Problem :
Currently all the APIs created in API Manager can be secured using OAuth2 token in API Gateway level. If the APIs need to be secured using mutual SSL, we do not provide a out-of-box solution to achieve the same. The proposed feature is to support such requirement and with this feature, users will be given the flexibility to select either OAuth2 token or mutual SSL or both as their API security option.

Proposed workflow for creating APIs that are secured using mutual SSL.

- In the publisher level, when the API security options is prompted in API Manage section, the mutual ssl option need to be selected.
- The client certificates that need to be used for accessing this API need to be uploaded when prompted in UI.
- When uploading the client certificates the user need to select the subscription tier the requests need to be throttled against. Throttling key will be <CN_OF_CERTIFICATE>_<API_IDENTIFIER>
- The relevant certificates will be stored in the database. Following is the proposed table schema,

```
CREATE TABLE IF NOT  `AM_API_CLIENT_CERTIFICATE_METADATA` (
        `TENANT_ID` INT(11) NOT NULL,
        `ALIAS` VARCHAR(45) NOT NULL,
        `API_ID` INTEGER NOT NULL,
        `POLICY_ID` INT(11) NOT NULL,
        `CERTIFICATE` BLOB NOT NULL,
        FOREIGN KEY (API_ID) REFERENCES AM_API (API_ID) ON DELETE
CASCADE ON UPDATE CASCADE,
FOREIGN KEY (POLICY_ID) REFERENCES AM_POLICY_SUBSCRIPTION
 (API_ID) ON DELETE CASCADE ON UPDATE CASCADE,
 PRIMARY KEY (`ALIAS`)
);
```

Proposed workflow for publishing APIs that are secured using mutual SSL.

- Relevant client certificates will be published to all the gateways that is selected for current API.
- When publishing API synapse artifacts, two new properties will be injected to APIAuthentication handler. APILevelPolicy and APISecurity will be passed to API AuthenticationHandler.APILevelPolicy is needed as, if it is oauth2 token flow relevant api level throttling policy will be retrieved from key manager level, since we do not have any token involved, we need to add this as property.
- Further when the API is secured only using mutual SSL, each API resource will not be having a filter mediation based on key type, the relevant api call will be send to production endpoint. Sample API resource definition looks like below,

```
  <resource methods="GET" url-mapping="/menu" faultSequence="fault">
```

```
    <inSequence>
      <property name="api.ut.backendRequestTime"
            expression="get-property('SYSTEM_TIME')"/>
      <send>
        <endpoint key="PizzaShackAPI--v1.0.0_APIproductionEndpoint"/>
      </send>
    </inSequence>
    <outSequence>
      <class
name="org.wso2.carbon.apimgt.gateway.handlers.analytics.APIMgtResponseHandler"/>
      <send/>
    </outSequence>
  </resource>
```

Proposed workflow when accessing the APIs that are secured using mutual SSL from store.

- The user need to add the client's private certificate to browser.
- When the API is invoked from the browser, server will send set of trusted certificates from its trust store.
- Based on that browser prompts the list of certificates that are available in browser end.
- User selects a certificate and if the mutual ssl authentication succeeds, axis2 message context will contain a property "ssl.client.auth.cert.X509" with the client certificate. If that property is missing, the API invocation will fails with an error message.
- Username of the current API invocation will be picked up from CN of the certificate.