Liberation SISTRA: Custody plug-in

Code: SISTRA-PLUGINCUSTODIA

DOCUMENT CONTROL SHEET

DOCUMENT / FILE

Title: Liberation SISTRA: plugin Custody	File Name / s:**SISTRA-PLUGINCUSTODIA-001.doc**
Code: **SISTRA-PLUGINCUSTODIA**	Software**: Word**
Date: 30/03/2010	
Version**1**	

RECORD OF CHANGES

Version	Pages	Reason for change
1	6	Document Creation
2	6	Added function to obtain query url of a document in custody

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

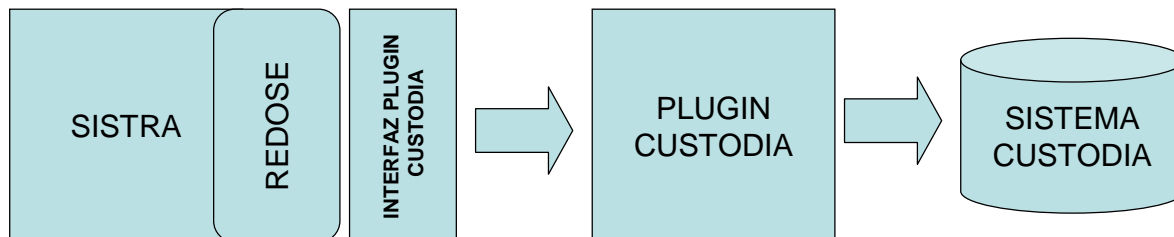# 1. Goal

The purpose of this document is to define actions to perform for allowing REDOSE integration with a signed document custody system..

2. Subclass

The scheme works as follows:

REDOSE module will synchronize through a custody plug-in to store signed documents in custody system (for models configured to be in custody). In this way, after inserting, modifying or deleting a signed document in REDOSE, this update will be propagated to the custody system.

3. Plug-in interface

Communication interface between SISTRA and custody plug-in is next:

Method	Description	Parameters	Result
custodiarDocumento	It inserts a document in custody system	DocumentoRDS: RDS document with document content and associated signatures	In custody document identifier
eliminarDocumento	It removes a document from custody system. If a document that does not exist is deleted, it will generate a warning (no exception).	codigoDocumentoCustodia: In custody document identifier	EstadoEnvio: shipment and a description status
obtenerUrlDocumento	It gets query url of document in custody.	codigoDocumentoCustodia: In custody document identifier	Query url (null if not allowed)

For more information consult javadoc.

REDOSE will call custodiarDocumento each time the document or its signatures are modified.

In case of non-transactional integrations, this function should return a different identifier each time. In this way, different custody versions will be stored in REDOSE and then remove process will delete old versions. It assures transactional logic because a roolback in REDOSE transaction would maintain equivalent document in custody.