

Service Design - New

- Deliverables Checklist :: Demo and Review Status
 - Brillio Technical Team
 - Experian Technical Team
- Overview
- Diagrams
 - High Level Diagram
 - High Level Diagram details
 - Components Diagram
 - Flow Diagram
- Important documentation links related

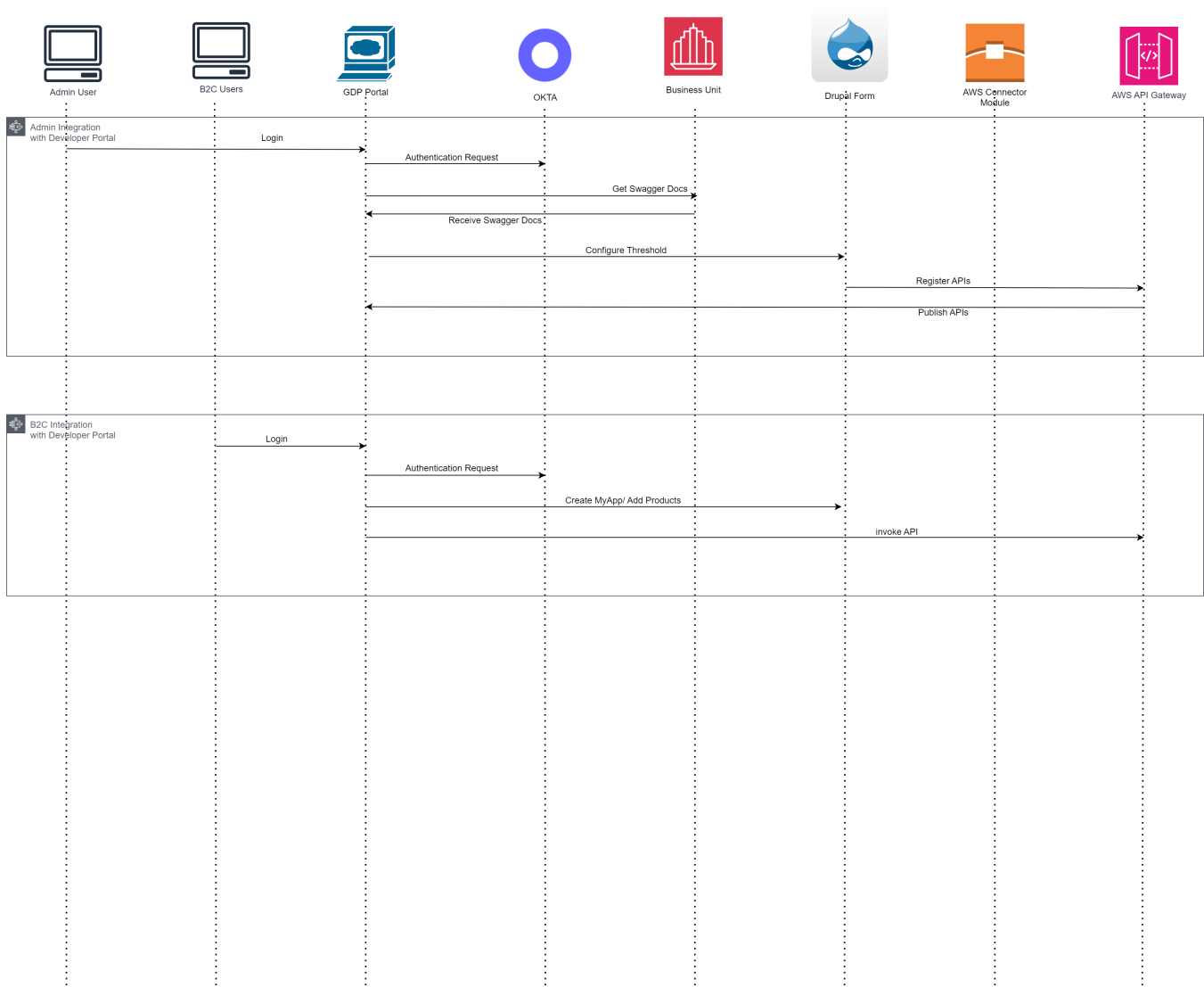
Deliverables Checklist :: Demo and Review Status

[illegible]

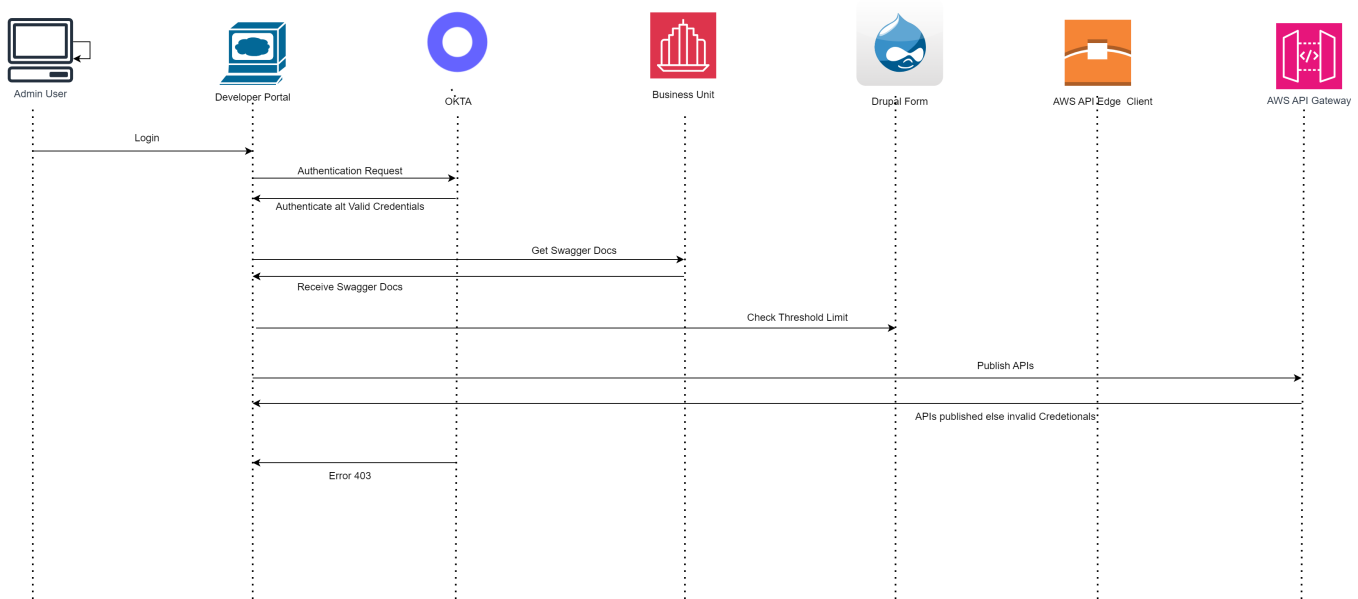
- Workflow-2: Applying Certificates to AWS Certificates Manager and AWS Enterprise Gateway. Steps as follows:
 - APICoE team gets approval from marketing team on 'DNS' entry
The APICoE raises a request for a new DNS entry via Email sent to the CSS team. A serviceNow ticket is created with the CName/DNS
 - CSS team creates and registers the domain and updates the environment based on ServiceNow approval
 - Certificate created by the APICoE team. 3 Certificates are manually downloaded and the content is copied into a pem file eg. 'certwith_chain.pem'
 - API CoE team applies the pem file to both, WAF and S3 bucket
 - Lambda function updateTruststore() gets triggered & update the AWS Certificates Manager
 - Lambda function applyCertificate() gets triggered & updates AWS API GW with the certificate
 - VPC endpoint will identify AWS API GW using <<us1-experian.com/buapigw>>
 - Rules needs to be defined as routes in Ent API GW: /buapigw/api1 | /buapigw/api2
- Workflow-3: Okta integration with AWS Cognito and AWS Dynamo DB
 - Redirect to OKTA or Authentication using JWT authentication type. OKTA authenticates the user. Returns error '403' if credentials are invalid
 - If authentication is successful then OKTA invokes the API. At this point OKTA also returns the 'authorization-code' to AWS Cognito. AWS Cognito receives token using authorization code from OKTA
OKTA returns the ID and accessToken
 - User is created in Cognito User Pool and pre-sign-up Lambda triggers
 - This calls the updateUserDetails() function to update AWS Dynamo DB
 - API GW validates access token with AWS Cognito Built-in JWT Authorizer and returns the token validity.
Web sends GET/PUT/POST/DELETE HTTP Requests to AWS API GW that invokes AWS Lambda service to call the relevant Microservice via MicroGateway and returns the HTTP Response to the Web

Components Diagram

Complete Flow



- **Workflow-1: Publish APIs to AWS Enterprise Gateway using Drupal CMS**



Flow Diagram

• Workflow-1: Publish APIs to AWS Enterprise Gateway using Drupal CMS. Steps are as follows

- Log into Developer Portal (<https://developer.experian.com>) as an admin

◦

Install a AWS Client app that will integrate Drupal and AWS API GW

- Get Swagger Docs from BU

◦

Open form "Enter API details". Enter information such as 'Region', 'Threshold limits'

- Upload the swagger doc. This will export the APIs to AWS API GW and list them
- API Gateway allows you to publish APIs to Developer Portal from where they are available for consumption by developers and consumers.

API Gateway also allows you to publish the APIs to the following destinations:

Service registries. This enables applications to dynamically locate an API Gateway instance that can process that API.

Integration Server. This is used in API first implementation approach.

- Customize the APIs before publishing them by way of AWS API GW
eg. -Restrict the exposure of specific resources, methods, and operations of an API to other applications.
-Define a custom gateway endpoint by customizing the URL of the gateway endpoint that your users will use to access the API.
- The following sections describe how you can activate an API, customize the gateway endpoint, and publish APIs to different destinations.
- Activating an API
 - You must first activate the API before publishing it to a portal so that the gateway endpoint is available for developers and consumers to invoke the API.
 - You must have the Activate/Deactivate APIs functional privilege assigned to perform this task. You can activate an API in the Manage APIs page. Alternatively you can also activate the API from the API Details page.
 - The Gateway endpoint is now available, which can be used by the consumers of this API. You can now publish the API to the required destination and expose the API for consumption by the consumers.
- Once the API is activated, you can define the custom gateway endpoints. For more information about gateway endpoints, see Gateway Endpoints.
- Once the API is activated, you can enable the tracer. For more information about how to enable the tracer and view the tracing details, see Trace API.
- Publishing an API to Developer Portal sends the SOAP and REST APIs to Developer Portal on which they are exposed for testing and user consumption. The process of publishing an API to Developer Portal is initiated from API Gateway and is carried out on the Developer Portal server. Doing this involves the following high-level steps:
 - You initiate the publish process by selecting the API to be published, specify the API endpoints to be visible to the consumers, and the Developer Portal communities in which the API is to be published.
 - API Gateway publishes the API to each of the specified Developer Portal communities.
 - During bulk publishing of APIs, the process continues even if API Gateway encounters a failure with Developer Portal.
- When publishing an API to the Developer Portal destination, keep the following points in mind:
 - The Developer Portal destination must be configured in API Gateway.
 - You must have the Publish to Developer Portal functional privilege.
 - You cannot publish an API if it is in inactive state. You have to activate the API before publishing it.

- 1) Log into Developer Portal (<https://developer.experian.com>)
- 2) Click on My Apps link in user navigation menu and choose the Region(Click on View My Apps button)
- 3) Click on the create new app button
A new form that will take the 'App Name' as an input and checkbox of APIs that could be listed in the app. 'Add App'
- 4) The 'client_id' and 'client_secret' gets generated from Drupal and stored in the Drupal DB
These will be used to make the calls from your application to the Experian API.
- 5) The 'New App' should get listed under the respective region under 'MyApps' with information from Drupal DB
APP Name | client_id | client_secrets | Operations (Edit/Delete) | Status (Approved)
- 6) Get an access token from AWS Cognito
Get your access token using the Client ID and Client Secret from the application created earlier along with your Developer Portal username and password.
The call to get the Oauth2 token is a POST request with a Content-Type which needs to be specified as JSON; the response will also be in JSON format:

```
curl -X POST https://sandbox-us-api.experian.com/oauth2/v1/token \
-H 'Accept: application/json' \
-H 'Content-type: application/json' \
-H 'Grant_type: password' \
-d '{"username": "<USERNAME>","password": "<PASSWORD>","client_id": "<CLIENT_ID>","client_secret": "<CLIENT_SECRET>"}
```

```
{
  "issued_at": "1478405901908",
  "expires_in": "1800",
  "token_type": "Bearer",
  "access_token":
    "eyJraWQiOiJSbSmpTMXJQqjdJODBHWjgybmNsSlZPQkF3V3B3ZTVYblNKZUdSZHdpclEYxliwidHlwIjoiaSldUliwiYWxnIjoiUlMyNTYifQ.",
    "eyJzdWIiOiJmcmVkZGlllwiRw1haWwiOiJmcmVkaXJpYy52YW5kZXJlbHN0QGv4cGVyaWFuLmNvbSIsIkZpcnN0TmFtZSI6ImZyZWRRkaWUilCjpc3MiOiJFWFBFUklBTilsikxc3ROyWY1IljoideFuZGVyIGVscylzslmV4cCI6MTUwOTAyNDkxMSwiaWF0IjoxNTA5MDE3NzExLCJqdGkiOiIYTdlYzJhZS00YzdiLTQ3Mzk0YmU0MS1hMDdjZTBmNTc2N2YifQ."
}

MINbv9JtA9JeBW0cp6mhRZ7xhOvu3ol8WDu73xInj14w1fZRXXPaHFR9
OsCgzGm-7s46vS2vyScd_MWYIFRWQ8TUgPZ6Gbdk43l_B4UJxu5Uujh1bhrWA1KCsr5p7LKNi6Pxhc76oVd2EAAd0l3X7um-d_flld1N4KA
XmYtXT_oU8DkIKHYiWH6L5Yx3Ue_kYQwXikqU0nXdvdab35KyFCza9XqSJEEVEubTSdTvvVluzv4AJxN5X-yEtzOtNtv_Yynj4KzdYv8tpuoF2
LGdzp4G0fOe8mLRNY3g4rCfAnpe0yc1h6LHh0TMhu2e8jalro4dx7b3VhLIXULm1RqPw",
"refresh_token": "3lb5SjC6AOux5R47ffobFFi8DhGIC2GO"
```

7) Send the access_token to Okta

```
curl -X GET
'https://sandbox-us-api.experian.com/businessinformation/businesses/v1/search' \
-H 'accept: application/json' \
-H 'authorization: Bearer
eyJraWQOiJBbSnpTMXJQJkdJODBHWFjgybmNsSIZPQkF3V3B3ZTVYbINKZUdSZHdpcEYxliwidHlwIjoiSldUliwiYWxnIjoiUlMyNTYifQ.'
eyJzdWl0OiJmcmVkZGlllwiRW1haWwiOiJmcmVkZXJpYy5yZW5kZXBjHN0QGV4cG9yYWFuLmNvbSlslkZpcnN0TmFtZSI6Im
ZyZWRkaWU
iLCJpc3MiOiJFWFBFBUIK1tIsLkxhc3ROYW11IjoIdmFuZGVyYIGVscylsImV4cCI6MTUwOTAyNDkxMSwiaWF0IjoxNTA5MDE3ZnE
xLCJqdG
kiOiI1YTdlYzJhZS00YzdiLTQ3MzktYmU0MS1hMDdjZTBmNTc2N2YifQ.'
MINbv9JtA9JeBW0cp6mhRZ7xhOvu3ol8WDu73xlnj14w1fZRX0
PaHFR9OsCgzGrn-
7s46vS2vyScd_MWYfFRWQ8TUgPz6Bgdh43l_B4UJxu5Uujh1bhrWA1KCsr5p7LKNl6Pxhc76oVd2EAd0lX37um-d_fl
lds1N4KAXmYtXT_oU8DklKH YiWH6L5Yx3Ue_kYQwXikqU0nXdvab35KyFCza9XqSJEEvEubTsdTvvVluzv4AJxN5X-
yEtzOtNTV_Yynj4Kz
dYv8tpu0F2LGdzp4G0fOe8mLRNY3q4rCfAnpe0yc1h6LHh0TMhu2e8jalro4dx7b3VhLIXUL
```

Important documentation links related

- Certificates Management and DNS registration [Certificates Management and DNS Registration](#)
- Okta Integration with WAF and API Gateway [Manual Okta Integration with WAF and API Gateway](#)
- Service Catalog [delete the page](#)
- Authorization/Authentication APIs Test cases scenarios
- Developer Portal Test cases scenarios
- CI/CD
- Terraform name conventions [Terraform Best Practices#NameConvention](#)
- Terraform scripts organization [Terraform Best Practices#StandardStructure](#)
- Bitbucket project link <https://code.experian.local/projects/APDE/repos/eits-enterpriseapigateway-aws/browse>