Introduction to HANA Cloud Integration Security
Securing your communications
# Table of Contents

**Introduction** ................................................................. 4  
Customer Installation: .............................................................. 4  
H CI Tenant: .............................................................................. 4  
Inbound Messages: ............................................................... 4  
Outbound Messages ............................................................. 4  

**Certificates and Public Key Infrastructure** .................................. 4  

**Initial requirements** ............................................................ 8  
Save the allocated Tenant details........................................... 8  
Ensure a Certificate Signing process is in place ..................... 8  
Install SAP GUI ................................................................. 8  
Install Eclipse & the HCI Tooling .......................................... 8  
Install Keystore Explorer ...................................................... 9  

**Certificate Storage in SAP Installations** ................................... 11  
Personal Storage Environments ........................................... 11  
  * The Server PSE .......................................................... 11  
  * The Anonymous Client PSE ....................................... 11  
  * SSL client SSL Client (Standard)/SSL Client PSE .......... 12  

**Creating certificates in SAP GUI** ........................................ 12  
Server PSE Configuration .................................................... 12  
  * Adding a certificate .................................................. 13  
Anonymous Client PSE Configuration .................................. 14  
Client PSE Configuration .................................................... 15  
  * Adding a certificate .................................................. 15  
Restart ICF ......................................................................... 18  

**Using certificates in SAP GUI** .............................................. 19  

**Creating Certificates with Keystore Explorer** .......................... 20  
Deploying Certificates to the HCI Tenant ................................ 26  

**Authentication options in HCI** ............................................ 27  
Basic Authentication ............................................................ 28  
  * HCI Inbound .......................................................... 28  
  * Requirements ....................................................... 29  
  * Process ................................................................. 30  
  * HCI Outbound ....................................................... 32  
  * Requirements ....................................................... 33  
  * Process ................................................................. 33  
Certificate Authentication Overview ...................................... 36  
  * HCI Inbound .......................................................... 36  
  * Requirements ....................................................... 37  
  * Process ................................................................. 38  
  * Outbound ............................................................ 38  
  * Requirements ....................................................... 39  
  * Process ................................................................. 39  
SAP Managed process .......................................................... 40
**Introduction**

This paper will look at common customer configurations around implementing network communication security in the HANA Cloud Integration (HCI) scenarios.

When beginning its best to consider the different scenarios and the frames of reference around terms so that the reader is clear about the different configurations:

**Customer Installation:**
Refers to an ERP system running Process Integration or Process Orchestration but it is not restricted to this, it could just as easily be a third party landscape sending messages to HCI.

**HCI Tenant:**
This is the installation that is running on the HANA cloud platform and is subscription licensed by SAP who provides the infrastructure to companies. This will be referred to as the HCI Tenant. Each customer will have one or more tenants.

This paper will focus around security protocols in general as well as security landscapes for the on premise installation and the HANA Cloud Infrastructure, specifically around the implementation of Basic and Certificate authentication for messages being transferred to and from the HCI Tenant and the Customer Installation since this is often the most complex portion of the environment to setup.

**Inbound Messages:**
These are the messages that are being received by the specified server. It will be clear in the documentation which server (HCI Tenant, Customer Installation) is being referred to.

**Outbound Messages**
These are the messages that are being delivered by the specified server. It will be clear in the documentation which server is being referred to.

**Certificates and Public Key Infrastructure**

Fundamental to secure communication on the Internet is the use of public key cryptography (PKI).

Wikipedia defines PKI as:

"A public key infrastructure (PKI) is a set of hardware, software, people, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates[1] and manage public-key encryption. The purpose of a PKI is to facilitate the secure electronic transfer of information for a range of network activities such as e-commerce, Internet banking and confidential email."

PKI thus manages certificates and the SSL/TLS protocol for infrastructure. The certificates use the X.509 standard and when used securely can verify the identity of the server and the client. Though SSL has been
replaced by the newer and stronger Transport Layer Security (TLS), the name SSL is still used though the protocol has been retired. Throughout this paper there will be references to SSL, however this will be synonymous with Transport Layer security.

SSL is based on a hierarchical model of trust where Certificate Authorities (CA’s), are the very fundamental entities on which both parties involved in a SSL communication must know and trust. If any of them does not know about them or does not trust them, SSL will not work since basically they cannot build trust on what client or server are claiming to be.

Certificates by themselves are much like business cards in that they provide a means of identity but do not prove the owner of the certificate is who they say they are. The actual encryption stage is done later when establishing a secure connection by asking the server to encrypt some data and see if the data supplied can be correctly decrypted using the supplied server certificate. Since only the server can encrypt data that can be unencrypted using the public key held in the certificate, this proves the server is the owner of the certificate and can be trusted. If the data is not decrypted correctly, then the server does not possess the correct public key and so is not the owner of the certificate.

The top of the certificate tree this is referred to as the Root Certificate and is issued by the Root Certificate Authority or the Root CA. Each intermediate certificate must be able to trust its issuer up to the point where the certificate the server or client uses to prove their identity is reached. This certificate is called the client certificate. The terms Root Certificate and Client Certificate will be used throughout this document so it is important they are well understood. The root certificate is the one everyone trusts and is normally shipped with the browser The intermediate certificates are trusted only because their trust is proven by the root certificate. Finally, the client certificate that is your certificate is the one used in your communications to prove your identity. Of course it is also possible for an intermediate CA to issue a certificate that can be used as proof of your identity.

Mathematically speaking that trust is computed as a digital signature bound to the SSL certificate.

No matter which tool you use, the process of generating SSL certificate is fundamentally as follows:

1. The customer generates a private and public key pair according to the rules of a given encryption algorithm and with a given size of bits. That key pair consists of a public and a private key and will make up the two fundamental pieces of the final SSL certificate.
2. The customer also establishes a subject for the certificate. This will assign an identity to the certificate and is the identity the CA is agreeing to certify. The subject will be a Distinguished Name (DN) and this name is guaranteed to be unique. The format of the name is required to be compliant with RFC 5280:4.1.2.6 and looks similar to the following example:

   \[ E=some.one@sap.com,CN=i00000,OU=SAP Trust Centre,O=SAP AG,L=Walldorf,ST=BW,C=DE \]

Here we have attributes for the Entity (E), the email name (CN) the organizational unit (OU), the
organization (O), the location (L), the state (ST) and the country (C).

3. The public key along with the name of the owner who is requesting the signing is sent as a signing request to the CAs in the form of the PKCS#10 Certificate Signing Request (CSR). This signing certificate authority or Root CA needs to be trusted by both the client and the server so that the common server can verify the certificate is signed.

4. The CA validates the information received and signs the CSR request as an act of trust on the given subject (typically this is the distinguished name of the CSR). The signed CSR eventually becomes the digital signed certificate that is returned back to the owner.

5. The private key and the signed certificate are installed on the server providing SSL services to connecting applications or to servers requesting the caller provide proof of their identity. The certificate (which contains the DN and public key, signed by the CA) can be made public to whoever requests it. Once the owner receives and installs that signed digital certificate, the owner will have a fully functional SSL certificate ready to be deployed on the HTTP server.

6. It is possible for the CA to delegate authority for issuing certificates to other servers and organizations.

This creates a chain of trust so that when a certificate is presented all the intermediate certificates up to the CA need to be present on the client (and the server if the server is performing mutual or client authentication) so that the chain to the issuing certificate authority can be verified. There will always be an unbroken chain to the CA from the issued certificate.

7. Verifying the certificate chain to ensure the client or server certificate is valid is sufficient to ensure the certificate has not been tampered with, however it does not establish if the server in fact the owner of the certificate. Later on the SSL certificate and private key will be used to validate the
server owns the certificate.

8. During the SSL handshake at the beginning of the connection establishment, the SSL server will send out its SSL digital certificate to the client that essentially contains the public key and the digital certificate signed by the CA. If client trusts the CA (i.e. it can verify the digital signature inside) and all the intermediate certificates are not expired, not tampered with and are present on the client certificate store then the client will ask the server to encrypt the subject of the certificate. When the response is received from the server, the client will attempt to decrypt the response using the server public key. If the data can be decrypted, then it will become the proof the SSL certificate is a valid and owned by the server.

This process of verifying the certificate is known as the server authentication. Note that same process can be also applied in reverse if required which is known as client authentication or if both the server and the client request a certificate exchange then this is termed mutual authentication. If the certificate exchange process seems familiar, it is the basis for Single Sign-on!
Initial requirements

To make this paper easier to understand and cut down on duplications, some pre configuration steps will be needed in order to make the process easier to implement.

Save the allocated Tenant details
After registering for SAP HCI, you will receive an email with the details of the HCI tenant that has been allocated for use. These details must be saved for future reference, as this will contain the tenant server address and the login details.

Ensure a Certificate Signing process is in place
Part of using SSL is to ensure that the server being communicated with is the server it purports to be. As was mentioned this requires a valid client certificate for the customer installation and/or the HCI Tenant depending on the direction(s) the messages will be flowing. This process will require signing by a certificate authority, ideally one of the certificate authorities trusted by the SAP Load Balancer. Valid Certificate authorities may be found on the SAP Website. To get the key pair signed will require accessing the certificate authority web site, uploading a signing request and downloading the resultant certificate. This process is normally highly restricted and will require the nominated person in the company to perform and/or a ticket be raised to the IT department to get the request authorized. If there is no Certificate Signing process in place then the process will take substantially longer as the certificate authority will be required to verify the company identity and nominate a contact person who is authorized to sign certificates for the company. If the SAP Load Balancer does not already recognize the Certificate Authority then additional time will be required to get the root certificate approved and installed on the SAP Load Balancer.

Some people may find the command line tools supplied by SAP to be preferable to the use of SAP GUI. Fortunately this is well documented already by SAP and there is extensive documentation on the SAP Developer Network on how to do this already. For example see http://scn.sap.com/docs/DOC-46819

Install SAP GUI
SAP GUI is a program that is used to access and manage SAP Servers. This is downloadable from the SAP Store. SAP GUI will be required to deploy and manage certificates in the customer installation.

Remember:
If your server is not listed in the default server list, and SAP GUI is running on OSX, use the Expert mode Expert settings to construct a connection string to connect to the server.

Install Eclipse & the HCI Tooling
Eclipse Luna SR2 for Java EE Developers (or later) is required and may be downloaded from the eclipse website.

Once downloaded it must be installed and the HCI Tooling downloaded from the HCP website.
From the help menu, use the “Install New Software” command to install the HCI tooling. This tooling is required to connect to the HCI tenant in cases where artifacts need to be deployed. The tooling may be downloaded from https://tools.hana.ondemand.com/hci by following the instructions on the site.

After the HCI Tooling is installed and Eclipse restarted, Eclipse must be configured to connect to the HCI tenant in order to allow artifacts to be deployed onto HCI Tenant. Opening the Eclipse Preferences, expanding the SAP HANA Cloud Integration node and choosing the “Operations Server” node perform this.

![Eclipse Preferences panel](image)

This panel will allow the tenant address to be entered as received in the on-boarding email together with the SCN user id assigned to the tenant account.

**Remember:**
- The Tenant address needs to be prefixed with https:// if it is not already prefixed
- Test the connection with the “Test Connection” button.
- The “Test Connection” button will only be enabled when a valid URL is entered.

**Install Keystore Explorer**

In order to deploy certificates to the HCI tenant and enable certificate generation for the Tenant it will be easier to do this through the Keystore Explorer application and the STRUST transaction than through the command line tools.

Keystore Explorer is an open source application for Windows, OSX and Linux that manages keystores with a GUI.

The application can be downloaded from the Keystore-Explorer website (http://keystore-explorer.sourceforge.net/)

**Remember:**
- Keystore Explorer requires the Java runtime to be installed
- Also required is the Unlimited Strength Jurisdiction Policy installed.
- On OSX, installing the policy files is not very well described. Open the file “/Library/Internet Plug-Ins/JavaAppletPlugin.plugin” by choosing “Show Package Contents” from the menu option.
Navigate to the “Contents/Home/lib/security folder and replace the files there with the files downloaded from the Oracle/Java website.
Certificate Storage in SAP Installations

Throughout the authentication processes in HCI there is a significant requirement to use certificates. The processes for creating and installing certificates need to be well understood for this to be accomplished.

There are two primary ways that a certificate can be generated. This may be done from SAP GUI or via the Keystore Explorer app. The certificates can also be generated and managed with the command line tools, however SAP document 46819 covers the command line tools that can be used to accomplish the same result but is not covered in the is paper due to the excellent documentation.

On the customer installation, SAP GUI can be used to create and manage certificates through the STRUST or STRUSTSSO2 transactions. However, when working with the HCI tenant, SAP GUI is not applicable, so in that case the Keystore Explorer application can be used to create the keystores and certificates and Eclipse with the HCI Tooling can be used to deploy the keystores to the HCI tenant.

Personal Storage Environments

Within the SAP installation, credentials and particularly certificates are managed using encrypted containers called Personal Storage Environments or PSE’s. In essence a PSE is a container for the root certificate and any intermediate certificates. The PSEs are managed and deployed using the STRUST transaction. A PSE contains both a key pair and optionally a certificate and certificate chain. SAP GUI is used to manage the PSE by generating Certificate Signing Requestes, creating a key pair and importing certificates.

The complexity of PSE’s is that there are a number of standard ones and knowledge of these is required in order to be able to utilize them with SSL. The standard PSE’s provide support for specific use cases around SSL deployment both for incoming and outgoing connections.

With this paper, creating a PSE will be used as this provides the best opportunity to understand the certificate creation, management and installation processes.

The Server PSE

Typically, there will be a single server PSE that will be used to provide certificates to incoming connections. This enables the server to encrypt traffic between itself and the caller using the PSE’s “own” certificate for which it owns both the certificate and the private key for the certificate. This PSE will also include the root certificate and any intermediate certificates of the “own” certificate in the PSE since this is needed to provide a chain of trust. When an incoming connection is made, the server responds with the certificate in the PSE to the caller. Whilst it is possible to create multiple PSE’s this would require transaction SMICM – “ICM Monitor” to set each PSE within the SSL configuration section and is outside of the scope of this paper since using alternate names in the signed certificate will provide the same functionality.

The Anonymous Client PSE

This PSE is for outgoing connections that do not perform mutual authentication; i.e. the connection will not be asked to provide a certificate. This PSE needs to have the root certificates and intermediate certificates of any servers the customer landscape will connect to in order to validate the certificates the remote system will supply. The private key and own certificate of this PSE will not be used and should not be used.
SSL client SSL Client (Standard)/SSL Client PSE
For outgoing connections that require mutual authentication this PSE contains a signed certificate and private key as well as any intermediate certificate chains. STRUST will be used to create a signed certificate in the PSE then when an outgoing connection is made that is using mutual authentication this certificate will be presented to the remote server to provide proof of identity. The remote server will need to have a common root certificate and intermediate certificates in order to verify the presented client certificate has not been tampered with.

These PSE’s are critical to understanding the role of PSE’s with the SAP Installation. The next step is to deploy signed certificates to the store using SAP GUI.

Creating certificates in SAP GUI

SAP servers will normally connect to other HTTPS servers to deliver data and also receive connections from other servers to integrate data. These will require certificates in most cases since the data being transmitted over the Internet needs to be encrypted and the connection validated. This section will cover creating a PSE for use by incoming SSL connections and also a PSE for mutual authentication of outgoing connections.

Start SAP GUI and execute transaction STRUST to open the trust manager.

As can be seen there are already a number of Personal Storage Environments (PSE’s) created, including the Server PSE, the anonymous SSL PSE and the client SSL PSE.

Each PSE will be utilized to show its function and how the PSE can be used to enable communication to and from the HCI tenant.

Server PSE Configuration
If the inbound connection will require validation, then this PSE needs to be configured to supply a valid certificate when requested. The server PSE is used to manage a certificate for the server for the case where inbound connections need to be able to trust the server. This will require that the “SSL Server Standard”
PSE have a certificate that is signed by a CA that is in common with the SAP Load Balancer list of approved CA's. This list can be obtained from the SAP website here

When creating the certificate, the Common Name (CN) portion of the owner DN must be the fully qualified name of the server, or a wildcard if there are multiple servers in the domain that will be sharing the certificate. For example the CN should be ourserver.domain.sap.corp or *.domain.sap.corp if the entire domain is to share the same private key and certificate. In either case a certificate request will need to be generated and sent to a Certificate Authority for signing. The resultant signed certificate will then be added to the PSE and will provide identity for the server.

Adding a certificate

Execute transaction STRUST and open the “SSL server Standard” PSE. Verify the “own” name CN is the domain name of the server, if it is not then the PSE will need to be changed or replaced and a new certificate generated. Examining the PSE owner certificate, it should be noted that the certificate is marked as “Self Signed”. This means that a Certificate Signing Request will need to be generated and a new certificate created to be used in the PSE. When a correctly signed certificate is in place the self signed message will disappear. The warning is to ensure that the server will at best show with a red warning when any browser calls the server and at worst all connections will be refused as the connections are not trusted.

From the Edit menu, choose the “Create Certificate Request” command or click on the button under the “Owner” label. This will generate a certificate-signing request that must be sent to the CA to be signed. Copy the CSR request text and submit it to the corporate signing process.

At this point the root certificate of the CA and any intermediate certificates will have to be imported into the PSE so that a chain of trust can be established. This is done through the use of the “Import” command on the “Certificate menu” or via the command in the Certificate panel.

Remember:
After importing a certificate click the “Add to Certificate List” button in the certificate section. Failure to do so will NOT add the certificate to the PSE.

Use the save command to save changes to the PSE after all certificates are imported.

When the signed certificate response is received back from the CA, this will need to be imported into the PSE. This is done through the use of the “Import Certificate Response” command on the “Edit” menu or via the command in the “Own Certificate” panel.

The “Self Signed” message under the owner certificate will now have disappeared. Save the PSE now.

This completes the configuration of the server SSL certificate. External applications will now be able to securely connect to the server and the server will respond with a valid certificate (assuming the client contains the root and intermediate certificates of the CA who signed the server certificate).

Anonymous Client PSE Configuration

If the outbound session will be connecting to a remote server and the remote server will not require the client prove its identity via a certificate, then the anonymous client PSE can be used. In this case it is sufficient to import the root certificate from the CA and intermediate CA’s that the server will be connecting to. This will provide validation that the server being connected to is whom it is purporting to be but the remote server will be unable to prove who is connecting.

This PSE is used when configuring the outgoing RFC destination. The RFC destination can be configured in transaction SM59. In transaction SM59, the PSE is set in the SSL configuration section when the SSL is activated.
Client PSE Configuration

If the outbound session will be connecting to a remote server and the remote server will request the client prove who it is, the client will be required to provide a valid certificate to the server. The Client PSE contains the certificate that will be used. Each outbound connection configured via SM59 is potentially connecting to a different server so each outbound connection could require a different certificate. The means that there will likely be multiple client PSE’s in the STRUST transaction. Each PSE is used in conjunction with the RFC outbound configuration settings where the outbound connection can specify an individual PSE if required.

If not already done so, open SAP GUI and execute transaction STRUST. Whilst the SSL Client SSL client (Standard) could be used, the preferred solution is to create a custom PSE and use that in the RFC destinations so it is obvious that a specific certificate is being used to connect to a specific destination.

Adding a certificate

From the Environment menu, select “SSL Client Identities”.

Use the “New Entries” button to create a new client PSE

Save the new PSE and note that there is now a new PSE in the list on the left but it is empty.
A Distinguished Name (DN) for the PSE now needs to be defined by right clicking over the PSE node and using the “Create” command. This will prompt for the DN for the PSE.

Remember:
- The Name field must correspond to the fully qualified domain name of the host server.
- When using the auto-generated name ensure it is the correct fully qualified domain name of the server.

At the moment there are no SSL certificates assigned to the PSE because the certificate list is empty. An existing PSE such as the “SSL client SSL Client (Standard)” that will already have a generated DN assigned to it so could be used as well. In this case, the certificate would need to be recreated taking care to ensure the CN uses the fully qualified domain name of the host server as would be done for a custom client PSE.

To create a signed certificate, as was done with the server certificate, a certificate-signing request will need to be made and the resulting signed certificate imported into the own certificate store of the PSE.

Open the newly created SSL client PSE.
From the Edit menu, choose the “Create Certificate Request” command or click on the button under the “Owner” label. This will generate a certificate request that must be sent to the CA to be signed. Copy the CSR request text and submit it to the corporate signing process.

At this point the root certificate of the CA and any intermediate certificates being used to sign the CR will have to be imported into the PSE so that a chain of trust can be established. This is done through the use of the “Import” command on the “Certificate menu” or via the command in the Certificate panel.

Remember:
- After importing the certificate, use the “Add to Certificate List” button to save the imported certificate. The certificate will then appear in the certificate list.
- Double click on the DN’s of the owner certificate or any certificate in the certificate list to view the certificate details in the Certificate panel.
- Use the save command to save changes to the PSE.

With the certificate chain in place all the way up to the root certifying authority, the certificate from the CA can be imported via the “Import Certificate Response” command on the “Edit” menu. This command can also be executed using the button in the “Own Certificate” panel. Once the certificate is imported the (Self Signed) text in the owner panel will disappear. Double clicking on the owner DN in the “Owner” panel will show the certificate received from the CA in the “Certificate” Panel.

Save the changes.

The result will be that the new client PSE will now contain a certificate that an outbound connection can use to provide identification.

**Restart ICF**

In order to get the Internet Connection Framework (ICF) to recognize the new PSE’s, the ICF will need to be soft restarted. Execute transaction SMICM, then from the Administration menu, ICM submenu, choose the “Exit Soft” submenu and choose “Global” to restart the ICF.

The certificates for SSL for inbound and outbound connections have now been setup using SAP GUI. Next the PSE’s will have to be attached to outbound connections in order that the outbound connection uses the correct certificate.
Using certificates in SAP GUI

To enable outbound SSL connections, the RFC(s) need to be configured. The configuration is per RFC and is setup using transaction SM59 – “Configuration of RFC Connections”.

Open SAP GUI and execute transaction SM59 to get the screen below:

As can be seen the list of RFC’s is shown together with the destination types. Expand the “HTTP Connections to External Server” node and choose the relevant RFC to edit.

Select the “Logon and Security” tab and use the “Security Options” panel to configure SSL. To enable SSL, select the “Status of Secure Protocol” panel and set the SSL radio button to “Active”. This will then enable the selection of a PSE in the outbound connection.

Remember:
- If the SSL setting is not set to active, then the connection is an unencrypted connection with all data plainly visible on the Internet.
- Basic Authentication over a non-SSL connection should never be used either since this would put the username and password in plain text and visible to anyone evedropping on the connection.
If the RFC destination is not using mutual authentication then the SSL Client (Anonymous) PSE must be used, since this is the only store that will not cause the external server to request a certificate from the customer installation. Otherwise the caller and the server will both be required to present certificates to prove they are who they say they are. In this case, the list of PSE’s able to authenticate the connection will be listed. These PSE’s are the same ones that were defined in the STRUST “Environment” menu with the “SSL Client Identities” command.

Setting the SSL certificate to the new PSE store will enable mutual authentication for outbound connections on this RFC.

Save the changes to the RFC destination and the RFC is now enabled for outbound SSL communication.

**Remember:**
- Use the Anonymous PSE to connect to servers requiring encryption but not proof of identity.
- Use the standard or custom client PSE when outbound connections require encryption and proof of identity.
- A certificate authority for which the server also has a root and intermediate certificates must sign the client PSE’s.
- The Anonymous PSE must contain the root certificates for the server the outbound connection is connecting to.

**Creating Certificates with Keystore Explorer**
To deploy a certificate to the HCI tenant requires creating a signed certificate and storing this in a JAVA keystore. This keystore will be deployed to the HCI tenant to allow outbound and inbound SSL connections.

To begin open Keystore Explorer and create a new Keystore by clicking on the “Create a new Keystore” button, or choosing “New” from the file menu.

The type of keystore will now need to be chosen.

HCI only supports the JCEKS keystore that is the newer and more secure version of the JKS keystore. So select the JCEKS option and click “OK”.

After the keystore is created, this is a good point now to create a password. From the Tools menu, select the “Set Password” option and enter a new password for the keystore.

Next, save the keystore file. Though there is nothing in the file, it is best to do it here. If you forget the passphrase or password that was used the keystore will not be able to be opened.

The next steps involve creating a signed certificate by creating a key pair then creating a Certificate Signing Request (CSR) to send to the certificate authority for signing.
From the Tools menu, execute the “Generate Key Pair” command.

Leave the defaults as a 2048 bit RSA key pair. Click OK and create the certificate that will be used with this key pair to be submitted to the Certificate Authority to sign.

The certificate should remain a SHA-256 RSA V3 formatted certificate with a randomly generated serial number.

When entering in the Name, this should consist of the following fields:

<table>
<thead>
<tr>
<th></th>
<th>Common name</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CN</td>
<td>Common name</td>
<td>The name of the HTI tenant server without the HTTPS portion. This will be the DNS name of the server.</td>
<td>MyHCITenant.neo.ondemand.com</td>
</tr>
<tr>
<td>OU</td>
<td>Organizational Unit</td>
<td>The name of your team or the team issuing the certificates</td>
<td>SAP Trust Center</td>
</tr>
<tr>
<td>O</td>
<td>Organization</td>
<td>The company name</td>
<td>SAP AG</td>
</tr>
<tr>
<td>L</td>
<td>Locality</td>
<td>The town where the company is located</td>
<td>Walldorf</td>
</tr>
<tr>
<td>ST</td>
<td>State</td>
<td>The province where the company is located</td>
<td>BW</td>
</tr>
<tr>
<td>C</td>
<td>Country</td>
<td>The country the company is located</td>
<td>DE</td>
</tr>
</tbody>
</table>

These 6 components form a unique address of which the 5 should already have been established when from the Certificate Authority.

Click “OK” and create a certificate alias. An alias is required to identify the certificate in the iFlow when using certificate based authentication.
Click “OK” again and enter the password to protect the certificate. This certificate needs to use the same password as the password protecting the store for HCI to be able to install the certificate on the tenant.

Click “OK” the final time create the certificate and note that it is now added to the certificate list in the key store.

Save the keystore now.

Next the certificate will need to be signed by a certificate authority. Right click over the key pair certificate that was just generated and choose the “Generate CSR” command.

Leave the defaults as a PKCS#10 SHA-256 certificate and enter in a filename or use the default one.
The generated signing request file may now be sent to the Certificate Authority to be signed.

At this point any root and intermediate certificates that need to be added to the store should be added now. These are the certificates of the CA the signing request was sent to and any intermediate certificates if the Root CA is not directly signing the certificate.

To import root and intermediate certificates, from the tools menu, use the “Import Trusted Certificate” command.

Remember to start with the root certificate for the CA, which will prompt you to trust that certificate. Since this is a root certificate it will not have any paths of trust so it is correct to trust the root. The certificate should be obtained from a trusted source to make sure this is the root certificate of the CA.

Click OK.
Click OK again and finally confirm the certificate is to be treated as trusted and after setting an alias the certificate will be added to the store.

Any intermediate certificates imported after the root certificate will then be trusted automatically since there is a path of trust now established to the primary root certificate.

Once the signing request response is received from the Certificate Authority, the “Import CA Reply” command may be used to import the response.
The password for the certificate will need to be entered and the certificate imported. Save the certificate store and this concludes adding a signed certificate to a certificate store. The certificate store is now in a state where it could be uploaded to the HCI tenant to provide SSL services when connecting to the customer landscape.

**Deploying Certificates to the HCI Tenant**
At the time of writing the keystore cannot be deployed via the HCI Cockpit in the browser. To deploy a keystore to the HCI tenant requires the use of Eclipse with the HCI Tooling.

Open Eclipse and connect to the HCI Tenant.

Right click over the tenant node name and choose the “Deploy Artifacts” command

Select the Keystore artifact and click “Next”.


Select the keystore file created earlier and enter the password in the passphrase box.

Click finish to deploy the keystore to the tenant.

Once the keystore is deployed, verify that it deployed correctly by viewing the Deployed Artifacts tab on the "Node Explorer" node root.

This concludes deploying a keystore to the HCI tenant.

**Authentication options in HCI**

This paper is focused on the scenario where the customer will configure their outbound connections to communicate with the HCI tenant and vice versa. A customer has two ways to connect their installation to their HCI Tenant. The first method is to use Basic Authentication and the second is to use certificates. These two methods will be discussed together with a real world scenario.

SAP also offers a service where the setup for the customer is performed by SAP and the customer provides minimal input. Generally speaking the latter scenario is of less value for these integration scenarios than customer configuration of HCI security. Where appropriate the alternative SAP option will be discussed to contrast the level of work required.
Basic Authentication

This means of authentication is accomplished by first setting up a secured tunnel between the customer installation and the SAP Load Balancer to ensure the data is encrypted when communicating over the Internet. Once a secure tunnel is in place then the user name and password can then be sent over the tunnel where it is authenticated against the SCN user identity owned by the HCI tenant. In the reverse case, the iFlow is configured to create a connection to the customer landscape with the supplied credentials artifact. The outbound connection is over SSL with the connection to the customer installation validated with a certificate but not used for authorization.

Basic authentication as its name implies is very simple. The username and password are obfuscated using a very simple technique, defined in Internet standard RFC 2617. This is itself not very desirable since the username and password can be trivially extracted from the obfuscated password. Whilst Basic authentication is supported and is easy to setup and configure in the landscapes, it should however only be viewed as a starting point before moving onto configuring the landscapes to communicate via mutual authentication using certificates.

HCI Inbound

Below is the system overview of requirements and data flows when transferring messages from the customer installation to the HCI tenant. This type of scenario is where the customer installation is creating purchase orders or RFI’s and needs to send them to their HCI tenant.

In this landscape the flow of authentication information originates on the customer installation and sends messages to the HCI Tenant.
The server in the customer installation connects to the SAP Load Balancer via SSL and verifies the Load Balancer certificate. With this in place the credentials are transferred to the Load Balancer that will then pass these credentials to the HCI landscape where they will be validated against the SDN database, the user roles assigned and the appropriate tenant, iFlow and package chosen.

The HCI landscape verifies the user identity via a lookup into the SCN accounts database. The iFlow just uses the fact the user is validated using basic authentication, the password and username is not visible to the iFlow.

Requirements
The following is required to enable Basic authentication when sending messages from the customer installation to the HCI Tenant

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communications user</td>
<td>This user will need to be created in SCN and will be used as the user to be authenticated. SAP will assign this user roles for HCI and validate the authentication request by validating the identity against the SCN user database</td>
</tr>
<tr>
<td>SAP Load Balancer root certificate</td>
<td>One of the root certificates in the HCI Load balancer will be required in order for the outbound connection to the load balancer to be authenticated.</td>
</tr>
</tbody>
</table>
This certificate will need to have all intermediate certificates installed in addition to the root certificate so that a full chain of trust can be verified. It is recommended the customer and SAP share a common Certificate Authority before commencing the integration work.

A list of the supported root certificates can be obtained from this [link](#).

**Process**

As documented above, the customer will add the SAP Load balancer root certificate and intermediate certificates to the anonymous PSE. These certificates can be obtained from the [SAP website](#).

To validate the certificate supplied by the SAP Load Balancer the anonymous PSE must contain the CA root certificate and any intermediate certificates. The SAP Load Balancer will be supply a certificate to the customer installation server and the customer installation server will verify the certificate to make sure the certificate supplied by the load balancer during the connection is valid.

The customer installation needs to be configured to use Basic Authentication when connecting to the SAP Landscape. This is done via configuring an RFC Destination using transaction SM59.

Each HTTP destination that will connect to the HCI Tenant will need to be configured to use Basic Authentication. Open the relevant RFC destination in the “HTTP Connection to External Server” and enable editing mode.
Choose the “Logon and Security” tab and use the “Logon with User” panel choosing the “Basic Authentication” option. Enter the communication username and password (obtained earlier). If not done so already enable SSL to use the Anonymous Client SSL PSE.

Save the changes and exit the transaction.

**Remember:**
- Check the SSL panel is correctly configured.
- Check the correct PSE is being used if SSL is being used.

The final step is to open the HCI cockpit, select the relevant package and iFlow and edit the sender element to enable Basic Authentication for the inbound connections.
Save and deploy the iFlow to enable the processing of messages from the customer installation to the HCI Tenant.

This completes the procedure to enable Basic Authentication for message transfer from a customer installation to the HCI Tenant.

**HCI Outbound**

This scenario handles the case where the HCI Tenant is generating messages that will then need to be delivered to the customer installation for processing such as invoice processing or purchasing integration onto an on-premise SAP ERP installation.

In this scenario the flow of authentication information originates on the HCI Tenant and messages are sent to the customer installation.
The HCI tenant will process incoming messages with an iFlow from a package. This iFlow will terminate with a receiver connecting to the customer installation. The delivery of the messages will be using an SSL connection with the customer installation root certificate being used to verify the connection to the customer landscape and the user name and password supplied by the iFlow as Basic Authentication.

Requirements

The following is required to enable Basic authentication when sending messages from the HCI Tenant to the Customer Installation

<table>
<thead>
<tr>
<th>Eclipse with HCI Tooling</th>
<th>Eclipse together with the HCI tools is required and Eclipse must be connected to the tenant using the HCI tooling.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Username and password</td>
<td>This user will be a customer installation user that will be processing the messages from the HCI tenant.</td>
</tr>
<tr>
<td>Server root certificate</td>
<td>The root certificate for the customer installation.</td>
</tr>
</tbody>
</table>

Process

Start eclipse and open the Integration Options perspective. Since the connection to the HCI tenant is already configured in Eclipse, there will be a node for the server already present. Click over the root and choose “Deploy Artifacts” then choose “User Credentials” to get the dialog below.
Select the “User Credentials” option and fill in the relevant fields, taking care to note the name field, as this will be required later on in the iFlow configuration step.

Click “Finish” to save and deploy the credential artifact.

Double clicking on the tenant node and clicking on the “Deployed Artifacts” tab can verify the deployment of the artifact.
The next step is to establish a secure outbound channel between the HCI Tenant and the customer installation. Obtaining the customer root certificate for the customer installation server does this. This is the root certificate from the CA that signed the customer server client certificate being used for incoming SSL connections. The HCI tenant will validate the connection to the customer installation with the customer client certificate.

The CA root certificate and any other intermediate certificates will need placing in a keystore. This key store can then be deployed as an artifact into the HCI landscape for the tenant. The creation and addition of certificates into a keystore is covered in an earlier section.

As above, right click over the tenant node and choose “Deploy Artifact” from the menu. Add the keystore file containing the certificates and the password for the keystore.

Click “Finish” to deploy the artifact to the HCI tenant.
Finally the iFlow needs to be configured so that the customer installation server will receive Basic Authentication requests from the HCI Tenant. To do this open the relevant package and iFlow in the HCI cockpit and select the receive channel terminating in the on premise landscape (labelled as ERP in the diagram below).

Edit the channel and enable the “Basic Authentication” check box then in the “Credential Name” use the name of the credential artifact deployed when deploying the username and password artifact.

This concludes setting up the HCI tenant to use Basic Authentication to communicate with a customer installation.

In the scenario where the customer has elected to have SAP manage the HCI Tenant, the customer will provide to SAP the above user credentials and customer installation root certificate(s). SAP will install the certificate and user credentials on the HCI Tenant and ensure the iFlow is correctly configured in order to communicate with the customer installation using Basic Authentication.

Certificate Authentication Overview
A stronger method of authenticating users is to use X.509 certificates. These certificates allow both parties in the conversation to verify their identities and have their identities bound to roles within the two landscapes.

This is the SAP preferred means to communicate between the customer installation and the HCI tenant due to its high security and inbuilt mechanisms to prevent many kinds of security attacks, though this comes at the cost of requiring a more complex skill set to implement.

HCI Inbound
Below is the system overview of requirements and data flows when transferring messages from the customer landscape to the HCI landscape using certificates. This type of scenario is the same as the one where Basic Authentication is used in the previous steps but removing the credentials and replacing them with a signed certificate.
In this landscape the flow of authentication information originates on the customer installation and messages are sent to the HCI Tenant.

The server in the customer installation connects to the SAP Load Balancer via SSL and verifies the Load Balancer certificate. The SAP Load Balancer verifies the certificate being presented by the customer landscape is valid. The certificate is considered a valid identity and thus the Distinguished Name (DN) of the certificate is the validated user identity. The certificate is then passed from the Load Balancer to the HCI Tenant to authorize the user by confirming the DN of the certificate in the iFlow receiver component matches the certificate DN. If they do not, then the iFlow will be aborted.

**Requirements**
The following is required to enable Certificate Authentication when receiving inbound messages from the Customer Installation to the HCI landscape.

| Customer client certificate | This certificate will be required to communicate with the SAP Load balancer and the backend HCI tenant. The private key used to sign the certificate is required in order to encrypt data when communicating with the Load Balancer |
| Load balancer certificate (and certificate chain) | The load balancer root certificate |
**Process**
The customer installation will be required to generate a certificate that has as a root a CA that is trusted by the SAP Load Balancer. This certificate will be added to a PSE and the PSE linked to an outbound RFC destination as explained in earlier steps.

The iFlow needs to be configured to ensure the user executing the iFlow is the user authorized to perform the flow. For this the distinguished name of the certificate is used in the iFlow sender component to form authorization checks. Either the certificate DN and issuer DN can be entered or the certificate imported into the iFlow Sender component.

Open the HCI cockpit and the relevant package and iFlow. Edit the iFlow, selecting the sender node.

![Image of HCI cockpit and iFlow configuration](image)

Whereas in the previous configurations, the “Authentication Type” was set to Basic Authentication, this time the “Authentication Type” must be set to “Certificate Based Authentication”. The “Assigned Certificates” list will now be available where the “Subject DN” and “Issuer DN” can be assigned or the certificate loaded manually.

After entering the requisite certificate information, the sender will need to be saved and the iFlow redeployed to the HCI tenant to enable certificate authentication.

This completes the procedure to enable certificate authentication for message transfer from a customer installation to the HCI Tenant.

**Outbound**

Below is the system overview of requirements and data flows when transferring messages from the HCI Tenant to the customer landscape. This type of scenario is the same as the one where Basic Authentication is used in the previous steps but removing the credentials and replacing them with a signed certificate.

In this landscape the flow of authentication information originates on the HCI Tenant and messages are sent to the customer installation.
The HCI Tenant connects to the Customer installation and verifies the certificate presented by the customer landscape is valid. The customer landscape requests the tenant provide a certificate to validate the identity of the customer tenant. The tenant certificate is then mapped to a user identity in the customer SAP system and the processing of the incoming data is performed under the identity of the user mapped to the certificate.

**Requirements**

The following is required to enable Certificate authentication when sending messages from the HCI Tenant to the Customer Installation

<table>
<thead>
<tr>
<th>Customer root certificate</th>
<th>This certificate will be required to communicate with the customer installation server.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tenant certificate and corresponding key pair.</td>
<td>The certificate to be used on the tenant. This will require the key pair used to generate the certificate as well as all the intermediate certificates in the certificate chain. Remember that the certificate needs to be trusted by a certificate authority shared between HCI tenant and the customer installation server.</td>
</tr>
<tr>
<td>Eclipse With HCI Tools</td>
<td>An Eclipse installation with the HCI tooling installed and connected to the HCI tenant.</td>
</tr>
</tbody>
</table>

**Process**

This process is the most complex to configure, both for SAP managed and for customer-managed installations since this will require configuration on both the HCI Tenant and the customer’s SAP installation.

Whilst less work is required for the SAP managed process, the majority of the work is still in the configuration of the inbound connections to the customer installation.
SAP Managed process
A tenant root certificate and all intermediate certificates will be provided to the customer by SAP. The customer will provide SAP with the root certificate and intermediate certificates for their installation (from the SSL Server PSE maintained in STRUST) as well as the WDSL endpoint so the messages being processed in HCI can be delivered to the customer installation.

The customer installation now needs to be configured to recognize the supplied Tenant certificate from the HCI tenant and map the incoming user in the certificate to a technical user.

Customer Managed Process
Where the customer elects to manage his or her own tenant the process will require more effort since this requires the customer generate a certificate and deploy it to the HCI tenant.

The customer starts by creating a keystore, generating a key pair and creating a certificate-signing request to have that request signed by certificate authority. The resulting signed certificate is then placed into the keystore along with the customer installation root certificate and intermediate certificates as documented previously.

The key store will now be deployed into the tenant using Eclipse and the Deploy Artifact command.

Start eclipse and open the Integration Options perspective. Since the connection to the HCI tenant is already configured in Eclipse, there will be a node for the server already present. Click over the root and choose “Deploy Artifacts” then choose “Keystore” to get the dialog below.

Select in the file containing the keystore and enter the password used to encrypt the keystore then click finish to deploy the keystore.

Double clicking on the tenant node and clicking on the “Deployed Artifacts” tab can verify the deployment of the artifact.
Finally, log into the HCI tenant and open the package, choosing the appropriate iFlow. Edit the iFlow and select the outgoing SOAP connector.

Uncheck “Basic Authentication” and “Connect without Client Authentication” and enter the alias of the signed certificate in the keystore. This is to enable HCI to choose the correct certificate in the keystore to authenticate with when connecting to the customer installation where there are multiple certificates.

Save and deploy the iFlow.

With the certificates deployed for use, the HCI Tenant can prove its identity and the customer installation can verify the HCI tenant.

The customer installation now needs to be configured to recognize the certificate from the HCI tenant and map the incoming user to a technical user.

Completing the configuration on the customer installation

In either the SAP managed process or the customer managed process, the result is that the customer will still be required to do configuration on their installation to support certificate authentication.
The first is to add the HCI tenant root certificates to the SSL Server PSE (if not already present) so that the incoming certificate can be validated. This is documented in earlier steps and updates the server PSE certificate list with the new root certificate.

All processing in SAP installations are executed under an identity so the roles and authorizations can be managed. This is performed by mapping the DN of the HCI tenant certificate to a technical user in the SAP landscape.

The certificate DN will now be mapped to a SAP user. Open SAP GUI and execute transaction SM30 to maintain the user to external id mapping. The customer can then map the certificate to the technical user that will be processing inbound messages. This will involve assigning the DN of the certificate to the SAP user in the User External ID view (VUSEREXTID).

Enter the table name of VUSEREXTID and click “Maintain.”

From the list of external ID types, choose an id type of (X) and open the mapping list table.
Click “New Entries” to add a new mapping.

In the external ID field add the DN name of the certificate from the HCI Landscape and in the user field choose a user in the SAP System with an appropriate role and authorization level.
Save the changes and exit. The inbound connection is now mapped to an identity.

This concludes the SAP managed installation configuration. As can be seen there is little for the customer to do to get HCI working within their installation.

Example HCI scenarios

In this scenario a fictitious customer Fundi-mental Industries (FI) has a SAP ERP installation and will send Sales Orders to their HCI Tenant for processing and delivery through HCI. They also elect to do the management of the processes by themselves rather than have SAP perform it.

The initial rollout will use basic authentication until the system is production ready, then the basic authentication will be upgraded to the more secure certificate authentication.

To start the project, FI have a SAP PI expert create an iFlow to take supplier messages and process the message into a new message for delivery to the FI server. This will be validated using the HCI tools to ensure the iFlow works with sample data.

HCI Inbound

On receipt of their tenant details, the customer begins by creating a new SCN user on the SAP Community Network portal. This is the communications user. FI informs SAP of the user identity and SAP configures the HCI tenant to authenticate and authorize this user.

FI can now configure their ERP system. Using transaction STRUST a root certificate is added to the SSL Client Anonymous PSE. This will be a root certificate supported by the SAP Load Balancer. These root certificates allow the FI to securely connect to the HCI tenant then pass the credentials over the encrypted connection safely.
Transaction SM59 is then used to first configure the HTTP destination with the communications user credentials and secondly enable SSL with SSL Client Anonymous PSE.

This completes the client configuration.

The HCI tenant iFlow is then configured to enable basic authentication for incoming messages by enabling basic authentication in the sender component of the iFlow.

With the basic authentication working and messages being delivered from the FI to the HCI landscape, there is now a desire to move to a more secure mechanism of message delivery. To do this FI will move to using certificate authentication mechanisms.

This will involve implementing the procedure earlier on where certificate authentication was enabled from the customer installation to the HCI tenant.

FI will create a signed certificate in a PSE using transaction STRUST from a CA that is also shared with the SAP Load Balancer. The destination RFC in transaction SM59 will be updated to use the new PSE. Finally the iFlow will be updated to use the DN of the certificate as the identity allowed to execute the iFlow.

This concludes how an example customer would configure inbound communication to the HCI tenant using either Basic Authentication or Certificate based authentication.

**HCI Outbound**
Later on there becomes a need for FI to not just send messages to the HCI tenant for processing but also receive messages from the HCI Tenant. The PI consultant is then recalled to create a second iFlow that takes messages from a 3rd party supplier processes them in the HCI Tenant and delivers them to the FI installation. Again these will be tested using HCI tools to ensure they match the specification supplied to the PI expert.

The focus is to get the integration working. The easiest way to do this is to use Basic Authentication. Eclipse is started and a connection to the HCI Tenant made. A credentials artifact is named, created and deployed. The credentials are the username and password of a user on the FI SAP system. The FI server will authenticate the user when inbound communication from the HCI Tenant is received. At the same time a keystore with the root certificates of the FI server will also be deployed to the tenant to ensure the tenant can securely connect to the FI landscape. Lastly the iFlow will need to be updated to ensure that the destination uses the name of the credential artefact to ensure the correct username and password are sent.

This completes the installation of Basic Authentication for outbound communication from the HCI Tenant to the FI installation. With the Basic authentication running successfully, the system is upgraded to use Certificate Authentication, as this is more secure.

FI will create a signed certificate, assign it an alias and place it in a keystore together with the intermediate certificates, remembering to keep the password for the keystore and the certificate the same. The keystore will be deployed to the HCI Tenant. The iFlow will be updated to use the certificate alias to connect to the FI server.

In the FI system, using SAP GUI, FI will assign a SAP user and map the user to the DN of the certificate deployed to the HCI tenant, remembering to install all root and intermediate certificates.
That completes the installation of Basic Authentication in an iFlow when transferring messages from the HCI Tenant to the FI landscape and the subsequent upgrading to use Certificate Authentication.

**Conclusion**

This paper has shown how to implement Basic and Certificate authentication for the HCI tenant and the customer installation.

The fundamentals of using certificates and Public Key Infrastructure have been explained and worked through, especially the configuration of the SAP installation and the HCI Tenant.

**Glossary**

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>PKI</td>
<td>Public Key Infrastructure</td>
</tr>
<tr>
<td>CA</td>
<td>Certificate Authority</td>
</tr>
<tr>
<td>CN</td>
<td>Common Name</td>
</tr>
<tr>
<td>DN</td>
<td>Distinguished Name</td>
</tr>
<tr>
<td>CSR</td>
<td>Certificate Signing Request</td>
</tr>
<tr>
<td>PSE</td>
<td>Personal Storage Environment</td>
</tr>
<tr>
<td>HCP</td>
<td>HANA Client Platform</td>
</tr>
<tr>
<td>HCI</td>
<td>HANA Client Infrastructure</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Sockets Layer</td>
</tr>
<tr>
<td>Certificate Chain</td>
<td>The hierarchical trust relationship for root and intermediate certificates where each certificate proves its validity by checking the signature with the parent certificate.</td>
</tr>
<tr>
<td>Root Certificate</td>
<td>The public certificate issued by a CA.</td>
</tr>
<tr>
<td>x.509</td>
<td>The specification for managing certificates and user identity</td>
</tr>
<tr>
<td>Intermediate Signing Authority</td>
<td>Delegated certificates issued by a CA that empower the issuer to issue new certificates on their behalf</td>
</tr>
<tr>
<td>Intermediate Certificate</td>
<td>The certificate trusted by the CA and issued on behalf of the CA.</td>
</tr>
</tbody>
</table>
## Summary

The table below covers the requirements that will enable the appropriate authentication when connection to/from the HCI tenant and the customer installation.

In the table below the term outbound refers to connections originating from the HCI tenant and the term inbound refers to the connections originating on the customer installation and terminating at the HCI Tenant.

<table>
<thead>
<tr>
<th>Basic</th>
<th>Outbound</th>
<th>Inbound</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Customer Root Certificate</td>
<td>Load balancer root certificate</td>
</tr>
<tr>
<td></td>
<td>Identity of Technical User</td>
<td>Tenant SCN Identity</td>
</tr>
<tr>
<td>Certificate</td>
<td>Customer Root Certificate</td>
<td>Customer certificate mapped to a user identity in the customer landscape.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Load Balancer root certificate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Customer certificate from a load balancer root certificate</td>
</tr>
</tbody>
</table>