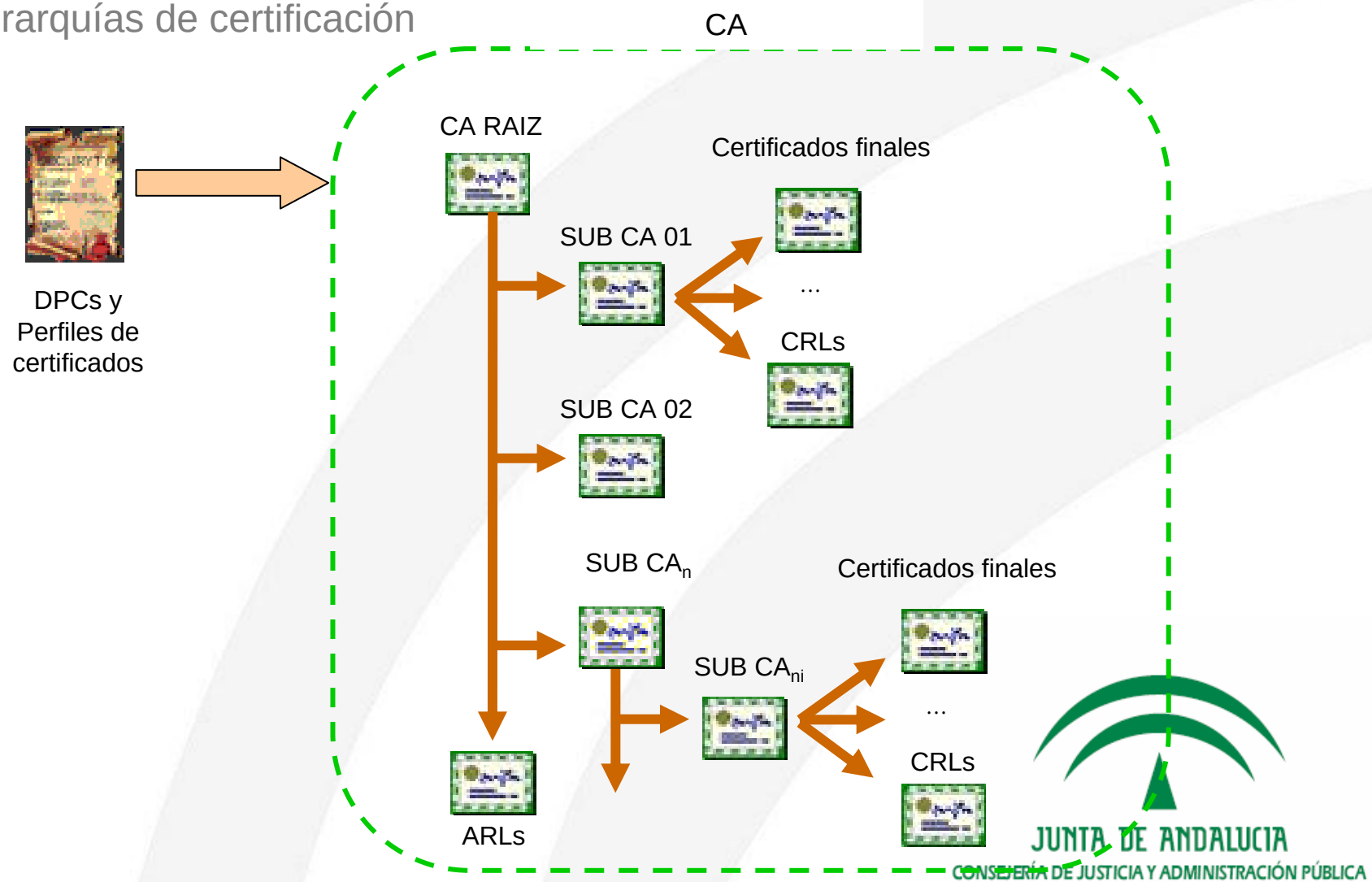
1.e. Infraestructura de clave pública (II)

Servicios de Certificación

- Consulta de estado de certificados: La CA debe publicar el estado de los certificados mediante las herramientas y protocolos pertinentes.
  - CRL (Certificate Revocation List)
  - OCSP (Online Certificate Status Protocol)
- Timestamping: Una transacción de firma puede incluir información complementaria que permita determinar el instante de realización de la firma. Dicho instante de tiempo debe estar acreditado por un tercero de confianza denominado Autoridad de Fechado Digital (TSA).
- Certificación de RA's: Formación y acreditación de Autoridades de Registro para la gestión de solicitudes de certificados. Se provee de las herramientas software y hardware necesarias a los registradores.
- Custodia a largo plazo: Almacenamiento y conservación de transacciones de firma en el tiempo proporcionando los métodos de disponibilidad pertinentes (según criterios de seguridad del BOE).
- Outsourcing: Suministro de servicios de CA para entidades privadas que lo necesiten. Incluyen un convenio de formación e implantación de RA's exclusivas para estas entidades privadas.

1.e. Infraestructura de clave pública (III)

Jerarquías de certificación



2. Formatos de firma

a. Tipos de firma electrónica.
b. Formatos de firma binarios. ASN.1.
c. Formatos de firma basados en XML.

2.a. Tipos de firma electrónica

Según la ley 59/2003 de firma electrónica

- Reconocidas: Estas firmas electrónicas son las únicas válidas legalmente ante terceros y equivalentes a la firma manuscrita tradicional. Una firma electrónica es reconocida cuando ha sido generada con un medio de creación de firmas seguro y utilizando un certificado electrónico reconocido.
- No reconocidas: No tienen validez legal, aunque técnicamente se pueden probar que son fiables. Son generadas por certificados internos o de PSCs no reconocidos por la Administración Española.

Según el número de firmantes

- Simple: En la estructura de firma electrónica existe un único firmante.
- Multifirma: En la estructura de firma electrónica existen varios firmantes.

2.a. Tipos de firma electrónica (I)

Según la jerarquía de la firma electrónica

- Jerárquica (counterSign): Firma de varios firmantes sobre un mismo documento, en un orden determinado de manera que cada uno firma sobre lo firmado anteriormente. Con ello cada firmante manifiesta conformidad con lo firmado por el anterior.
- Paralela (coSign): Cuando en la estructura de firma electrónica no existe ni orden ni jerarquía, lo que realmente importa es que un conjunto de N personas firmen un mismo documento.

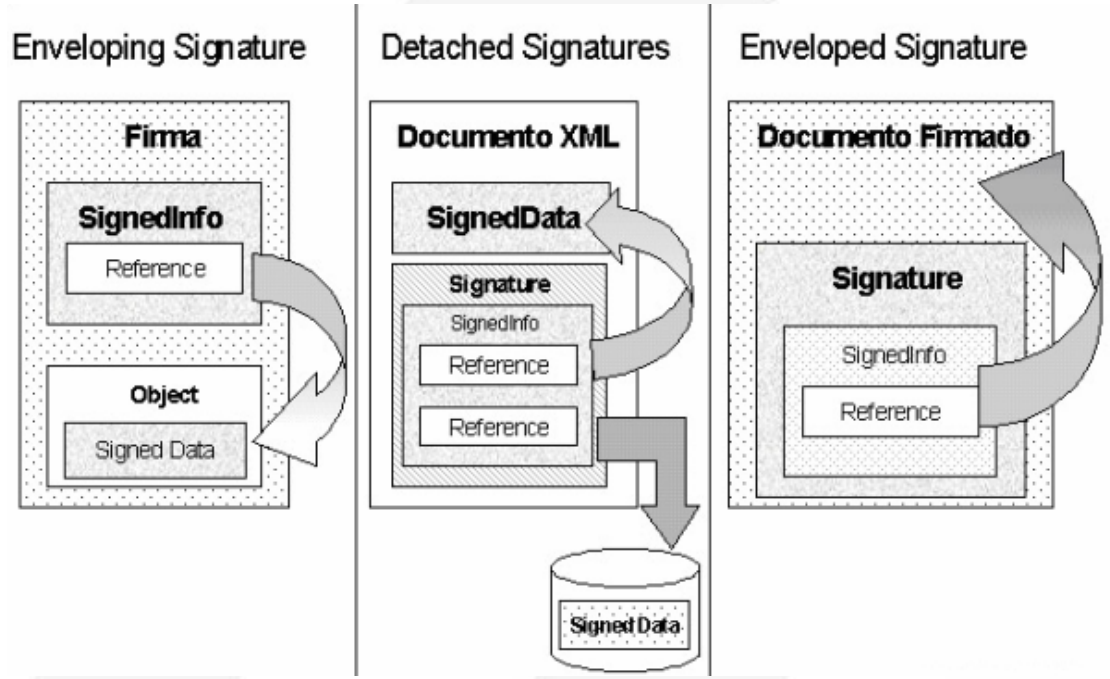
Según la ubicación de la información firmada

- Explícita: La estructura de firma electrónica es independiente del documento firmado, es decir, se obtienen dos ficheros, uno con la firma y otro con la información original.
- Implícita: La estructura de firma electrónica incorpora el documento firmado. Se suele utilizar cuando los documentos a firmar son pequeños.

2.a. Tipos de firma electrónica (II)

Según la relación entre la firma XML y el documento original

- Detached: El documento original se mantiene separado de la firma, manteniendo la firma una referencia al documento.
- Enveloped: El documento original y la firma quedan ligados en un único XML. El documento engloba a la firma.
- Enveloping: El documento original queda contenido en la firma. El documento a firmar debe ser XML.



JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

2.b. Formatos de firma binarios. ASN.1

ASN.1 (Abstract Syntax Notation One)

- Es un lenguaje de descripción de datos con un propósito muy específico: rapidez en el proceso y mínimo tamaño para acelerar su transmisión.
- Su uso está muy extendido en criptografía, si bien no se limita a ella: X.25, LDAP, UMTS, etcétera.
- ASN.1 permite indicar el significado de los datos, pero no cómo se deben codificar dichos datos para su transmisión. Existen múltiples tipos de codificaciones:
  - **BER (Basic Encoding Rules) :** Reglas de codificación básica. Denominadas sintaxis de transferencia en el contexto de ASN.1 especifican las secuencias de octetos exacta para codificar un elemento dado.
  - **DER:** Codificación distinguida, variante BER especialmente indicada para criptografía reduciendo redundancia y asegurando que para una información existe una única forma de representarla.
  - **XER:** Codificación XML, pensada para transcodificación directa entre ASN.1 y XML.
  - **CER:** Codificación canónica, variante de BER que evita incoherencias mediante una mayor restricción.
  - ...

2.b. Formatos de firma binarios. ASN.1(I)

ASN.1 (Abstract Syntax Notation One) – RFC 3852 (CMS)

5.1. SignedData Type
The following object identifier identifies the signed-data content type:

id-signedData OBJECT IDENTIFIER ::=
*{ **iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs7(7) 2** }*
The signed-data content type shall have ASN.1 type SignedData:

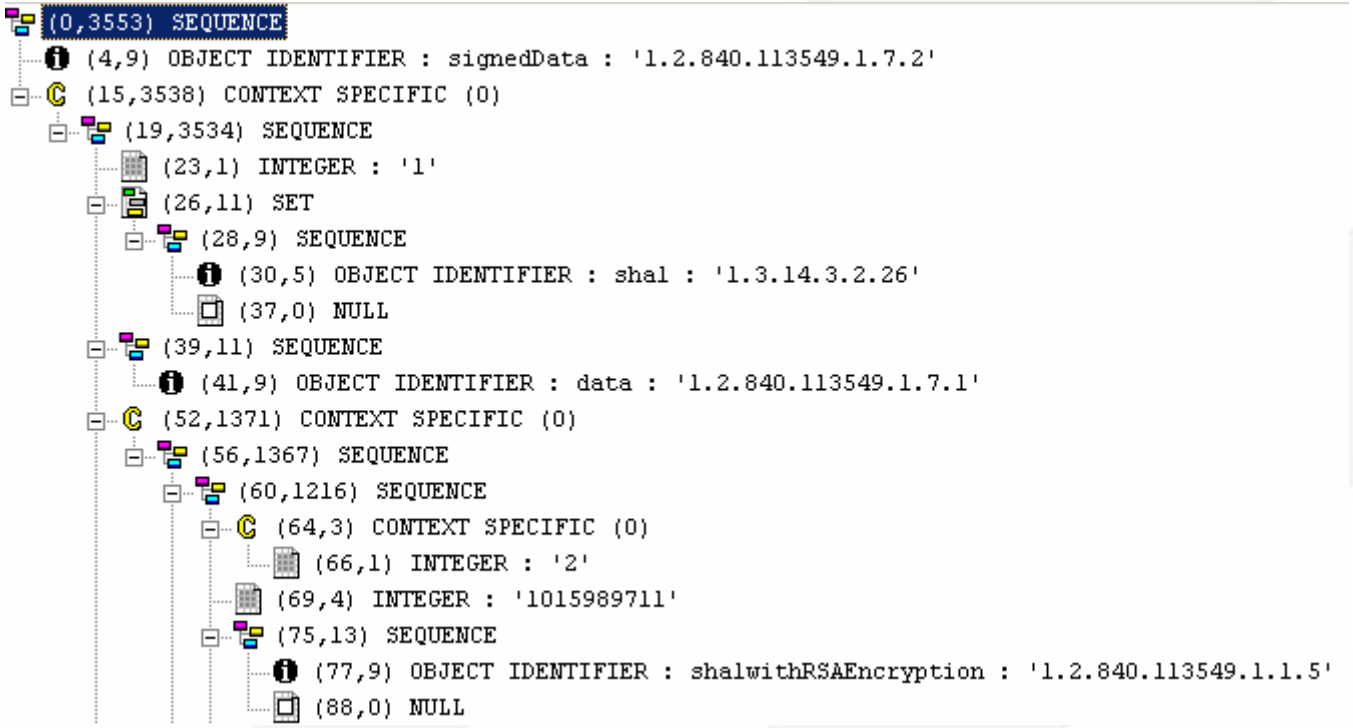
```
SignedData ::= SEQUENCE {
  version CMSVersion,
  digestAlgorithms DigestAlgorithmIdentifiers,
  encapContentInfo EncapsulatedContentInfo,
  certificates [0] IMPLICIT CertificateSet OPTIONAL, crls [1] IMPLICIT

    RevocationInfoChoices OPTIONAL, signerInfos SignerInfos }
```

JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

2.b. Formatos de firma binarios. ASN.1(II)

ASN.1 (Abstract Syntax Notation One)



JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

2.b. Formatos de firma binarios. ASN.1(III)

Formato PKCS#7 (Public-Key Cryptography Standard #7)

- PKCS#7 describe una sintaxis genérica para contenido sobre el cual se ha aplicado una operación criptográfica, como firmas digitales o sobres digitales.
- La sintaxis está basada en el estándar ASN.1.
- El formato acepta recursión, de modo que un objeto PKCS#7 puede estar dentro de otro objeto PKCS#7.
- Desarrollado por los laboratorios RSA.

2.b. Formatos de firma binarios. ASN.1(IV)

Formato CMS (Cryptographic Message Syntaxt)

- Básicamente CMS es la adecuación como estándar RFC (3369) de la normativa PKCS#7 de RSA.
- Se usa tanto en firmas como en sobres digitales.
- Muy similar a PKCS#7, aunque existen incompatibilidades entre ambos formatos, especialmente en los sobres digitales.

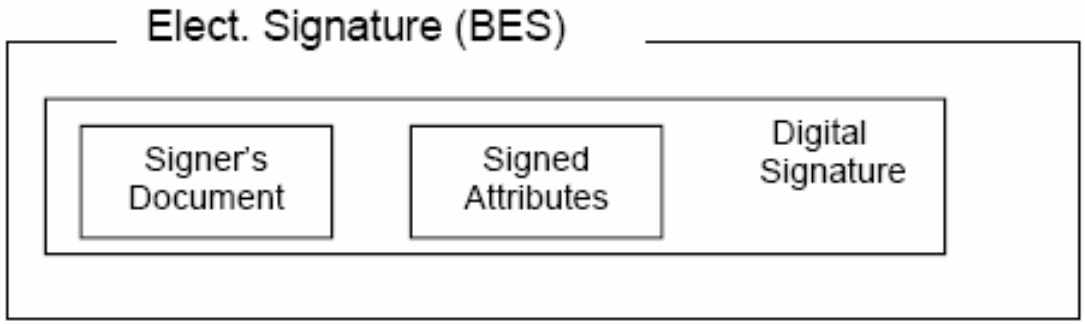
Formato CAdES (CMS Advanced Electronic Signature)

- CAdES es una evolución de CMS para acomodarse a las necesidades legales impuestas por la Comisión Europea.
  - Se han definido extensiones opcionales para el archivo longevo.
  - Se ha alineado la terminología con XAdES.
  - Se introducen referencias a las políticas de firma.
  - Se han actualizado los identificadores OID de ASN.1.
  - Se han corregido errores respecto a CMS.

2.b. Formatos de firma binarios. ASN.1(V)

Variantes CAdES (ETSI TS 101 733)

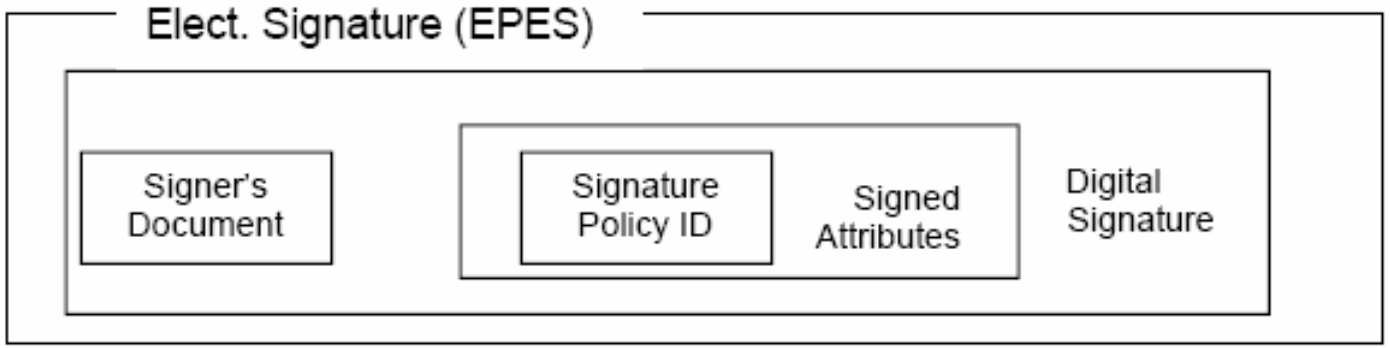
- CAdES-BES (básico): Tres campos obligatorios: documento, atributos (firmados) y firma digital. Se emplea para firma digital simple.



2.b. Formatos de firma binarios. ASN.1(VI)

Variantes CAdES (ETSI TS 101 733)

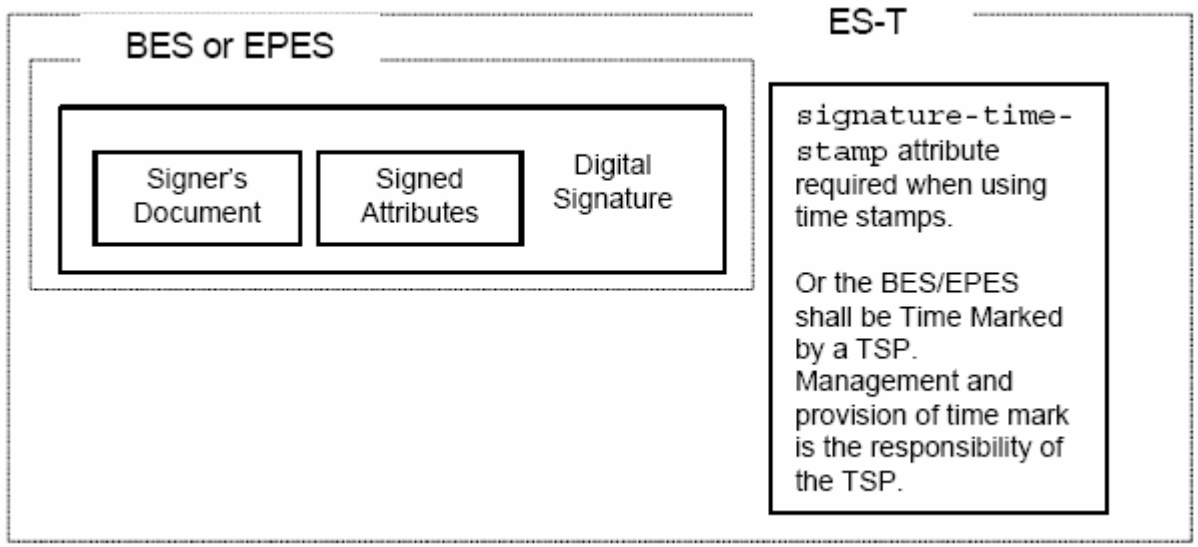
- CAdES-EPES (firma basada en política explícita): Se incluye como campo adicional un identificador de política.



2.b. Formatos de firma binarios. ASN.1(VII)

Variantes CAdES (ETSI TS 101 733)

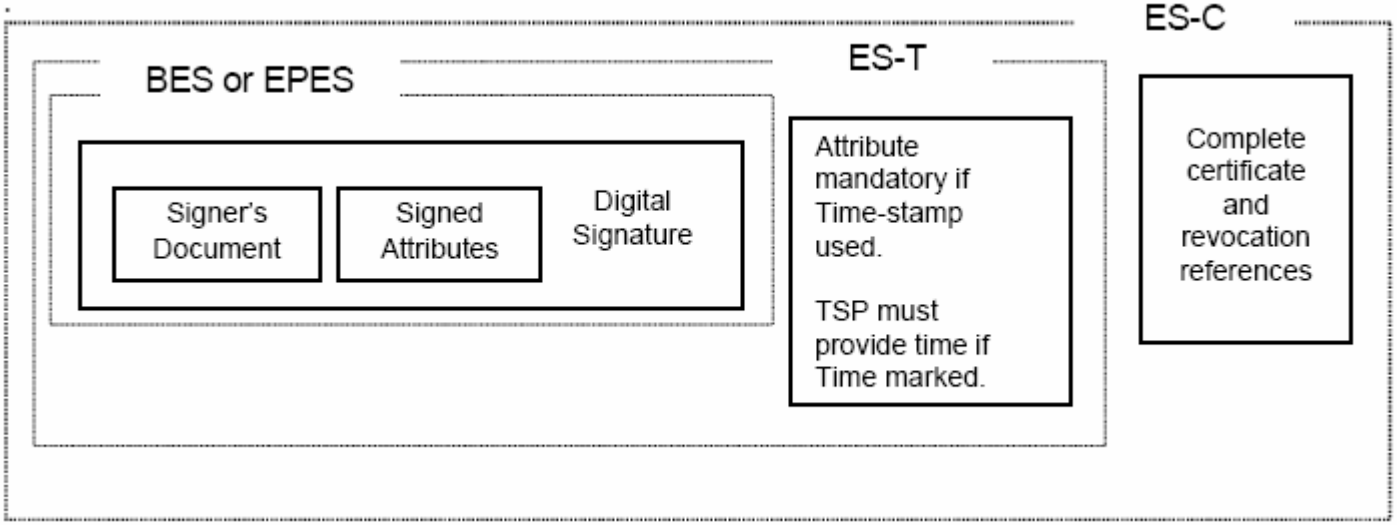
- CAdES-T: CAdES-BES/EPES + sello de tiempo. Permite probar que el documento firmado existió en un momento determinado.



2.b. Formatos de firma binarios. ASN.1(VIII)

Variantes CAdES (ETSI TS 101 733)

- CAdES-C: CAdES-T + referencias a los datos de validación (certificados y revocaciones).

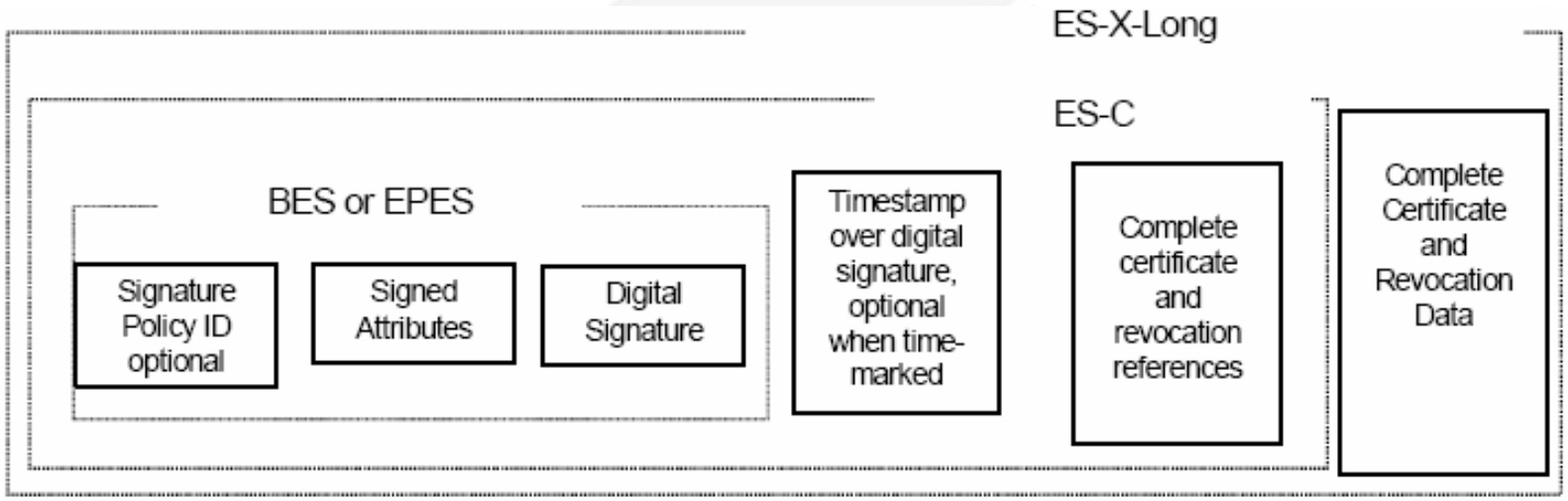


JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

2.b. Formatos de firma binarios. ASN.1(IX)

Variantes CAdES (ETSI TS 101 733)

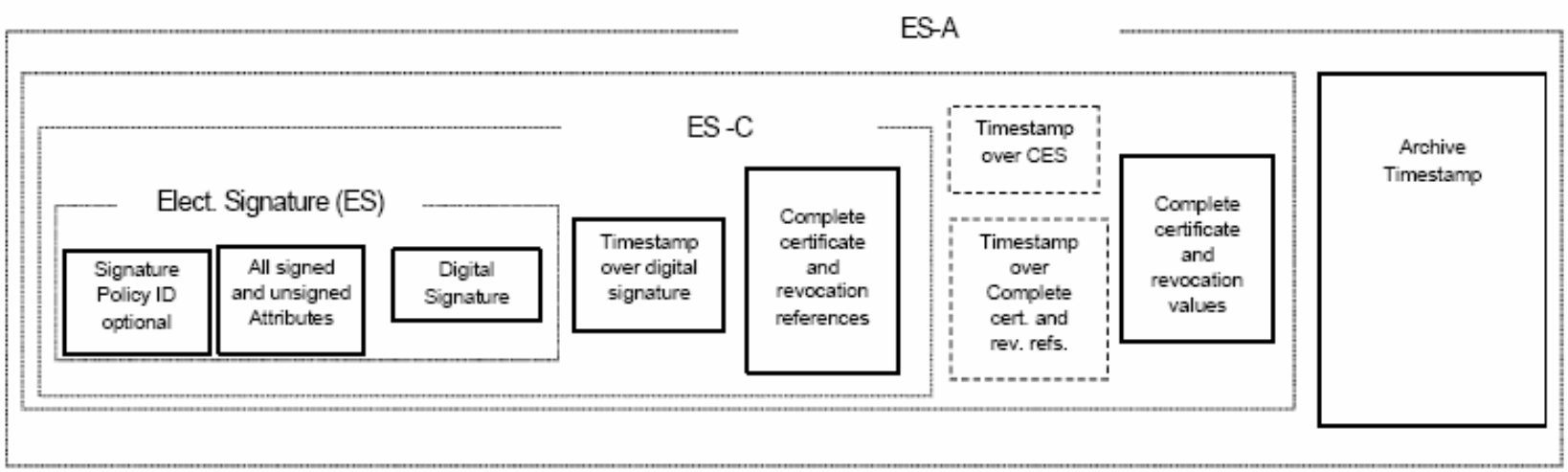
- CAdES-X-L: Como CAdES-C pero con datos de validación, no referencias. Firma longeva.



2.b. Formatos de firma binarios. ASN.1(X)

Variantes CAdES (ETSI TS 101 733)

- CAdES-A: Construido sobre CAdES-XL, añade la posibilidad de incorporar sellos de tiempo periódicos.



JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

2.c. Formatos de firma basados en XML

XMLDSig (RFC 3275)

- Equivalente de PKCS#7 para XML.
- Admite firma en el mismo XML que contiene los datos o firmas en un XML separado.
- Admite una gran versatilidad en cuanto a algoritmos de cifrado/firma, huellas digitales y codificación.
- Al igual que PKCS#7 incluye los datos adicionales necesarios (certificados, sellos de tiempo, etcétera)

2.c. Formatos de firma basados en XML (I)

XMLDSig (RFC 3275)

Elementos del XML

- Signature: Encapsula la firma digital.
- SignedInfo: Contiene la información necesaria para la creación y validación de firma.
- CanonicalizationMethod: Especifica el algoritmo de transformación canónica aplicado al código XML de SignedInfo antes de realizar el cálculo de su firma digital.
- SignatureMethod: Algoritmo utilizado para el cálculo de la firma digital.
- Reference: Referencia a la información o documento que se encuentra firmado.
- KeyInfo: Elemento opcional que permite recuperar la clave pública del firmante.
- Object: Elemento opcional que permite incluir información complementaria en la firma, los datos firmados, por ejemplo.

2.c. Formatos de firma basados en XML (II)

XMLDSig (RFC 3275)
Elementos del XML

```
<Signature ID?>
  <SignedInfo>
    <CanonicalizationMethod/>
    <SignatureMethod/>
    (<Reference URI?>
      (<Transforms>)?
      <DigestMethod>
      <DigestValue>
    </Reference>)+
  </SignedInfo>
  <SignatureValue>
  (<KeyInfo>)?
  (<Object ID?>)*
</Signature>
```

2.c. Formatos de firma basados en XML (III)

XAdES (ETSI TS 101 903)

- Es una variante de XMLDSig creada para adecuar la tecnología a las necesidades legales de la legislación europea respecto a firmas digitales.
- Respecto a XMLDSig añade los siguientes campos:
  - Sobre la firma (campos firmados): Fecha de la firma, certificado de firma, identificador de la política de firma, lugar de firmado y función del firmante.
  - Sobre la firma (campos no firmados): Contrafirma.
  - Sobre los datos firmados (campos firmados): Formato del objeto firmado, tipo de compromiso, sello de tiempo de todos los objetos de datos, sello de tiempo de cada objeto de datos (individualmente).

2.c. Formatos de firma basados en XML (IV)

Variantes XAdES (ETSI TS 101 903)

- XAdES-BES: Añade a XMLDSig una serie de atributos que aportan consistencia a la firma electrónica.

```
<Signature ID ?>
     <SignedInfo>
          <CanonicalizationMethod />
          <SignatureMethod />
          (<Reference URI?>
               (<Transforms>)?
               <DigestMethod>
               <DigestValue>
          </Reference>)+
     </SignedInfo>
     <SignatureValue>
          (<KeyInfo>)?
          (<Object ID ?>)*
          (<QualyfingProperties>)
</Signature>
```

```
<QualyfingProperties>
     <SignedProperties>
          (<SigningTime>)?
          (<SigningCertificate>)?
          (<SignerRole>)?
               ...
     </SignedProperties>
     <UnsignedProperties>
          (<CounterSignature>)?
          (<CompleteCertificateRefs>)?
          (<SignatureTimeStamp>)?
               ...
     </UnsignedProperties>
</QualyfingProperties>
```

JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

2.c. Formatos de firma basados en XML (V)

Variantes XAdES (ETSI TS 101 903)

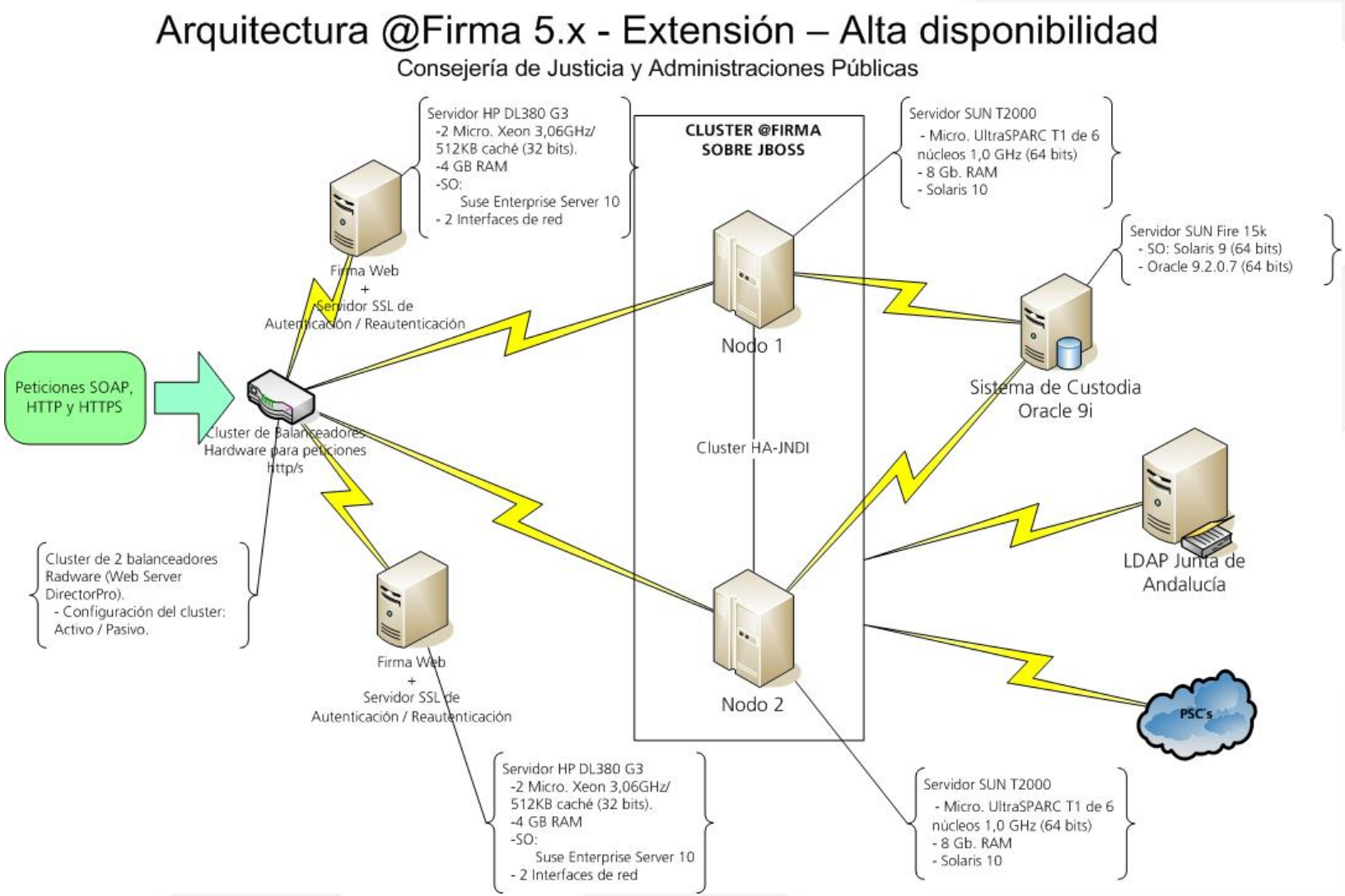
- XAdES_EPES: Se incluye como propiedad adicional un identificador de política
- XAdES-T: Añade un sello de tiempo para la firma para evitar el repudio. Es un nuevo campo no firmado.
- XAdES-C: Añade referencias a los datos completos necesarios para la validación (la cadena de certificación completa y su estado de revocación). Incorpora dos nuevos campos no firmados.
- XAdES-X: XAdES-C + información extendida para validación.
- XAdES-X-L: XAdES-X extendido para longevidad. Añade datos adicionales para la validación para no tener ninguna dependencia externa. Dos campos nuevos no firmados (certificados y revocaciones).
- XAdES-A: Incluye protección contra la obsolescencia de los algoritmos propietarios mediante un sello de tiempo de archivo.

3. Arquitectura @firma

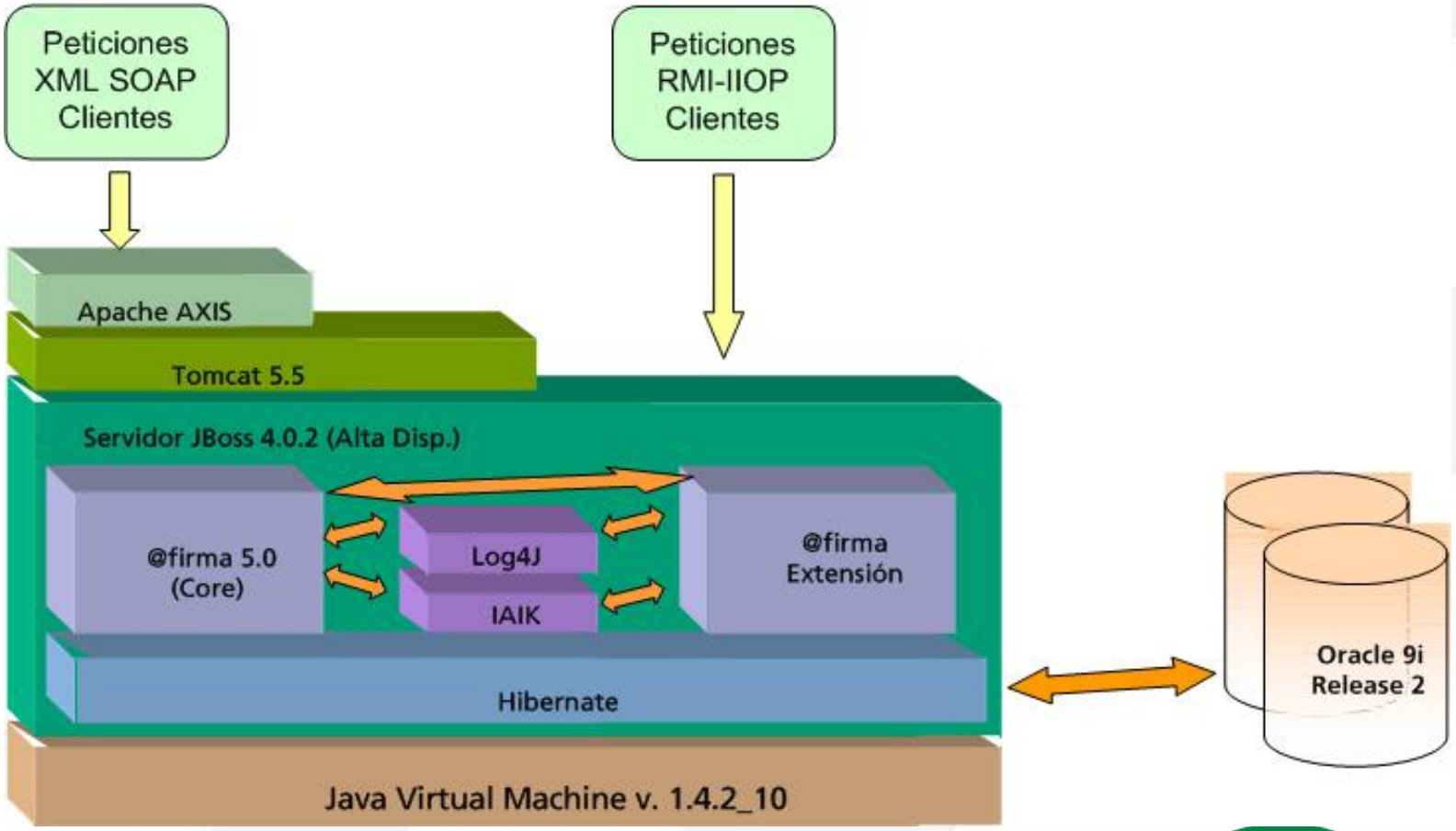
a. Arquitectura física.
b. Arquitectura lógica.

JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

3.a. Arquitectura física



3.b. Arquitectura lógica



JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

4. El cliente de firma

a. El instalador: estrategia de despliegue y mecanismo de distribuciones.
b. El cliente de firma: plugin XAdES.
c. Evolución: la versión 2.3.5.
d. Ejemplos del CD de desarrollo de @firma v5.

4.a. El instalador: estrategia de despliegue y mecanismo de distribuciones

- Responsable del despliegue de las librerías del cliente de firma.
- Interfaz Javascript disponible en instalador.js.
- Dispone de métodos para instalación, actualización y desinstalación.
- Instalación versiones anteriores 2.3.5.
  - Las librerías Java se copian en una ruta de la carpeta de perfil del usuario
    - Windows XP: C:\Documents and Settings\<usuario>\
    - Linux: /home/<usuario>/
  - Las librerías nativas se copian en carpetas de sistema. Requiere permisos de administrador del equipo.
    - Windows XP: C:\Windows
    - Linux: Seleccionable por el usuario
- Instalación en 2.3.5 y superiores.
  - La instalación completa se realiza en la carpeta de perfil del usuario.
  - No se requieren privilegios de administrador.

4.a. El instalador: estrategia de despliegue y mecanismo de distribuciones (I)

- Si la versión del cliente y la del servidor de aplicaciones difieren se ejecuta el proceso de instalación.
- Dicho comportamiento puede provocar actualizaciones frecuentes del cliente que perjudican la experiencia de usuario.
- A partir de la versión 2.3.5 está disponible el mecanismo de distribuciones, el cual permite alojar varias versiones independientes en la carpeta de perfil del usuario.
- Es necesario configurar la variable distinctDistroDir en scriptfirma.js (aplicación integrada con la extensión) o en el fichero constantes.js (integración nativa).
- Nomenclatura: <Indentificador organismo><X_Y_Z>
- Ejemplo: JA2_3_5

4.a. El instalador: estrategia de despliegue y mecanismo de distribuciones (II)

```
<html>
<head>
<script type="text/javascript" language="javascript" src="constantes.js"></script>
<script type="text/javascript" language="javascript" src="common-js/time.js"></script>
<script type="text/javascript" language="javascript" src="common-
    js/appletHelper.js"></script>
<script type="text/javascript" language="javascript" src="common-
    js/instalador.js"></script>
[…]
</head>
<body onload="cargarAppletInstalador()">
[…]
<a href="#" onclick="alert('Cliente instalado:' + isInstalado()); return false;">
Comprobar instalación cliente (true=correcta, false=no instalado)
</a>
<a href="#" onclick="alert('Plugin XAdES instalado:' + isInstalado('xades')); return
    false;">
Comprobar instalación plugin XAdES (true=correcta, false=no instalado)
</a>
<br>
[…]
</body>
</html>
```

JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

4.b. El cliente de firma: plugin XAdES

- Permite seleccionar el formato de firma a utilizar.
  - **CMS** (sin plugins)
  - XADES
  - XMLDSIGN
- Módulo de cifrado: Soportados AES, CAST5, IDEA, SERPENT, 3DES, RC5 y TWOFISH.
- Para incorporar soporte en el instalador para el plugin XAdES se agregarán las lineas en version.properties.

```
plugins.XAdES.version.mayor=2
plugins.XAdES.version.minor=3
plugins.XAdES.jars=xom.jar,XMLSIGtools.jar,xades-plugin.jar
plugins.XAdES.version.bDate=2008/03/17 08\:51
plugins.XAdES.version.build=0
plugins.XAdES.install.jars=xades-plugin.zip
plugins.XAdES.class=com.telventi.afirma.cliente.signatureformat.XAdESSignatureFormat
plugins.names=XAdES
```

- La instalación del plugin debe realizarse explícitamente.

```
<a href="#" onclick="instalar('xades'); return false;">
Instalar el plugin XAdES
</a>
```

JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

4.b. El cliente de firma: plugin XAdES (I)

Firma web

- En el proceso de firma web una parte de una página web (como un formulario o la página entera) puede ser firmada digitalmente.
- Dicho proceso consta de los siguientes pasos
  - Se compone un HTML (sin imágenes).
  - Se muestran al usuario los datos a firmar.
  - Se le solicita permiso para firmarlo.
  - Se le pide al usuario que seleccione un certificado con el que firmar.
  - Se solicita (si es necesario) la contraseña para acceder a la clave privada del certificado.
  - Se firma el HTML generado.
- En los campos modificables por el usuario se firman los valores seleccionados por el mismo. Esto incluye también adjuntos.
- El cliente no firma las imágenes.
- El cliente recoge todos los estilos CSS del documento definidos mediante LINK o STYLE.
- No es equivalente a la firma web de la extensión.

4.b. El cliente de firma: plugin XAdES (II)

```
<script type="text/javascript" language="javascript" src="constantes.js"></script>
<script type="text/javascript" language="javascript" src="../common-js/time.js"></script>
<script type="text/javascript" language="javascript" src="../common-js/appletHelper.js"></script>
<script type="text/javascript" language="javascript" src="../common-js/instalador.js"></script>
<script type="text/javascript" language="javascript" src="../common-js/firma.js"></script>
<script type="text/javascript" language="javascript" src="../common-js/htmlEscape.js"></script>
<script type="text/javascript" language="javascript" src="../common-js/utils.js"></script>
<script type="text/javascript" language="javascript" src="../common-js/styles.js"></script>
<script type="text/javascript" language="javascript" src="../common-js/firmaWeb.js"></script>
[…]
<script type="text/javascript" language="javascript">
function enviar()
{
     clienteFirma.initialize();
     clienteFirma.setShowErrors(false);
     var formulario = document.getElementById("formulario");
     var ruta = webSign(formulario, document);
     if(!clienteFirma.isError())
     {
             var fichero = document.getElementById("fichero");
             fichero.value = ruta;
             return true; // Enviar
     }
     else
     {
             alert("No se ha podido firmar: "+clienteFirma.getErrorMessage());
             return false;
     }
}
</script>
```

JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

4.b. El cliente de firma: plugin XAdES (III)

Firma electrónica

- Similar al proceso de firma web, aunque los datos a firmar no tienen por qué ser HTML.
- Existen varias alternativas a la hora de configurar la información a firmar,
  - Fichero. Se establece qué fichero firmar mediante el método setFileUri, el cual recibe la ruta al fichero a firmar.
  - Datos codificados base 64. Se establece la cadena a firmar codificada en base 64 con el método setData.
  - Hash codificado base 64. Se establece el hash codificado en base 64 con el método setHash.

4.b. El cliente de firma: plugin XAdES (IV)

```
<script type="text/javascript" language="javascript">
    function enviar() {
          var fichero= document.getElementById("fichero");
          clienteFirma.initialize();
          clienteFirma.setFileuri(fichero.value);
          clienteFirma.setShowErrors(false);
          firmar();
          if(!clienteFirma.isError())
          {
                    var firmaB64 = document.getElementById("firmaB64");
                    firmaB64.value = clienteFirma.getSignatureBase64Encoded();
                    return true; // Enviar
          }
          else
          {
                    alert("No se ha podido firmar:
    "+clienteFirma.getErrorMessage());
                    return false;
          }
    }
</script>
```

JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

4.b. El cliente de firma: plugin XAdES (V)

Firma masiva

- La firma masiva sólo permite firmar a partir de hashes codificados en base 64. No admite multifirma.
- Se invocará el método addMasiveHash tantas veces como sea necesario.
- Las firmas se recogen mediante el método getSignaturesBase64Encoded, el cual devuelve las firmas de los hashes codificadas en base 64.

```
clienteFirma.initialize();
clienteFirma.addMassiveHash(hash1);
clienteFirma.addMassiveHash(hash2);
clienteFirma.sign();
if(!clienteFirma.isError())
{
	var signaturesArray= clienteFirma.getSignaturesBase64Encoded().split("!");
	for(i=0; i<signaturesArray.length; i++)
	{
		var signature= signaturesArray[i];
		[...]
	}
}
```

4.b. El cliente de firma: plugin XAdES (VI)

Co-firma

- Permite a varios usuarios firmar un mismo documento.
- Para ello es necesario pasar al cliente la firma electrónica de los demás firmantes.
- Es posible proporcionar dicha firma de varias maneras,
  - Mediante un fichero que contenga la firma codificada en base 64. Para ello se ejecutará el método setElectronicSignatureFile, al cual se le pasará la ruta del fichero como parámetro.
  - Indicándolo directamente en base 64 con el método setElectronicSignature.
  - Caso de no especificarse se pedirá al usuario que elija un fichero con la firma base 64.
- Para realizar la firma se invoca al método coSign.

4.b. El cliente de firma: plugin XAdES (VII)

Contra-firma

- Permite a un usuario firmar las firmas de otros usuarios.
- Es similar a la co-firma, aunque indicando qué firmas contra-firmar. Puede ser necesario conocer la estructura de firmantes del documento.
- La estructura de firmantes puede conocerse con el método getSignersStructure, el cual retorna una cadena que contiene los nombres de los firmantes separados por un retorno de carro (\n en JavaScript). Al comienzo del nombre hay tantos tabulados (\t) como nivel ocupe el firmante en el documento.
- Se puede especificar qué firmantes firmar de diferentes maneras
  - Firmar todas las hojas (firmas no contra-firmadas): counterSignLeafs.
  - Firmar todos los nodos: counterSignTree.
  - Firmar todos los nodos de un firmante: setSignersToCounterSign.
  - Firmar nodos (firmas) determinados: setSignerToCounterSign.

4.c. Evolución: la versión 2.3.5

- Última versión disponible del cliente de firma.
- Si bien la numeración puede sugerir una revisión menor, incorpora importantes mejoras y correcciones de errores.
  - Soporte para Windows Vista.
  - Soporte mejorado para Linux. Guadalinex v3 y v4.
  - Mecanismo de distribuciones.
  - Instalación simplificada y sin privilegios de administrador.
- Soporte revisado para Windows Vista SP1, Windows XP SP2 y SP3, Windows 2000 SP4, Guadalinex v3, v4 y v5 (no soportado), con navegadores Internet Explorer 6, 7, Mozilla Firefox 1.5, 2.0 y 3.0 (no soportado), con Sun JRE 1.5 y 1.6.

4.c. Evolución: la versión 2.3.5

Matriz de compatibilidad

	Sun JRE 1.5				Sun JRE 1.6			
	IE6	IE7	Ff1.5	Ff2.0	IE6	IE7	Ff1.5	Ff2.0
Windows 2000 SP4	OK	N/D	OK	OK	OK	N/D	OK	OK
Windows XP SP2, SP3	OK	OK	OK	OK	OK	OK	OK	OK
Windows Vista, SP1	N/D	OK	OK	OK	N/D	OK	OK	OK
Guadalinex v3	N/D	N/D	OK	OK	N/D	N/D	OK	OK
Guadalinex v4	N/D	N/D	*	*	N/D	N/D	OK	OK
Guadalinex v5	N/D	N/D	x	x	N/D	N/D	x	x

*No soportado JRE 1.5 instalado desde el repositorio

+ Firefox 3 no está soportado en ninguna configuración

x Guadalinex v5 no está soportado en ninguna configuración

JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

4.d. Ejemplos del CD de desarrollo de @firma v5

- En el disco de desarrollo de @firma v5 se incluyen ejemplos que permiten familiarizarse con la API del cliente de firma de ficheros y con los conceptos relacionados.
- Los ejemplos se encuentran ubicados en la ruta Cliente\web-instalador
  - demoInstalador.html.
  - demoFirmaWeb-01.htm.
  - demoFirmaMasiva.html.
  - demoMultifirma.html.
  - demoCifrado.html.
  - demoSobreDigital.html.
  - demoVisorNodosMultifirma.html.

5. Integración con @firma v5

a. Mecanismos de seguridad. Acceso seguro y métodos de autorización.
b. Módulos. Custodia, firma y validación.
c. Mensajes de entrada y respuesta en servicios. Esquemas de validación.
d. Descripción de servicios de la plataforma.
   - Validación.
   - Firma electrónica.
   - Servicios de custodia.
e. Fachada de tickets.

5.a. Mecanismos de seguridad. Acceso seguro y métodos de autorización

- El acceso a los servicios de la plataforma por las aplicaciones está securizado mediante el establecimiento de una conexión segura (SSL) entre el servidor de aplicaciones y el servidor de firma, y su autorización con alguno de los mecanismos configurados para la aplicación.
- Para establecer una conexión segura el cliente Webservice de la aplicación debe estar configurado para tener acceso a la clave pública del servidor de firma.
- La clave pública de cada entorno @firma de la Consejería de Justicia y Administración Pública es proporcionada en el proceso de alta de la aplicación.

5.a. Mecanismos de seguridad. Acceso seguro y métodos de autorización (I)

- En el alta de una aplicación es posible configurar alguno de los siguientes métodos de autorización:
  - Mediante usuario y password. La autorización de uso de un servicio web se encuentra condicionada a la especificación en la llamada al servicio de una pareja usuario-clave dada de alta previamente en la configuración de la aplicación. Es posible especificar en el alta varias parejas usuario-clave. Corresponde al modo de securización UsernameToken del fichero de configuración de los ejemplos del disco de desarrollo de @firma v5.
  - Mediante certificado. Con este método se empleará un certificado digital para la validación de la llamada a un servicio web de la plataforma. Es posible proporcionar en el alta varios certificados digitales para una aplicación. Corresponde al modo de securización BinarySecurityToken del fichero de configuración de los ejemplos del disco de desarrollo de @firma v5.

5.b. Módulos. Custodia, firma y validación

- Los servicios de la plataforma están disponibles en la dirección https://servidor_de_firma/afirmaws/services
- Los servicios se encuentran agrupados funcionalmente en módulos.
  - Módulo de custodia. Incluye los relacionados con el acceso a la custodia de la plataforma.
    - AlmacenarDocumento
    - EliminarContenidoDocumento
    - ObtenerContenidoDocumento
    - ObtenerContenidoDocumentoId
    - ObtenerIdDocumento
    - ActualizarReferencia
    - ObtenerTransaccionesPorFecha
    - ObtenerTransacciones
    - …

5.b. Módulos. Custodia, firma y validación (I)

- Módulo de firma. Agrupa los servicios relacionados con las distintas modalidades de firma.
  - ValidarFirma
  - FirmaServidor
  - FirmaServidorCoSign
  - FirmaServidorCounterSign
  - FirmaUsuario3FasesF1
  - FirmaUsuario2FasesF2
  - FirmaUsuarioBloquesF1
  - ValidarFirmaBloquesCompleto
  - …

5.b. Módulos. Custodia, firma y validación (II)

- Módulo de validación. Contiene los servicios relacionados con la validación de certificados.
  - ValidarCertificado
  - ObtenerInfoCertificado

5.c. Mensajes de entrada y respuesta en servicios. Esquemas de validación

- Los parámetros de entrada de los servicios y los resultados de su ejecución se transmiten mediante mensajes XML.
- Ejemplo de mensaje de entrada

```
<?xml version="1.0" encoding="UTF-8"?>
        <mensajeEntrada targetNamespace="https://afirmaws/ws/firma"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="https://localhost/afirmaws/xsd/mfirma/ws.xsd">
        <peticion>FirmaServidor</peticion>
        <versionMsg>1.0</versionMsg>
        <parametros>
                    <idAplicacion>appPrueba</idAplicacion>
                    <idDocumento>177</idDocumento>
                    <firmante>RSA.2048</firmante>
                    <idReferencia>idReferencia</idReferencia>
                    <algoritmoHash>SHA1</algoritmoHash>
                    <formatoFirma>CMS</formatoFirma>
        </parametros>
</mensajeEntrada>
```

JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

5.c. Mensajes de entrada y respuesta en servicios. Esquemas de validación (I)

- Ejemplo de mensaje de salida/respuesta

```
<?xml version="1.0"?>
<mensajeSalida xmlns="https://afirmaws/ws/firma"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:SchemaLocation="https://localhost/afirmaws/xsd/mfirma/ws.xsd ">
       <peticion> FirmaUsuario3FasesF1 </peticion>
       <versionMsg>1.0 </versionMsg>
       <respuesta>
                  <Respuesta>
                             <estado>true</estado>
                             <descripcion>Proceso de fase 1 de firma de usuario en 3
fases realizado correctamente.</descripcion>
                             <idTransaccion>1150302756695137</idTransaccion>
                             <hash>Z3IUyA4ZiemW6cbMlgunG+wPqT8=</hash>
                             <algoritmoHash>SHA1</algoritmoHash>
                  </Respuesta>
       </respuesta>
</mensajeSalida>
```

JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

5.c. Mensajes de entrada y respuesta en servicios. Esquemas de validación (II)

- Para cada módulo se encuentra definido un XML-Schema que permite validar los mensajes de entrada y de respuesta de los servicios del módulo.
- En el CD de desarrollo de @firma se encuentran disponibles las definiciones de todos los módulos.
  - KitIntegraciónWS\xsd\mcustodia
  - KitIntegraciónWS\xsd\mfirma
  - KitIntegraciónWS\xsd\mvalidacion

5.d. Descripción de servicios de la plataforma Validación.

ValidarCertificado

- El servicio ValidarCertificado es el responsable de la validación de certificados X509 y el DNI electrónico.
- Es necesario especificar alguno de los modos de validación siguientes.
  - Validación simple (0). Comprueba la caducidad, integridad y confianza del certificado.
  - Validación intermedia (1). Realiza las comprobaciones de la validación simple y el estado de revocación del certificado.
  - Validación compleja (2). Realiza las comprobaciones de la validación intermedia y valida la cadena de confianza al completo.
- Permite recuperar la información asociada al certificado. El campo TIPOAFIRMA permite identificar el tipo de certificado (Guía de Certificados Reconocidos por @firma).
- Empleado para implementar el proceso de autenticación de las aplicaciones.

5.d. Descripción de servicios de la plataforma Validación (I).

ObtenerInfoCertificado

- Permite extraer la información de un certificado mediante la aplicación del mapeo definido para su tipo. Este proceso verificará que el tipo de certificado se encuentra definido en la plataforma y que la aplicación que realiza la petición tiene acceso a dicho tipo de certificado.

```
<Campo>
        <idCampo>subject</idCampo>
        <valorCampo>
                EMAIL=test@test.com,CN=FEREN
                TEST,givenName=FEREN,SN=TEST,serialNumber=11111111H,OU=000000000,OU=COLEGIO
                NOTARIAL DE TEST,OU=NOTARIO - PRUEBAS,O=CONSEJO
                GENERAL DEL
                NOTARIADO,L=BARCELONA,ST=BARCELONA,C=ES
        </valorCampo>
</Campo>
<Campo>
        <idCampo>tipoCertificado</idCampo>
        <valorCampo>ANCERT PF FERN</valorCampo>
</Campo>
<Campo>
        <idCampo>versionPolitica</idCampo>
        <valorCampo>31</valorCampo>
        </Campo>
<Campo>
```

5.d. Descripción de servicios de la plataforma
Firma electrónica. Formatos de firma y tipos de firma

- La plataforma @firma v5 soporta múltiples formatos de firma electrónica.
  - PKCS7 (PKCS7)
  - **CMS (CMS)**
  - CAdES (CADES-BES, CADES-T) (5.2)
  - XMLDSignature (XMLDSIG)
  - XAdES (XADES-BES, XADES-T)
- En el cálculo de la firma es posible utilizar actualmente los siguientes algoritmos de hash.
  - MD2
  - MD5
  - SHA
  - **SHA1**
  - SHA256
  - SHA384
  - SHA512

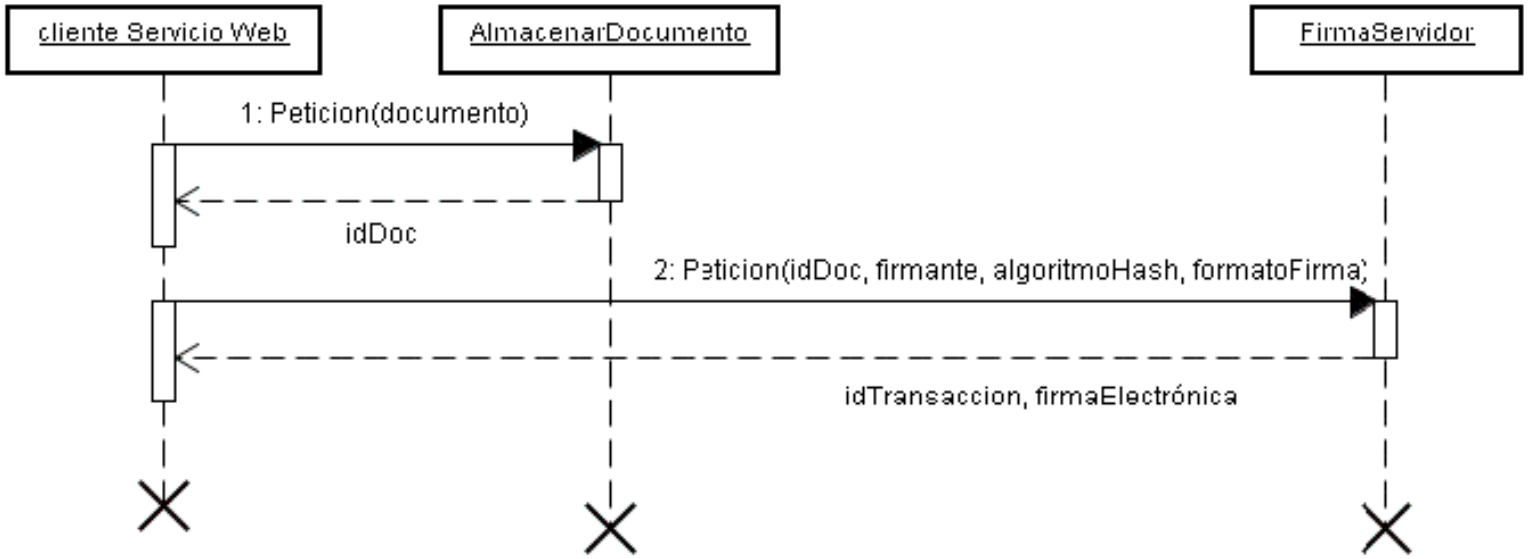
5.d. Descripción de servicios de la plataforma Firma electrónica (I)

Firma electrónica en servidor

- Permite a una aplicación cliente realizar una firma electrónica con el certificado de firma de servidor especificado en la llamada al servicio.
- Es posible realizar firmas con varios certificados X509 custodiados en la plataforma y configurados para la aplicación.
- Existen tres modalidades de firma de servidor.
  - Firma simple. Se genera una firma electrónica con un formato determinado a partir de unos datos. Requiere el registro previo del documento.
  - Firma en paralelo (cosign). A partir de una firma electrónica existente se añade un nuevo valor de firma al envoltorio que es la firma electrónica en sí.
  - Firma en cascada (countersign). Se realiza una firma sobre el valor de la firma electrónica de un firmante concreto, dentro del envoltorio de la firma electrónica.
- Por defecto se asocia a cada aplicación dada de alta el certificado de servidor correspondiente a la plataforma.

5.d. Descripción de servicios de la plataforma Firma electrónica (II)

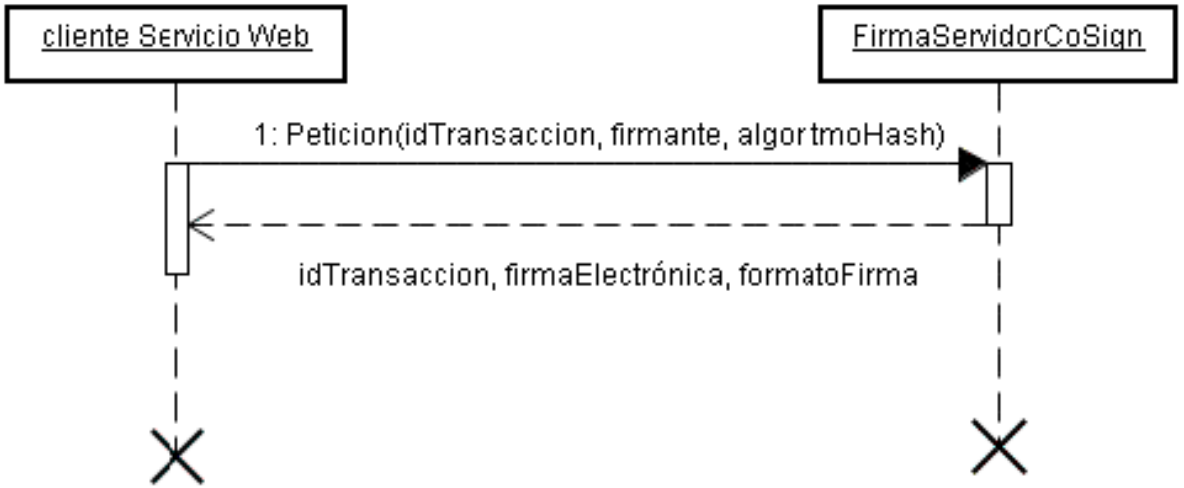
Firma servidor simple



JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

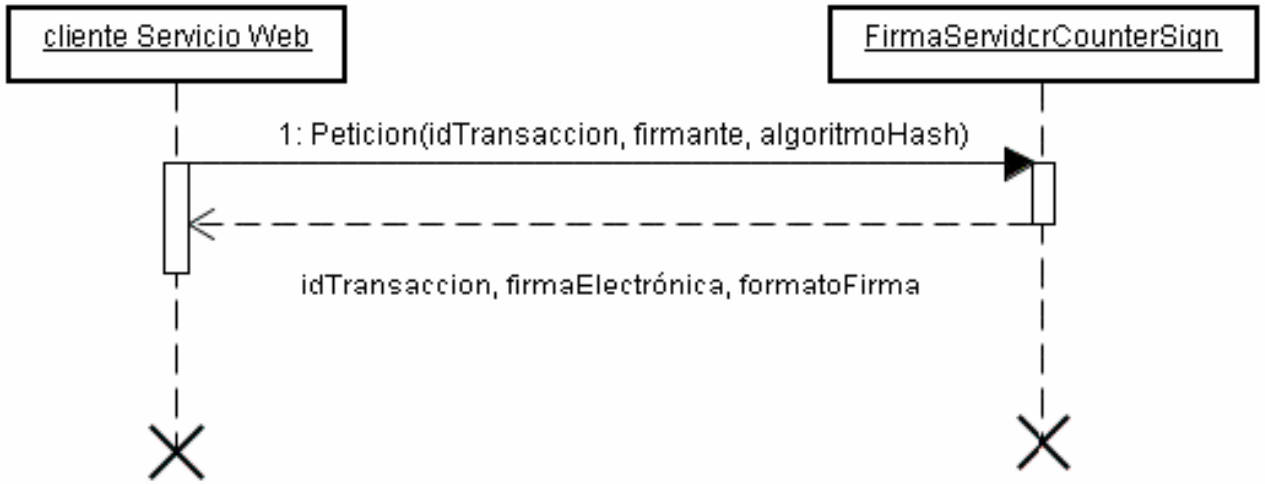
5.d. Descripción de servicios de la plataforma Firma electrónica (III)

Firma servidor Cosign



5.d. Descripción de servicios de la plataforma Firma electrónica (IV)

Firma servidor Countersign



JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

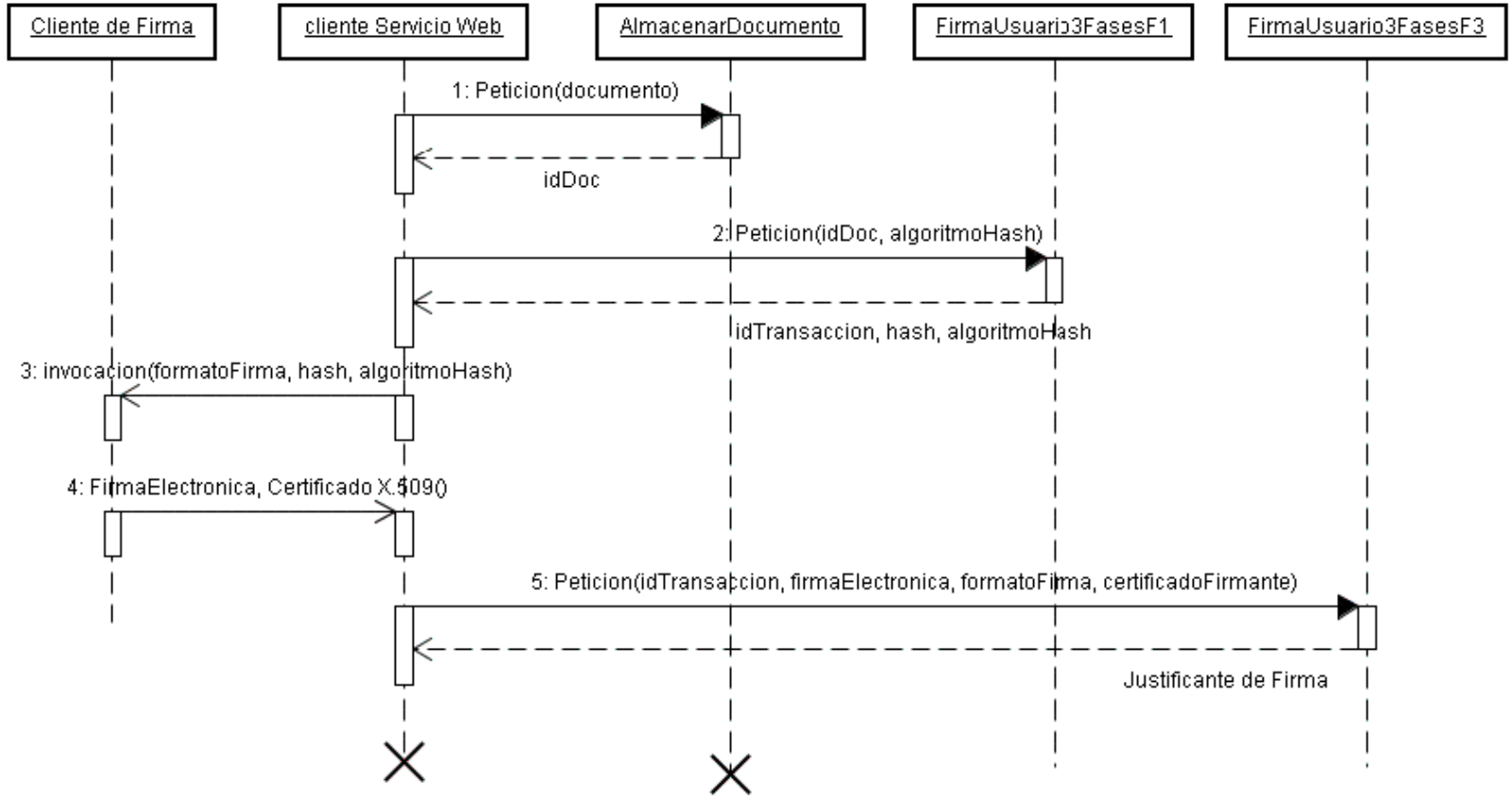
5.d. Descripción de servicios de la plataforma Firma electrónica (V)

Firma electrónica de usuario en 3 fases

- Permite a las aplicaciones ofrecer a sus usuarios un servicio de generación de firma electrónica en el cliente.
- Requiere el uso del cliente de firma de la plataforma.
- Existen tres modalidades de firma de usuario en 3 fases.
  - Firma simple. La información recuperada de la plataforma y pasada al cliente de firma será el hash de un documento registrado previamente.
  - Firma Cosign. La información recuperada de la plataforma y pasada al cliente de firma será una firma electrónica. Se añade un nuevo valor de firma al envoltorio pasado como parámetro.
  - Firma Countersign. La información recuperada de la plataforma y pasada al cliente de firma será una firma electrónica. Se realiza una firma sobre el valor de la firma electrónica de un firmante concreto, dentro del envoltorio.

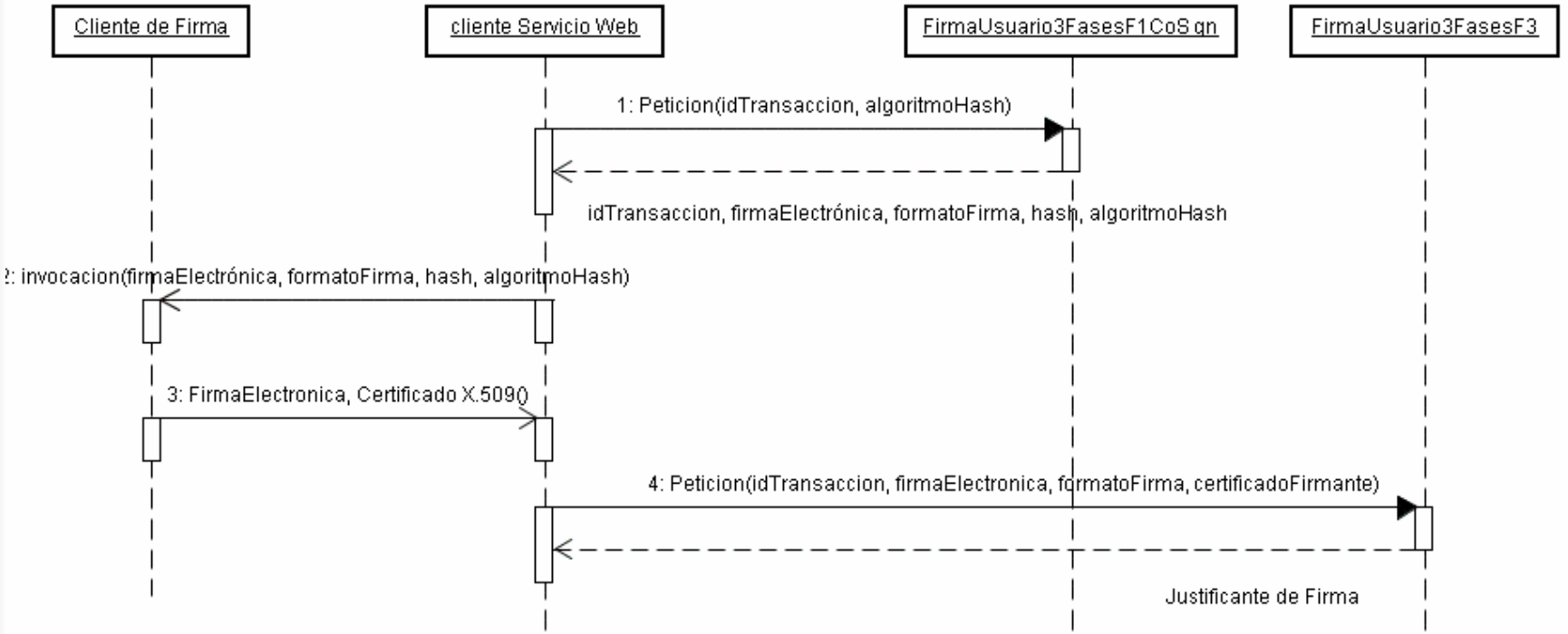
5.d. Descripción de servicios de la plataforma Firma electrónica (VI)

Firma electrónica de usuario en 3 fases simple



5.d. Descripción de servicios de la plataforma Firma electrónica (VII)

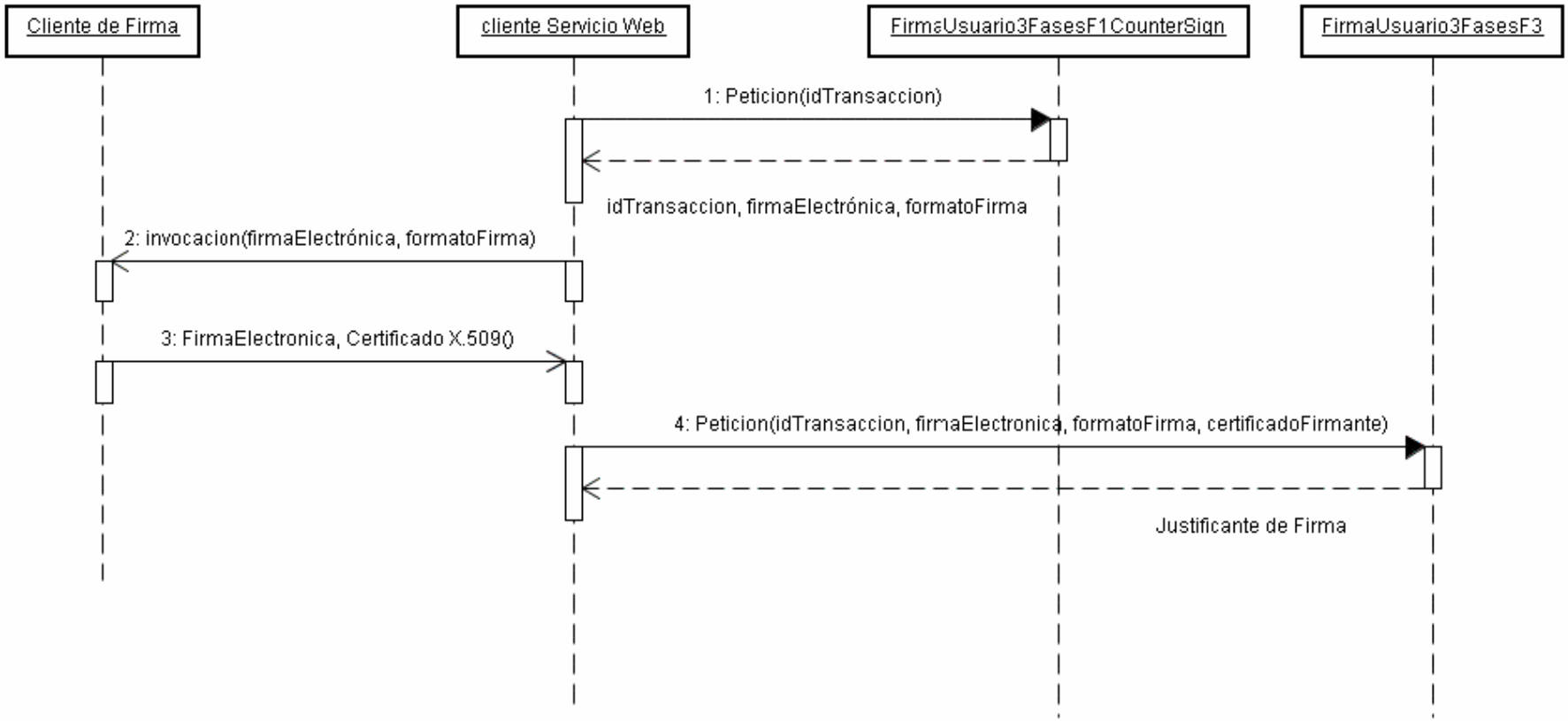
Firma electrónica de usuario en 3 fases Cosign



JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

5.d. Descripción de servicios de la plataforma Firma electrónica (VIII)

Firma electrónica de usuario en 3 fases Countersign



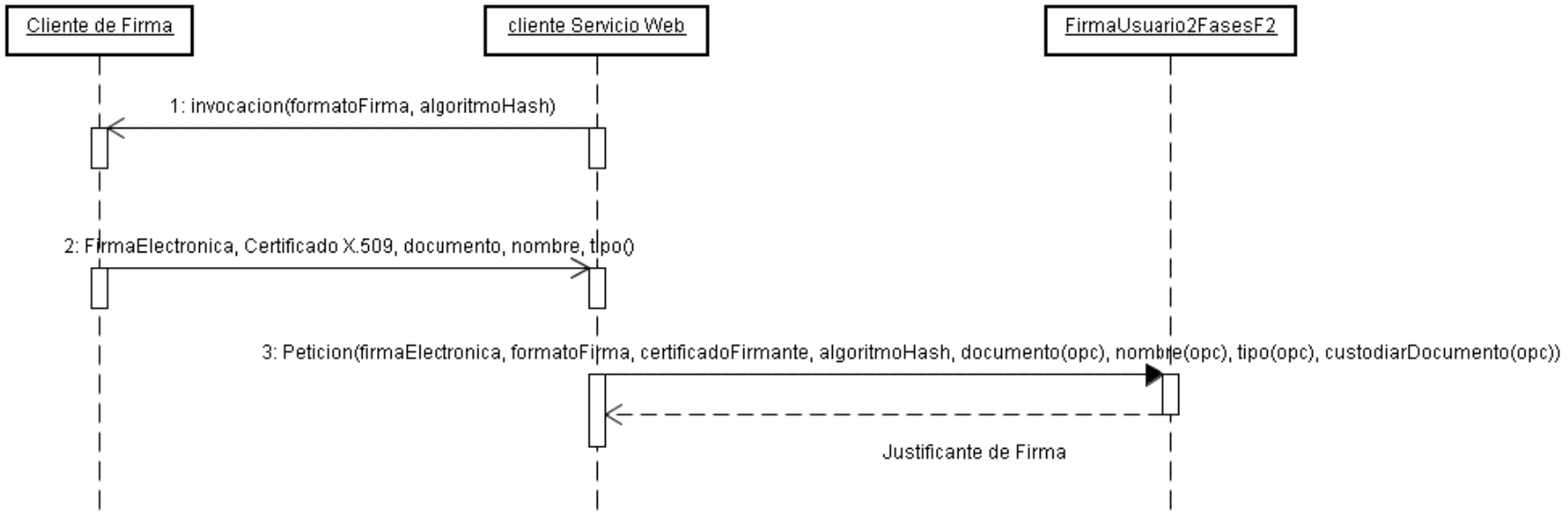
5.d. Descripción de servicios de la plataforma Firma electrónica (IX)

Firma electrónica de usuario en 2 fases

- Requiere el uso del cliente de firma de la plataforma.
- Los datos a firmar pueden no ser almacenados en la plataforma.
- Modo de firma introducido en la versión 5 de @firma. Permite la firma de ficheros de gran tamaño.
- Se recomienda el uso de dicho tipo de firma frente a la firma de usuario en 3 fases. Reduce el consumo de recursos de la plataforma.

5.d. Descripción de servicios de la plataforma Firma electrónica (X)

Firma electrónica de usuario en 2 fases



JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

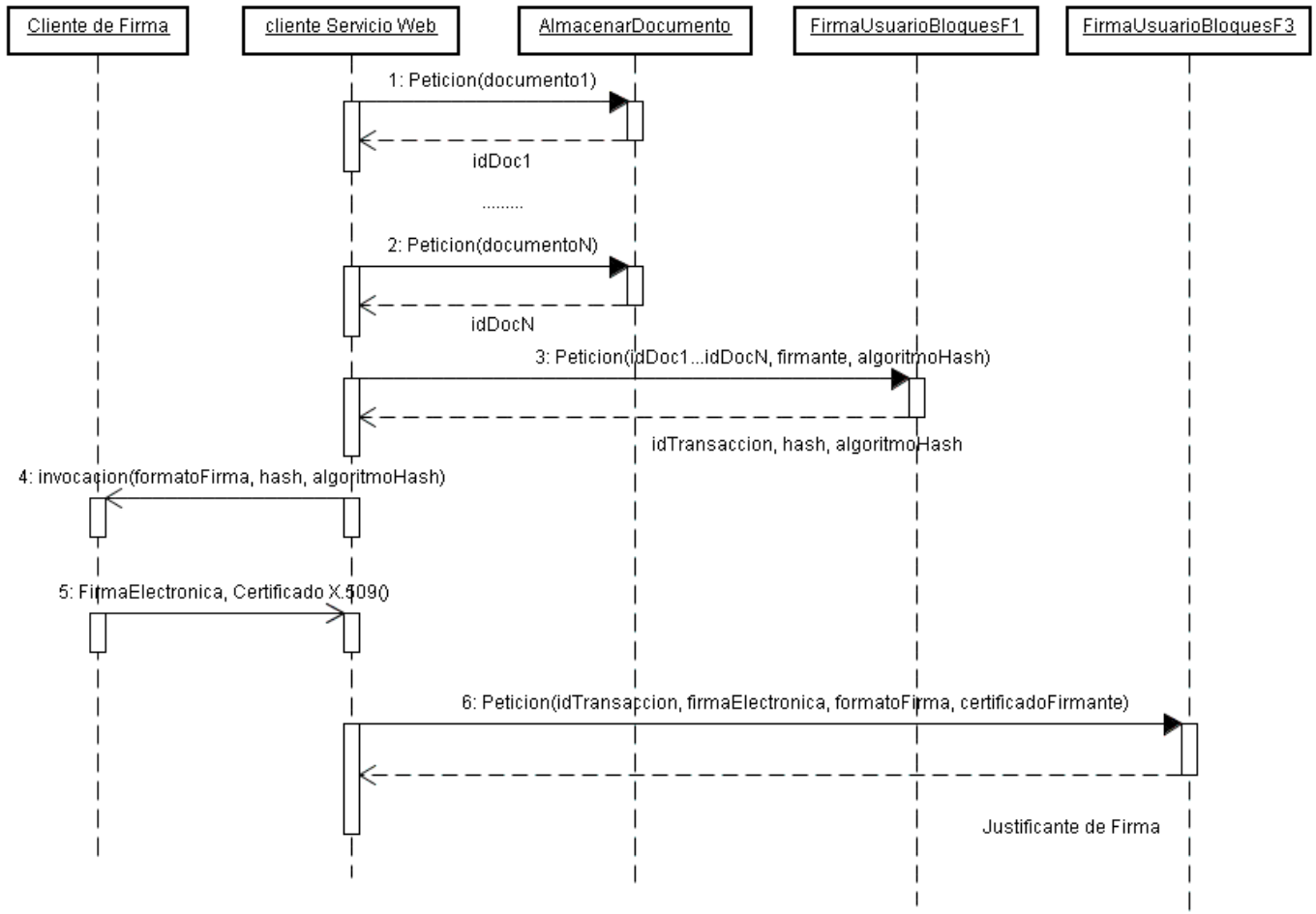
5.d. Descripción de servicios de la plataforma Firma electrónica (XI)

Firma electrónica de usuario por bloques

- Facilita los procesos de firma en los que existan múltiples documentos a firmar, al evitar al usuario la repetición de los pasos de firma para cada documento.
- Posee todas las características del proceso de firma individual.
- El bloque consta de la firma electrónica de servidor de los documentos que lo componen. El hash del bloque es utilizado para completar el proceso de firma.
- Los bloques de firma generados por @firma v5 difieren de los de @firma v4, por lo que existen servicios diferenciados para cada uno de ellos.
- Se ha mejorado la construcción de bloques respecto a @firma v4, permitiendo en la generación de un bloque la incorporación de otros bloques y de documentos firmados por otros usuarios e incluidos en otros bloques (multifirma).

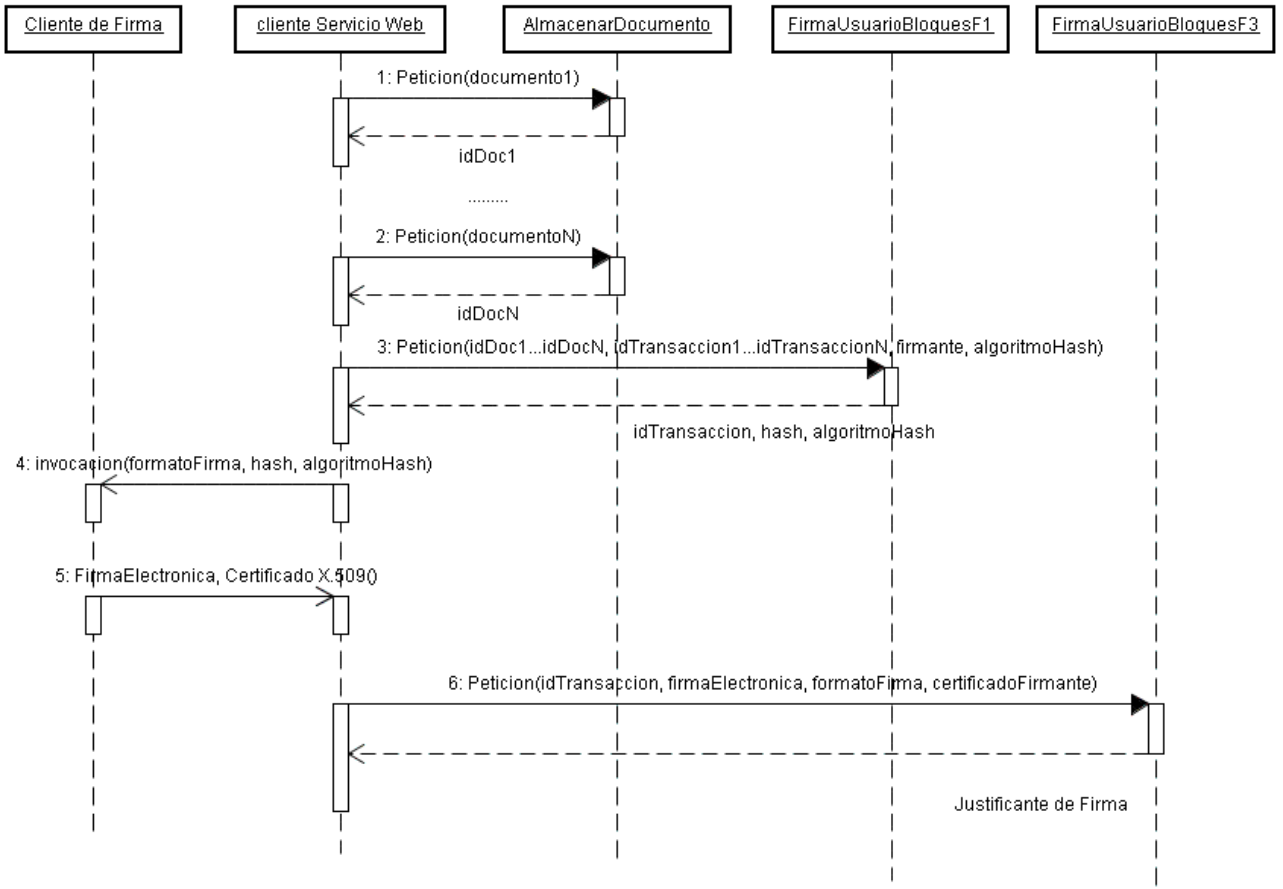
5.d. Descripción de servicios de la plataforma Firma electrónica (XII)

Firma electrónica de usuario por bloques



5.d. Descripción de servicios de la plataforma Firma electrónica (XIII)

Firma electrónica de usuario por bloques multifirmados



5.d. Descripción de servicios de la plataforma Firma electrónica (XIV)

Servicios adicionales sobre firmas electrónicas de usuario por bloques

- Existen servicios específicos según la versión de la plataforma en la que se generó la firma.
  - ObtenerIdDocumentosBloqueFirmas.
  - ObtenerIdDocumentosBloqueFirmasBackwards.
  - ObtenerInformacionBloqueFirmas.
  - ObtenerInformacionBloqueFirmasBackwards.

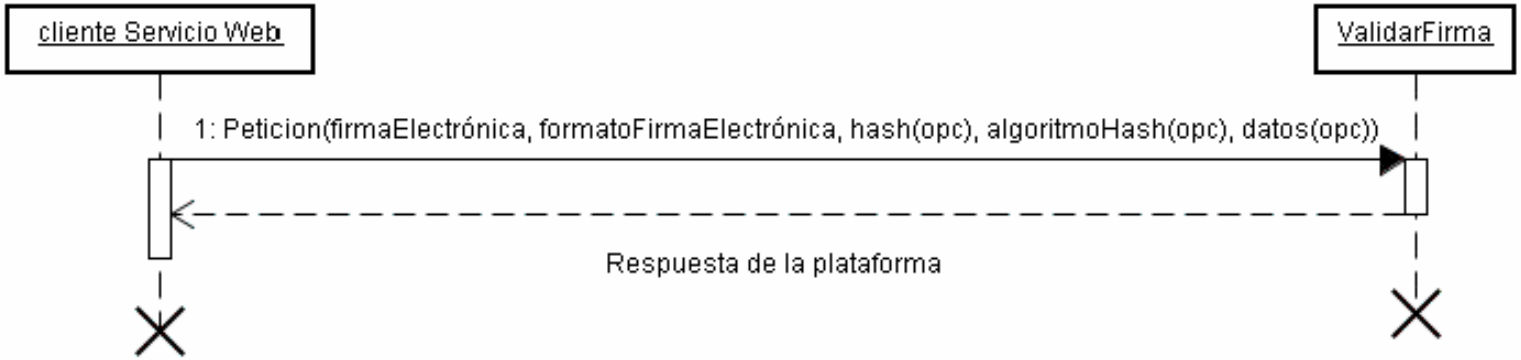
5.d. Descripción de servicios de la plataforma Firma electrónica (XV)

Validación de firma electrónica

- Permite realizar una validación completa de la firma electrónica proporcionada al servicio: validación de la firma, validación del certificado X509 y soporte del certificado en la plataforma.
- Puede ser utilizado tanto para validar firmas electrónicas generadas por la plataforma como aquellas ajenas a @firma 5, siempre que su formato sea soportado.

5.d. Descripción de servicios de la plataforma Firma electrónica (XVI)

Validación de firma electrónica



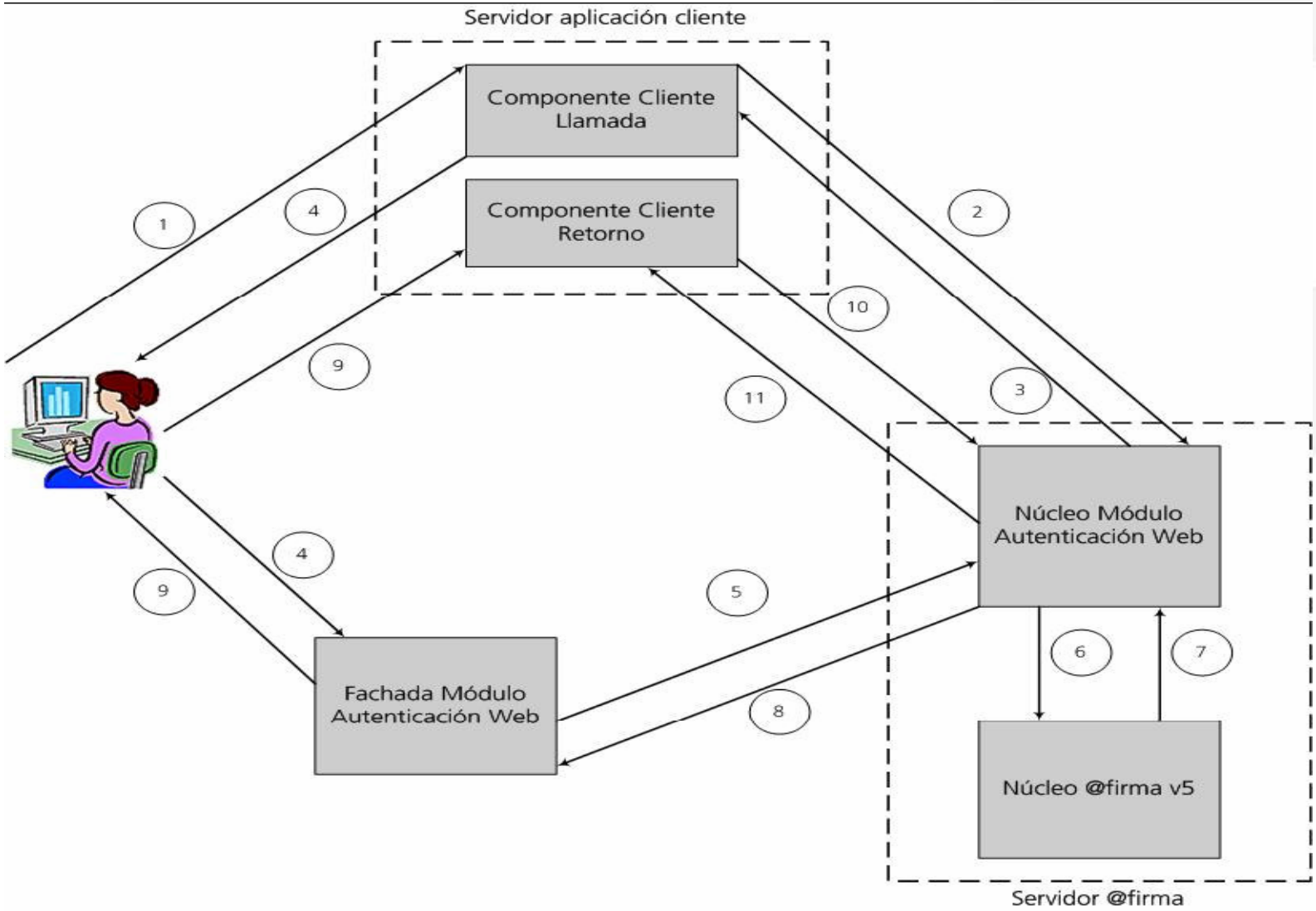
5.d. Descripción de servicios de la plataforma
Servicios de custodia

- Custodia es la base de datos de la plataforma @firma 5 empleada para almacenar información referente a transacciones de firma electrónica.
- Las aplicaciones cliente tienen acceso única y exclusivamente a las transacciones de firma electrónica llevadas a cabo por ellas.
- Servicios disponibles
  - Almacenar un documento (AlmacenarDocumento).
  - Eliminar el contenido de un documento (EliminarContenidoDocumento).
  - Actualizar referencia externa de transacción (ActualizarReferencia).
  - Obtener el identificador de un documento firmado (ObtenerIdDocumento).
  - Obtener el contenido de un documento (ObtenerContenidoDocumento).
  - Obtener el contenido de un documento firmado (ObtenerContenidoDocumentoId).
  - Obtener las transacciones de firma electrónica de una aplicación (ObtenerTransacciones).
  - Obtener las transacciones de firma electrónica en un rango de fechas (ObtenerTransaccionesPorFecha).

5.d. Descripción de servicios de la plataforma Servicios de custodia (I)

- Obtener las transacciones de firma electrónica por referencia (ObtenerTransaccionesReferencia).
- Obtener la firma electrónica de una transacción (ObtenerFirmaTransaccion).
- Obtener el bloque de firmas de una transacción (ObtenerBloqueFirmas).

5.e. Fachada de tickets



5.e. Fachada de tickets (I)

1) El navegador cliente accede al Componente Cliente de Llamada de la aplicación en la que desea autenticarse.

2) El Componente de Llamada realiza una llamada Web Service al núcleo del módulo de Autenticación Web pasándole como parámetros: Identificador de aplicación Y Identificador de sesión Web.

3) El núcleo del módulo de Autenticación Web inicia una transacción de autenticación y genera un ticket o identificador de transacción.

4) Dicho ticket será empleado en el componente de llamada, que junto con el identificador de aplicación, identificador de sesión Web y URL del componente de retorno será lo único que circule entre Aplicación <-> Usuario <-> Fachada del módulo de Autenticación Web.

5) Se almacena el ticket unto con el identificador de aplicación en la sesión Web y posteriormente se envía al usuario el ticket generado, junto con el identificador de aplicación y sesión Web, que lo redirige al componente Fachada del módulo de Autenticación Web para obtener su certificado.

6) El componente fachada obtiene el certificado, el ticket de la llamada, el identificador de aplicación y de sesión y realiza una petición de validación de certificado asociado a ticket al núcleo del módulo de Autenticación Web.

5.e. Fachada de tickets (II)

7) El núcleo del módulo de Autenticación verifica que el ticket generado es válido y procede a realizar la validación del certificado mediante el núcleo de la plataforma.

8) La respuesta obtenida es asociada al ticket, indicado en la solicitud de validación, por el núcleo del módulo de Autenticación Web.

9) El componente fachada recibe el resultado de la validación y redirige el flujo de comunicación, a través del usuario, al componente de retorno. A la URL indicada le serán concatenados un código de finalización, el identificador de ticket y sesión Web.

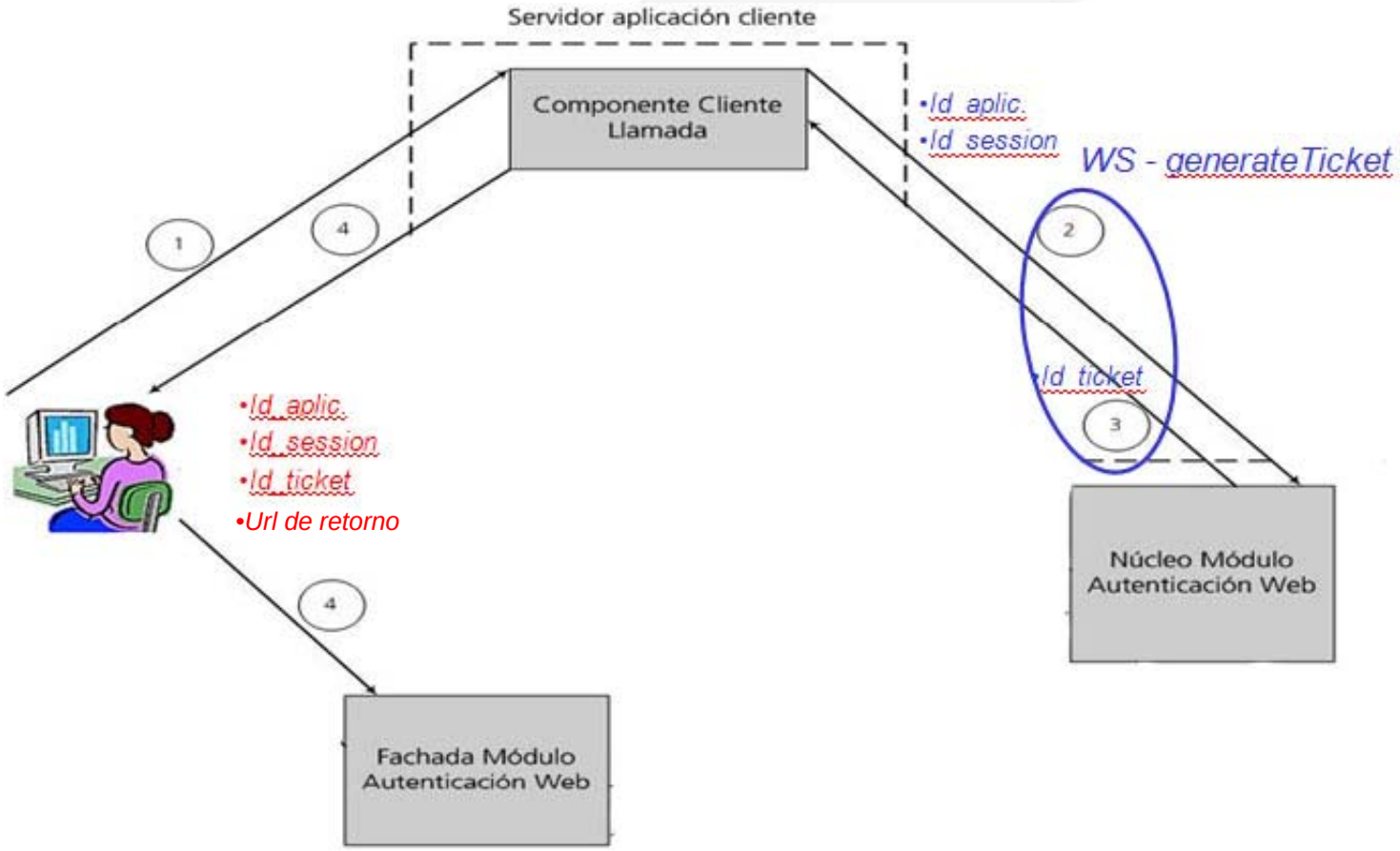
10) El componente de retorno de la aplicación recibe el código de finalización, comprueba que es correcto, valida los parámetros presentados en la URL con los almacenados en sesión y, si todo es correcto, procederá a consultar al núcleo del módulo de Autenticación Web los datos del usuario asociados al ticket presentado mediante una llamada Web Service.

11) El núcleo del módulo de autenticación verifica si existe información disponible para el ticket solicitado y se la devuelve a la aplicación. La información devuelta será el resultado de la autenticación y mediante la cual la aplicación podrá iniciar la fase de autorización.

5.e. Fachada de tickets (III)

Componente Cliente - Solicitar Ticket

Solicitamos un id_ticket al núcleo y se inicia una autenticación



5.e. Fachada de tickets (IV)

SolicitarTicket

1) Llamada WS al modulo de autenticación para obtener un nuevo ticket

Datos Necesarios : Id Aplicación e ID session → Estructura XML (input) → Llamada al servicio → Estructura XML (output) → Objeto Map

2) Extraemos el código de ticket, lo guardamos en la request

3) Codificamos los campos de la URL y Redirigimos el flujo de comunicación hacia el interfaz web del módulo de autenticación para realizar la validación del certificado

```
https://"+authServerIP+":"+authServerSecurePort+"/authenticationFacade?
                        action=validateCert
                        &ticketId=ticketId
                        &appId=appId
                    &webSessionId=sessionId
                &comeBackURL=comeBackURLEncoded
```

JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

5.e. Fachada de tickets (V)

SolicitarTicket

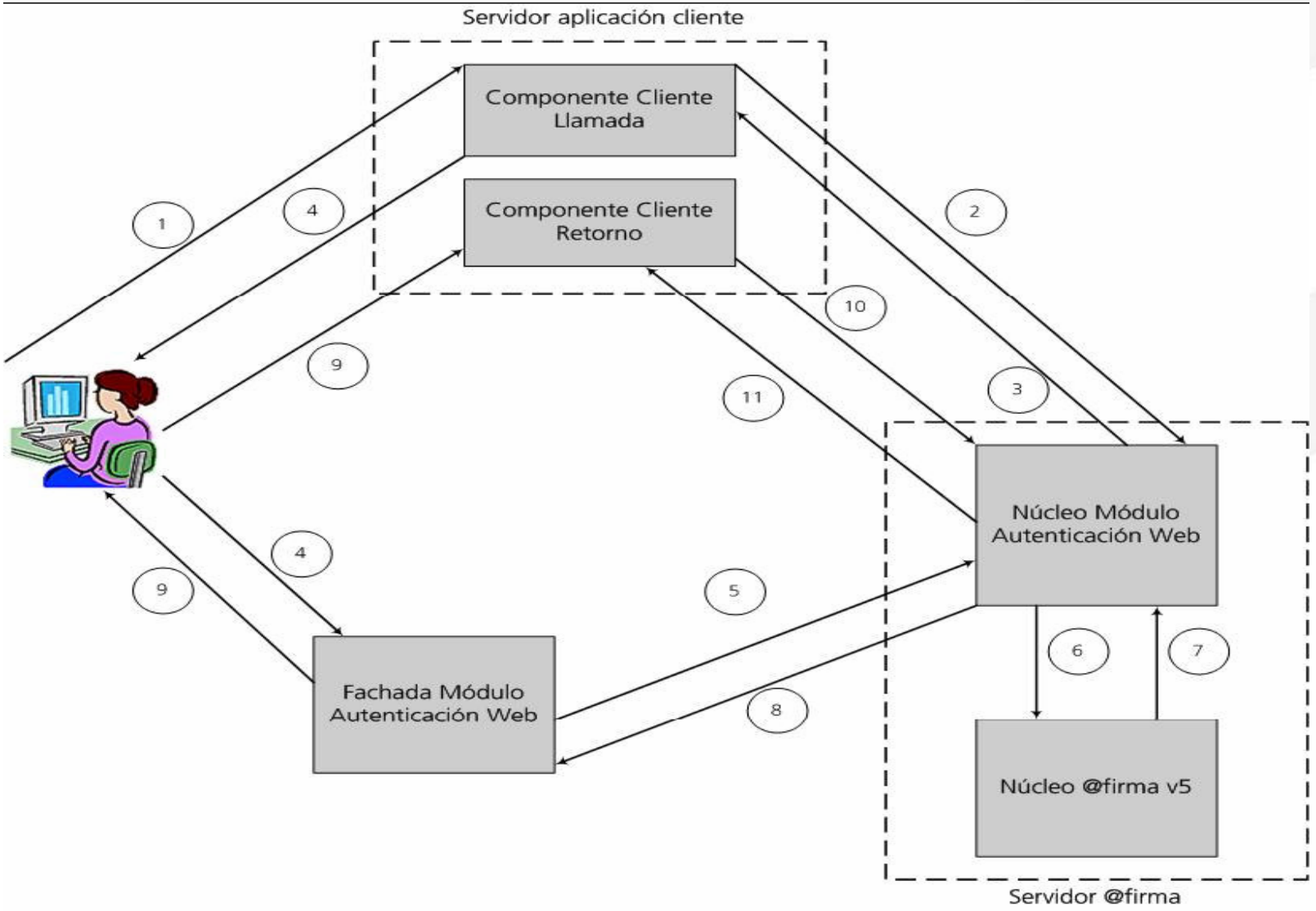
Comprobar código fuente de la aplicación de prueba

es.andaluciajunta.cjap.autenticacion.SolicitarTicket.java

```
//Efectuamos una llamada web service al modulo de autenticación para obtener
    un nuevo ticket
UtilidadesAutFachada util = new UtilidadesAutFachada();
Map ticketResp = util.generateTicket(appId,sessionId);
```

En UtilidadesAutFachada podemos ver generateTicket(appIf,sessionId), Método que crea el XML para realizar la petición del servicio, obtiene un XML de respuesta y lo mapea a un MAP

5.e. Fachada de tickets (VI)

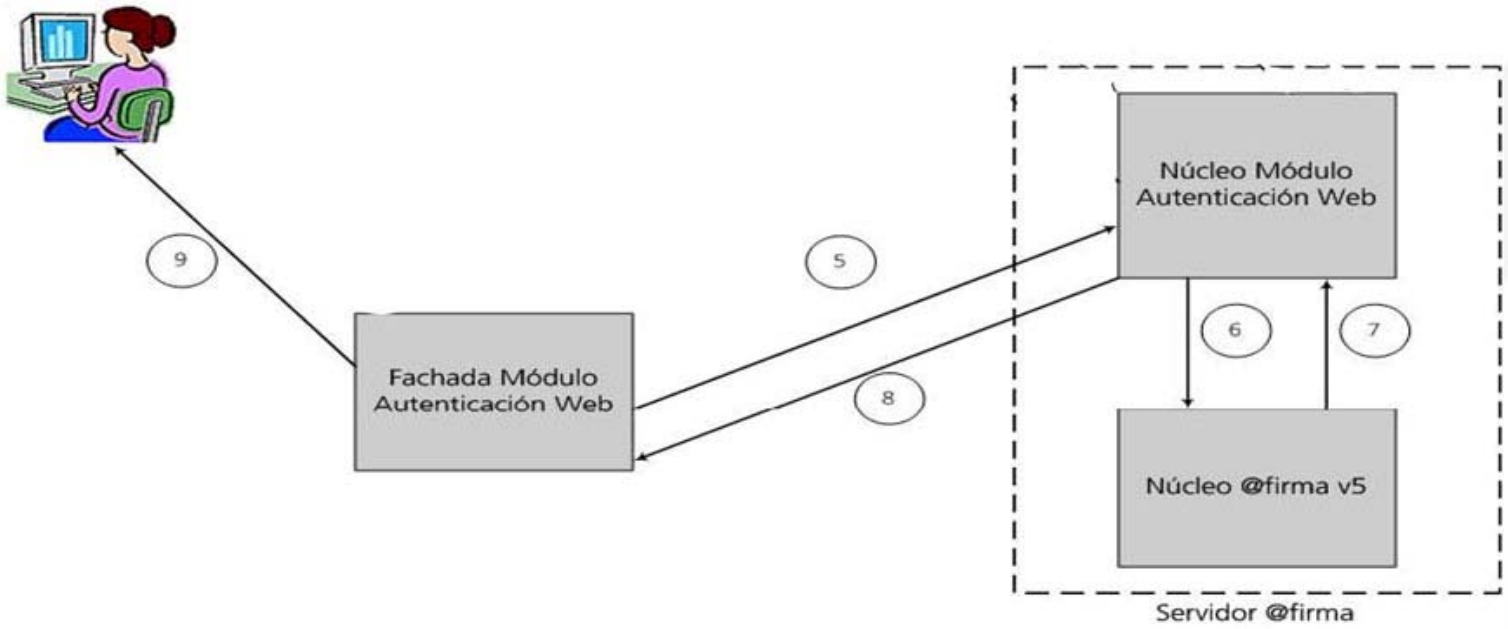


5.e. Fachada de tickets (VII)

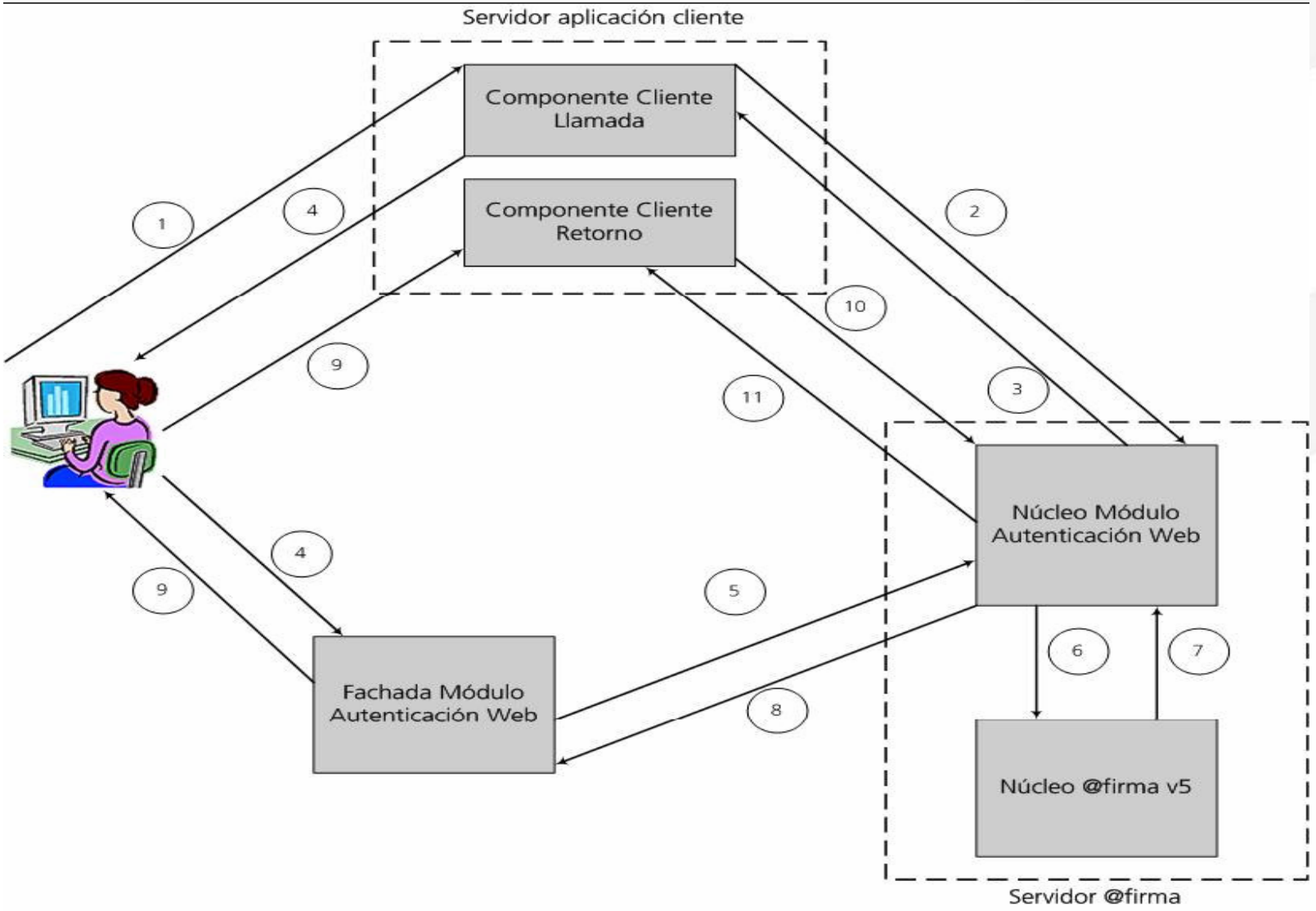
Fachada Modulo Autenticación Web ticket

Envía el certificado

Devuelve código de confirmación



5.e. Fachada de tickets (VIII)



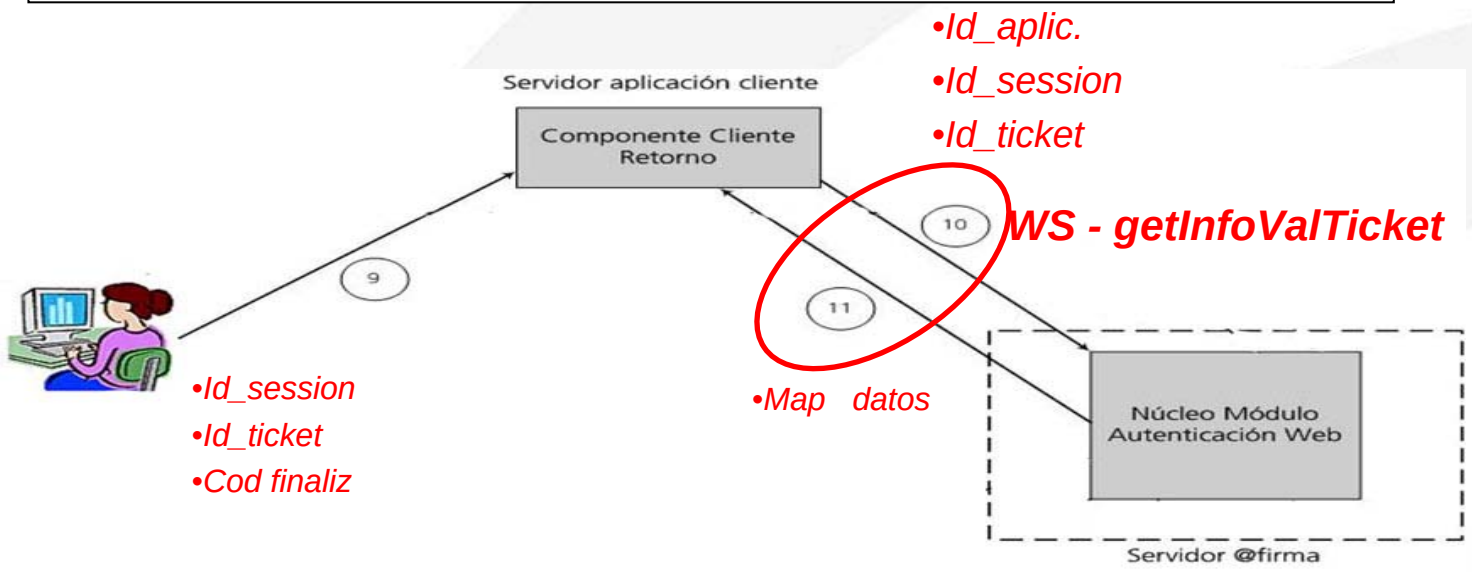
5.e. Fachada de tickets (IX)

Componente Retorno - Datos Ticket

1) Recogemos los datos de la request

2) Acudimos al nucleo y extraemos los datos

3) Mapeamos al objeto que deseemos



5.e. Fachada de tickets (X)

DatosTicket

1) Establecimiento de variables extraídas de la petición por la request

`rErrorCode` *`(error devuelto por módulo),`* `rTicketId, rAppId y rSessionId`

2) Establecimiento de variables extraídas de la sesión (sTicketId y sAppId) y borradas después de la sesión

3) Comprobación de los campos (rErrorCode, rTicketId, rAppId y rSessionId)

4) Si todo OK, Llamada WS al modulo de autenticación para obtener la información de validación de certificado asociada al ticket de la petición

Datos Necesarios : rTicketId, rAppId y rSessionId → Estructura XML (input) → Llamada al servicio → Estructura XML (output) → Objeto Map

5) Tratar el objeto Map para obtener los datos del certificado

JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

5.e. Fachada de tickets (XI)

```
//Efectuamos una llamada web service al módulo de autenticación para
   obtener la información de validación de certificado asociada al
   ticket de la petición
   UtilidadesAutFachada util = new UtilidadesAutFachada();
   ticketValInfoResp =
   util.getInfoValTicket(rTicketId,rAppId,rSessionId);
-------------------------------------------------------------------
Map resValInfo =
   TransformersFacade.getInstance().parseResponse(valInfo,"ValidarCertif
   icado",TransformersConstants.version10);
-------------------------------------------------------------------
  validacion = UtilidadesAutFachada.mapeoValInfoResultadoVal(resValInfo);
   if ((validacion == null)||(! "0".equals(validacion.getResultado()))){
         errorComp = UtilidadesAutFachada.resultadoVal(validacion);
   }else{  user = UtilidadesAutFachada.mapeoValInfoUser(resValInfo);
                request.getSession().setAttribute("usuario",user);   }
```

JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

5.e. Fachada de tickets (XII)

Comunicación segura

- **Por usuario/ password**
  - Además de llamar al servicio hay que incluir los parámetros de usuario/password, para que ningún otro servicio pueda utilizar el identificador,
  - **authorizationMethod = UsernameToken**
  - Clear/digest – Por defecto debe siempre enviarse digest
  - Pueden existir distintos usuarios/password
- **Por certificado**
  - Sería necesario enviar a la administración de la aplicación la parte pública del certificado, para identificar la petición.
  - **authorizationMethod = BinarySecurityToken**
- **Ambas comunicaciones con el núcleo deben tener el mismo sistema de seguridad ya que sólo existe una configuración, si podrían utilizar user/pass o certificados distintos**

5.e. Fachada de tickets (XIII)

Adaptación de la aplicación

1) Incluir en nuestra aplicación las **librerías** que no tengamos previamente en nuestro proyecto
2) Incluir la carpeta **xml_parameter**,
3) Incluir y modificar los siguientes .**properties**
    a) **afirma5ServiceInvoker.properties**
        Modificar → **com.trustedstore** y **com.trustedstorepassword**
    b) **transformers.properties**
        Modificar → **xmlTemplateFilesPath**
    c) .properties de la aplicación
        Incluir → authServer = https://ws116.juntadeandalucia.es/
        Incluir → comeBackURL = http://aaa.xxxxxxxxxxxxxxx.zzz/
        Incluir → appId = BBB.ccccc

JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

5.e. Fachada de tickets (XIV)

Adaptación de la aplicación (I)

Ahora podríamos nosotros desarrollar nuestro código partiendo del ejemplo o utilizar el código que tenemos incluyéndolo en nuestra aplicación.

IMPORTANTE, es un código de ejemplo, puede ser revisado y adaptado a las necesidades y tecnologías utilizadas en nuestra aplicación.

4) Copiar el paquete es.andaluciajunta.cjap.autenticacion;

- **SolicitarTicketServlet.java**,
  - Modificar → archivo de propiedades de la aplicación

```
static ResourceBundle bun =ResourceBundle.getBundle(“xxxxxxxxxx");
```

  - Modificar → dirección de retorno.

```
private static final String comeBackURL =
bun.getString("comeBackURL")+"/autFachadaTicket/DatosTicket";
```

  - Modificar → página de error de autenticación

```
response.sendRedirect("callClientComponent.jsp");
```

JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

5.e. Fachada de tickets (XV)

Adaptación de la aplicación (II)

- Incluir en el web.xml el servlet de SolicitarDatos
- UtilidadesAutFachada.java,
    - Modificar → el procedimiento,

```
public static CertificateUser mapeoValInfoUser(Map valInfo)
```

realiza la conversión del objeto map obtenido al objeto que nosotros utilicemos en nuestra aplicación, por lo que será necesario adaptarlo a los datos que necesitamos del certificado, y el tipo del objeto, además será necesario incluir los cambios en las clases desde las que se llame.

- ResultadoValidacion.java
- ValidacionSimple.java

5.e. Fachada de tickets (XVI)

Adaptación de la aplicación (III)

- DatosTicketServlet.java,
  - Puede ser el sustituto natural del RetornoCertificado (aplicaciones de ejemplo anteriores), si se utiliza struts, puede convertirse en un Action. Si continua como serlvet ponerlo en el web.xml sino en el strusts-config.xml
  - A esta clase una vez realizada las comprobaciones básicas se le puede añadir las comprobaciones de nuestra propia aplicación

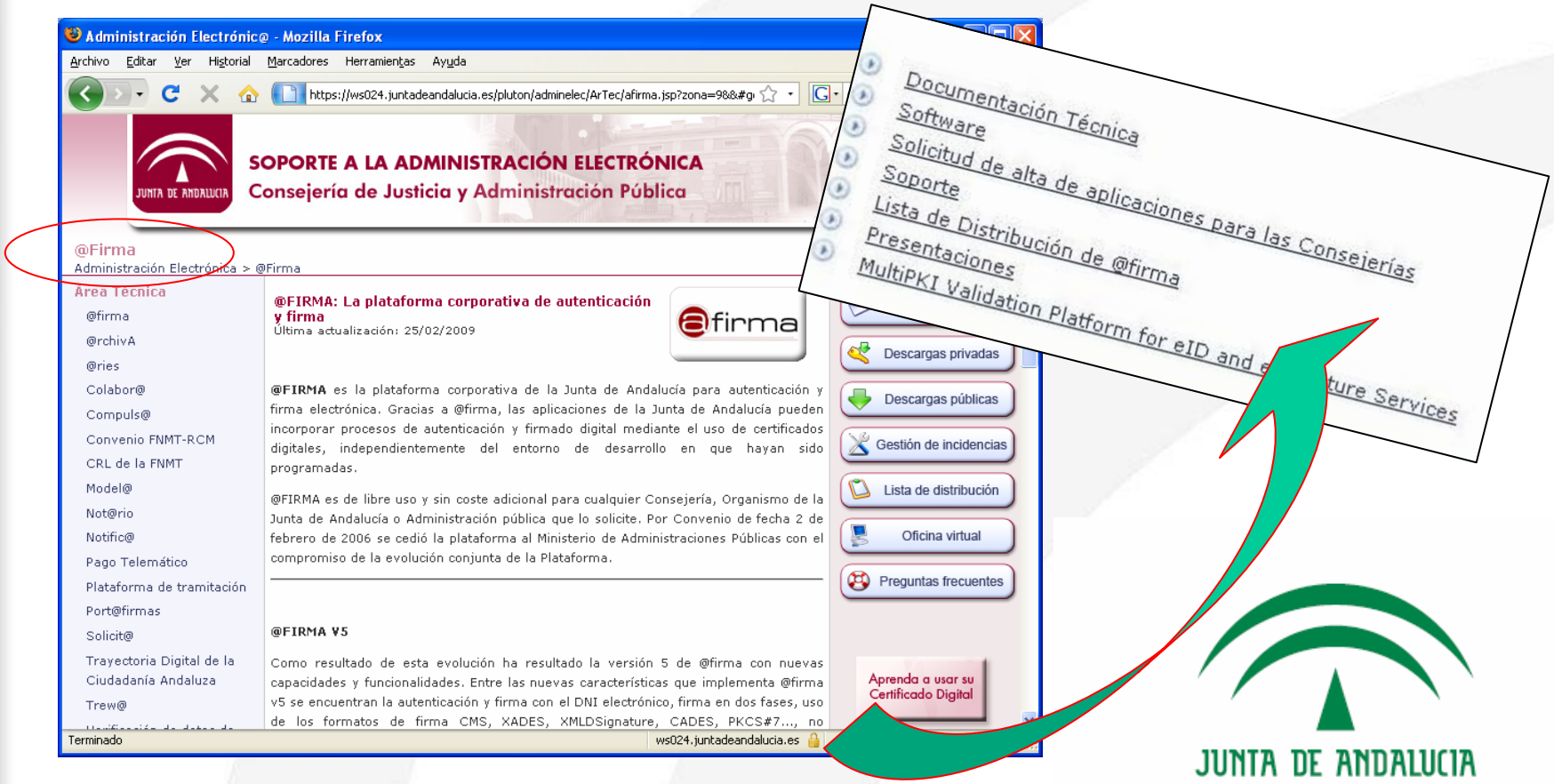
6. Preparación del entorno de pruebas

a. Recursos para el desarrollo.
b. Proceso de alta.
c. CD de desarrollo de @firma.

6.a. Recursos para el desarrollo

Web de Plutón - Recursos

https://ws024.juntadeandalucia.es/pluton



JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

6.a. Recursos para el desarrollo (I)

Gestor de Incidencias (iTracker)

https://ws025.juntadeandalucia.es/itracker

JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

6.b. Proceso de alta

Datos de configuración

Solicitud de Alta de Aplicaciones en @firma v5

Desde aquí puede acceder al manual para dar de alta una aplicación en @firma, donde se explica detalladamente los requisitos necesarios para solicitar el alta.

Enviar a la cuenta de correo de la Oficina de Administración Electrónica soporte.admonelectronica@juntadeandalucia.es escribiendo en el asunto "SOLICITUD ALTA AI información:

- Datos del solicitante: Nombre, apellidos, teléfono, e-mail.
- Nombre de la empresa.
- Consejería para la que se desarrolla y responsable de la Consejería.
- Nombre de la aplicación.
- Descripción de la aplicación.
- Responsable de la aplicación: Nombre, apellidos, teléfono, e-mail.
- ¿Requiere custodia de documentos? Sí/No (El uso de custodia de documentos limitará el tamaño de los ficheros a firmar a un máximo de 8Mb). En caso de no u conservar el mismo. Se recomienda no usar custodia de documentos por motivos de rendimiento y al no tener que enviarlo a través de la red para su custodia en
- Tipo de Autorización de acceso:
  - Autorización mediante usuario/clave
  - Autorización basada en certificados digitales (Recomendado)
- Política de TimeStamp:
  - Con TimeStamp
  - Sin TimeStamp
- Plataforma @firma en la que se solicita el alta: Desarrollo / Producción.
- Fecha prevista de puesta en producción.
- Volumen aproximado de accesos/día.
- Tipo de Servicio @firma solicitado:

JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

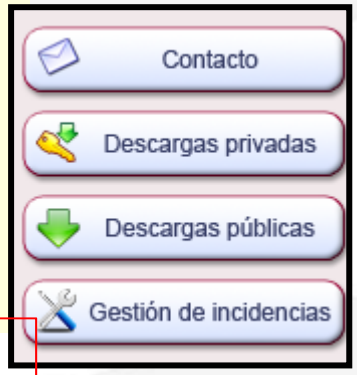
6.c. CD de desarrollo de @firma

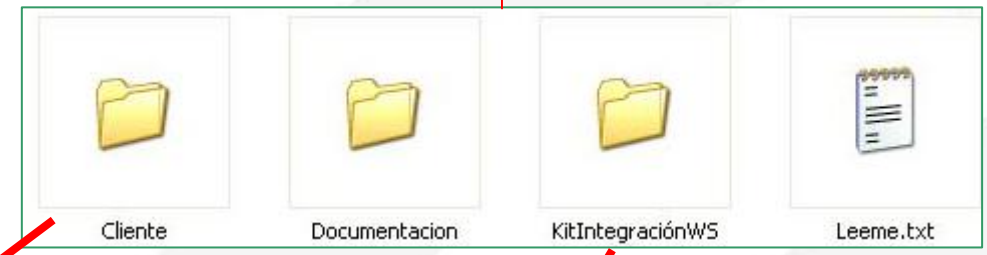
El Cd se obtiene a partir de las descargas privadas de Plutón.

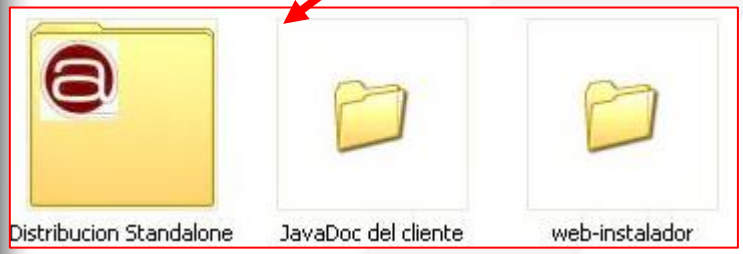
Enviando un email a soporte.admonelectronica@juntadeandalucia.es

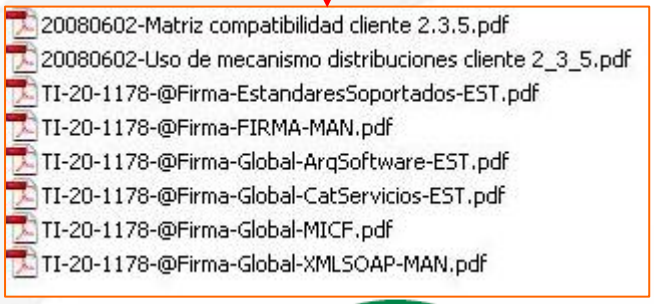
Con los siguientes datos: Nombre, apellidos, cargo y DNI de la persona que va a realizar la descarga. Indicando que se desea la descarga del CD de desarrollo

Este CD puede ser solicitado directamente por las empresas









JUNTA DE ANDALUCIA
CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA

6.c. CD de desarrollo de @firma (I)

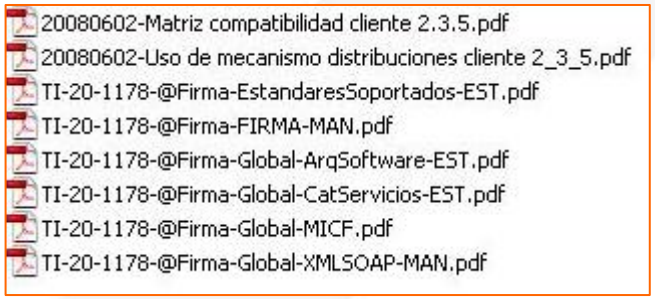
Carpeta Cliente

- Distribución Integrador: Contiene los ficheros a los que se hace referencia en el Manual del Integrador del Cliente de Firma (TI-20-1178-@Firma-Global-MICF) y algunos ejemplos de integración y uso del cliente.
- Distribución Standalone: Contiene una utilidad que instala el cliente de firma en el ordenador sin necesidad de descargarlo desde internet.
- JavaDoc del Cliente: Contiene el api del applet cliente.

6.c. CD de desarrollo de @firma (II)

Carpeta Documentación

- Documentación de la plataforma.



6.c. CD de desarrollo de @firma (III)

KitIntegracionWS

Contiene la información necesaria para conectarse con los WS de la plataforma. Se incluyen también ejemplos de uso de todos los servicios.

- **Ejemplos** → Ejemplos de uso de todos los WS de la plataforma. Para su uso leer el archivo README.
- **SOAP** → Ejemplos de mensajes SOAP de petición y respuesta.
- **Wsdl** → Ficheros de descripción de cada uno de los servicios web que publica la plataforma.
- **Xml** → Ficheros de ejemplo de xml de entrada para cada servicio web. También incluye el xml de respuesta.
- **Xsd** → XSchema general que han de cumplir los xml de entrada y salida de cada servicio web.