Smartcard/PKI based Web SSO Authentication Using OpenSSO Enterprise 8

(Emphasis on using browser certificates, PIV, CAC and PKCS#15/eID Smartcards)

Ramesh Nagappan
Sun Microsystems

This is an unofficial (unedited draft) document for illustrating the usage of Smartcard/Browser based PKI certificates and keys for authentication and single sign-on (SSO) using Sun OpenSSO Enterprise 8 (Formerly referred as Sun Java System Access Manager). This document identifies the technical pre-requisites, architectural scenarios and configuration/deployment steps using Sun OpenSSO Enterprise and ISV partner technologies.

OpenSSO supports the use of PKI certificates from Browser or Smartcard/Token based PKI credentials for authentication and enabling Web SSO by determining the revocation status of the certificate through the use of the Online Certificate Status Protocol (OCSP), Certificate Revocation Lists (CRLs) and matching the certificate to a pre-existing certificate entry in LDAP.

Pre-requisites

1. Sun OpenSSO Enterprise 8 or above
2. Sun GlassFish Enterprise v2.1 or Sun Web Server 7.0 (or above)
    a. Must be configured with an NSS Keystore
    b. PKCS#11/HSM based Keystore (optional).
        - Sun Cryptographic Accelerator (SCA-6000)
3. Sun Java System Directory Server EE6 or Sun OpenDS (Bundled with OpenSSO 8)
    a. Repository for user accounts and its corresponding PKI certificate entries (optional).
4. PKI Provider
    a. Certificate and Validation Authority
        - Certificate Authority: Cybertrust / Entrust / Microsoft / Verisign
        - OCSP Responders: Tumbleweed / Corestreet OCSP Validator
    b. Root CA Certificates and CRLs
        - FBCA SSP CA certificates and CRLs (For PIV cards)
        - DoD CA/ECA root certificates and CRLs (For CAC cards)
        - Govt PKI Root CA certificates and CRLs (For eID cards)
    c. OCSP Signing certificate
5. Smartcard Reader
6. Smartcard client middleware – Browser Plug-in (PKCS#11 or MS-CAPI)
    a. ActivIdentity (ActivClient PKI 6.0 / CAC 6.0 or above)
    b. GemAlto (GemSAFE)
    c. OpenSC PKCS#11 (OpenSC.org)
7. Smartcards provisioned with PKI certificates

Architectural Strategies

OCSP based Certificate Validation

In this strategy, OpenSSO determines the revocation status of the certificate by issuing a real-time status request and confirms the status by accepting the response from the OCSP responder. OpenSSO 8 supports OCSP based certificate validation by sending OCSP request validation to an

OCSP responder URL (Validation authority or CA) specified in the PKI certificate credential (On the Smartcard) – usually available as an Authority Information Access (AIA) extension attribute (RFC3280). If the AIA attribute is not present, OpenSSO will send the OCSP request to designated OCSP responder URL specified in the OpenSSO Certificate Module configuration (Figure 1).

OpenSSO 8 supports issuing signed OCSP requests by making use of OCSP signing certificates stored in the Web container's NSS keystore or HSM.

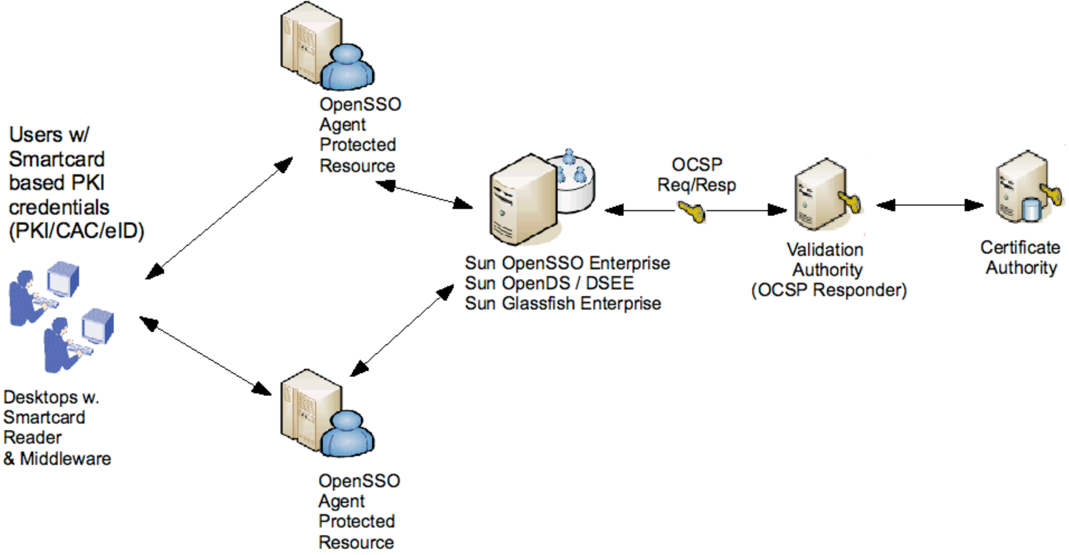


Figure 1: Logical Architecture - OCSP based Validation Strategy

Matching PKI certificates in LDAP/CRL Repository

In this strategy, OpenSSO determines the validity of the PKI certificate by matching the public-key certificate against the user's LDAP directory entry stored in a local or remote LDAP repository. OpenSSO uses the X.509 attributes from the certificate (ex. SubjectDN attributes including uid, emailAddress, serialNumber etc) for searching and retrieving the stored certificate entry in LDAP (Refer Figure-2).

If the user's certificate matches the retrieved certificate – the authentication is considered successful. As a pre-requisite, the cardholder's public-key certificate from the Smartcard must be pulled out and then stored as an *userCertificate;binary* attribute of the user account in the LDAP directory.

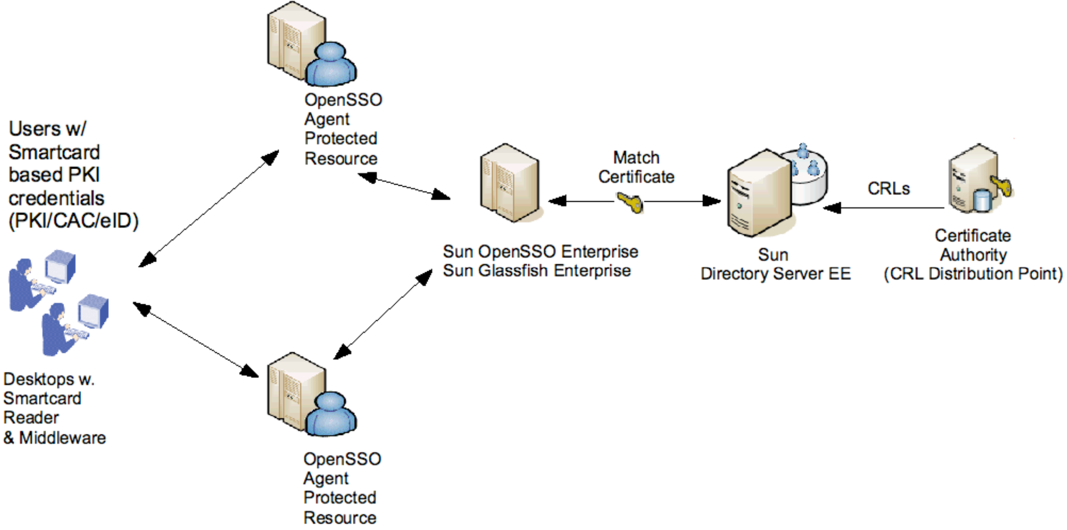


Figure 2: Logical Architecture - Matching to LDAP/CRL entries

OpenSSO also supports matching certificates to CRLs in an LDAP repository. This means OpenSSO uses the Issuer's DN attribute for searching CRLs in LDAP repository. If the certificate is identified on the CRL; the user authentication is denied. As a pre-requisite, the CRLs must be imported into the LDAP directory. If the user's certificate includes a *CRLDistributionPointsExtension or IssuingDistributionPointExtension* attribute that identifies the location of CRL distribution points where the CRLs are available.

In a real-world scenario, OCSP based certificate validation is overwhelmingly preferred as a best practice over matching certificates using LDAP or CRLs as they require caching them locally, frequency of updates and concerns related to timestamps, authenticity and integrity.

Configuration and Deployment

1. Choosing a Web container and Keystore for OpenSSO

The OpenSSO server is required to be deployed on a Web container that supports a FIPS conformant keystore (ex. NSS Keystore or a FIPS-140 compliant HSM). Using Sun Glassfish v2.1 Enterprise server or Sun Web server 7.0 provides support for using NSS Keystore or a PKCS#11 compatible HSM keystore.

Make sure the Web container for NSS keystore or configuration of a HSM keystore before installing OpenSSO. An easy way to verify NSS keystore in Glassfish, check for the existence of key3.db and cert8 in Glassfish config directory. To use PKCS#11 based HSM keystore in Glassfish Enterprise; it can be done using *'modutil'* utility. For example:

```
modutil -dbdir /opt/SUNWappserver/domains/domain1/config
                                      -nocertdb    -add "Crypto Provider"
                                                          -libfile     /usr/lib/libPkcs11Provider.so
```

To list the currently available security modules (secmod.db):

```
modutil -list –dbdir /certDirectory
```

2. Deploy OpenSSO WAR

Install the OpenSSO deployable WAR in the Web Container and make sure all the installation pre-requisites are complete and configuration is verified to include a default realm using a LDAP compliant directory server (ex. Sun Java System Directory Server EE).

For detailed instructions, on installing OpenSSO and pre-requisites refer to Sun OpenSSO Enterprise 8 Installation Guide.

3. Configure Web Container for SSL with Client-certificate authentication

Configure SSL by creating a dedicate HTTP listener to support SSL based communication including client-certificate authentication. Make sure the Glassfish / Web server is configured with SSL certificates issued by a trusted certificate authority (CA). Enable client-certificate option to ensure both the client and server exchange the certificates for ensuring trusted communication.

To enable SSL using Trusted CA (not self-signed) issued valid certificates, we need to use the following steps in Glassfish Enterprise / Sun Web server using *'certutil'* utility:

a. Create a Certificate signing request (CSR) - For example, to create a CSR with key type RSA with key size 2048 and the following Subject DN:

```
certutil  -R  -k RSA –g 2048 -s "CN=pivone.east.sun.com, OU=MDE
                                   O=Sun Microsystems, L=Mountain View,
                                       ST=Massachusetts, C=US" -p "777-555-8888"
                                                              -o mycert.csr -d /certDirectory
```

b. Send the CSR file to your CA for processing and signing the request. The CA will issue a CA signed certificate (For SSL) along with CA’s Root chain certificates.

c. To install this certificate, you need add the CA issued certificate to the Glassfish’s certificate database:

```
certutil  -A  -n Server-Cert  -t  "u,u,u"  -i  mycert.crt  -d  /certDirectory
```

In addition, you must add the CA’s Root chain certificates:

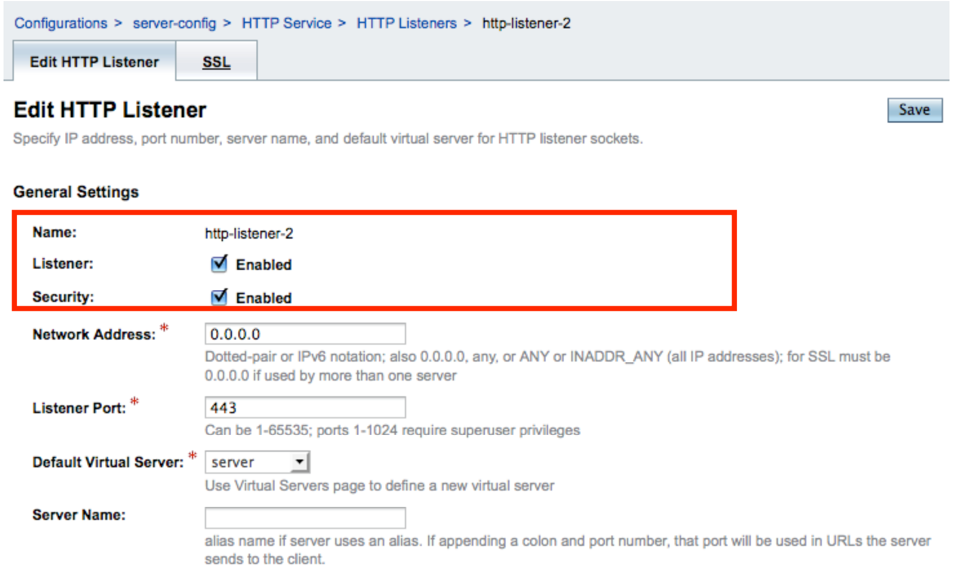
```
certutil –A –n CARootCertificate –d . –i CARoot.crt –t “CT,CT,CT”
```

d. To list all the certificate entries:

```
certutil -L -n Server-Cert -d  /certDirectory
```

After installing the SSL certificates, configure a HTTP Listener in the Glassfish Enterprise / Web Server to support SSL communication and enable Client authentication.

In Glassfish Enterprise, Open application server administration console and from the left side menu select Configuration> HTTP Service> HTTP Listeners> and create New HTTP Listener (ex. http-listener-2) and check the “Listener” and “Security” check boxes. Then select the SSL Tab, and specify the certificate nickname “Server-Cert”, and then make sure that you have checked “Client Authentication” check box.


Configurations > server-config > HTTP Service > HTTP Listeners > http-listener-2
Edit HTTP Listener
SSL
Save

Edit HTTP Listener

Specify IP address, port number, server name, and default virtual server for HTTP listener sockets.

General Settings

Name: http-listener-2

Listener: ☑ Enabled

Security: ☑ Enabled

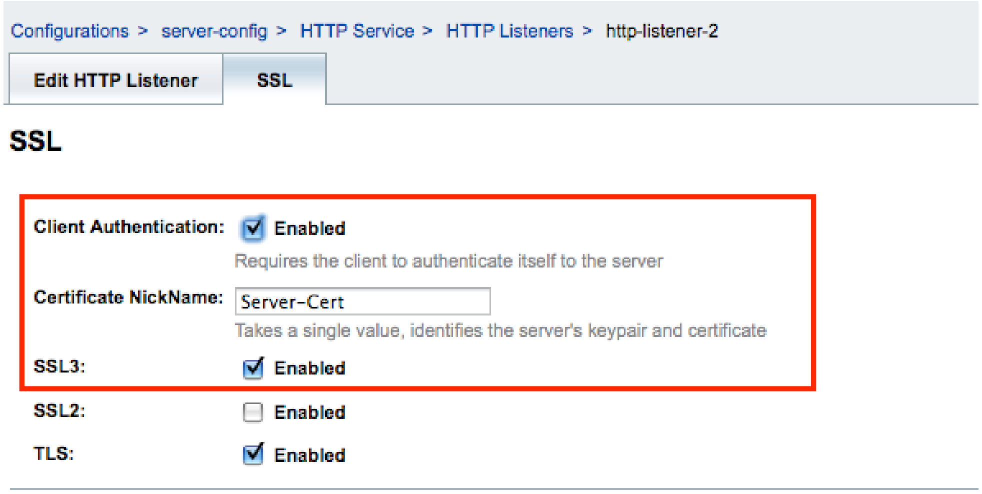
Network Address: * 0.0.0.0
Dotted-pair or IPv6 notation; also 0.0.0.0, any, or ANY or INADDR_ANY (all IP addresses); for SSL must be 0.0.0.0 if used by more than one server

Listener Port: * 443
Can be 1-65535; ports 1-1024 require superuser privileges

Default Virtual Server: * server
Use Virtual Servers page to define a new virtual server

Server Name:
alias name if server uses an alias. If appending a colon and port number, that port will be used in URLs the server sends to the client.

Figure 3: Add a HTTP Listener to support SSL Security (Glassfish Enterprise v2.1)


Configurations > server-config > HTTP Service > HTTP Listeners > http-listener-2
Edit HTTP Listener
SSL

SSL

Client Authentication: ☑ Enabled
Requires the client to authenticate itself to the server

Certificate NickName: Server-Cert
Takes a single value, identifies the server's keypair and certificate

SSL3: ☑ Enabled

SSL2: □ Enabled

TLS: ☑ Enabled

Figure 4: Enable “Client Authentication” and “Certificate NickName” (Glassfish Enterprise v2.1)

Once complete, restart the Glassfish Enterprise server. For more detailed instruction on configuring SSL and Client certificate authentication refer to the Glassfish Administration Guide and Sun Java System Web Server Administration guides.

4. Install Smartcard/PKI Issuing CA’s Root and OCSP Signing Certificates

To support trusting the Smartcard based PKI certificates, all relevant Root CA certificates of the PKI credential issuing CA must be installed in the Web container’s certificate database. This include the Root CA and OCSP signing certificates (if required) that support trusting the user’s PKI certificate credentials issued on the Smartcard.

```
certutil -A -n "OCSPSigningCertificate" -d . -i OCSPSigner.cer -t "CT,CT,CT"
```

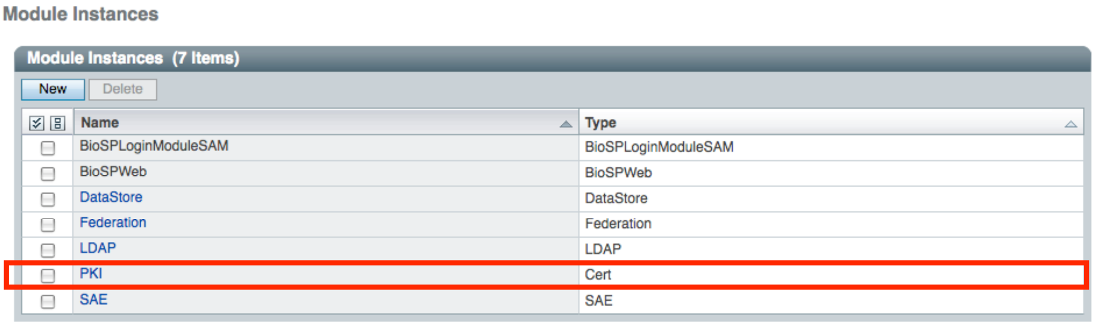
For PIV cards, the Root CA certificate bundle and OCSP signing certificates can be obtained from the Managed Services’s PKI SSP. In case of DoD PKI, these certificates can be obtained from the ECA PKI provider supporting the DoD agency.

Verify the list of certificates loaded in to the Web container’s certificate database and the restart the Web container server.

5. Configure the OpenSSO Certificate Authentication Module

The OpenSSO Certificate Authentication module enables a user to log in using PKI certificate credential made available through a Web client (Browser keystore configured with a user certificate or configured to use a Smartcard Token as keystore). The user is granted or denied access to a protected resource by determining the state of PKI certificate based on whether or not the provided certificate is valid through the use of OCSP, CRLs and matching the certificate to a pre-existing entry in LDAP.

To configure the Certificate Module, login to the OpenSSO administration console using *amadmin* , select the “Access Control” tab, select your default “Realm”, select “Authentication”. Click on “Module Instances” and click on “New” to create a Module instance. Assign a name to the module (ex. PKI) and select “Certificate” as type (Refer Figure 5).



Module Instances

Module Instances (7 Items)

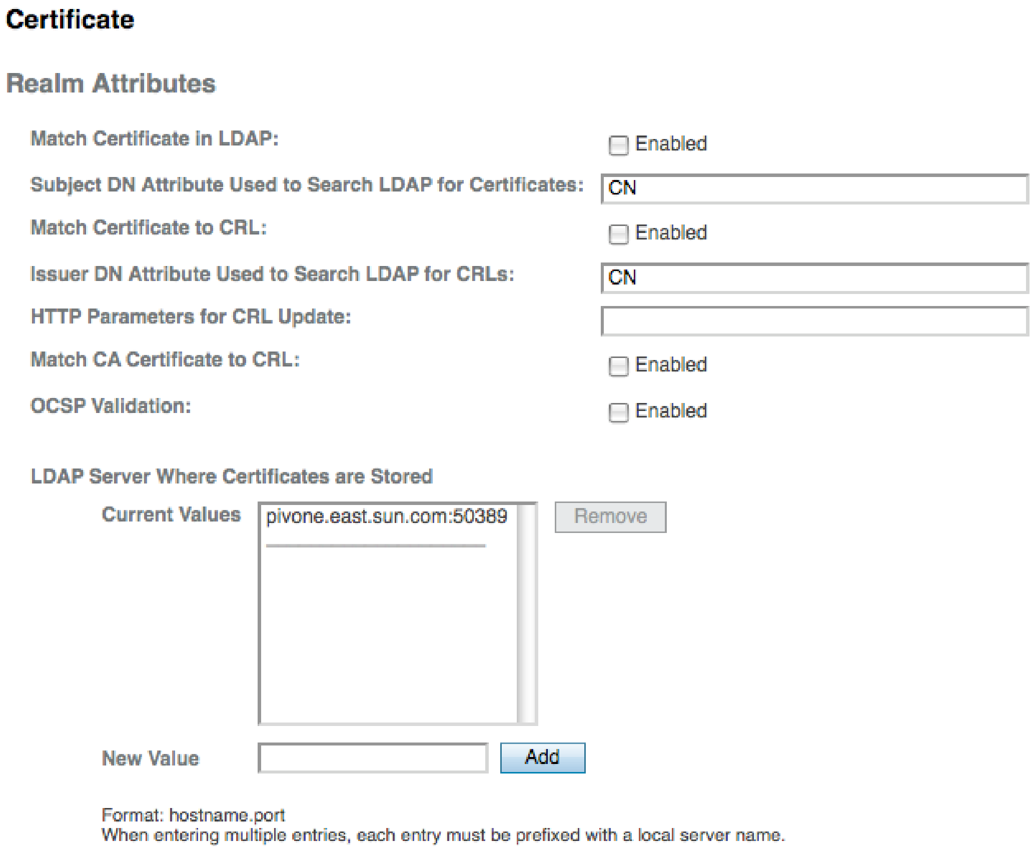
□	Name	Type
□	BioSPLoginModuleSAM	BioSPLoginModuleSAM
□	BioSPWeb	BioSPWeb
□	DataStore	DataStore
□	Federation	Federation
□	LDAP	LDAP
□	PKI	Cert
□	SAE	SAE

Figure 5: Create a new Module instance using "Certificate"

6. Configuring OCSP based Certificate Validation

If the CA issuer of the user's PKI credential supports the use of OCSP for certificate status verification, OpenSSO support sending OCSP requests to the OCSP responder. In most cases, the user's PKI credential will specify an AIA attribute showing an OCSP responder URL. If the CA issuer is an SSP (ex. FBCA, ECA), and they would use an OCSP validation authority that aggregates access to multiple CAs and manage the OCSP requests on behalf of the CAs.

To configure "Certificate Module" and its attributes, go to the Module instances and select the newly created Module instance "PKI". OpenSSO will redirect you to the configuration of "Certificate Module" realm attributes (Figure 6).



Certificate

Realm Attributes

Match Certificate in LDAP: □ Enabled

Subject DN Attribute Used to Search LDAP for Certificates: CN

Match Certificate to CRL: □ Enabled

Issuer DN Attribute Used to Search LDAP for CRLs: CN

HTTP Parameters for CRL Update:

Match CA Certificate to CRL: □ Enabled

OCSP Validation: □ Enabled

LDAP Server Where Certificates are Stored

Current Values: pivone.east.sun.com:50389

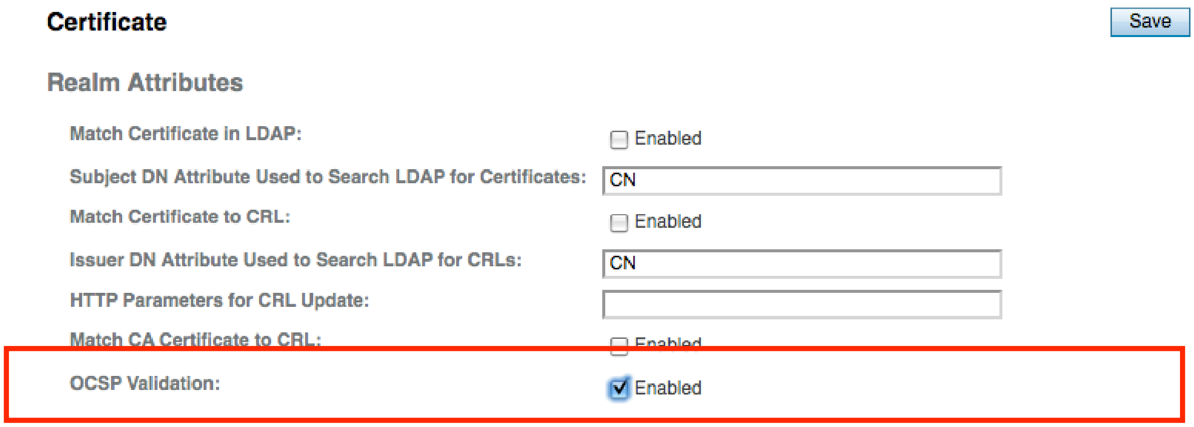
New Value:

Format: hostname.port
When entering multiple entries, each entry must be prefixed with a local server name.

Figure 6 : Configuring the OpenSSO Certificate Authentication Module

You may need to configure OpenSSO for one of the following options depending on the location of the OCSP responder of the PKI provider or using OCSP validation authority.

a. Use Certificate's AIA Attribute for CA's OCSP Responder
If the user's PKI credential includes an AIA attribute that specifies the actual OCSP responder URL, then select "OCSP Enabled" checkbox.



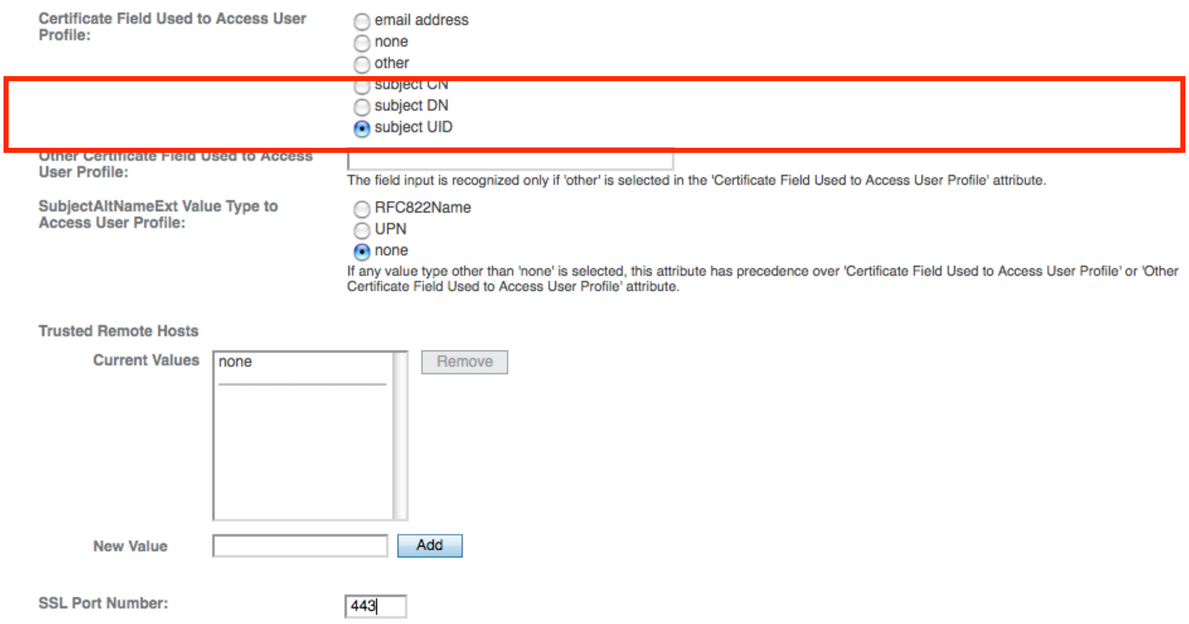
Certificate Save

Realm Attributes

Match Certificate in LDAP:	□ Enabled
Subject DN Attribute Used to Search LDAP for Certificates:	CN
Match Certificate to CRL:	□ Enabled
Issuer DN Attribute Used to Search LDAP for CRLs:	CN
HTTP Parameters for CRL Update:	
Match CA Certificate to CRL:	□ Enabled
OCSP Validation:	☑ Enabled

Figure 7: Enabling OCSP

Make sure to choose the appropriate certificate field to access the OpenSSO user account profile (ex. emailAddress, subject UID). If the OpenSSO user profile does not contain the specified X.509 attributes may need to extend the LDAP schema and make sure the specified attribute exists. Also, specify the SSL port number of the Web container. The port number must match the port number used for the web container's SSL client authentication HTTP listener port. Once complete, click on the Save button and then directly go to Step 9.



Certificate Field Used to Access User Profile:	○ email address ○ none ○ other ○ subject CN ○ subject DN ● subject UID
Other Certificate Field Used to Access User Profile:	The field input is recognized only if 'other' is selected in the 'Certificate Field Used to Access User Profile' attribute.
SubjectAltNameExt Value Type to Access User Profile:	○ RFC822Name ○ UPN ● none If any value type other than 'none' is selected, this attribute has precedence over 'Certificate Field Used to Access User Profile' or 'Other Certificate Field Used to Access User Profile' attribute.

Trusted Remote Hosts

Current Values: none

SSL Port Number: 443

Figure 8: Additional attributes to access OpenSSO user profile.

The above configuration is tested and works well for National eID cards (based on PKCS#15 standard) particularly Belgium eID (BELPIC), Finnish FINEID, Swedish Posten eID.

b. Using a OCSP Validation Authority

To enforce OpenSSO to use an "OCSP Validation Authority URL" overriding the AIA attribute specified in the PKI certificate, in addition to following the steps mentioned in "Option (a) Use Certificate's AIA Attribute for OCSP Responder" - you must add the following lines in your AMConfig.properties (located in the OpenSSO deployment directory of the Web container).

```
com.sun.identity.authentication.ocspCheck=True
com.sun.identity.authentication.ocsp.responder.url=<OCSP ValidN Auth URL>
com.sun.identity.authentication.ocsp.responder.nickname=<Cert NickName>
```

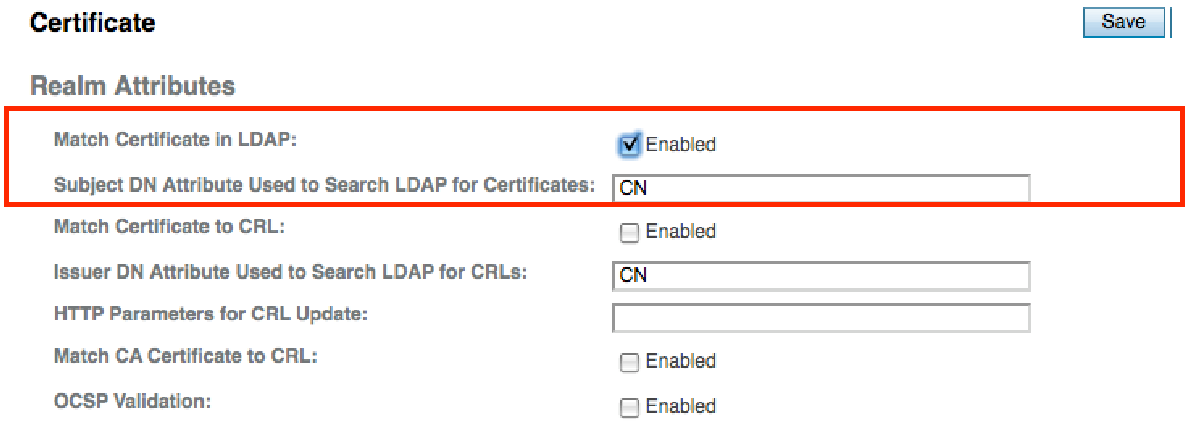
After setting these values restart the Web container to take effect. The above values are usually provided by the PKI provider (FBCA SSP (for PIV cards) and ECA (DoD CAC cards), who manages the OCSP responder or the validation authority (OCSP validation authority ex. Corestreet and Tumbleweed).

7. Configuring Match to LDAP/CRLs based Certificate Validation

OpenSSO supports certificate validation by matching user's public-key certificate stored in a local or remote LDAP repository or matching the certificates to CRLs in a LDAP repository.

a. Match certificate to LDAP

If you choose the match certificate in LDAP strategy, go to the "Module instances" in Authentication tab and select the newly created Module instance "PKI" to initiate Certificate module configuration , select "Match Certificate in LDAP" checkbox "Enabled" and specify a "Subject DN" attribute that allows searching and identifying the certificates in LDAP (For ex: CN, uid, emailAddress, serialNumber).



Certificate

Realm Attributes

Match Certificate in LDAP: ☑ Enabled

Subject DN Attribute Used to Search LDAP for Certificates: CN

Match Certificate to CRL: □ Enabled

Issuer DN Attribute Used to Search LDAP for CRLs: CN

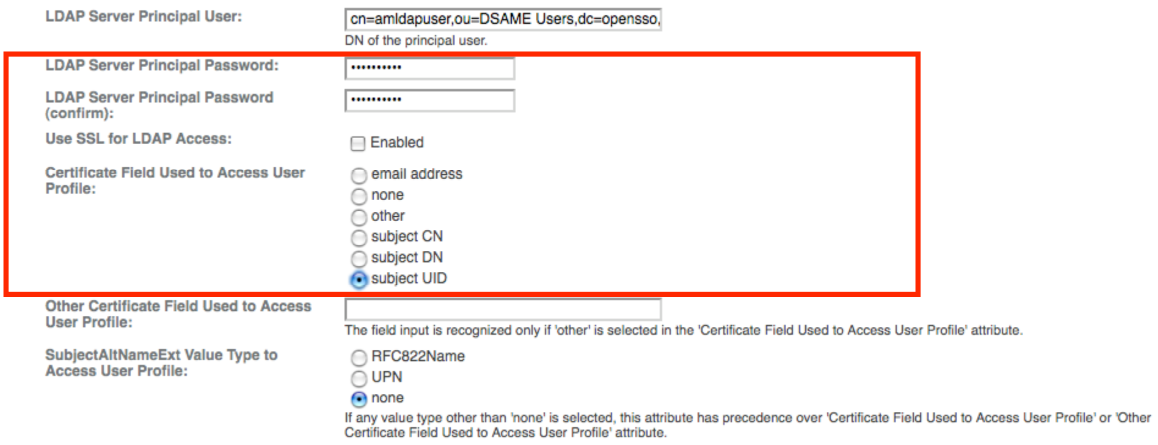
HTTP Parameters for CRL Update:

Match CA Certificate to CRL: □ Enabled

OCSP Validation: □ Enabled

Figure 9: Enabling Match Certificate to LDAP

Then, specify the password for the LDAP principal and identify the Subject DN attribute from the certificate than can be used for accessing the OpenSSO user account profile. Once complete save the configuration.



LDAP Server Principal User: cn=amldapuser,ou=DSAME Users,dc=opensso,
DN of the principal user.

LDAP Server Principal Password: ••••••••••

LDAP Server Principal Password (confirm): ••••••••••

Use SSL for LDAP Access: □ Enabled

Certificate Field Used to Access User Profile:
○ email address
○ none
○ other
○ subject CN
○ subject DN
● subject UID

Other Certificate Field Used to Access User Profile:
The field input is recognized only if 'other' is selected in the 'Certificate Field Used to Access User Profile' attribute.

SubjectAltNameExt Value Type to Access User Profile:
○ RFC822Name
○ UPN
● none
If any value type other than 'none' is selected, this attribute has precedence over 'Certificate Field Used to Access User Profile' or 'Other Certificate Field Used to Access User Profile' attribute.

Figure 10: Additional attributes to access OpenSSO user profile

As it is critical to have the certificates stored in LDAP, it becomes important to know the steps involved:

i. Convert the user's certificate from PEM to DER format.
The easiest way is to use 'openssl' utility – here is an example:

```
openssl x509 -myCertificate.pem
                   -inform PEM -outMyCertificate.der -outform DER
```

ii. After conversion to DER, using the 'ldif' utility you need to create an LDIF file from the DER file. The ldif utility is usually located in the ~/bin/slapd/server directory of your LDAP.

```
ldif -b "usercertificate;binary"
                       < outMyCertificate.der   > myCert.ldif
```

The contents of the LDIF file will look like this:

```
usercertificate;binary::MIIE8HEEBFagAwIBAgIEOAOR7jANBgkqhkiG9w0BAQ
QFADCByTELMAkGA1UEBhMCVVMxFDASBgNVBAoTC0VudHJ1c3QubmV0MUgwRgYDVQQL
 MxFDASBgNVBAoTC0VudHJ1c3QubmV0MUgwRgYDVQQLMxFDASBgNVBAoTC0VudHJ1c3Q
 . . .
```

iii. Next, modify the existing LDAP user account entry to include the user's certificate. To support modifying the LDAP entry to include the LDIF file, you need to prepend the user's entry DN entry for the certificate attribute.

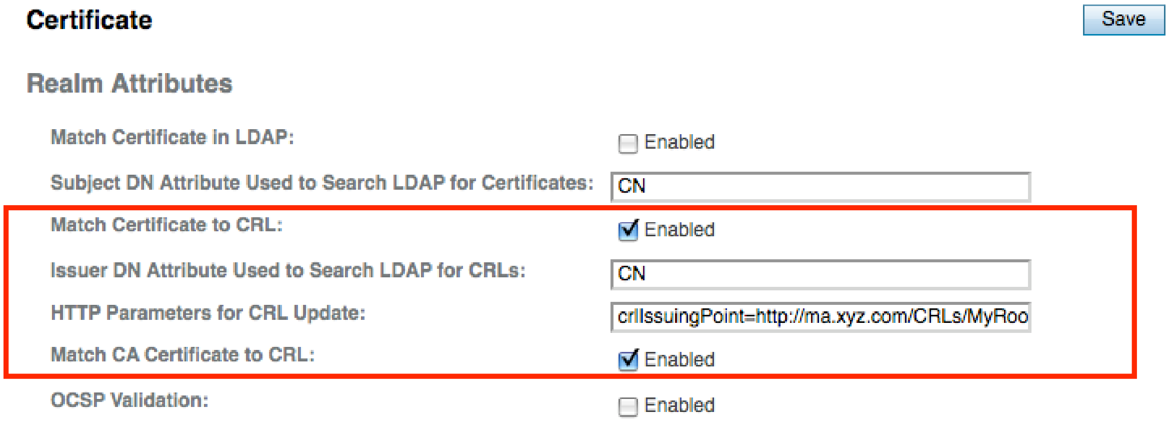
```
# entry-id: ramesh
dn: uid=ramesh,ou=People,dc=opensso,dc=java,dc=net
 usercertificate;binary::MIIE8HEEBFagAwIBAgIEOAOR7jANBgkqhkiG9w0BAQ
 QFADCByTELMAkGA1UEBhMCVVMxFDASBgNVBAoTC0VudHJ1c3QubmV0MUgwRgYDVQQL
 MxFDASBgNVBAoTC0VudHJ1c3QubmV0MUggYDVQQLMxFDASBgNVBAoTC0VudHJ1c3Q
 . . .
```

iv. Now, run the ‘ldapmodify’ command to add the certificate (in LDIF) to an existing directory entry.

```
ldapmodify -r -h <directory-hostname>
                 -p port -f myCert.ldif -D
                             cn=Directory Manager -w password
```

b. Match certificate to CRLs

If you choose to enable match certificate to the CRLs in LDAP, then select “Match Certificate in CRL” checkbox “Enabled” and specify a “Subject DN” attribute that allows searching and identifying the certificates in LDAP (for example – CN, uid, emailAddress, serialNumber). Also, if the configured CRL or matched user certificate has *IssuingDistributionPointExtension* or *CRLDistributionPointsExtension*, then OpenSSO Certificate module automatically updates the local CRLs. To download the CRL from the CRL distribution point, you need to specify the HTTP URL of *CRLDistributionPointsExtension* or *IssuingDistributionPointExtension* in “HTTP Parameters for CRL update”.



Certificate

Realm Attributes

Match Certificate in LDAP: ☐ Enabled

Subject DN Attribute Used to Search LDAP for Certificates: CN

Match Certificate to CRL: ☑ Enabled

Issuer DN Attribute Used to Search LDAP for CRLs: CN

HTTP Parameters for CRL Update: crlIssuingPoint=http://ma.xyz.com/CRLs/MyRoo

Match CA Certificate to CRL: ☑ Enabled

OCSP Validation: ☐ Enabled

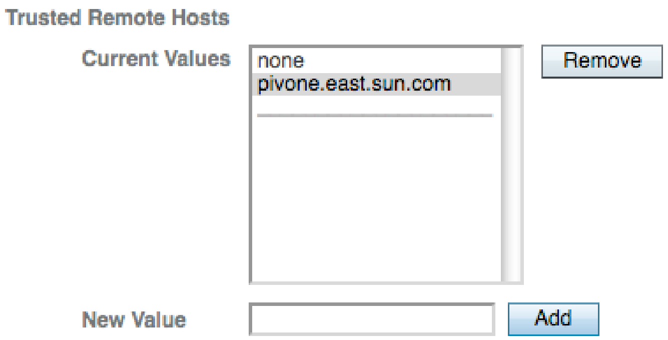
Figure 11: Matching Certificate to CRLs

Once complete save the configuration.

Regarding importing CRLs to LDAP, you may follow the steps as described for adding certificate entry instead of using *usercertificate;binary* attribute you need to *certificaterevocationlist;binary* attribute.

8. Managing SSL Termination and Load balancer

In case of load balancing and SSL termination managed by front-end switches and load balancers it becomes important to identify the a list of trusted hosts that send the certificates that can be trusted by OpenSSO. To support this scenario, you may add the list of trusted hosts that are authorized to send certificates.



Trusted Remote Hosts

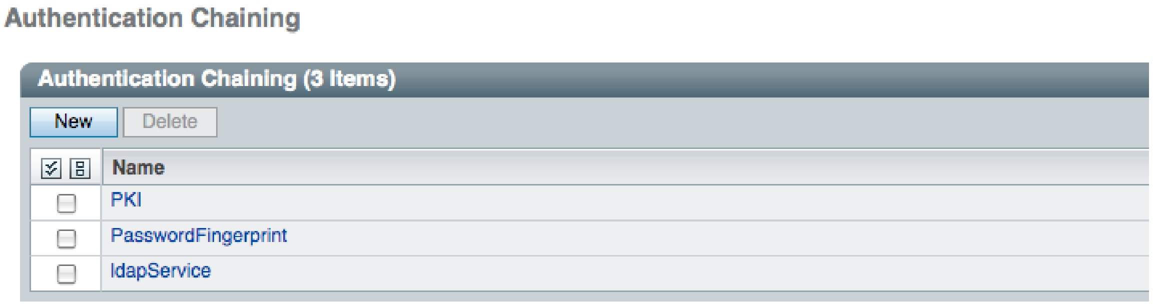
Current Values
none
pivone.east.sun.com

New Value

Figure 12: Identifying Trusted Hosts supporting SSL termination

9. Configure the OpenSSO Authentication Chain for Certificate Module.

This is last step with OpenSSO to deploy the Certificate Module for use with protecting resources. To create an authentication chain, go to the OpenSSO administration console, click on "New", assign a name to the authentication chain (ex. PKI) and choose the "PKI" module instance and select "Required".



Authentication Chaining

Authentication Chaining (3 Items)

☑	Name
□	PKI
□	PasswordFingerprint
□	ldapService

Figure 13: Authentication Chain configuration

Now the OpenSSO certificate module is available for use with PKI certificate credentials.

10. Client Configuration: Browser Certificates or Smartcard Client Middleware

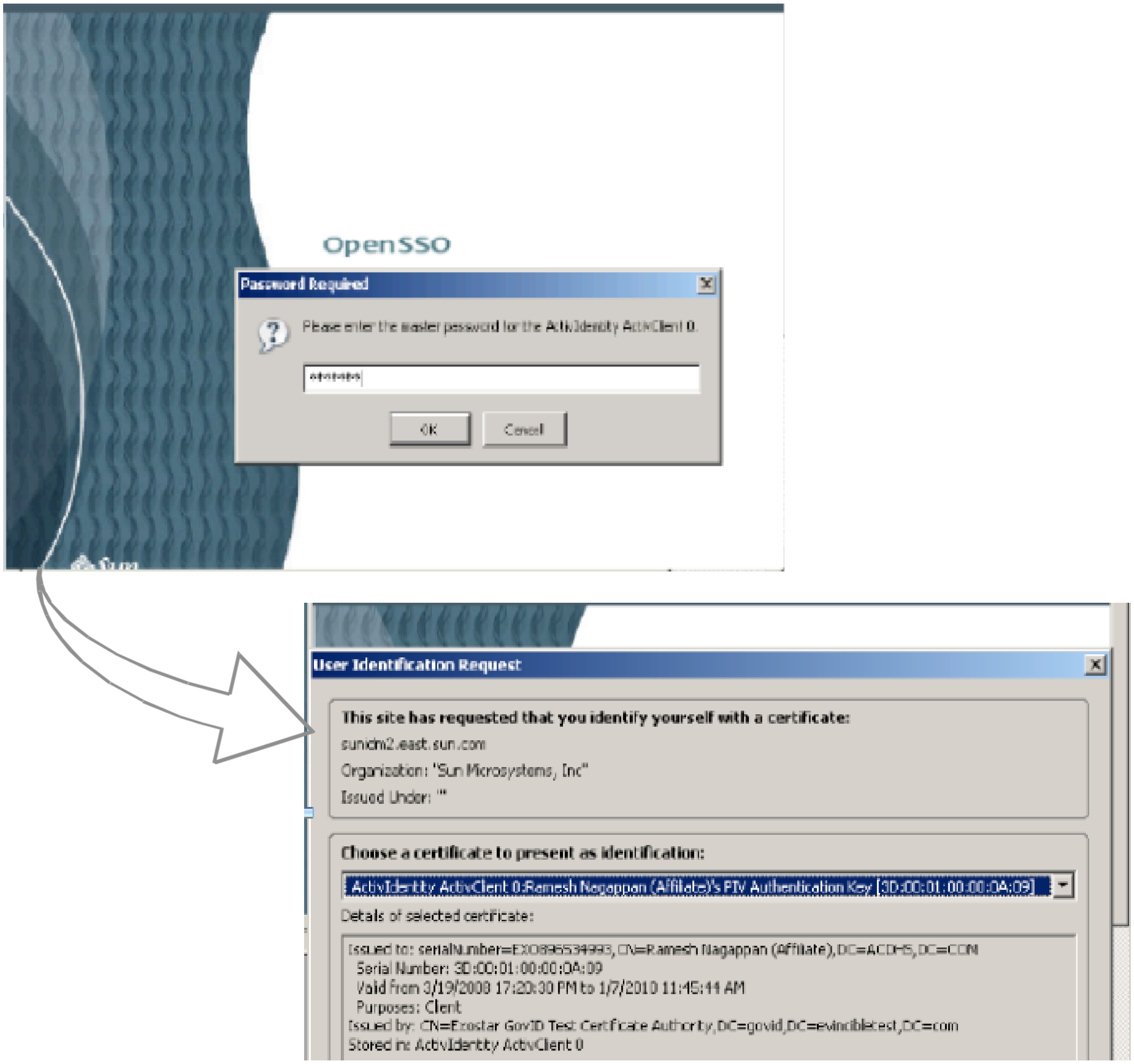
To support client authentication using PKI certificates, it is critical to install user's certificates in the browser or make use of a Smartcard client middleware that provide PKCS#11/MSCAPI based interfaces for accessing PKI certificate credentials on a Smartcard (ex. CAC, PIV, eID). The Smartcard client middleware helps represent the Certificate module callback and also prompts the user for Smartcard insertion and PIN during authentication.

a. To install user's PKI certificates, it is recommend following the instructions of using certificates with your Web browser (Internet Explorer / Firefox / Mozilla).
b. In case of using Smartcard based PKI certificates, you must plugin your USB Smartcard reader and then start installing Smartcard client middleware. The Smartcard middleware automatically detect the reader and install the required browser plugins for supporting PKI based authentication.
    i. Refer to your Smartcard client middleware provider's installation guide for detailed instructions.

ii. In case of using OpenSC PKCS#11 Plugin, refer to the installation and usage documentation specific to your target operating system at : http://www.opensc-project.org/opensc/wiki/OperatingSystems

11. Verify and Test the Configuration and Deployment

Try https://<GlassFish>:443/opensso/UI/Login?module=PKI , you will be prompted (To insert Smartcard in case of browsers configured to use Smartcard based PKI certificates) and to enter PIN for accessing the card.



Please enter the master password for the ActivIdentity ActivClient 0.

******

This site has requested that you identify yourself with a certificate:

sunidm2.east.sun.com

Organization: "Sun Microsystems, Inc"

Issued Under: ""

Choose a certificate to present as identification:

ActivIdentity ActivClient 0:Ramesh Nagappan (Affiliate)'s PIV Authentication Key [3D:00:01:00:00:0A:09]

Details of selected certificate:

```
Issued to: serialNumber=EXO896534993,CN=Ramesh Nagappan (Affiliate),DC=ACDHS,DC=COM
  Serial Number: 3D:00:01:00:00:0A:09
  Valid from 3/19/2008 17:20:30 PM to 1/7/2010 11:45:44 AM
  Purposes: Client
Issued by: CN=Exostar GovID Test Certificate Authority,DC=govid,DC=evincibletest,DC=com
Stored in: ActivIdentity ActivClient 0
```

Figure 14: Test driving Smartcard/PKI Authentication

For all troubleshooting, refer to the OpenSSO logs in debug mode. In case of using OCSP and CRLs, you may need to verify the URL access to the OCSP responder, CRL distribution points and check the requirements for signed OCSP requests and validity of certificates.