@firma

Seminario de Integración de Aplicaciones v5

Sevilla, 18 de Diciembre de 2.007

1. Introducción

2. Arquitectura de @Firma v5

3. Nuevas funcionalidades y capacidades de @Firma v5

4. Desarrollo de nuevas aplicaciones con @Firma v5

1. Introducción
2. Arquitectura de @Firma v5
3. Nuevas funcionalidades y capacidades de @Firma v5
4. Desarrollo de nuevas aplicaciones con @Firma v5
5. Ruegos y Preguntas

Introducción

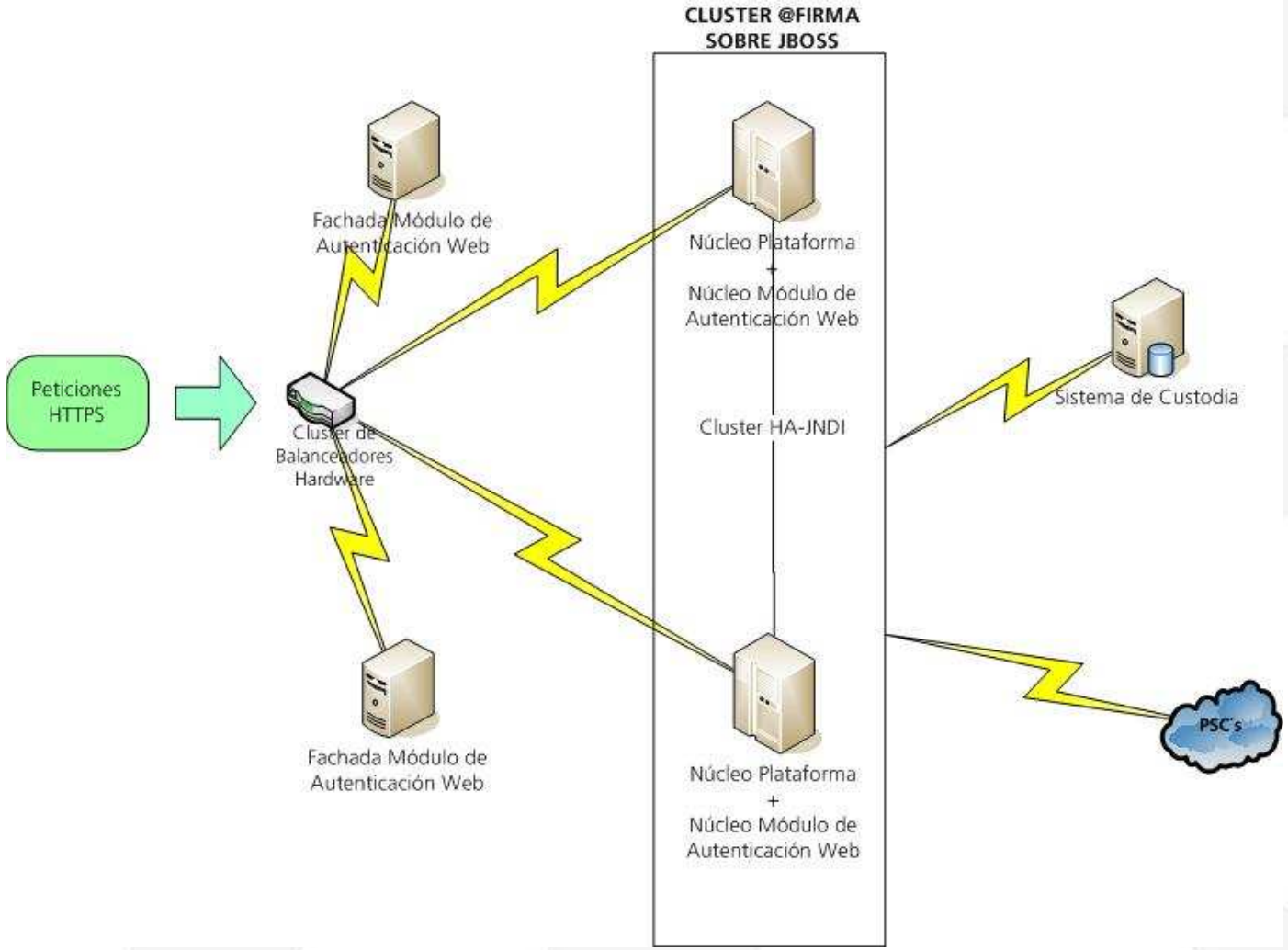
- ¿Qué es @Firma?

  Es una plataforma electrónica que utiliza certificados digitales X.509 v3 según las principales recomendaciones y estándares internacionales (RFC 2360, 3280, ETSI TS 101 733 v1.5.1, etc.) para la generación y validación de firmas digitales en múltiples formatos (CMS, XADES, XMLDSignature…), así como la validación avanzada de certificados digitales para garantizar en todo momento la integridad y validez de los mismos en el momento de la realización de una firma.

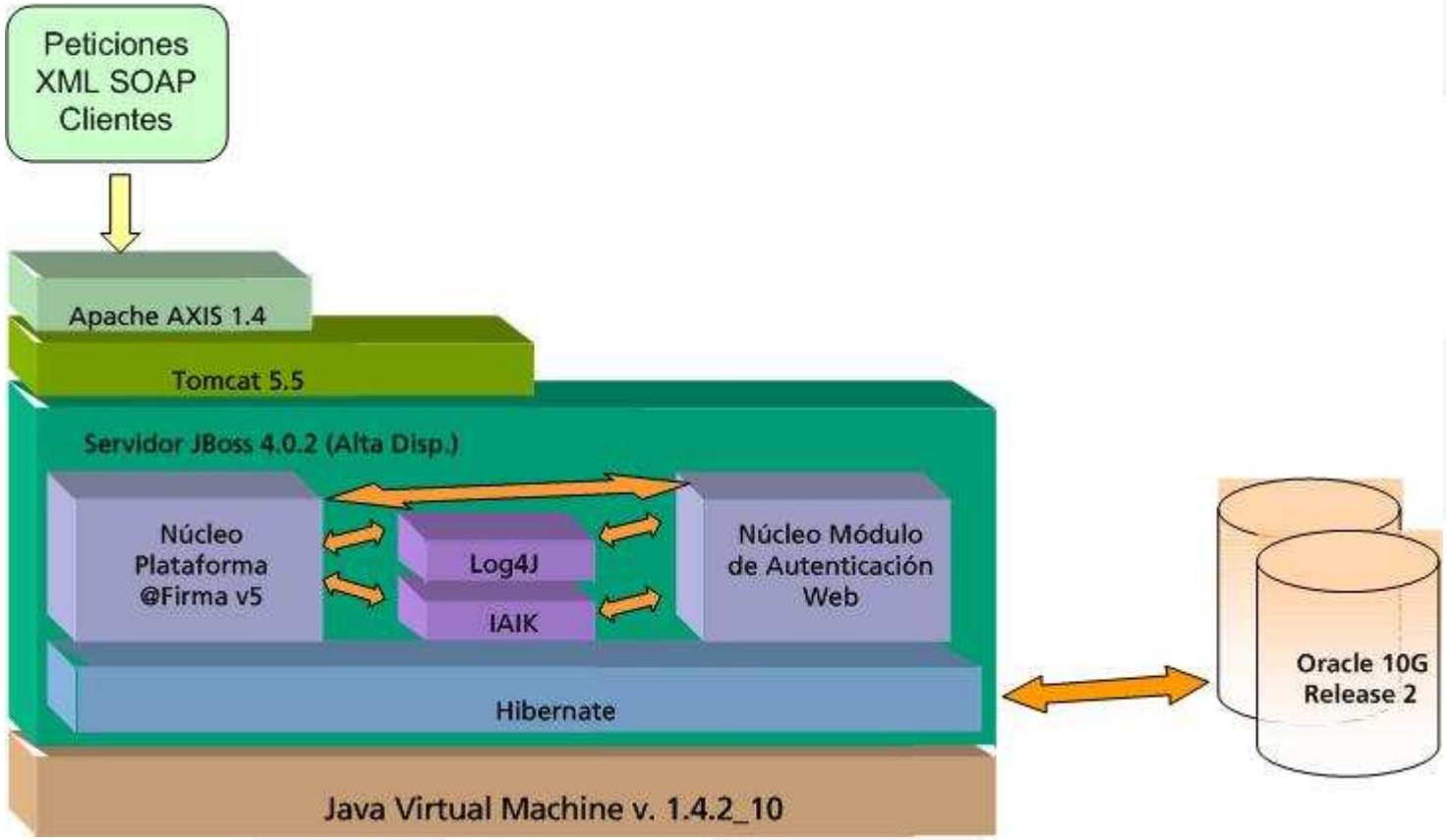
- Versiones de @Firma
  - V3.x (Fielato, Consejería de Hacienda)
  - V4.x (Junta de Andalucía, C. Justicia, C. Medio Ambiente, etc.)
  - V5.x (Ministerio de Administraciones Públicas)

1. Introducción
2. Arquitectura de @Firma v5
3. Nuevas funcionalidades y capacidades de @Firma v5
4. Desarrollo de nuevas aplicaciones con @Firma v5
5. Ruegos y Preguntas

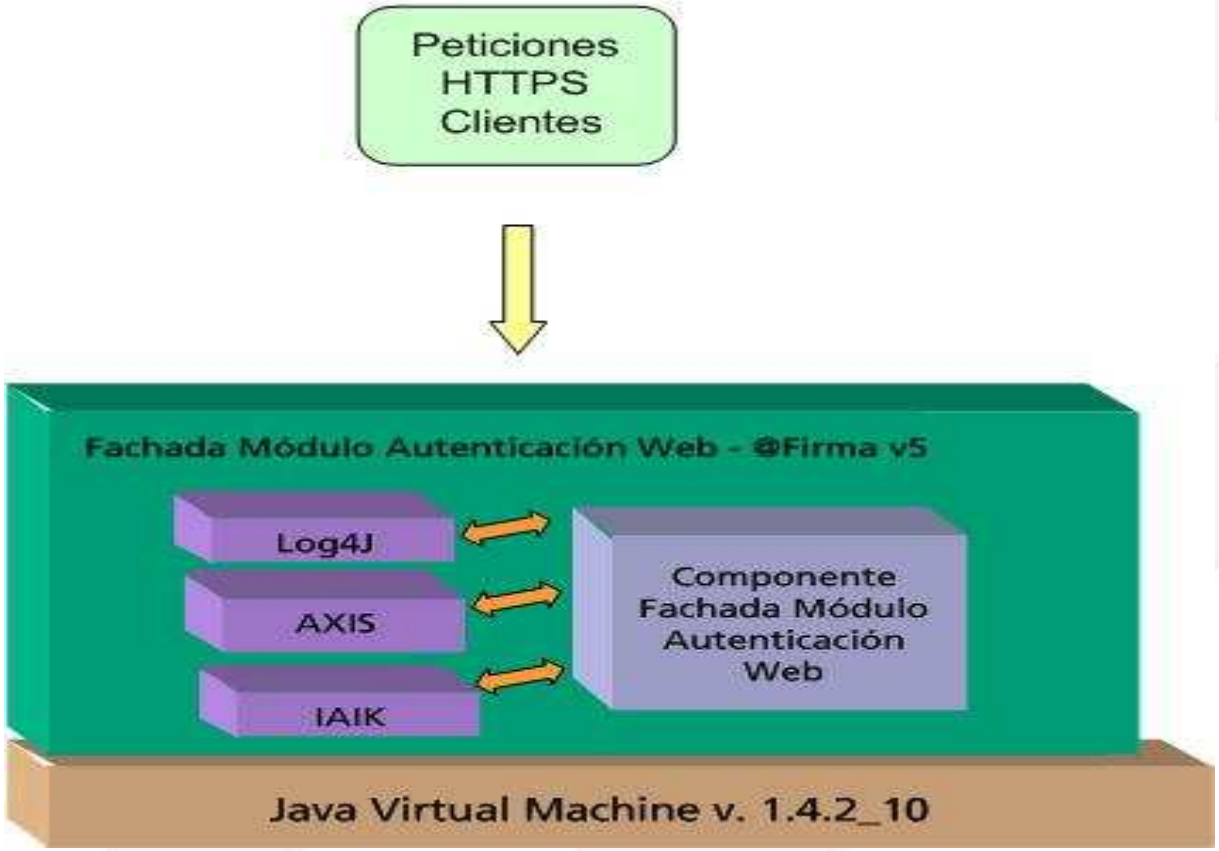
Arquitectura Física



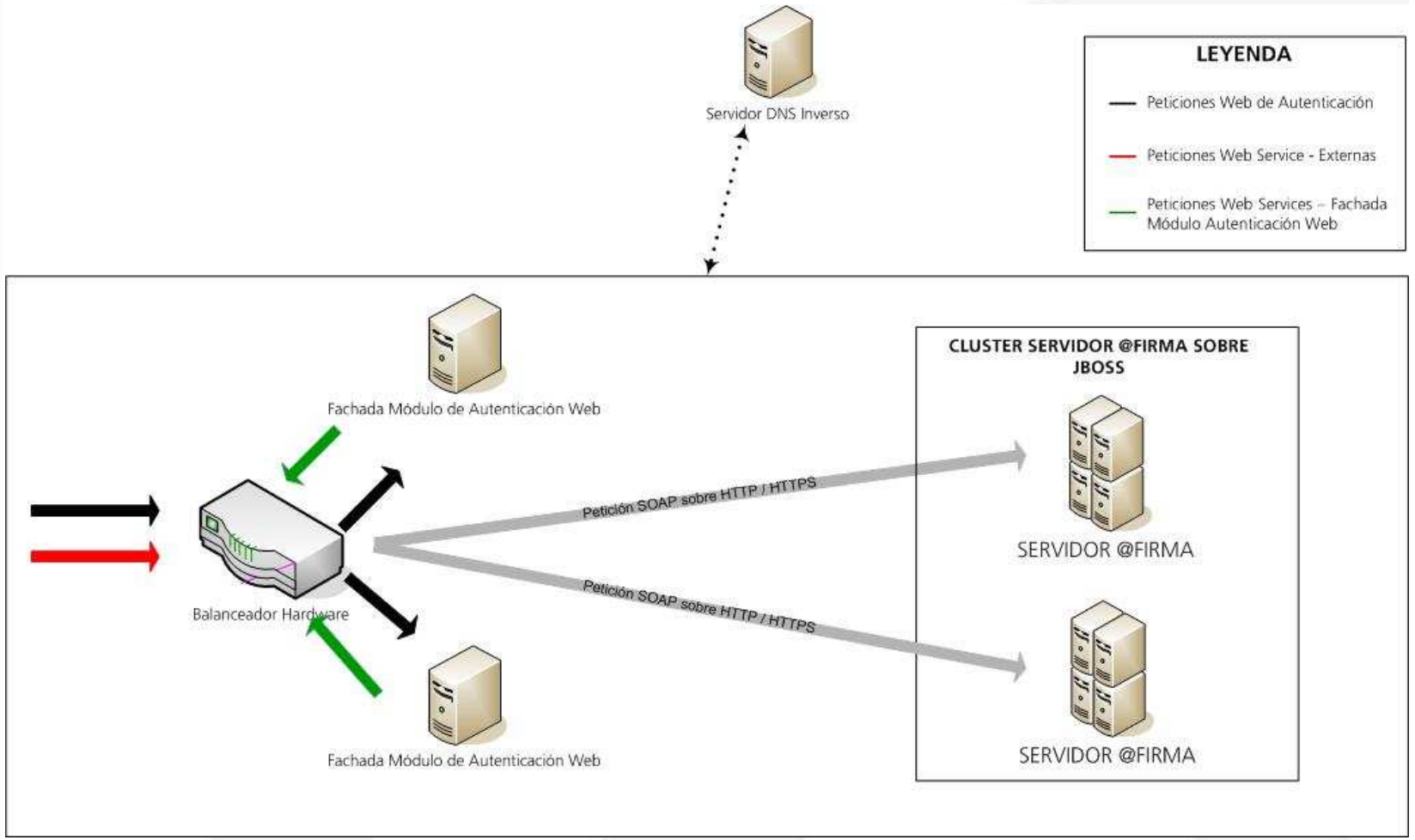
Arquitectura lógica – Núcleo



Arquitectura Lógica – Fachada Módulo de Autenticación Web



Arquitectura lógica – Configuración en Alta Disponibilidad



1. Introducción
2. Arquitectura de @Firma v5
3. Nuevas funcionalidades y capacidades de @Firma v5
4. Desarrollo de nuevas aplicaciones con @Firma v5
5. Ruegos y Preguntas

Servicios Web publicados por el núcleo de la plataforma (I)

- Módulo de Validación
  - Validación de Certificados.
  - Obtención de información de cerificados.

- Módulo de Firma
  - Firma y multifirma de ficheros.
  - Firma en bloque de ficheros y transacciones de firmas.
  - Validación de firmas:
    - Firmas.
    - Multifirmas.
    - Bloques de firmas.
  - Consultas sobre bloques de firma.

Servicios Web publicados por el núcleo de la plataforma (II)

- Módulo de Custodia
  - Sobre documentos:
    - Almacenamiento.
    - Borrado.
    - Consulta mediante varios criterios.
  - Sobre transacciones de firmas y bloques de firma:
    - Consultas mediante varios criterios.
    - Actualización de referencia externa.

- Módulo de Autenticación Web
  - Generación de tickets.
  - Actualización de tickets.
  - Consulta de tickets.

Cliente de firma digital

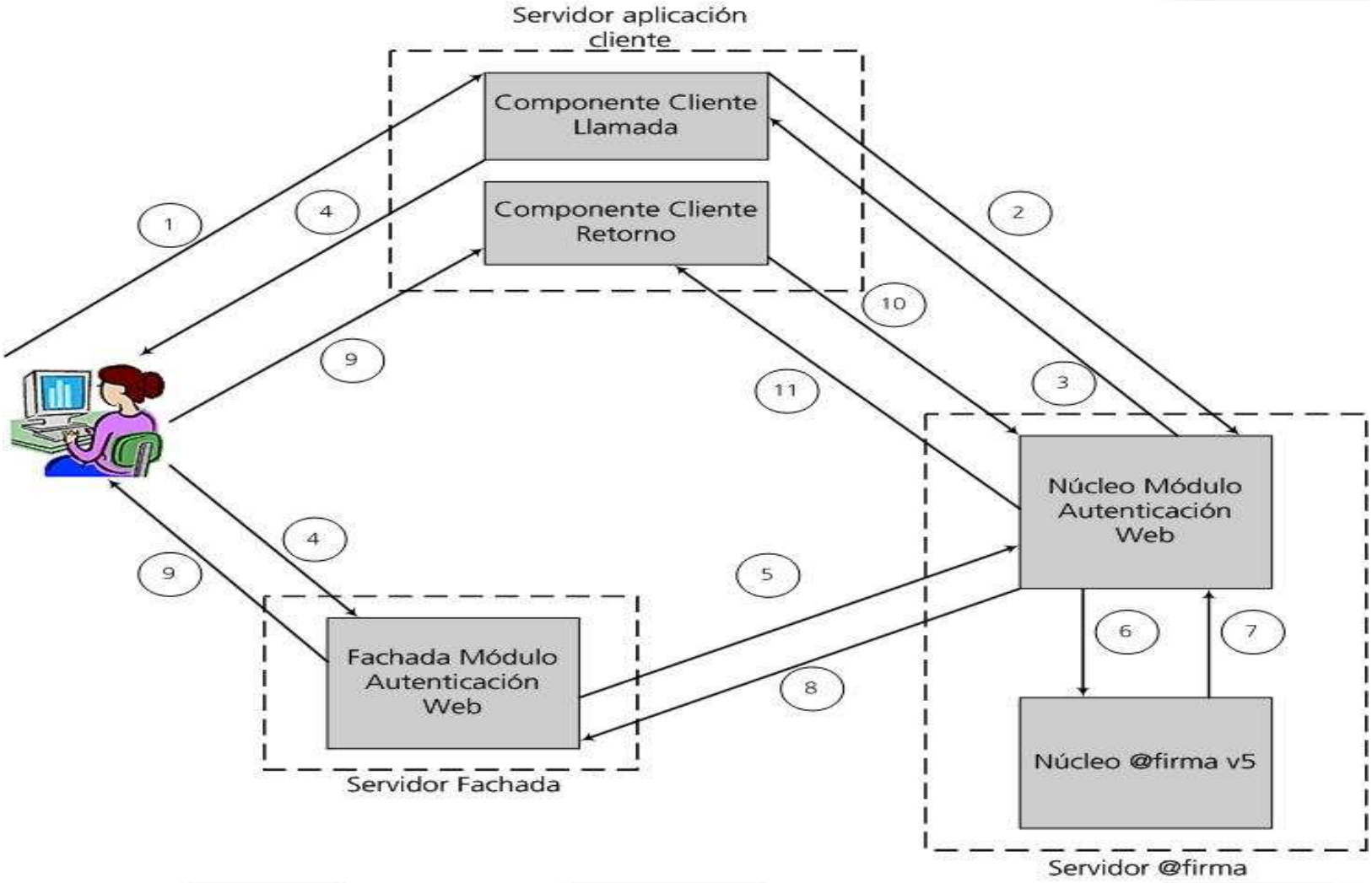
- Soportado en varios sistemas operativos y navegadores Web.
- Generación de diversos formatos de firma:
  - Binarios: PKCS7, CMS, …
  - XML: XMLDSig, XAdES, …
- Modos de firma:
  - Co y counter para formatos de firma binarios.
  - Permite realizar firma de ficheros, datos, Web y firma masiva.
- Funcionalidad criptográfica:
  - Generación de varios tipos de sobre digital (cifrado, firmado, cifrado y firmado).
  - Cálculo de varios tipos de resumen (MD5, SHA1, …).
  - Cifrado simétrico (TripleDES, AES, CAST5, TWOFISH, …).

1. Introducción
2. Arquitectura de @Firma v5
3. Nuevas funcionalidades y capacidades de @Firma v5
4. Desarrollo de nuevas aplicaciones con @Firma v5
5. Ruegos y Preguntas

Utilidades de desarrollo

- Apache – AXIS para java y C/C++. Motor SOAP que incluye herramientas para la generación de clientes a partir de un fichero WSDL.
- Java Web Services Development Pack (JWSDP).
- WASP sobre ECLIPSE.
- Gsoap . Permite generar clases cliente C/C++ a partir de un fichero descriptor WSDL.
- Microsoft Visual Studio .NET.
- NuSOAP. Permite desarrollar Web Services bajo el lenguaje PHP.
- …

- Autenticación Web mediante tickets



Validación de certificados mediante OCSP (I)

- OCSP (Online Certificate Status Protocol) se encuentra definido mediante el estándar RFC 2560.
- Permite obtener el estado de revocación de un certificado de forma on line.
- Protocolos de comunicación usados: HTTP y HTTPS .
- Ventajas respecto a la validación mediante CRL's:
  - Más fiable.
  - Más eficiente.
- Inconvenientes respecto a la validación mediante CRL's:
  - No todos los prestadores ofrecen este método de validación.

Validación de certificados mediante OCSP (II)

- Pasos para validar un certificado:

1. Formar la petición: OCSPRequest.
2. Firmar petición (opcional, soportado a partir de la versión 5.2).
3. Enviar mediante HTTP o HTTPS.
4. Esperar respuesta OCSPResponse.
5. Validar firma de la respuesta (opcional).
6. Obtener estado de revocación de la respuesta.

- Acciones previas:

  - Aplicación cliente:
    - Obtener clave pública del certificado de firma usado por el servidor OCSPResponder y establecer como certificado de confianza.
    - Establecer certificado firmante de peticiones.
  - Aplicación servidora:
    - Incorporar clave pública del certificado de firma de peticiones usado por cliente al almacén de confianza empleado.

Firma de ficheros (I)

• Código cliente:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTML>
<HEAD>
    <META http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <TITLE>Ejemplo Firma de ficheros en dos fases</TITLE>
    <script type="text/javascript" src="./common-js/instalador.js"></script>
    <script type="text/javascript" src="./common-js/appletHelper.js"></script>
    <script type="text/javascript" src="./common-js/time.js"></script>
    <script type="text/javascript" src="./common-js/utils.js"></script>
    <script type="text/javascript" src="constantes.js"></script>
```

Firma de ficheros (II)

• Código cliente (continuación):

```
<script type="text/javascript">
    function submitForm() {
        if(clienteFirma){
            clienteFirma.setFileuri(document.getElementById('file').value);
            clienteFirma.setSignatureFormat(document.getElementById('signFormat').value);
            clienteFirma.setSignatureMode('EXPLICIT');
            ok = clienteFirma.sign();
            if(ok){
                document.getElementById('eSignature').value =
                                                    clienteFirma.getSignatureBase64Encoded();
                document.getElementById('signerCert').value =
                                                    clienteFirma.getSignCertificateBase64Encoded();
                document.formulario.submit();
            } else {
                alert('Se produjo un error en la firma...');
                document.formulario.reset();
            }
        } else {
            alert('Error cargando el cliente de firma.');
        }
    }
</script></HEAD>
```

Firma de ficheros (III)

• Código cliente (continuación):

```
<BODY onload="cargarAppletFirma();">
<h2 align="center"><strong>Firma de usuario en dos fases</strong></h2>
<form action="/firma/afirma5FirmaUsuarioDosFases" method="post"
                                    enctype="multipart/form-data" id="formulario" name="formulario">
   <input type="hidden" name="afirmaService" value="FirmaUsuario2FasesF2">
   <input type="hidden" id="eSignature" name="eSignature" value="">
   <input type="hidden" id="signerCert" name="signerCert" value="">
   <table width="90%">
     <tr>
        <td width="20%" align="center"><b>Fichero a firmar: </b></td>
        <td><input type="file" id="file" name="file" value=""></td>
     </tr>
     <tr>
        <td width="20%" align="center">Formato: </td>
        <td><select id="signFormat" name="signFormat">
                  <option value="CMS" selected="selected">CMS</option>
                  <option value="XADES">XAdES</option>
               </select></td>
      </tr>
```

Firma de ficheros (IV)

- Código cliente (continuación):

```
      <tr>
        <td width="20%" align="center"><b>Referencia externa:</b></td>
        <td><input type="text" name="refId" value=""></td>
      </tr>
      <tr>
        <td width="20%" align="center"><b>Custodiar el documento firmado:</b></td>
        <td> <input type="radio" name="storeDoc" value="true" checked> Si
             <input type="radio" name="storeDoc" value="false"> No
        </td>
      </tr>
      <tr>
        <td width="50%" align="center">
          <input type="button" name="boton1" value="Firma en dos Fases“
            onClick="submitForm()"></td>
        <td><input type="reset" name="boton2" value="Limpiar formulario"></td>
       </tr>
   </table>
</form>
</BODY>
</HTML>
```

Firma de ficheros (V)

• Código servidor:

```
…
public class TwoPhasesUserSignWSServlet extends HttpServlet {
    …
    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        //Captura parámetros de la petición
        …
        twoPhasesUserSignResp = this.twoPhasesUserSignWS(requestParams);
        out.println(this.procesarRespuesta(twoPhasesUserSignResp));
    }

    private String twoPhasesUserSignWS(Map requestParams) throws Exception {
        //Método que realiza la llamada web service al núcleo de la plataforma para finalizar
        //y custodiar la transacción de firma
        …
        //Construir parámetro XML de entrada requerido por el servicio a partir de los
        //parámetros obtenidos de la petición
        xmlInput = this.buildTwoPhasesUSerSignRequest(requestParams);

        // Invocar el servicio web
        return invokeWebService(xmlInput);
    }
```

Firma de ficheros (VI)

- Código servidor (continuación):

```
    private String buildTwoPhasesUSerSignRequest(Map requestParams) {
        //Método que construye el parámetro de entrada requerido por el servicio web
        // FirmaUsuario2FasesF2 a partir de los parámetros de la petición

        xmlInput = “<?xml version=\"1.0\" encoding=\"UTF-8\"?> <mensajeEntrada “ +
            “xmlns=\”https://afirmaws/ws/firma\” ... > “ +
            “<peticion>FirmaUsuario2FasesF2</peticion><versionMsg>1.0</versionMsg>” +
            “<parametros><idAplicacion>” + requestParams.get(“idAplicacion”) +
            ”</idAplicacion>” + … ;

        return xmlInput;
    }

    private String procesarRespuesta(String twoPhasesUserSignResponse) {
          //Método que construye código html que muestra el resultado de la operación
          //a partir de la respuesta obtenida del núcleo de la plataforma
          …
          return htmlCode;
    }
}
```

Firma de página Web (I)

• Código cliente:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<HTML>
<HEAD>
   <META http-equiv="Content-Type" content="text/html; charset=UTF-8">
   <link rel="stylesheet" href="css/estilo2.css" type="text/css">
   <TITLE>Ejemplo Firma de ficheros en dos fases</TITLE>

   <script type="text/javascript" src="./common-js/instalador.js"></ script >
   <script  type="text/javascript" src="./common-js/appletHelper.js"></ script >
   <script type="text/javascript" src="./common-js/time.js"></ script >
   <script type="text/javascript" src="./common-js/utils.js"></ script >
   <script type="text/javascript" src="constantes.js"></ script >
   <script type="text/javascript" src="../common-js/firma.js"></script>
   <script type="text/javascript" src="../common-js/htmlEscape.js"></script>
   <script type="text/javascript" src="../common-js/utils.js"></script>
   <script type="text/javascript" src="../common-js/styles.js"></script>
   <script type="text/javascript" src="../common-js/firmaWeb.js"></script>
```

Firma de página Web (II)

- Código cliente (continuación):

```
<script type="text/javascript">
    function submitForm(element){
       if(clienteFirma){
          clienteFirma.setSignatureFormat('CMS');
          clienteFirma.setSignatureMode('IMPLICIT');
          var signElement = firmaWeb(element, document);
          if(!clienteFirma.isError()){
             document.getElementById('eSignature').value =

clienteFirma.getSignatureBase64Encoded();
             document.getElementById('signerCert').value =

clienteFirma.getSignCertificateBase64Encoded();
             document.formulario.submit();
           } else {
             alert('Se produjo un error en la firma...');
             document.formulario.reset();
           }
       } else {
          alert('Error cargando el cliente de firma.');
       }
    }
</script>
```

Firma de página Web (III)

• Código cliente (continuación):

```
<body onload="cargarAppletFirma();">
<form action="/firma/afirma5FirmaUsuarioDosFasesWeb" method="post"
                                        enctype="multipart/form-data" id="formulario" name="formulario">

   <input type="hidden" id="eSignature" name="eSignature" value="">
   <input type="hidden" id="signerCert" name="signerCert" value="">

   <div class="titulo">
      <div class="poneresquina">
         <div class="ponerimagen">Escribenos</div>
      </div>
   </div>

   <div class="contenido">
      Su dirección de correo electrónico: <br/>
      <input name="email" maxlength="90" size="50" value="" type="text"/><br/>
      Fichero adjunto 1:<br/>
      <input type="file"/><br/>
      Fichero adjunto 2:<br/>
      <input type="file"/><br/>
```

Firma de página Web (IV)

• Código cliente (continuación):

```
        Motivo de su consulta: <br/>
        <select name="motivo">
                <option value="0">-- Elegir opción --</option>
                <option value="1">Quiero patrocinar o poner publicidad en vuestro sitio web</option>
                <option value="2">Quiero colaborar con vosotros</option>
                <option value="3">He detectado un problema en el sitio web</option>
                <option value="10">Otros</option>
        </select><br/>
        Su mensaje: <br/>
        <textarea name="mensaje" id="mensaje" class="formens" cols="60" rows="5"></textarea>

        <input name="btnFirmar" id="botonFirmar" value="Firmar formulario" class="boton"
                type="button" onclick="submitForm(document.getElementById('formulario'))"/><br/>
    </div>
    <input name="op" value="add" type="hidden"/>
    <input name="idzona" value="" type="hidden"/>
</form>
<p>¡Gracias por colaborar!</p>
</body>
</html>
```

Ruegos y Preguntas