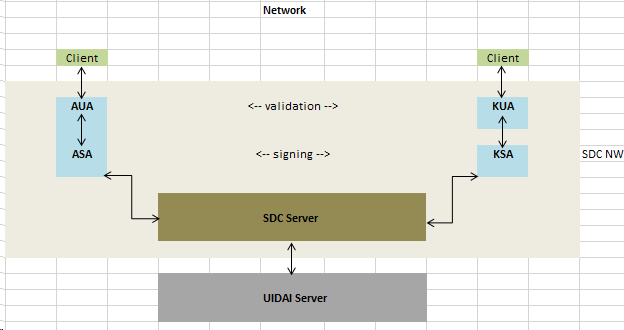
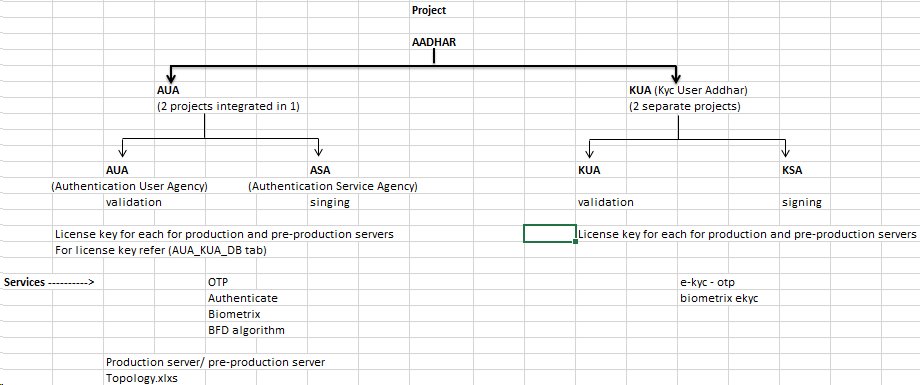
|  |  |  |
| --- | --- | --- |
| **Sr.No.** | **Index** | **Page** |
| 1 | Network | 2 |
| 2 | Project Structure | 2 |
| 3 | API service flow | 3 |
| 4 | Access AUA – KUA servers | 4 |
| 5 | Errors | 4 |
| 6 | License Keys and Deployment | 5 |
| 7 | Start / Stop JBoss server | 7 |
| 8 | MOU process | 10 |
| 9 | Transaction Count Details | 11 |
| 10 | Configuration of HOST file | 12 |
| 11 | Update Application level certificates | 13 |
| 12 | Update Sever level SSL certificates | 15 |

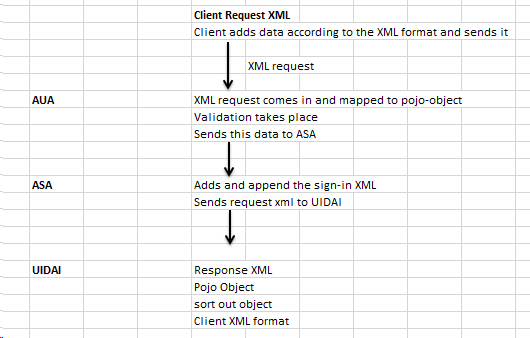
**Network:**



**Project structure:**



**API service flow:**



The above flow is of authentication, Same flow is for kuai.e.eKYC also.

**Access AUA KUA servers:**

For ip address, username & password credentials for each server refer Topology.xlxs file.

* Use wincScp to get UI of server for deployment purpose.
* Use putty to get Server console to start/stop servers.

**Errors:**

There are 2 types of error:

1. **3 – digit**: UIDIA error – refer PDF

For error from UIDAI copy and paste <Rar> tag element from response xml and Base64 decode it to verify the specific error.

1. **4 – digit:** AUA – KUA error – refer PDF
2. **Telnet:**Check logs for any other errors (especially 1009) which occurs when there is no connection between the SDC server and UIDAI server.

Resolution: If connection timed out:

* Open putty and enter credentials like ip - address, username and password.
* Cmd -> “*sudosu*”
* telnet ip**port**

E.g. “*telnet 10.66.32.22* ***443***”

If it says:- “try connection…” 🡪 then there is no connection between the current ip server and the uidai server.

If it says:- “escape characters” 🡪 then there is connection.

* If no connection is there contact SDC to inform telnet unable to perform connection between SDC and UIDAI servers. Provide them server addresses for whichever servers there is no connection.
* **Traceroute**: To check where our server is reaching/hitting before the Aadhar server use below commands (firewall issue).

“traceroute 10.66.32.22”

After the above commands it will show which ip it is able to reach.

1. **HandshakeError:**

Replace the below jars in specified folder as **java/lib/jdk/jre/lib/security**.

* + localpolicy.jar
  + US\_export\_policy.jar

The above jar files are present in D://Desktop folder/JCM

**License Keys and deployment:**

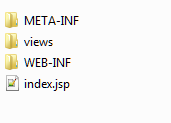
AUA:

* **License Keys:**
* There are 2 license keys
  + Aua license key
  + Asa license key



* License keys are mapped in aadhar.properties file as above which is present in war (resources directory) file.
* In AUA just copy paste the license keys as it is provided.
* There is no need of creating a new war file for updating license keys. There is a shortcut to do this.

1. Copy the deployed working war file from jboss from the server
2. Extract it (right click 7 zip 🡪 extract to /aua)
3. Inside the extracted folder go to path aua/WEB-INF/classes/aadhar.properties
4. Open aadhar.properties and copy paste the provided license keys.
5. Move back to the folder where META-INF, views, WEB-INF etc. are present as shown below and then select all and add archieve it and edit the name before saving.



1. Edit aua.zip to aua.war and save it.This is the new war file created. Deploy the new war in the jboss and restart the server.

* **Deployment with changes:**
* During any changes in the application make sure to do some changes as per local to the production or pre – production server. Edit the dispacther – servlet.xml. Update database, username, and password i.e.mysql database configuration.

KUA:

* Here unlike aua, the properties file is configured outside the project on server. Only the path of this properties file is configured inside the aadhar.properties file (present in war file resources folder).
* There are 2 war files:
  + Kua.war
  + Ksa.war
* Both the war files has its own properties file which is present in server – path/kuaksa\_props/folder
  + Request folder 🡪 adds signature 🡪 picks from this folder and encodes its request
  + Response folder 🡪 decodes
* **License Keys:**
* There also there are two types of license keys

1. KUA license key 🡪 in properties file its configured for **lk=ENC(**encoded license key**)**

****

1. KSA license key 🡪**asalk=ENC(**encoded license key**)**

****

* In KUA/KSA we need to **encode** key and copy paste license keys it.
* **Encoding licensekeys:**
  + Use tool 🡪**jasypt – 1.9.2 – dist**
  + Copy paste the **bin** path of the above tool folder in command prompt.
  + Copy license key and use the below command line to encrypt license keys:
  + Incmd🡪**encrypt.bat input=**licensekey **password=**jasyptsecret
  + You will get the output as encoded license key. Copy and paste this license keys in properties file in ENC(encoded license key).
  + This same license key encoding process is used in the project which you can verify by analysing the code com.uid.ksa.util🡪 AadharConstant.java file.
* **Deployment:**
* If no code change only license keys needs to be updated just update he license keys as describe above.
* With new codes changes just make some changes in dispacther – servlet.xml. for url verify the web.xml

**Start/Stop Jboss server:**

* **Stop the server:**

NOTE: Use shortcut CRTL + R to find the previously performed commands:

1. Find the running pid of the jboss server:

**ps –f|grepjboss**

You will find the numeric root pid. E.g. 19425

1. Kill the pid:

**sudo kill -9 pid**

E.g.sudo kill -9 19425

1. Also kill the pid service if jboss**+exists** during startup. Kill the previous piddisplayedbefore**+exit** message displayed.
2. **Start the server:**

Note: Use shortcut CRTL + R to find the previously performed commands:

1. First navigate to the jboss **bin** folder.
2. To start jboss:

**Sudo nohup ./standalone.sh –b 0.0.0.0 &**

or

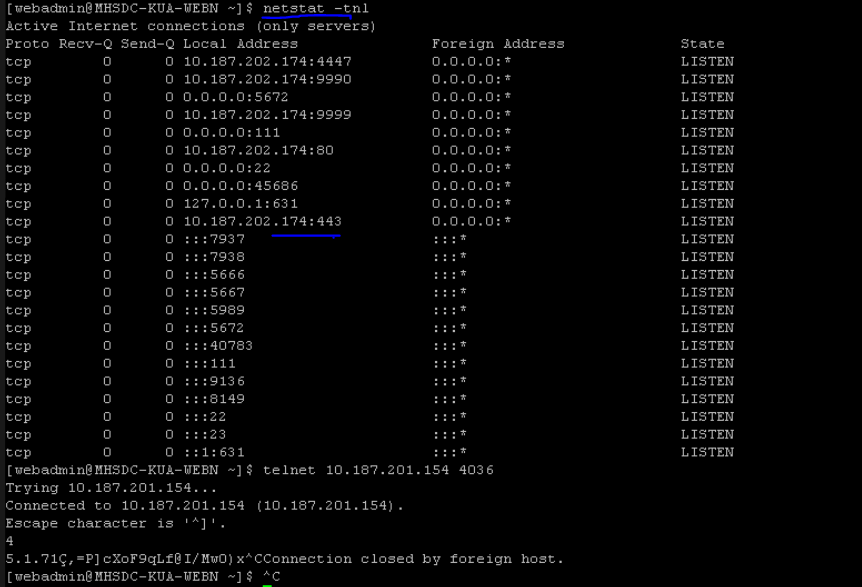
**for kua 2.1 prod start server**

nohup ./run.sh -Djava.rmi.server.useLocalHostname=false -Dremoting.bind\_by\_host=false -Dfile.encoding=UTF-8 -Djava.rmi.server.hostname=10.187.203.39 -b 0.0.0.0 &



1. Now check whether the services are started on browser using the api link provided in the mail.

Verify if the JBoss is running on the current server IP where port no: 443:



**NOTE:**These license keys are updated every 3 month for pre – productionaua/kua applications and every year for production server application.

To check server and db space and memory utilization:

* **df –h🡺**check server space
* **free –g 🡺**check memory utlization

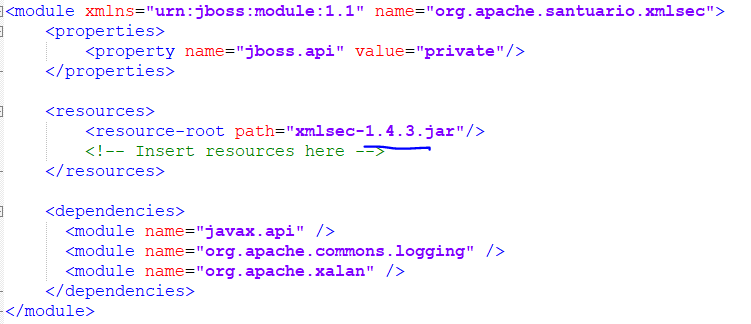
When new **JBoss-as-7.1** server deployed/installed with **JDK 7** make following changes:

1. Replace **1.5.1 xmlsec jar** with **1.4.3** and change the same in the **module.xml**. Rename **.jar.index** with 1.4.3 and restart the server.

jboss-as-7.1.1.Final\modules\org\apache\santuario\xmlsec\main

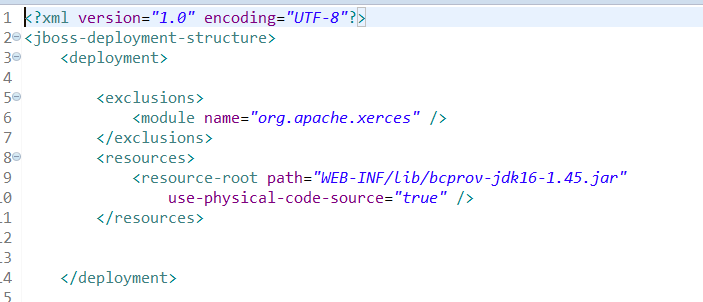
It resolved the below error.

**Error:**Caused by: java.lang.ClassCastException: org.jcp.xml.dsig.internal.dom.DOMXMLSignatureFactory  
cannot be cast to javax.xml.crypto.dsig.XMLSignatureFactory



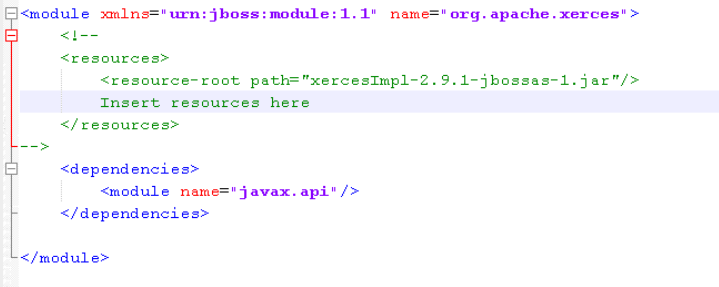
1. Error: JCE cannot authenticate the provider BC

Change in WebContent/META-INF/**jboss-deployment-structure.xml** in eclipse project and the mentioned jar in the classpath and the **WEB-INF/lib**



1. To prevent XXE injection we need to use the JDK jars. So to prevent the jboss jars to not be used during server startup comment out the module.xml file in jboss server files:

jboss-as-7.1.1.Final\modules\org\apache\xerces\main



**MOU Process:**

If a new client needs to register as a sub – aua to use our services we need to follow the below process:

1. Contact mantrayla**PMU** unit (e.g. Hemant, Bhavesh) and ask them to contact the client and they will contact clients for details. They would fill up the DIT form and approve it to send MOU to us for that client.
2. UIDAI will forward us sub – aua codes to enter it at our end.
3. Now we need to follow work at our end to register that sub – aua in our database.
4. Access the server for aua/kua and open **HeidiSQL** tool for database entry.
5. First register them in the pre – production database environment named – **kuaqadb**&**auaqadb**. Same will be applicable for production also when sub –aua is successful testing on pre – production.
6. In **auaqadb**:
   1. Open **auaqadb** database.

In **aua\_test** table add new client by duplicating the above row (right click on last row and use duplicate row option) and just rename and add the following column data.

* + 1. **dept\_code** (as on your own or provided)
    2. **dept\_name** (full name)
    3. **dept\_desc**(description)
  1. **aua\_users**table:

Select any row and duplicate it and create a new row. Add

* + 1. **aua\_user\_code** (provided by UIDAI)
    2. **dept\_code** (as entered from above step a.i)
    3. **password (on your own)**
  1. Add device in **auth\_devices** table:

Select any row and create a duplicate new row. Add

* + 1. **device\_code** (add on your own or provided)
    2. **aua\_user\_code** (from step – b.i)

Fill up the remaining rows as specific to sub – aua.

* 1. Now finally we need to share these with the client.
     1. aua\_code/kua\_code
     2. device\_code
     3. password
     4. ivconstant (from code specific)
     5. Encryption certificate (provided by DIT- JME.cer).
     6. PDFs

1. In **kuaqadb**:
   1. Open **kuaqadb**.

Create code in the kycgenericdatabase.

Create duplicate row in **aua\_subaua** table and add:

* + 1. **code** (provided by uidai).
    2. Fill the remaining rows like name,dept, address as provided by sub –aua

Once the sub – aua is successful in testing on pre – production environment then only uidai will provide the code for production environment for that specific sub – aua.

For production environment use database aua and kua and follow the above process to enter the sub –aua data in database.

**Note:**

1. We also have to ask client their server ip’s to register them and whitelist in our database for ip – binding.
2. Pre – production code starts with S... and Production code start with P...

**Transaction count details:**

For transaction related details from below tables:

1. For authentication transactions:
   1. **Aua** database: auth and otp transactions from *aua\_txn\_auth\_audit\_2\_0* and *aua\_txn\_otp\_audit* respective tables.
2. For ekYC transactions:
   1. **Kua** database: *kyctxnauditdetails\_2\_1* table.

**Configuration of host file:**

* The below configuration is same for aua and kua servers.
* Present in root/etc folder for each server.
* UIDAI gives us an IP e.g. = **10.5.32.17** (this ip is provided by uidai). But we do not provide our ip we instead bypass it by configuring it in properties file and then mapping that url in host file.
* When we hit the url using javacode (client.creates()) it comes to Aadhar.propertiesurl and then replace it with the host file ip address.
* In AUA 🡪
  + aadhar.properties file has url under stage parameter.

E.g.: authServerUrl = https\://**authpretest**.uidai.gov.in/uidaiauthserver/2.5/

In above url you can name authpretest as any name.

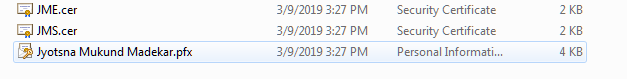
* + Now we need to configure this on the server pc host file. Just add an IP then add the above url as shown below:

10.5.32.17 **authpretest**.uidai.gov.in **authpretest**.uidai.gov.in

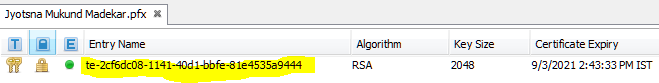
**Update application levelCertificates:**

There are two types of certificates needs to be updated at application level:

1. **Uidaiceritificate:**
   * Uidai provides certificates that needs to be share with the client i.e. sub – aua for pid data encryption.
   * There are different certificates for production and the pre – production server environment.
   * These certificates can be downloaded from uidai websites download section.
2. **Signing certificate:**
   * This certificate is shared to us (AUA/KUA) from DIT mantralaya which is to replaced in our project application.
   * This has 3 different certificates as shown below which is used for signing our request xml to be forwarded to uidai for authentication as well as eKYC.



* + The .pfx file is to be replaced in our folder application (currently JYOTSNA\_MUKUND\_MADEKAR.pfx). This is private certificate that should not be shared with anyone. You need to open this certificate with the provided password and copy alias name as shown below. Configure this .pfx file name, password and alias name in the properties file.



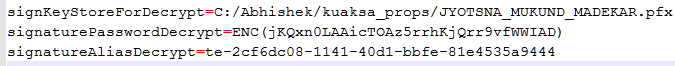
* + Also remember to encrypt the password using jasypt tool as with secret key=”jasyptsecret”.

Encryption of password is same as encryption of license keys explained earlier.

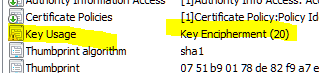
* + 1. For encryption:



* + 1. For decryption:



* + The **JME**(encryption )and **JMS**(signature) needs to be shared with the uidai to map at their end. These are public certificate and signing certificates.
  + Also rename and replace the digital sign certificate JME.cer as **digisign.cer** in request folder.
  + Kuadigisign.cer / JME.cer. If you open this certificate you will get the info in details tab.



* + In KUA/ KSA .pfx and kuadigisign.cer file should be in request folder (which is outside the war file).
  + In AUA .pfx file should be in resource folder (inside war file).
  + These certificates have validity.
  + The .pfx file is the master signature certificate. It is responsible for both encryption and signing.

**Basic rules of DSC:**

1.       It should be procured from a valid certification authority as per Indian IT Act (see http://cca.gov.in/cca/index.php?q=faq-page#n41)

2.       It should be a class II or class III certificate.

3.       The certificate should be procured from your Organization name. ‘O’ element should be present in the certificate under ‘Subject’.

4.       The Key Usage should be ‘Digital Signature’. It should not be SSL, Server Certificates or any other certificates.

5.       For signing purpose, please share the certificate having "key usage" value as "Digital Signature, Non-Repudiation (c0)".

6.       For encryption purpose, please share the certificate having "key usage" value as "Key Encipherment".

You can verify the encrypted password from url:

<https://www.devglan.com/online-tools/jasypt-online-encryption-decryption>

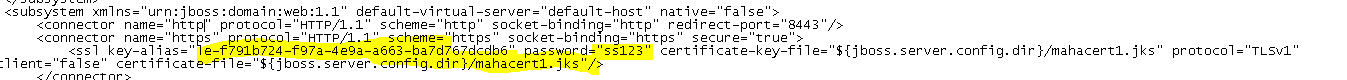
* + Also you need to share the JME.cer certificate with sub-aua for the encryption process.

**Update Server level SSL certificate (.jks):**

**Jboss new SSL certificate configuration:**

In Jboss-as-7.1 update the new certificate by pasting the new **.jks** file in the **standalone/conf** directory. Configure the new credentials in the **standalone.xml**as shown below such as:

* **Alias-name:** obtained by opening the certificate (.jks) file in the keyStoresoftware tool
* **Password:** provided by the vendor to open the certificate
* **Path:** where the certificate has been saved (here in conf directory)



In Jboss-5.0.G.A.the configuration file is present in the **default/deploy/jbossweb.sar/server.xml** file. Here we don’t need to configure the alias-name. Just update the password and the path of the certificate. Also the path of the certificate will be default/deploy directory.

Restart the server and check the status.

