Indra

Liberation SISTRA: Signature module

Code: DSI-LIBSISTRA-FIRM

Govern
de les Illes Balears

Unión Europea

Fondo Europeo de
Desarrollo Regional

DOCUMENT CONTROL SHEET

DOCUMENT / FILE

Title: Liberation SISTRA: Signature module	File Name / s:**SISTRA-PLUGINFIRMA-002.doc**
Code: **DSI-LIBSISTRA-FIRM**	Software**: Word**
Date: September 2010	
Version**: 2**	

RECORD OF CHANGES

Version	Pages	Reason for change
1	9	Document Creation
2	9	Distinction between internal format when storing/retrieving signatures from DB and external format in webservices api

DISTRIBUTION OF DOCUMENT

Name	Staff

DOCUMENT CONTROL

PREPARATION	REVISED / APPROVED	ACCEPTED	ACCEPTED
Rafael Sanz	José Vicente Juan Pérez	**Jerome Navarrete**	**Bernat Alberti**

Complete with the name, signature and date

Customers only

INDEX

1. Goal

The purpose of this document is to define actions to perform in order to allow that SISTRA platform can use signature implementations from DGTIC and @firma.

2. Signature modules

2.1. DGTIC signature

Solution based on a signature java API hiding digital signature implementation details.

It is necessary to build an applet using aforementioned API to be able of performing signatures from Web clients taken in consideration that:

- installed certificates should be displayed through a pull-down
- a text box for entering pin should be shown
- button for signature procedure starting should exists

Signed document custody system from DGTIC should be used for digital signature stamp in DGTIC. All signed documents will be stored in this custody system.

2.2. @Firma

Provides two types of services according to the client:

- Web Client: applet based solution (installer applet + signer applet + javascript files). Applet wouldn't appear in web page. When signature button is pressed, it would appear a window for certificate selection and another for entering a pin afterwards.
- Server applications: next web services are offered for applications. Application should be registered and application id should pointed out in each call to the web service to be able of using these services:
    - digital signature (delegated signature: application certificate is registered in @firma server indicating to be signed with this certificate
    - signature verification
    - certified data query

The operation of adding a time stamp to a digital signature is performed from @FIRMA platform, that is from the server (never from a client). Signatures with time stamp can only be generated through FirmaServidor service. The format in which the signature is returned is called XAdES-T

3. Signing operation in SISTRA platform

3.1. Web client signature

Next processes are performed in web front-end:

- forms and annex signature
- process signature
- acknowledgment of receipt signature

3.2. Sign-in in server

Next processes should be performed in server part:

- validation of signatures generated in client
- proof of registry signature by telematic registry api
- signature storage in REDOSE
- file generation storing a digital signature to be downloaded by citizen

4. Proposed Solution

4.1. Signature in Web client

Two functioning modes (DGTIC or @firma) exist in web front-end. Page changes depending on each mode as operational and visual appearance differs in each case.

Items to be set depending on operating mode are:

- applets / js loading
- explanatory signature

This development will be done within tasks of platform release developed by DGTIC.

4.2. Signature in server

A signature generic interface is defined on the server side for server operations. It's necessary to define two implementations or plug-ins for each signature mode.

Plug-in for DGTIC signature will be implemented within tasks of platform release developed by DGTIC.

Signature plug-in with @Firma should be implemented by organization that will use it.

5. Signature plug-in interface

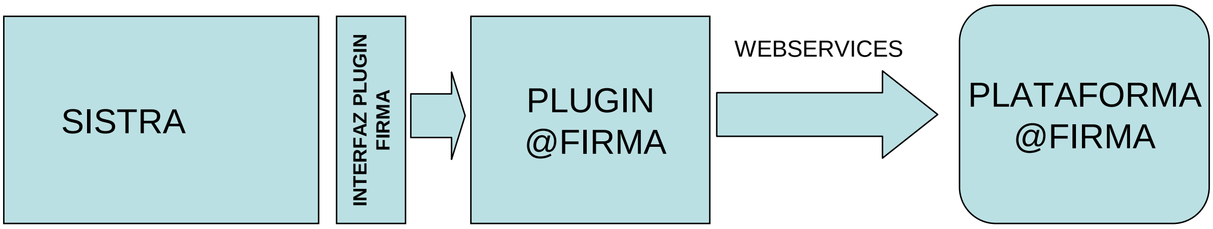
Signature plug-in interface for signature operations in server is next:

Method	Description	Parameters	Result
getProveedor	It gets signature provider	No	DGTIC or @FIRMA
Sign	It performs a digital signature	datos: data to be signed nombreCertificado: certificate to use parámetros: vendor specific parameters	FirmaIntf: digital signature
verificarFirma	It checks a signature	data: signed data FirmaIntf: digital signature	true / false
parseFirmaFromHtmlForm	It creates a signature from data string received from HTML form	signatureHtmlForm: signature string received from HTML form	FirmaIntf: digital signature
parseFirmaFromBytes	It creates a signature from an array of bytes retrieved from DB	firma: bytes with signature content formatoFirma: signature format	FirmaIntf: digital signature
parseFirmaToBytes	It serializes the signature in a byte array to store in DB	firma: signature data	Byte array with signature
parseFirmaFromWS	It creates a signature from a byte array coming from web services API	firma : bytes with signature content formatoFirma: signature format	FirmaIntf: digital signature
parseFirmaToWS	It serializes the signature in a byte array for use in web services APIs	firma: signature data	Byte array with the signature
parseFirmaToFile	It serializes the digital signature in a file downloadable by user	datosFirmados: signed data	FicheroFirma: file generated data

6. Signature plug-in implementation

6.1. Plug-in @firma

@firma plug-in must implement necessary calls to @firma platform webservices internally to support defined operations.



@firma implementation allows using different signature formats (CMS, XMLDsig, CADES, XADES, etc.). Therefore a signature will be stored together with its format in RDS to be able of retrieving it correctly later (parseFirmaFromBytes method).

We suggest to use a format generating a signature stamp (eg XAdES-T) for server signature (offered by the plugin).

CMS format will be used (default in @firma) on the client side as time stamp cannot be generated. Available formats on client are CMS (default) and CADES-BES.