eID Trust Service, Digital Signature Service e Identity Provider

Análisis de arquitectura y solución de firma digital del documento de identidad electrónico (eID) de Bélgica y definición de los cambios requeridos para adaptar la solución de eID Trust Service, eID Digital Signature Service y el eID Identity Provider al SNCD de Costa Rica

Rolosa HyJ S.A. - MICITT

11 de Agosto de 2014

Resumen

El presente documento presenta un análisis de la arquitectura actual de la solución de firma digital del eID Trust Service, eID Digital Signature Service y eID Identity Provider de Bélgica adaptado para el SNCD de Costa Rica.

Tabla de contenido

Introducción

El Trust Service provee dos servicios principales:

eID Trust Service Portal

Vía este portal web, los ciudadanos pueden revisar la validez de sus certificados

eID Web Service

Vía este web service SOAP, los Proveedores de Servicios pueden revisar la validez de rutas de certificados. Este web service está construido de acuerdo al estándar W3C XKMS2.

A parte de estos dos artefactos, el eID Trust Service también viene con un portal de Administración que permite a los administradores utilizar y configurar el eID Trust Service a través de una interfaz web.

El Digital Signature Service provee dos servicios principales:

eID Digital Signature Service Portal

Vía este portal web, los ciudadanos pueden firmar documentos y verificar firmas en documentos existentes

eID Digital Signature Service Web Service de validación de firmas

Vía este web service SOAP, los Proveedores de Servicios pueden verificar las firmas en digitales de documentos. Este web service está construido de acuerdo al estándar OASIS DSS.

El Identity Provider ofrece 3 protocolos para la comunicación de aplicaciones Web de los proveedores de Servicios:

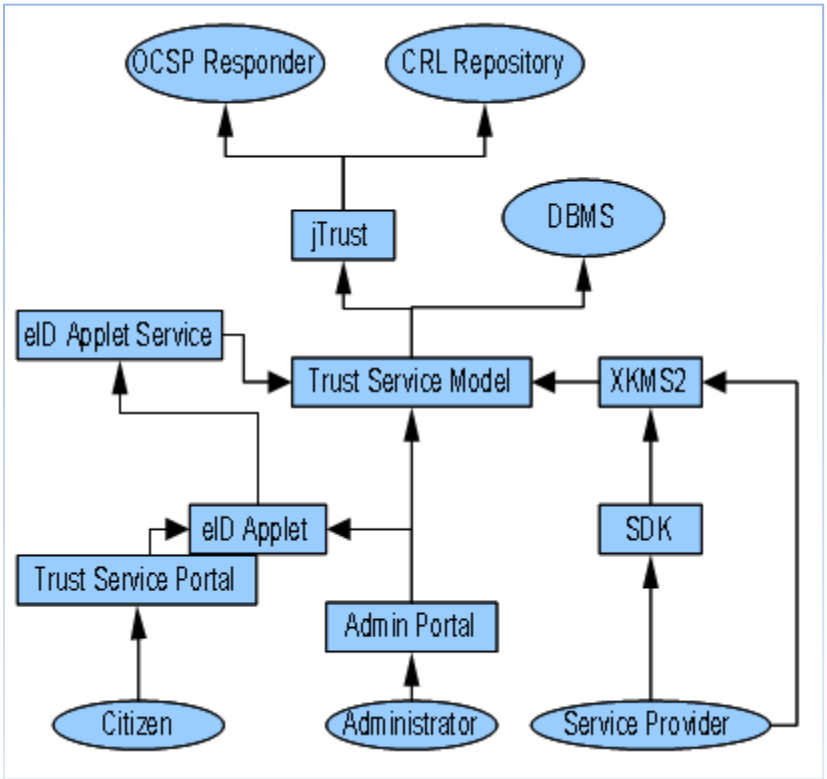
SAML v2.0

OpenID v2.0

WS-Federation v1.1

Arquitectura

La arquitectura de el eID Trust Service se muestra en la siguiente figura:



OCSP Responder y CRL Repository

El eID Trust Service utiliza servicios existentes de validación-PKI para la validación de rutas de certificados.

jTrust

Este componente de software maneja la validación-PKI, utilizando OCSP en línea y servicios de CLR. El diseño interno de este componente permite fácilmente agregar enlazadores de confianza externos proveyendo escalabilidad. El eID Trust Service realiza tal tarea por ejemplo agregando un enlazador de confianza personalizado manteniendo una cache CRL.

Solamente se necesita una configuración limitada puesto que la librería jTrust utiliza los URI's que se encuentran dentro de los Certificados para OCSP y validación CRL.

El código fuente de este componente puede ser encontrado en:

http://dcfd-mw-applet.googlecode.com

Para obtener una copia del código se debe utilizar un cliente SVN y realizar la operación de Checkout:

svn checkout http://dcfd-mw-applet.googlecode.com/svn/trunk/

Trust Service Model

El eID Trust Service Model es un módulo de software escrito en tecnología EJB3. Este módulo administra dominios de confianza, configuración del servicio, cache de revocación de certificados y más. El módulo jTrust utiliza este cache de revocación de certificados durante la validación de la ruta de certificados.

XKMS v2.0

La interfaz primaria de el eID Trust Service para Proveedores de Servicios es un web service el cual está basado en el estándar W3C XKMS v . El protocolo de enlace utilizado es HTTP SOAP. Dependiendo de la configuración de políticas de seguridad de el eID Trust Service (configurable por un administrador) una aplicación puede utilizar este web service sin necesidad de autenticación de servicio, con autenticación TLS unilateral o autenticación a nivel de mensajes utilizando respuestas XKMS2 firmadas.

SDK

El SDK del eID Trust Service permite a los Proveedores de Servicio la fácil integración de el eID Trust Service en aplicaciones (web). El SDK consiste de:

- Referencia de la implementación de un cliente XKMS2 basado en Java

- Una implementación .NET (C#) 2.0, disponible para las plataformas Microsoft y Mono.

eID Applet y eID Applet Service

Utilizado estos componentes genéricos (Java-Applet), los desarrolladores pueden integrar fácilmente la funcionalidad eID en aplicaciones web. El eID Applet provee un completo soporte para las funcionalidades de la tarjeta inteligente. Esto comprende identificación, autenticación y firma digital. El eID Trust Service utiliza el eID Applet dentro del eID Trust Service Portal y del eID Trust Service Admin Portal. El código fuente para el eID Applet puede ser encontrado en:

Trust Service Portal

Utilizando el eID Trust Service Portal, los ciudadanos pueden verificar la funcionalidad de sus tarjetas inteligentes y validar sus certificados digitales fácilmente. El portal está traducido a muchos lenguajes. El usuario tiene la posibilidad de seleccionar el lenguaje manualmente. Si no se elige manualmente, la configuración del explorador web será usada. El portal además incluye un panel de información, explicando los certificados y la tarjeta en general. Esta información puede ser modificada por un administrador en el eID Trust Service Admin Portal.

Admin Portal

Los productos eID Trust Service, eID Digital Signature Service y eID Identity Provider proveen de un Admin Portal independiente que permite a los administradores configurar cada uno de dichos productos.

Un Administrador se autentica a si mismo dentro de cada Admin Portal utilizando su tarjeta inteligente. Si aun no existe ningún administrador registrado, el primer usuario que se autentique exitosamente, será registrado como un administrador. Un administrador tiene la posibilidad de aprobar registros de administradores adicionales. Un usuario puede solicitar ser un administrador a través de una autenticación exitosa en el Admin Portal. Después de realizarlo, un administrador existente tendrá la posibilidad de aprobar esta solicitud. El administrador solo será autenticado utilizando la verificación de clave publica durante el inicio de sesión en el Admin Portal, una validación-PKI para un administrador es innecesaria.

Para el eID Trust Service la siguiente configuración es posible:

- Agregar, aprobar y remover administradores.
- Configuración de la Política de seguridad del Web Service para ser usada si los mensajes XKMS2 salientes deberían ser firmados o no y configuración del keystore necesario para este fin.
- Configuración de un proxy de red HTTP utilizado por el eID Trust Service para acceder al CRL Repository y al OCSP responder.
- Configuración de tareas de detección de desfase de tiempo, NTP y TSP están soportados.
- Configuración del mensaje de información a ser mostrado en el eID Trust Service Portal, esto para todos los lenguajes que el eID Trust Service Portal soporta.
- Administración de dominios de confianza diferentes, Esto consiste en:
    - Agregar y remover dominios de confianza
    - Agregar y remover dominios virtuales de confianza
    - Establecer el dominio de confianza por defecto. Este es el dominio de confianza que será utilizado en el eID Trust Web Service si no se especifica ningún otro.
    - Seleccionar puntos de confianza dentro de un dominio de confianza
    - Seleccionar puntos de confianza dentro de un dominio virtual de confianza
    - Configurar el intervalo de actualización de CRL por punto de confianza
    - Configurar restricciones de certificados por dominio de confianza
    - Configurar si el cache de CRL puede ser utilizado por dominio de confianza
    - Realizar una actualización manual del cache CRL por punto de confianza. Esto disparara una actualización en todos los CAs bajo ese punto de confianza.
    - -Realizar una actualización manual del cache CRL por cada CA.

Para el eID Digital Signature Service la siguiente configuración es posible:

- Agregar, aprobar y remover administradores.
- Configuración de validación PKI, URL del Trust Service Web Service
- Configuración de un proxy de red HTTP
- Configuración de parámetros del TSP

Descripción General

En esta sección proveeremos de una descripción general del diseño del eID Trust Service. El diseño del eID Trust Service está inspirado en el ECPV [HM2003].

Entidades de Dominio

Aquí se describen las entidades de dominio utilizadas en el eID Trust Service. Se utiliza JPA v1.0 de la arquitectura JAVA EE.

Dominio de Confianza

Un Dominio de Confianza cubre cierta topología PKI. Por ejemplo la estructura PKI de SNCD de Costa Rica, la cual consiste en puntos de confianza Raíz Nacional, Política Persona Física y SINPE. Diferentes dominios de confianza pueden ser definidos y configurados por un administrador. Para cada dominio de confianza se puede configurar si cache de CRL es permitido o no. Un dominio de confianza por defecto puede ser asignado, el cual será utilizado en las solicitudes al web Service XKMS2 del eID Trust Service si no se especifica un dominio de confianza explícitamente.

Dominio Virtual de Confianza

Un Dominio Virtual de Confianza consiste en un conjunto de Dominios de Confianza ya definidos dentro del eID Trust Service. Una validación positiva dentro un dominio virtual de confianza especificado se traduce a sí misma en una validación positiva dentro de al menos uno de los dominios de confianza de el dominio virtual de confianza. De esta manera por ejemplo, se puede definir un dominio virtual de confianza para Europa, conteniendo todos los dominios de confianza de sus miembros.

Punto de Confianza

Un Punto de Confianza corresponde con una Autoridad de Certificación de Raíz (CA). Un Punto de Confianza esta enlazado siempre a uno o muchos dominios de confianza. Un punto de confianza es utilizado por el validador PKI como punto inicial de la validación de la ruta de certificado. El intervalo de

actualización de la cache CRL puede ser configurado por punto de confianza. Cuando el intervalo ha pasado, una actualización de la cache CRL de la CA correspondiente al punto de confianza será realizada, al igual que todas las CAs dependientes.

CA - Autoridad de Certificación

Una cache de revocación de certificados es mantenida por cada CA. Un CA no está necesariamente enlazada a un punto de confianza. Para cada CA, un registro es mantenido, de la validez de sus registros de revocación de certificados. Este intervalo de validez corresponde a los campos **thisUpdate** y al **nextUpdate** de el CRL especifico que alimenta esa cache. La validez del intervalo es crucial para prevenir que la validación PKI tome decisiones basada en registros de certificados de revocación expirados.

Registro de Cache de Revocación de Certificados

Cuando un certificado es revocado, esto será manifestado en la cache de revocación de certificados correspondiente a la CA. Este registro contiene el emisor, numero de serie y fecha de revocación del certificado.

Administrador

El eID Trust Service Admin Portal, permite habilitar muchos administradores. La Autenticación es basada en la tarjeta inteligente utilizando la clave publica de el certificado de autenticación. Un Administrador puede tener un estado Pendiente si aun no ha sido aprobado por algún otro administrador.

Configuración de desfase de Reloj

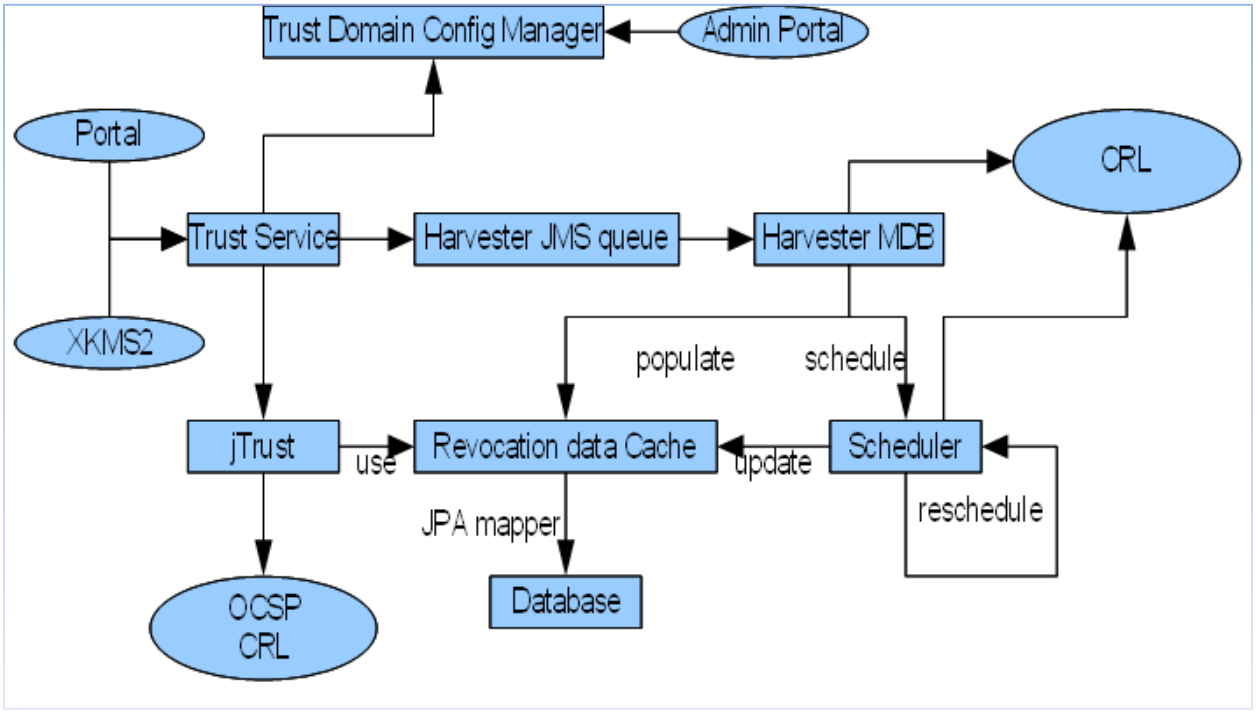
El eID Trust Service cuenta con una tarea de detección de desfase de reloj. El intervalo de la tarea puede ser configurado utilizando una expresión cron. NTP y TSP son soportados.

Configuración de WS Security

Se puede configurar si las respuestas XKMS2 del eID Trust Service deberían ser firmadas, así mismo el keystore que será utilizado para este propósito.

eID Trust Service Model

El diseño del eID Trust Service Model se muestra a continuación:



La implementación utiliza el framework EJB3.0.

Punto de Entrada

El punto de entrada viene con un componente facade de TrustService. Este componente provee funcionalidad para validad una ruta de certificado dada. La lógica de validación provee una estrategia de respaldo hacia la fuente de datos de revocación de certificados utilizada. Primero una verificación es realizada si para un CA especifico, datos locales de revocación de certificados están disponibles en la cache de revocación de certificados. Si no, una validación-PKI clásica será realizada utilizando el siguiente proceso: El OCSP responder es consultado, si no existe para este CA o el request ha fallado, el último recurso es utilizar el CRL de la CA.

Al mismo tiempo el harvester es informado de la falta de datos de revocación de certificado para la CA. La comunicación entre el componente del front-end y el harvester es realizada utilizando una cola JMS.

Si están disponibles registros de revocación de certificados (y son utilizables, de acuerdo a la verificación de actualización) para la CA, entonces esos serán utilizados. En este caso, no es necesario para el eID Trust Service consultar servicios PKI en línea de OCSP y/o CRL.

Es importante mencionar que para cada interacción con el Trust Service, se requiere de la Cadena de Certificados de Confianza completa.

El eID Applet soporta actualmente la preparación de solicitudes al TrustService considerando los certificados necesarios para construir una Cadena de Certificados de Confianza apropiada. Cualquier otro procedimiento que interactúe con el TrustService por medio del WebService debe garantizar la presencia de una Cadena de Certificados de Confianza apropiada para que el TrustService realice la validación correspondiente.

Harvester

El harvester es implementado como un EJB3 Message Driven Bean (MDB). Este recibe mensajes desde los componentes del front-end a través de la cola JMS de harvester, iniciando el procesamiento del mismo.

El harvester descarga el CRL de la CA especificada, verifica su validez y procesa la cache de datos de revocación de certificados. Después de esto, el harvester activa la cache de revocación de certificados para la CA especificada y v validaciones de ruta de certificado futuras pueden utilizar estos datos.

La ventaja de un harvester basado en MDB es que la arquitectura JMS provee reintentos automáticos en caso de falla en la descarga o procesamiento del CRL. Así de este modo, no es necesaria ninguna lógica extra en el harvester para soportar esta funcionalidad.

Además de recolectar CRLs para datos de revocación, el harvester-MDB también maneja la tarea de detección de desfase del reloj.

Scheduler

El scheduler es responsable de la frecuencia de actualización de la cache de revocación de certificados y la detección de desfase de reloj. Esta manejado por el Java EE Timer Service. El beneficio de este es el aspecto transaccional. EL scheduler reprogramara también le Java EE Timer al finalizar sus tareas.

El intervalo de actualización de la cache de revocación de certificados es establecida por cada punto de confianza. El scheduler puede ser activado manualmente por un administrador a través de una reprogramación del Java EE Timer.

El intervalo de validez, mantenido por cada CA, es crucial para prevenir validaciones PKI tomando decisiones basadas en datos de revocación desactualizados (en el caso de que algo haya salido mal con el scheduler). Como tanto el harvester y el scheduler son manejados por un administrador de transacciones, se garantiza plenamente la consistencia entre los registros de la cache y la validación del intervalo.

A parte de programar actualizaciones de cache de revocación de datos, el scheduler también maneja la tarea de detección de desfase de reloj. Cuando el correspondiente Java EE Timer es disparado, la configuración de detección de desfase es obtenida de la base de datos y una solicitud NTP o TSP es realizada.

Cache de Revocación de Certificados

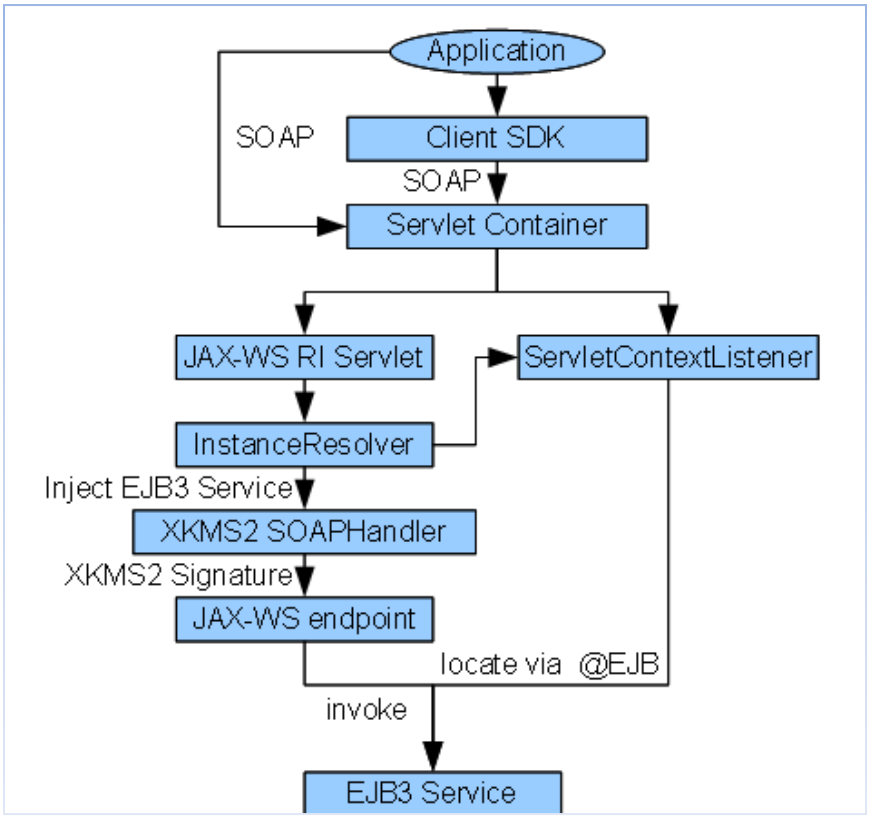
Debido a las actividades coordinadas del harvester y del scheduler, la cache de revocación de certificados solamente contiene datos para las CAs que son usadas activamente por las aplicaciones del Proveedor de Servicios. El sistema agrega automáticamente nuevos CAs a la cache de datos de revocación, durante la ejecución. Esto tiene un impacto positivo en el mantenimiento del eID Trust Service. Un mantenimiento automático de la cache ocurre transparentemente para las aplicaciones de los Proveedores de Servicio.

Tiempo

La correctitud del tiempo es crítica en una validación-PKI (revisiones actuales de OCSP y CRL). El eID Trust Service Model contiene una tarea de detección de desfase de reloj (administrada por un Java EE Timer) que revisa periódicamente la correctitud de el reloj local de la maquina. Esto puede realizarse utilizando un servicio NTP o TSP. Ubicación, tiempo de espera y desfase máximo pueden ser configurados por un administrador a través del eID Trust Service Portal.

JAX-WS Web Service Runtime

El eID Trust Service usa JAX-WS 2.1 RI en tiempo de ejecución para el web service XKMS v2. La configuración de JAX-WS en tiempo de ejecución se muestra a continuación:



La implementación usa el framework EJB 3.0.

El cliente SDK provee integración de el eID Trust Service y aplicaciones SOA. Un cliente Java – SDK y un cliente .net v2.0(C#)-SDK son provistos.

El entorno JAX-WS se ejecuta dentro del contenedor del servlet. Se utiliza un Java EE ServletContext Listener, en conjunto con inyección EJB3 estándar , subsecuentemente, un JAX-WX RI InstanceResolver se utiliza para inyectar estas referencias al servicio dentro de los JAX-WS endpoints. Esta configuración permite la independencia del servidor de aplicaciones utilizado.

La firma WS-Security opcional a la respuesta XKMS2 es agregada utilizando un JAX-WS SOAPHandler que es configurado dentro de la cadena de endpoints del JAX-WS handler.

A pesar que el JAX-WS tiene una API de integración para contenedores Java EE, se eligió no utilizar esta API. También se decidió implementar el web service XKMS2 como POJO JAX-WS endpoint en lugar de un

EJB3 JA-WS endpoint. Esto para prevenir una dependencia mayor con el servidor de aplicaciones Java EE (por ejemplo al utilizar JBossWS)

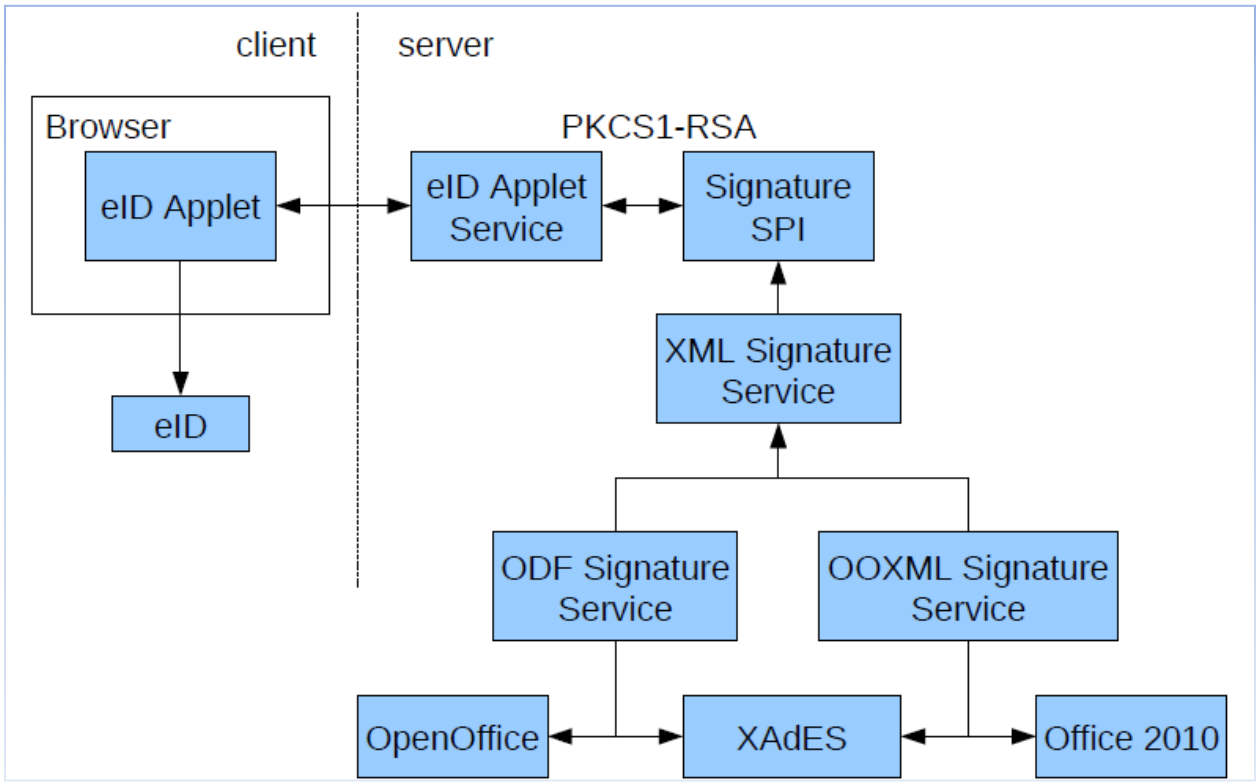
Portales Web

Los portales web utilizan JSF como tecnología para la vista (como en un modelo MVC). Se utiliza JSF 1.2 RI. Las plantillas son provistas por Facelets. Richfaces son utilizados por sus características AJAX para componentes JSF.

El enlace con los componentes del back-end se logra utilizando tecnología JBoss-Seam.

Arquitectura de Firmado de Documentos Digitales

El Applet Service pone a disponibilidad la Interfaz correspondiente para la implementación de firma digital. En la siguiente figura se puede apreciar la implementación necesaria para extender el servicio de firmado para formatos XML tanto para ODF (Open Office) como OOXML (Microsoft Office).



Aspectos de Aplicación Java EE

Seguridad

Como modelo de seguridad en el eID Trust Service Admin Portal, se utiliza un modelo de dos-trampas. A parte de que el contenedor del servlet conoce sobre el login/logout, los componentes del modelo utilizan el framework de seguridad EJB3. Un módulo de login JAAS asigna los roles necesarios a un Director, así de esta manera, el interceptor de autorización EJB3 puede realizar su trabajo.

Este modelo de seguridad tiene el beneficio de que la penetración en el front-end no implica la penetración de el componente de servicio puesto que está separado debido a la asignación independiente de roles que el módulo de login JAAS realiza.

Persistencia

El mapeo necesario para lograr persistencia esta realizado utilizando Java EE JPA 1.0 API. El beneficio de esto es la mantenibilidad del esquema de base de datos. Todos los metadatos (ej. índices) son expresados usando anotaciones de Java 5.

El motor de JPA (Hibérnate) tiene que ser capaz de interpretar estos metadatos correctamente para la generación del esquema de base de datos. Esto es importante para mantener las entidades JPA como una fuente autentica para el esquema de base de datos.

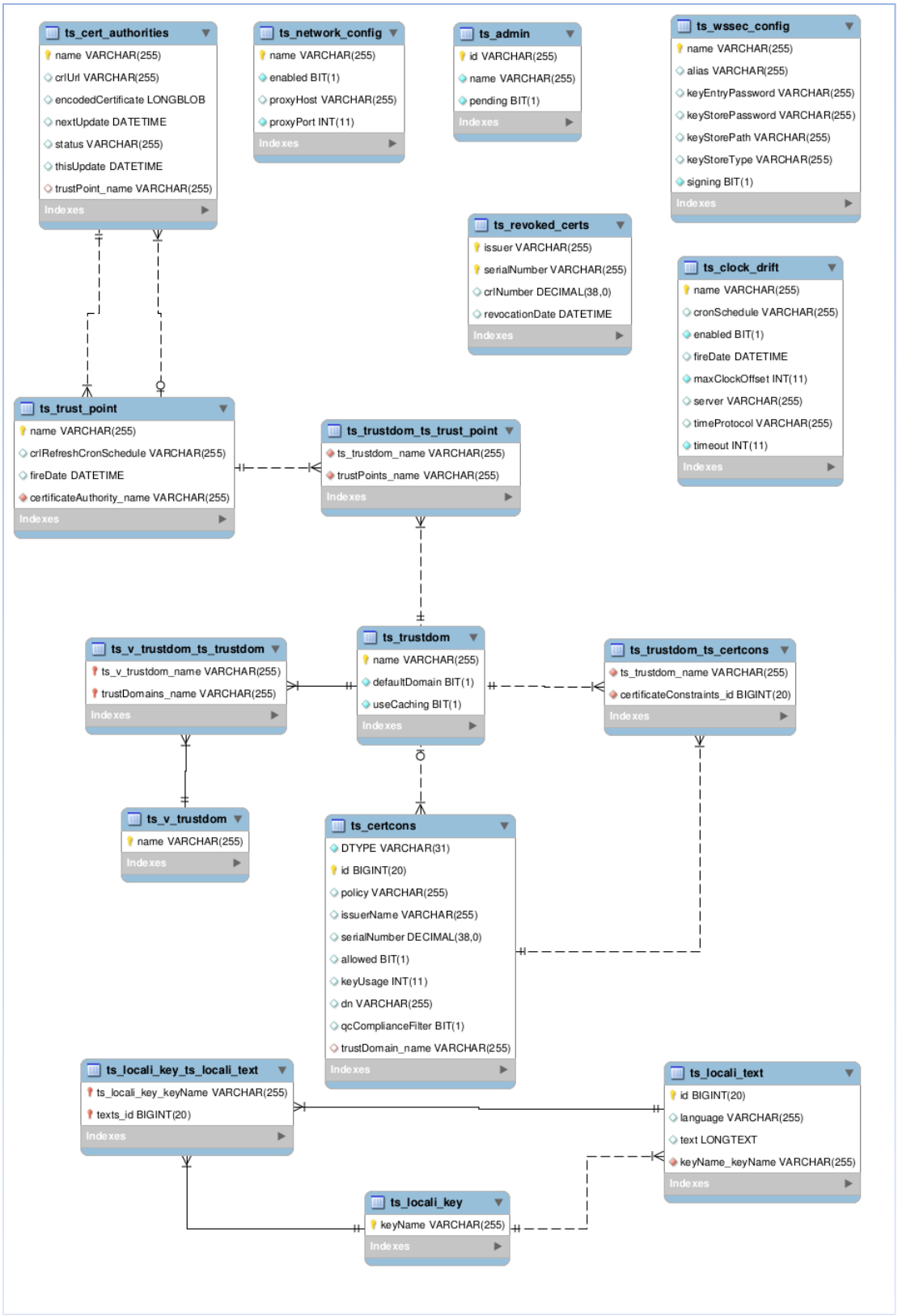
Se provee de distribuciones personalizadas para MySQL y Oracle, además de la habilidad de ejecutarse encima una distribución HSQL-Jboss estándar.

La base de datos del Trust Service almacena información sobre:

- Cache de Revocación de Certificados,
- Puntos de Confianza,
- Dominios de Confianza,
- Dominios virtuales de Confianza y
- Configuración en general.

La Implementación esta delegada al Framework de Hibernate para ORM por medio de Javax.Persistence

El Diagrama de Entidades se muestra a continuación:

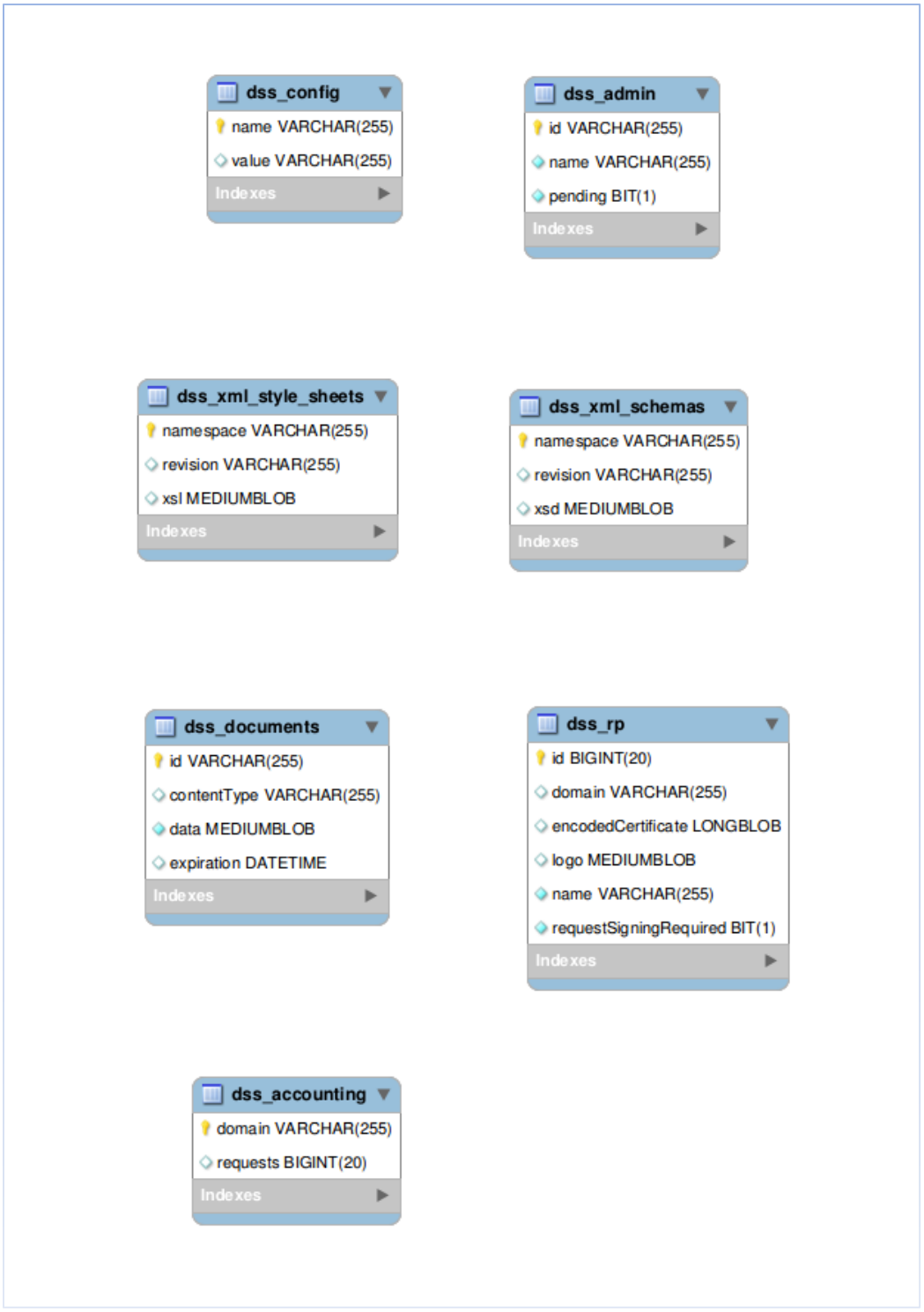

ts_cert_authorities
name VARCHAR(255)
crlUrl VARCHAR(255)
encodedCertificate LONGBLOB
nextUpdate DATETIME
status VARCHAR(255)
thisUpdate DATETIME
trustPoint_name VARCHAR(255)
Indexes
ts_network_config
name VARCHAR(255)
enabled BIT(1)
proxyHost VARCHAR(255)
proxyPort INT(11)
Indexes
ts_admin
id VARCHAR(255)
name VARCHAR(255)
pending BIT(1)
Indexes
ts_wssec_config
name VARCHAR(255)
alias VARCHAR(255)
keyEntryPassword VARCHAR(255)
keyStorePassword VARCHAR(255)
keyStorePath VARCHAR(255)
keyStoreType VARCHAR(255)
signing BIT(1)
Indexes
ts_revoked_certs
issuer VARCHAR(255)
serialNumber VARCHAR(255)
crlNumber DECIMAL(38,0)
revocationDate DATETIME
Indexes
ts_clock_drift
name VARCHAR(255)
cronSchedule VARCHAR(255)
enabled BIT(1)
fireDate DATETIME
maxClockOffset INT(11)
server VARCHAR(255)
timeProtocol VARCHAR(255)
timeout INT(11)
Indexes
ts_trust_point
name VARCHAR(255)
crlRefreshCronSchedule VARCHAR(255)
fireDate DATETIME
certificateAuthority_name VARCHAR(255)
Indexes
ts_trustdom_ts_trust_point
ts_trustdom_name VARCHAR(255)
trustPoints_name VARCHAR(255)
Indexes
ts_v_trustdom_ts_trustdom
ts_v_trustdom_name VARCHAR(255)
trustDomains_name VARCHAR(255)
Indexes
ts_trustdom
name VARCHAR(255)
defaultDomain BIT(1)
useCaching BIT(1)
Indexes
ts_trustdom_ts_certcons
ts_trustdom_name VARCHAR(255)
certificateConstraints_id BIGINT(20)
Indexes
ts_v_trustdom
name VARCHAR(255)
Indexes
ts_certcons
DTYPE VARCHAR(31)
id BIGINT(20)
policy VARCHAR(255)
issuerName VARCHAR(255)
serialNumber DECIMAL(38,0)
allowed BIT(1)
keyUsage INT(11)
dn VARCHAR(255)
qcComplianceFilter BIT(1)
trustDomain_name VARCHAR(255)
Indexes
ts_locali_key_ts_locali_text
ts_locali_key_keyName VARCHAR(255)
texts_id BIGINT(20)
Indexes
ts_locali_text
id BIGINT(20)
language VARCHAR(255)
text LONGTEXT
keyName_keyName VARCHAR(255)
Indexes
ts_locali_key
keyName VARCHAR(255)
Indexes

La unicidad de los CAs que se almacenan en la tabla ts_cert_authorities es dada utilizando el "Subject" y el "Serial Number" definidos para el punto de confianza que define la CA como un nombre compuesto. Esto permite mantener varios CAs que comparten el mismo "Subject" con distinto "Serial Number" y diferente URL para los CRLs asociados.

La base de datos del Digital Signature Service almacena información sobre:

- Esquemas XML,
- Hojas de Estilo XML para el visualizador de archivos XML,
- Referencia temporal de Documentos a ser eliminados periódicamente,
- Relying Parties (Proveedores de Servicios) y
- Configuración en general.

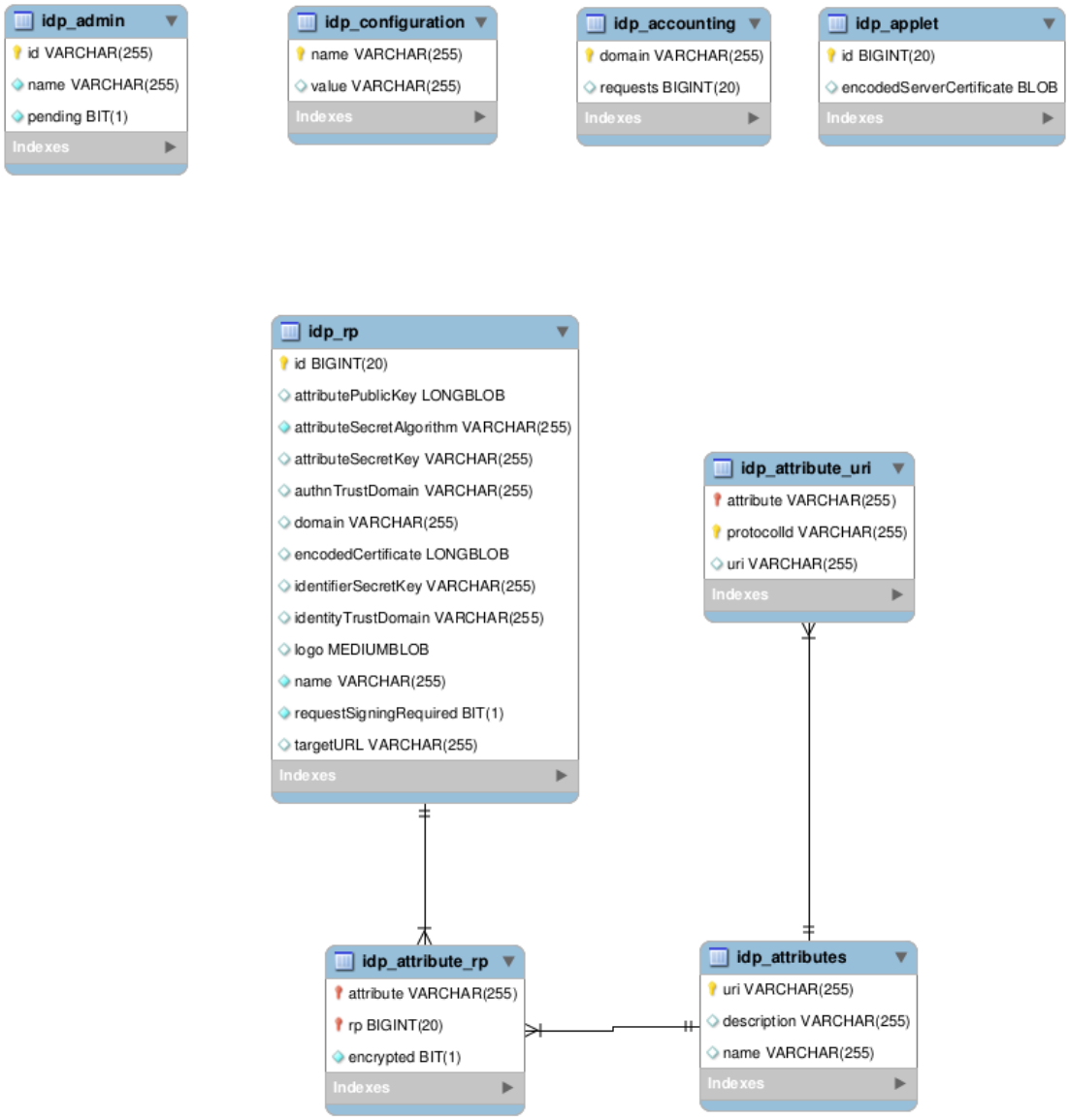
El Diagrama de Entidades se muestra a continuación:


dss_config
name VARCHAR(255)
value VARCHAR(255)
Indexes
dss_admin
id VARCHAR(255)
name VARCHAR(255)
pending BIT(1)
Indexes
dss_xml_style_sheets
namespace VARCHAR(255)
revision VARCHAR(255)
xsl MEDIUMBLOB
Indexes
dss_xml_schemas
namespace VARCHAR(255)
revision VARCHAR(255)
xsd MEDIUMBLOB
Indexes
dss_documents
id VARCHAR(255)
contentType VARCHAR(255)
data MEDIUMBLOB
expiration DATETIME
Indexes
dss_rp
id BIGINT(20)
domain VARCHAR(255)
encodedCertificate LONGBLOB
logo MEDIUMBLOB
name VARCHAR(255)
requestSigningRequired BIT(1)
Indexes
dss_accounting
domain VARCHAR(255)
requests BIGINT(20)
Indexes

La base de datos del Identity Provider almacena información sobre:

- Relying Parties (Proveedores de Servicios)
- Atributos de los Proveedores de Servicios
- Configuración en general.

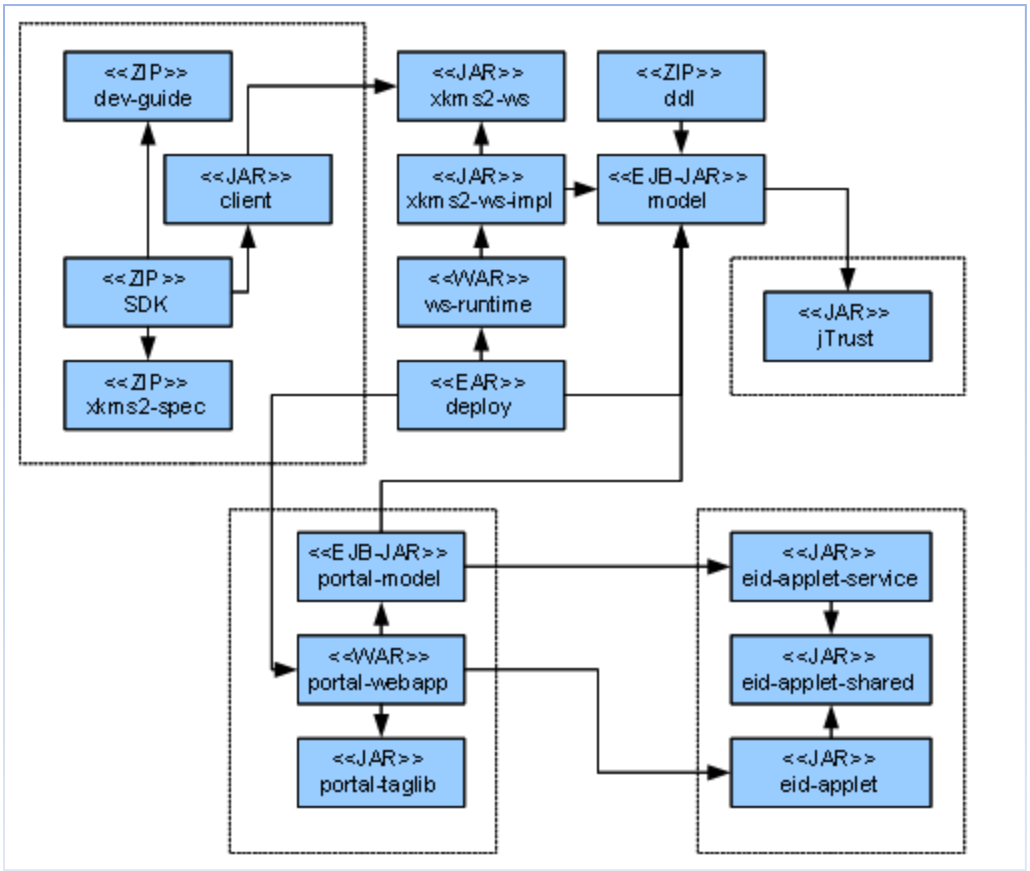
El Diagrama de Entidades se muestra a continuación:



Sistema de Compilación

Apache Maven v2 es utilizado como sistema de compilación. Para el despliegue de la aplicación Java EE en un servidor de aplicaciones local, simplemente de debe ejecutar **mvn boss:deploy.**

El sistema de compilación está dividido de manera que módulos Java EE diferentes son producidos como artefactos individuales, como se muestra a continuación:



A través del archivo pom.xml que se encuentra en la raíz, se agrupan diferentes artefactos en perfiles de compilación, se administran las dependencias de software y las versiones del proyecto.

Configuraciones de Compilación

Diferentes perfiles de Maven activan diferentes configuraciones. Por defecto se activa la configuración de desarrollo. Otros perfiles son por ejemplo **prod-mysql** y **prod-oracle**, los cuales permiten crear un artefacto que contiene una distribución JBoss MySQL u Oracle personalizada. Otro perfil llamado SDK disparara la generación de un archivo ZIP comprimido que contiene el SDK, los javadocs, un directorio con todas las dependencias necesarias y una guía de desarrollo. Este perfil puede ser disparado vía: **mvn -Denv=sdk clean install**

Especificación del Web Service

eID Trust Service XKMS2

Esta sección presenta un vistazo de el eID Trust Service-XKMS2 web service. Para una especificación detallada de XKMS v2.0 dirigirse a http://www.w3.org/TR/xkms2/

El eID Trust Service solamente cubre una porción de la especificación XKMS v2.0 ya que solamente soporta peticiones de validación.

Es su configuración más reducida, un cliente puede especificar una ruta de certificados para validación. El eID Trust Service realizara una validación-PKI en contra del dominio de confianza que se encuentra configurado por defecto. La petición XKMS2 será:

```
soap:Envelope/soap:Body/xkms2:ValidateRequest/xkms2:QueryKeyBinding/ds:KeyInfo/ds:X509
Data/ds:X509Certificate
```

donde el cliente agrega los elementos X509-Certificate en la ruta de certificados a ser validada. Después de la validación, el eID Trust Service XKMS web Service responde con la siguiente respuesta XKMS2:

```
soap:Envelope/soap:Body/xkms2:ValidateResult/xkms2:KeyBinding/xkms2:Status/@StatusValu
e=http://www.w3.org/2002/03/xkms#Valid
```

En caso de que la validación fuera invalida, una respuesta similar conteniendo el elemento StatusValue:

será retornada. Adicionalmente el elemento Status contendrá 1 o más razones URIs XKMS2 indicando el porqué de que la validación a hallado. En el siguiente enlace se pueden encontrar más detalles del elemento Status http://www.w3.org/TR/xkms2/#XKMS_2_0_Section_5_1

Es importante resaltar que se asume que el cliente siempre provee de la ruta de certificados completa y que el eID Trust Service no manea la construcción de rutas de certificados.

Si el cliente desea validad sus certificados en contra de un Dominio de Confianza distinto al dominio de confianza configurado por defecto, se tiene la opción de especificar el nombre de ese dominio de confianza. El cliente, en ese caso, deberá agregar a la petición, el siguiente elemento (como indica la especificación XKMS2):

```
xkms2:ValidateRequest/xkms2:QueryKeyBinding/xkms2:UseKeyWith@Application=urn:be:fedict
:trust:trust-domain@Identifier=<trust-domainname>
```

Autenticación de Servicio

La autenticación de Servicio para el cliente del web service tiene 3 opciones diferentes:

- El software cliente no requiere ningún servicio de autenticación. Este es el caso en el que el software cliente tiene una conexión dedicada con el eID Trust Service.
- El software cliente utiliza autenticación TLS unilateral. El cliente tiene esta opción al especificar el certificado del servidor eID Trust Service para validación.
- Autenticación de servicio utilizando respuestas XKMS2. Esta opción tiene que ser configurada explícitamente dentro del eID Trust Service Admin portal, puesto que al agregar la firma WS-Security tiene un impacto significativo en el performance del servicio.

Revocación de Datos

Es posible instruir al eID Trust Service XKMS2 web Service que devuelva los datos de revocación (repuesta OCSP y/o CRLs) durante la validación PKI. Esto puede ser útil para rutas de certificados de no-repudio donde estos datos serán utilizados para finalizar una firma XAdES. Un elemento **xkms2:MessageExtension** es introducido conteniendo el elemento ETSI XAdES v1.3.2 que contendrá estos datos en **RevocationValues** si así es requerido.

Si los datos de revocación son requeridos, el eID Trust Service no utilizara datos de revocación almacenados en la cache local, pero realizara una validación -PKI en línea (OCSP/CRL) puesto que se requiere datos de revocación recientes para la respuesta. Esto es importante para firmas XAdES.

Para especificar que una petición retorne los datos de revocación utilizados, el siguiente elemento tiene que ser añadido a la petición:

```
xkms2:ValidateRequest/xkms2:RespondWith/urn:be:fedict:trust:revocation-data
```

Validación Histórica

Es posible realizar una validación histórica en rutas de certificados. Para esto se debe agregar el elemento:

```
xkms2:ValidateRequest/xkms2:QueryKeyBinding/xkms2:TimeInstant
```

que contiene el tiempo en el cual se desea validar una ruta de certificados dada. Los datos de revocación (respuestas OCSP y/o CRLs) deberán ser añadidos por el mismo cliente en la petición XKMS2. Para esto, la misma extensión utilizada para devolver datos de revocación será usada, siendo este un elemento ETSI XAdES v1.3.2.

Certificados TSA

Es posible validar certificados TSA utilizando el eID Trust Service XKMS2 web service. Esto es importante para firmas XAdES-T.

Para esto, otra extensión **xkms2:MessageExtension** ha sido introducida, nombrada **TSAMessageExtension** . Esta extensión contiene un elemento de tipo **xades:EncapsulatedPKIDataType**

que contiene el toquen de sellado de tiempo codificado retirado por una petición de sellado de tiempo. Este toquen tendrá que contener la ruta de certificado del certificado TSA utilizado.

El eID Trust Service ejecutará una validación-PKI de la ruta de certificado contenida, contra el dominio de confianza de sellado de tiempo que está configurado en el eID Trust Service y especificado con:

```
xkms2:ValidateRequest/xkms2:QueryKeyBinding/

xkms2:UseKeyWith@Application=urn:be:fedict:trust:tsa@Identifier=<tsa-trust-domain-name>
```

Adicionalmente se verificará que el certificado TSA contiene la extensión **ExtendedKeyUsage** con identificador de propósito de la clave **id-kp-timeStamping**.

eID Digital Signature Service POST

Esta sección presenta un vistazo de el eID Digital Signature Service POST web service. Este protocolo puede ser utilizado por Proveedores de Servicios para integrar la firma de documentos como parte de su flujo de trabajo.

Solicitud de Firma

Puesto que el Proveedor de Servicio dispone de un documento a ser firmado, la aplicación web puede enviar una solicitud de firma utilizando un HTTP POST a través del navegador web del usuario final. Los siguientes parámetros POST están disponibles:

- El parámetro **SignatureRequest** (requerido) deberá contener el documento que necesita ser firmado codificado en base64.
- El parámetro **target** (requerido) deberá contener el URL del componente de la aplicación web al que se enviara de retorno el documento firmado.
- El parámetro **language** (opcional) deberá contener el código del lenguaje que el eID DSS utilizara durante la creación de la firma.
- El parámetro **ContentType** (opcional) deberá contener el tipo de contenido del documento que se envía. Por defecto es text/xml.
- El parámetro **RelayState** (opcional) es una referencia para mantener información por el proveedor de servicio. El mismo valor es retornado por el eID DSS dentro del mensaje de respuesta.

Firma de Servicio

Opcionalmente la aplicación web puede agregar una firma de servicio a la solicitud de firma. Los siguientes parámetros deberán ser agregados:

ServiceSigned: contiene el URL codificado y la lista separada por comas de parámetros POST firmados por la aplicación web. Esta lista deberá contener al menos los siguientes elementos: **SignatureRequest**, **target**, y si es posible **language, ContentType** y **RelayState**.

ServiceSignature: contiene la firma creada por el Relying Party, codificada en base64. Esta firma puede ser utilizada por el eID DSS para verificar la integridad y autenticidad del mensaje de solicitud. El algoritmo a usarse para la firma tiene que ser SHA1-RSA.

eID IDP SAML v2.0

La aplicación web del Proveedor de Servicio puede enviar una solicitud de autenticación SAML v2.0 codificada en Base64 utilizando HTTP Post a través del navegador del cliente según la especificación OASIS URN SAML v2.0. La respuesta de autenticación que el Identity Provider retorna esta codificada en Base64. También es posible enviar solicitudes de autenticación SAML v2.0 de manera similar al protocolo HTTP-POST o HTTP-Redirect pero al punto de entrada del protocolo de enlace de artefactos.

eID OpenID v2.0

Se ha agregado soporte para OpenID para integración transparente en plataformas como Drupal. Un documento dinámico de identidad YADIS, se utiliza para iniciar una autenticación OpenID.

eID WS-Federation v1.1

Se dispone de soporte para el modelo de solicitantes pasivos Web de WS-Federation. Este soporte provee integración transparente con aplicaciones ASP.net utilizando Windows Identity Foundation.

Cambios requeridos

Applet

Los principales cambios requeridos que corresponden al Applet para la adaptación de la solución eID Trust Service y Digital Signature Service para el SNDC de Costa Rica se listan a continuación por orden de relevancia:

I. Adecuar el Applet Manifest para compatibilidad con Java 1.7

El plugin vigente de Java al inicio del presente proyecto es de la versión 1.7 y requiere de atributos adicionales para permitir la ejecución de Applets que necesiten niveles de seguridad más elevados, estos son:

- Application-name
- Codebase

II. Implementar el procesamiento del IdentificationRequestMessage e IdentificationDataMessage

El Trust Service y el DIgital Signature Service requieren intercambiar información de Identidad entre el Applet y el Applet Service. Si bien para el presente proyecto no existe información de identidad (como nombre, dirección, fotografía, etc.), los certificados propios existentes en las tarjetas digitales serán utilizados para ese propósito.

III. Soportar Múltiples certificados para las Autoridades de Certificación

La PKI del Banco Central prevée la existencia de múltiples certificados para una Autoridad de Certificación, válidos en periodos de tiempo que pueden solaparse. El Applet debe garantizar la preparación de una cadena de certificados de confianza correspondiente al emisor de cada uno de los certificados pertenecientes a esta jerarquía, comenzando por el certificado a extraer de la tarjeta inteligente de Firma Digital.

IV. Implementar el procesamiento del SignCertificatesDataMessage y SignatureDataMessage

El Digital Signature Service requiere intercambiar información de los certificados de la cadena de confianza para la PKI del Banco Central entre el Applet y el Applet Service. Sus respectivas clases Handler deben ser adecuadas de forma correspondiente.

V. Adecuar la implementación del TSPTimeStampService

El Digital Signature Service permite la creación de firmas XAdES-X-L con persistencia en el tiempo para lo cual necesita consumir el servicio de Sellado de Tiempo del Banco Central de Costa Rica.

VI. Modificar firmado PKCS11

El Digital Signature Service requiere que el firmado PKCS11 sea compatible totalmente con las directrices RSA que validen según **http:www.w3.org/TR/xmldsig-core/#sec-CoreValidation**

Trust Service

Los principales cambios requeridos para la adaptación de la solución eID Trust Service para el SNDC de Costa Rica se listan a continuación por orden de relevancia:

I. Adecuar la inicialización de puntos de confianza y dominios de confianza

La PKI del Banco Central presenta una lista jerárquica de Autoridades de Certificación. Esta jerarquía debe ser integrada por defecto como parte de la inicialización de datos en instalaciones nuevas del Trust Service, como Puntos de Confianza y Dominios de Confianza Válidos.

II. Adecuar el procesamiento de la validación de Certificados

La PKI del Banco Central presenta certificados adicionales en comparación con la PKI que se tiene originalmente para eID de Bélgica.

III. Asegurar la coexistencia de múltiples certificados para una Autoridad de Certificación

La PKI del Banco Central prevee la existencia de múltiples certificados para una Autoridad de Certificación, en periodos de tiempo que pueden solaparse. Esta existencia se representa como múltiples Puntos de Confianza para los cuales se podrán disponer de datos de revocación independientes.

IV. Priorizar validación OCSP

La validación de Certificados debe ser realizada en primer lugar utilizando datos de referencia del servidor OCSP que se encuentran en los certificados digitales. La alternativa fail-safe se designa al procesamiento de CRLs.

V. Adecuar la Interfaz de usuario

La Imagen institucional del Micitt y el idioma español deberán ser provistos en el proyecto.

Digital Signature Service

Los principales cambios requeridos para la adaptación de la solución eID Digital Signature Service para el SNDC de Costa Rica se listan a continuación por orden de relevancia:

I. Adecuar la inicialización de los servicios

El Digital Signature Service requiere intercambiar información de Identidad entre el Applet y el Applet Service. Puesto que para el presente proyecto no existe información de identidad, los certificados propios existentes en las tarjetas digitales serán utilizados para ese propósito.

II. Adecuar el procesamiento de la validación de Certificados

La PKI del Banco Central presenta certificados adicionales en comparación con la PKI que se tiene originalmente para eID de Bélgica.

III. Habilitar compatibilidad con Office 2013

A la fecha del presente proyecto, el DSS no genera firmas digitales que sean reconocidas como válidas por Office 2013. La actual estructura del archivo XML de firma digital OOXML requiere de modificaciones para que la nueva versión de Office 2013 reconozca la firma digital generada por el DSS como válida. Estas modificaciones comprenden la generación de nodos adicionales dentro del documento XML de firma digital y modificación de la lógica de procesamiento de facets para la firma digital OOXML.

IV. Adecuar la Interfaz de usuario

La Imagen institucional del Micitt y el idioma español deberán ser provistos en el proyecto.

Identity Provider

Los principales cambios requeridos para la adaptación de la solución eID Identity Provider para el SNDC de Costa Rica se listan a continuación por orden de relevancia:

I. Adecuar el Procesamiento de mensajes de identidad.

El Identity Provider requiere intercambiar información de Identidad entre el Applet, el Applet Service y la aplicación web del Proveedor de Servicios. Puesto que para el presente proyecto no existe información de identidad, ciertos atributos necesitan ser explícitamente excluidos del procesamiento.

II. Adecuar los protocolos de intercambio de información

El Identity Provider provee Endpoints para que el Proveedor de Servicios solicite una Autenticación únicamente, Datos de Identidad únicamente o Autenticación + Datos de Identidad. Puesto que para el presente proyecto no existe información de identidad, los protocolos creados para proveer datos de Identidad deben ser deshabilitados explícitamente ya que no existe ninguna equivalencia de información con la que esta almacenada en las tarjetas del SNDC del Banco Central.

III. Adecuar la Interfaz de usuario

La Imagen institucional del Micitt y el idioma español deberán ser provistos en el proyecto.