Document Attributes

| Attribute | Value |
| --- | --- |
| Application ID /  Application Name | • OPUS- MOTS ID: 18257  • OPUS Mobile- MOTS ID: 18257 |
| Owner | Madas, Arun (am6489) |
| Owner Contact Information |  |
| Other Attribute |  |

## 

Revision History

The following table lists the revision history of this document:

| Author | Date | Version # | Revision Description |
| --- | --- | --- | --- |
| am6489 | 12/07/2015 | 0.1 | Initial Draft |
| am6489 | 12/20/2015 | 0.2 | Update doc after 1st HLD walk through comments. |
| am6489 | 12/22/2015 | 0.3 | Updated doc with comments from 2rd HLD walk through |

Table of Contents

[Overview 3](#_Toc406166738)

[Problem Statement 3](#_Toc406166739)

[Design Decisions 3](#_Toc406166741)

[Functional Overview 6](#_Toc406166742)

[Middleware Design 7](#_Toc406166743)

[External Interfaces 8](#_Toc406166744)

[Component Diagram 8](#_Toc406166745)

[Performance Design 9](#_Toc406166746)

[Security Design 9](#_Toc406166747)

[UI Design 9](#_Toc406166748)

[UI Design - Wireframe 9](#_Toc406166749)

[UI Design - Screen Element Definition and Data Mapping 10](#_Toc406166750)

[UI Action Processing - Sequence Diagram 11](#_Toc406166751)

[Asynchronous/Batch Processing - Sequence Diagram 13](#_Toc406166752)

[Externalized Configuration 13](#_Toc406166753)

[Database Tier Design 14](#_Toc406166754)

[Logical Data Model 14](#_Toc406166755)

[Database Interfaces 14](#_Toc406166756)

[Batch Interfaces 14](#_Toc406166757)

[Replication 14](#_Toc406166758)

[Alternative Designs 14](#_Toc406166759)

[Assumptions/Risks 14](#_Toc406166760)

[Pre-Production Disaster Recovery Planning 14](#_Toc406166761)

[Other Plans and References 14](#_Toc406166762)

[Acceptance & Approvals 15](#_Toc406166763)

## Overview

The High Level Design (HLD) describes how an application will implement the architectural concept and meet the requirements allocated to that application. The High Level Design describes the functionality the application will provide, the overall design for providing that functionality and meeting the nonfunctional requirements, and the rationale for choosing that design. The High Level Design also describes how the application will cooperate and interface with other applications to provide an integrated solution that achieves the architectural concept.

The High Level Design covers these topics:

* Problem Statement
* Design Decisions
* Alternative Designs
* Assumptions/Risks
* Other Plans/References

## Problem Statement

The objective of this project is to enhance OPUS to sell Microsoft Office Products.

AT&T will sell only the licenses. The actual provisioning of service, accounting, delivering the functionality all reside inside of Microsoft.

If a customer has already a CRU account, Customer can order any office 365 on to any of the CTNs on this account. The Office 365 products will be modelled as SOCs. OPUS will be creating NEW flow (similar to add feature) to be able to add office 365 SOC.

The following functionalities will be supported by OPUS

**New Office 365**

- Ability to add Office 365 product to an existing CRU customer (on a CTN).

**Change Office 365**

- Increase/Decrease the license quantity on EXISTING office 365 Products on the CTN.

- Add additional office products to the account. This account may have other set of products.

**Cancel Office 365**

- This will be a use case where is customer can zero the license quantity meaning discontinue a office 365 product.

**Cancel the CTN or Porting out the CTN**

- Customer trying to cancel the CTN, in such cases there will be M&P to warn the Customer of losing the office 365 subscription with ATT. In such cases OPUS will have to notify other systems such as CSI-DCM and TLG.

**Discount**

- Existing wireless discount framework will be leveraged for office 365 orders.

- There is no expectation to build P times Q promotions/discounts.

## Assumptions

1. All Microsoft O365 products will be added to TLG as Bolt-on SOCs.
   1. For large business CRU customers, OPUS shall re-use the permission for feature, to add/edit/delete Microsoft O365 products (No MRE impacts for 286365. MRE will not be sending any new permissions).
2. There shall be only one CAPES form to fill during the order. It is assumed that CPC catalog will indicate to OPUS to use the same form for all the products and all types of activities.
3. There are no CSS impacts for this project.
4. Today in OPUS, when a feature is added during the order before adding an equipment to the cart (or any other non-zero item that gets added to the cart), OPUS auto rings out such order and no receipt is printed. However, if the feature is added after items are added to the cart, rep is expected to ring out the order to complete the transaction and a receipt is printed which includes “Feature Change” line entry.

The above statement results in the below three assumptions

* 1. MO 365 order will also follow the same rule and will be rung out automatically if MO 365 items are added to the cart before adding equipment/items which require ring out.
  2. If bundles/promo are to be applied based on a MO 365 product purchase and some other item purchase, rep will have to ensure (**M&P**) that other items which need ring out will have to be added to the cart before adding MO 365.

For example:

CPC has an offer

“Purchase MO 365 with iPad- get $100 off”

In OPUS-

If user adds in the below order, customer will not get the promotion

MO 365 - Auto ring out

Then adds Ipad to the cart

However, if user adds in the below order, customer will get the promotion

IPad is be added first

Then MO 365 cart

Order is rung out together, thus applying the promotion.

* 1. For a MO 365 order, receipt will be printed only if the MO 365 order is added after adding equipment/items which require ring out to the cart.

1. It is assumed that the promotions capabilities in CPC will be used in this project to effect promotions which will target equipment discounts. Current solutions are not capable of applying/communicating service discounts MRCs to subscribers(on feature)
2. OPUS shall retrieve the product catalog from CPC on a nightly basis. This pricing catalog will have standard/RACK pricing for the office 365 products. OPUS is not expected to retrieve discounted prices from CPC in realtime (CPC custom discounts are out of scope for OPUS).
3. When a CTN or BAN is being cancelled or port out which has MO365 subscription on it, OPUS is not expected to have any logic specific to MO 365. OPUS shall continue to process the cancel/port-out requests BAU. There will be no MO 365 specific messaging for the customers.
4. If an order is placed from BOOST, there will be no CAPES unique id in CAPES system. BOOST does not talk to CAPES. So, after the order in BOOST, if the customer walks into store, OPUS will have to create a new form asking all provisioning details. However, if the order was placed from PHX, OPUS can retrieve the provisioning details from CAPES.
5. OPUS will be capable of onboarding any other PxQ products which have requirements IDENTICAL to those listed in this project for MS O365 (in terms of # of forms, pricing structure, product structure, dependencies, systems to be intimated etc). If there are any PxQ products whose requirements differ from those listed for MSO 365, in this project BRs, accommodating them in OPUS will require code changes.
6. It is assumed that CAPES will have some history of any MO365 products or any other PxQ AMS products if the orders are placed from OPUS or PHX. However, if the order is placed from BOOST, CAPES will not have such order details.
7. When a CTN with MO 365 product does a ToBR activity, OPUS does not have to drop the MO 365 products from the account. TLG will ensure that the MO 365 products are not carried over to the Non-CRU account.
8. OPUS assumes that there is no impact to printed/emailed receipt footer content.
9. There will not be any Office 365 products which will “conflict” with any other products/feature. Also, all Office 365 products are compatible with all other Office 365 products.

## Component List

This section lists each component that will be created, modified, or removed.

|  |  |  |
| --- | --- | --- |
| AmsProductsServicesBean.java | created |  |
| AmsProductsServices-ejb-jar.xml | created |  |
| AmsProductsServices-weblogic-ejb-jar.xml | created |  |
| AmsProductServicesDAO.java | created |  |
| AmsProductServicesDAOImpl.java | created |  |
| CapesServicesDAO.java | created |  |
| CapesServicesDAOImpl.java | created |  |
| CapesTemplateDBRecord.java | created |  |
| CapesWebServicesManager.java | created |  |
| CapesWebServicesUtils.java | created |  |
| CapesWSException.java | created |  |
| InquireMobileProductsOrderQryCmd.java | created |  |
| ManageMobileProductsOrderQryCmd.java | created |  |
| AMSProductsServicesManagerEvent.java | created |  |
| AmsTemplateDownloadEvent.java | created |  |
| AmsProductsServicesHome.java | created |  |
| AmsProductsServicesRemote.java | created |  |
| AmsCategory.java | created |  |
| AmsProduct.java | created |  |
| AmsProductCatalog.java | created |  |
| AmsProductDataInfo.java | created |  |
| AmsProductPrice.java | created |  |
| AmsProductPriceList.java | created |  |
| AmsProductSubscription.java | created |  |
| AmsProductSubscriptions.java | created |  |
| AmsSubCategory.java | created |  |
| AmsCategory.java | created |  |
| AmsProductsRequestInfo.java | created |  |
| AmsProductsResponseBase.java | created |  |
| AmsProductsResponseInfo.java | created |  |
| CapesRequestInfo.java | created |  |
| CapesResponseInfo.java | created |  |
| AmsConstants.java | Created |  |
| AmsTemplate.java | created |  |
| AmsTemplateAttribute.java | created |  |
| AmsTemplateGroup.java | created |  |
| InquireMobileProductOrderResponse.java | created |  |
| InquireMobileProductsOrderRequest.java | created |  |
| ManageMobileProductsOrderRequest.java | created |  |
| ManageMobileProductsOrderResponse.java | created |  |
| PriceDetails.java | created |  |
| ProductOrderDetails.java | created |  |
| AmsProductReviewAndSubmit.jsp | created |  |
| AmsProductSubscription.jsp | created |  |
| CAPESProvisioningForm.jsp | created |  |
| AmsProductCSSummaryAjax.jsp | created |  |
| AmsProductsServicesBean.java | created |  |
| AmsProductsServices-ejb-jar.xml | created |  |
| AmsProductsServices-weblogic-ejb-jar.xml | created |  |
| AmsProductsServicesBean.java | created |  |
| AmsProductsServices-ejb-jar.xml | created |  |
| customerservice-struts-config.xml | Modified |  |
| customerservice-tiles-defs.xml | Modified |  |

## Design Decisions

|  |  |  |  |
| --- | --- | --- | --- |
| **Req. ID** | | **Design Element** | **Trace-To** |
| 286365\_OPUS\_OM\_HLD\_01 | | HLD44 OPUS/OM shall support single ‘on/off’ project flag 286365\_AMS\_PRODUCTS enable/disable all 286365 functionality for both OPUS & OM to support the newly added functionality. Default value of the flag will be “OFF” as we deploy in OFF Mode.  OPUS/OM shall use single OM jsp’s for all new screens for both OPUS and OM while utilizing separate existing opus screens for non-new screens for updates.  OPUS/OM shall be enhanced to introduce a mechanism to ensure better handling of store configurable.  OPUS shall create two new columns in the store\_configurable\_list table.  Column 1 -  Indicate the purpose of the store configurable  Allowable values: “Functional” ; “deployment management”  Column 2 could indicate the time until when the store configurable will be active  **ALTER** **TABLE** STORE\_CONFIGURABLE\_LIST **ADD** ( PURPOSE **VARCHAR2**(100));  COMMENT ON COLUMN STORE\_CONFIGURABLE\_LIST.PURPOSE  IS 'PRE1602/Functional/Deployment Management’;  **ALTER** **TABLE** STORE\_CONFIGURABLE\_LIST **ADD** ( TIME\_UNTIL\_ACTIVE **DATE**);  COMMENT ON COLUMN STORE\_CONFIGURABLE\_LIST.TIME\_UNTIL\_ATIVE  IS ’Date until this configurable is active';  For this project the values will be  Purpose: ‘Functional’  TIME\_UNTIL\_ACTIVE: 12/31/2099??  Disclaimer : The usage of how these columns will be used for rollout or other functional use cases with-in opus will be addressed later. | 286365\_OPUS\_SR\_370  286365\_OPUS\_SR\_380 |
| **1 . Customer Summary Enhancements** | | | |
| 286365\_OPUS\_OM\_HLD\_02 | | OPUS shall create a new additional pod in Mobile Information to display the subscriptions for the CTN. This pod will be visible only to CRU customers based on account type and customer summary MRE permission values to determine FAN/FAN Liability type  The list of products subscriptions for the CTN in context will be determined by the response from the CSI InquireMobileProductsOrder api call for the CTN in context. The Enterprise config for the project for supported product families will determine the products displayed.  **High level design and Algorithm**   1. The information in customer summary pod is populated by leveraging the CPC catalog. 2. The CPC catalog object cloned in FE would serve as a reference point for the sorting and organization of the products list from DCM. 3. The InquireMobileProductsOrder call returns a list of applicable products spread across all the product families. The only way to effectively display the CS with its organized data is to match the response with the Catalog and then remove the unmatched products. 4. The products now sorted into their respective categories would serve as a easy object for use in the display   In case we are unable to retrieve the quantity information from DCM we would show a generic message to the user    The CS pod would serve as a launch point for other activities , such as view subscriptions if there are prior subscriptions or Order products if there are no prior subscriptions | 286365\_OPUS\_SR\_010,  286365\_OPUS\_SR\_020,  286365\_OPUS\_SR\_030, 286365\_OPUS\_SR\_050  286365.SR.OPUS.80, |
| **2. Catalog Selection** | | | |
| 286365\_OPUS\_OM\_HLD\_03 | | Opus/OM shall ensure to validate the BAN status, CTN status via the Telgence getMobilelevelData to do prevalidation for the Catalog page where the user can increase/decrease quantity and add new quantity  OPUS shall leverage CSI api InquireMobileProductsOrder to obtain the current products list from DCM. The information is already populated from the CS page and thus on this page the information previously gathered will be leveraged and the existing product information will be show to the user.      The CPC catalog object continues to be important for the dispay of the available products section.  The View subscription page consists of the three following sections   1. Existing produts section- that represents he list of the products/families 2. New Products- the section that represents the products added 3. Product Catalog- A filtered product catalog, that does not have the existing products listed in them. The catalog object is used once again as a reference object to filter out the existing products and used to display the relevant product catalog   The information gathered in the CS POD is re utilized and further DCM calls are not made until the CTN in context is changed/rungout.  At any given point the drop down represents the subscription in that particular product family. Changes made by changing the drop down, resets everything and the user has to start all over again. MO 365 SOCS display are only restricted to MO 365 ordering flow.  HLD46 | 286365\_OPUS\_SR\_070, 286365\_OPUS\_SR\_100,  286365\_OPUS\_SR\_110,  286365\_OPUS\_SR\_200  286365\_OPUS\_SR\_210, 286365\_OPUS\_SR\_260  286365\_OPUS\_SR\_025 |
| 286365\_OPUS\_OM\_HLD\_04 | | View All CTN  On load of the page we will just show the search by ctn & the view ALL ctns option.  2. if its search by ctn, we will take the input and verify that its part of the same BAN using the ScopeKey.SUMMARY\_MOBILES\_ON\_ACCOUNT\_LIST, which will be loaded up when a customer look up is preformed.  if this key is not present we will make TLGAPI Call: appQuery.getBanAndCtnList and verify the ctn against the BAN.  3. make the DCM call and show the results.  4. if its view all, we will make the CAPES call and for the returned CTNS we will make the individual DCM calls.  Note: Add an enterprise config with DCM threshold = 10(to start) and limit calls to DCM based on threshold.      When CAPES does not return any CTNs or if there are no products assosciated in a single CTN search scenario, OPUS will display an informational message as represented in callout D.    HLD46 | 286365\_OPUS\_SR\_060 |
| **3. Provisioning Information** | | | |
| 286365\_OPUS\_OM\_HLD\_05 | | FR13 OPUS shall be enhanced to be able to download the CAPES form template from CAPES on a nightly basis. Each template carries a version number and unique templateId and will be used to download on a nightly basis. When no changes exist, no templates will be updated in the opus database. Please see diagrams below for detailed explanation. | 286365\_OPUS\_SR\_120 |
| 286365\_OPUS\_OM\_HLD\_06 | | The form content shall be retrieved dynamically from CAPES before rendering the form content on the GUI.  OPUS will render the respective provision forms with the data which is returned from SearchFormRequest API.  The custom tag AmsElementTag .java is used to render the individual HTML elements. This tag maps the “Attr type” for each of the elements in the CAPES form to the HTML input elements.  CAPES to OPUS Attribute Type mappings are maintained in common java file AmsConstants.java and data fetched from previously submitted capes forms in backend are converted to represent a common object structure with decoded HTML Attribute types for use in the AmsElementTag.java  **Disclaimer :** All CAPES Templates to be used in OPUS will have to be first run in QC environment before releasing to production. We have seen XML’s coming incorrectly in some cases (it could be an assumption at this time). | 286365\_OPUS\_SR\_130 |
| 286365\_OPUS\_OM\_HLD\_07 | | OPUS shall be able to prepopulate the details available in OPUS for the customer in the CAPES form.  CAPES maintains a unique id for each attribute in the CAPES form.  **Pre-populate Provision Info:**  4 possible user actions are available on catalog selection page:   * Add New product in catalog page = Activation form * Remove Existing product in catalog page = Deactivation form * Increase the quantity on an existing product = QuantityIncrease form * Decrease the quantity on an existing product = QuantityDecrease form   OPUS shall track the user actions on catalog page based on the above 4 actions and appropriately dynamically create an instance of the template form (cached in the frontend, from list of latest version templates downloaded from capes in opus database) and use SearchForm API to query based on the CTN.  Current form of SearchFormAPI does not support querying up for 4 formTypes (above 4 user actions) in one single api call. OPUS will perform 4 back and forth OPUS🡨🡪CAPES SearchFormAPI calls with limit=25 to fetch past submitted records on that CTN to proceed with element attributeid mapping for prepopulation.  OPUS shall use the modifiedDate attribute to sort through results to get one unique most latest record (off the limit count = 25) and use the same for pre-population of data.  Prepopulation of provision info page is dynamic. Template stored in the database with no values are merged against the unique record fetched from the SearchForm request api to map the AttributeIds (unique values in capes templates) to render the UI with prepopulated values.  **Pre-populate Seller Info:**  OPUS would store a local application java code mappings of pre-defined set of seller info attributes (including seller attid, seller first name, seller last name, seller email address, seller title, seller phone fields) CAPES ATTRID vs Mapping of OPUS unique field names to identify for pre-population with opus session values of the current logged in rep.  OPUS would map the these “Attr id” in the CAPES forms to the data that is available in the session.  Any change in the CAPES id and the attribute mapping would require a code change in OPUS. | 286365\_OPUS\_SR\_140  286365\_OPUS\_SR\_230  286365\_OPUS\_SR\_235  286365\_OPUS\_SR\_270 |
| 286365\_OPUS\_OM\_HLD\_08 | | OPUS shall allow the user to review the O365 selections made in OPUS.  OPUS will go through all the changes to the existing products and the additions of new products and will present them as displayed below. | 286365\_OPUS\_SR\_150  286365\_OPUS\_SR\_240  286365\_OPUS\_SR\_290 |
| 286365\_OPUS\_OM\_HLD\_09 | | Once the customer completes making selection on their MO 365 products (add/change), OPUS shall take the customer to the CAPES form page and ensure that OPUS collects all the necessary data from the customer.  OPUS will dynamically present respective forms such as Activation, Deactivation, Quantity Increase and Quantity Decrease forms depending upon the customer selections on their MO 365 products. | 286365\_OPUS\_SR\_220 |
| **4. Review and Submit** | | | |
| 286365\_OPUS\_OM\_HLD\_10 | Click on Submit order on “Review and Submit” page,it invokes submit dispatch in AmsProductAction.java.    Call AmsProductsServicesBean.java processAmsProductsService method through AmsProductManager.java manager class.    Invoke processSubmitTransaction() method for the submit request input.  Capes Submission:  Invoke submitCapesProvisionInfo() for capes call processing.  Call createCapesForm() for New form creation.Passing Capes Uniqueid from CreateCapesForm Response and status “Initiated” call validateCapesForm() for validations.for any errors throw error object and call cancelCapesForm() passing “capes Unique ID”  .  Call submitCapesForm() passing Capes Unique ID and status “validate”. for any errors throw error object and call cancelCapesForm() passing “capes Unique ID”  TLG Call submission:  By checking capes response for submitcapesform status “submitted” Call addSoc() TLG call to store AMOCS socs.  If TLG call fails call cancelForm() method invocation by passing “capes unique id” and throw error object.  DCM Call submission:  By checking TLG call Boolean status invoke DCM manageMobileProductsOrder() method.  If DCM call fails roll back TLG call by calling deleteSoc().  Roll Back capes call by calling  cancelForm() method .  OPUS DB log for AMOCS.  If DCM call success invoke saveAmsProductTransActions() method in AmsProductServicesDAOImpl.java file.  Create AMS\_PRODUCT\_TRANS\_DATA table,P\_AMS\_PRODUCT\_TRANS\_DATA  Procedure to log AMOCS Transaction details.  Once the calls to CAPES, DCM and TLG are successful, OPUS will call TLG banActvConv.setMemoInfo() to create a memo note with the following details for all increase or decrease license O365 orders.   1. Date of Quantity change 2. New Unit Price 3. Previous Unit Price 4. New Quantity 5. Previous Quantity 6. Who submitted the change   Separate table will be created to track all the transaction info with unique capes id for tracking and debugging purposes.  Table name = ams\_product\_trans\_data  Name           Null     Type  -------------- -------- -------------------------------------  CS\_ID NOT NULL Varchar(20)  CTN            NOT NULL VARCHAR2(10)  BAN            NOT NULL VARCHAR2(30)  FAN            NOT NULL VARCHAR2(10)  TRANSACTION\_ID NOT NULL VARCHAR2(15)  SOC\_CODE                VARCHAR2(15)  QUANTITY                VARCHAR2(4)  FORM\_TYPE      NOT NULL VARCHAR2(12)  ACTIVITY NOT NULL VARCHAR(30)  TRANS\_DATE     NOT NULL DATE | | 286365\_OPUS\_SR\_160  286365\_OPUS\_SR\_170  286365\_OPUS\_SR\_180  286365\_OPUS\_SR\_190  286365\_OPUS\_SR\_250  286365\_OPUS\_SR\_300  286365\_OPUS\_SR\_310  286365\_OPUS\_SR\_320  286365\_OPUS\_SR\_340  286365\_OPUS\_SR\_350  286365\_OPUS\_SR\_360 |
| 286365\_OPUS\_OM\_HLD\_11 | OPUS shall have a new enterprise level configurable to handle AMS software products that has to be supported by OPUS. Enterprize config which have pipe seperated list of product families which will be mapped with Category attribute of cpc response.  OPUS – AMS CPC nightly job will download all the data from CPC into opus DB but the frontend cache that loads the data from opus DB will look at this enterprise config to load the cache to the applicable ones. OPUS will cache the product family configured in EC data (earlier downloaded from CPC into opus DB) into ServletContext so only one user request will make a single DB catch to load into FE cache until a server restart. On-demand RC can be used to trigger any cache updates if needed.  Example:  SUPPORTED\_AMS\_PRODUCT\_FAMILY = “Microsoft office 365| other Product Family name”.  For 1602 release the value will be SUPPORTED\_AMS\_PRODUCT\_FAMILY = “Microsoft office 365”.  Note:  The configurable is being built to support onboarding of any new PxQ products in the future identical to Microsoft product structure. | | 286365\_OPUS\_SR\_040 |
| 286365\_OPUS\_OM\_HLD\_12 | Office 365 Product Catalog:  OPUS shall have a nightly job to pull the office 365 catalog details from CPC.  Note:   * This pricing catalog will have standard/RACK pricing for the office 365 products. * OPUS Shall create 3 tables AMS\_PRODUCTS, AMS\_PRODUCTS\_PRICE\_LIST and AMS\_PRODUCTS\_PRICE.   When price changes on an existing soc(product code), CPC will create a new DCM product code. When price changes on existing soc(product code) with tiered pricing, CPC will use a new product code even when adding a new tier or changing price on existing tier. CPC will not leverage modification of pricing on existing product code. This is agreed CPC design.  When customer has a SOC with grand-fathered/expired pricing info from CPC and no new effective pricing was defined for that SOC, modifying the quantity (increase/decrease) should continue to honor the same old price. Currently phoenix queries CPC on demand and if SOC is grandfathered, CPC continues to return the old pricing to be honored. OPUS picks up existing SOC data from DCM and can leverage the same without going to CPC.  NOTE: how to handle expired products or grandfathered products  as OPUS will delete all the data everyday and refresh new data.  CPC returns only active products and will not return any expired or grandfathered products.  Option 1- delete all and refesh new data  Option 2 – find expired products and retain them during refresh  **TABLE** AMS\_PRODUCTS  (  product\_code VARCHAR2(10 ) **NOT** **NULL**,  catalog\_product\_name VARCHAR2(100) **NOT** **NULL**,  product\_description VARCHAR2(250) ,  product\_category VARCHAR2(50 ) **NOT** **NULL**,  product\_sub\_category VARCHAR2(50) **NOT** **NULL**,  efffective\_date Date ,  expiry\_date Date ,  product\_uom VARCHAR2(20) **NOT** **NULL**,  friendly\_name VARCHAR2(250),  product\_id VARCHAR2(10 ) **NOT** **NULL**,  vendor\_name VARCHAR2(100 ),  product\_class VARCHAR2(20 ) **NOT** **NULL**,  pp\_app\_id VARCHAR2(20 ),  dcm\_indicator VARCHAR2(10 ) **NOT** **NULL**,  product\_availability VARCHAR2(10 ) **NOT** **NULL**,  payment\_options VARCHAR2(40 ) **NOT** **NULL**,  sales\_channel VARCHAR2(100 ),  sip\_eligible VARCHAR2(20 ) **NOT** **NULL**,  margin\_required VARCHAR2(20 ),  price\_unit VARCHAR2(20 ),  product\_status VARCHAR2(20 ) **NOT** **NULL**,  product\_type VARCHAR2(20 ),  act\_contact\_form VARCHAR2(100 ),  act\_frequency VARCHAR2(100 ),  deact\_contact\_form VARCHAR2(100 ),  deact\_frequency VARCHAR2(100 ),  qty\_inc\_contact\_form VARCHAR2(100 ),  qty\_inc\_frequency VARCHAR2(100 ),  qty\_dec\_contact\_form VARCHAR2(100 ),  qty\_dec\_frequency VARCHAR2(100 ),  **CONSTRAINT** PK\_AMS\_PRODUCTS **PRIMARY** **KEY** (product\_code)  AMS\_PRODUCTS\_PRICE\_LIST  (  price\_list\_id VARCHAR2(50 ) **NOT** **NULL**,  product\_code VARCHAR2(50 ) **NOT** **NULL**,  price\_list\_name VARCHAR2(100 )**NOT** **NULL**,  country\_code VARCHAR2(2 ) **NOT** **NULL**,  price\_uom VARCHAR2(10 ) **NOT** **NULL**,  effective\_date **DATE** **NOT** **NULL**,  price\_term VARCHAR2(10 ) **NOT** **NULL**,  price\_type VARCHAR2(10 ) **NOT** **NULL**,  currency\_code VARCHAR2(3 ) **NOT** **NULL**,  billing\_system VARCHAR2(40 ) **NOT** **NULL**,  billing\_code VARCHAR2(20 ) **NOT** **NULL**,  apply\_to\_qty VARCHAR2(20 ),  price\_override\_eligible VARCHAR2(1 ),  **CONSTRAINT** PK\_AMS\_PRODUCTS\_PRICE\_LIST **PRIMARY** **KEY** (price\_list\_id,product\_code),  **CONSTRAINT** FK\_AMS\_PRODUCTS\_PRICE\_LIST **FOREIGN** **KEY** (product\_code) **REFERENCES** AMS\_PRODUCTS(product\_code)  AMS\_PRODUCTS\_PRICE  (  price\_list\_id VARCHAR2(10 ) **NOT** **NULL**,  product\_code VARCHAR2(10 ) **NOT** **NULL**,  price\_amount VARCHAR2(10 ),  effective\_date **DATE** **NOT** **NULL**,  min\_qantity VARCHAR2(5 ) **NOT** **NULL**,  max\_qantity VARCHAR2(5 ),  min\_override\_amount VARCHAR2(5 ),  max\_override\_amount VARCHAR2(5 ),  custom\_amount VARCHAR2(10 ),  **CONSTRAINT** PK\_AMS\_PRODUCTS\_PRICE **PRIMARY** **KEY** (price\_list\_id,product\_code, min\_qantity),  **CONSTRAINT** FK\_AMS\_PRODUCTS\_PRICE **FOREIGN** **KEY** (product\_code,price\_list\_id) **REFERENCES** AMS\_PRODUCTS\_PRICE\_LIST(product\_code,price\_list\_id)   * Also for CPC Sorting order OPUS shall create 2 tables   AMS\_CATEGORY and AMS\_SUBCATEGORY  **TABLE** AMS\_CATEGORY  (  AMS\_CATEGORY VARCHAR2(25) **NOT** **NULL**,  DESCRIPTION VARCHAR2(150),  DISPLAY\_SEQUENCE NUMBER,  CREATE\_DATE **DATE**,  CREATE\_BY VARCHAR2(10) ,  UPDATE\_DATE **DATE**,  UPDATE\_BY VARCHAR2(10),  **CONSTRAINT** PK\_AMS\_CATEGORY **PRIMARY** **KEY** (AMS\_CATEGORY)  **TABLE** AMS\_SUBCATEGORY  (  AMS\_SUBCATEGORY VARCHAR2(25) **NOT** **NULL**,  AMS\_CATEGORY VARCHAR2(25) **NOT** **NULL**,  DESCRIPTION VARCHAR2(150),  DISPLAY\_SEQUENCE NUMBER,  CREATE\_DATE **DATE**,  CREATE\_BY VARCHAR2(10) ,  UPDATE\_DATE **DATE**,  UPDATE\_BY VARCHAR2(10),  **CONSTRAINT** PK\_AMS\_SUBCATEGORY **PRIMARY** **KEY** (AMS\_SUBCATEGORY),  **CONSTRAINT** FK\_AMS\_SUBCATEGORY **FOREIGN** **KEY** (AMS\_CATEGORY) **REFERENCES** AMS\_CATEGORY(AMS\_CATEGORY)  GeneralEventProcessorMDBean will invoke POS2OPUSAMSProductsServicesManagerJMSQueue which in turn will call AMSProductsServicesManagerEvent.event method.  Implement bean method to get nightly feed from cpc and insert values with stored procedure call. | | 286365\_OPUS\_SR\_090 |
| 286365\_OPUS\_OM\_HLD\_13 | Receipts(Printed and emailed):  If a customer purchases MO365 product after adding any item to the cart (DF or CC), OPUS shall be enhanced to be able to print and email receipt with MO 365 product detail after the transaction is completed.  Note:   * MO 365 is similar to a feature. Today, receipt is printed for feature changes, only if the feature is added after any other item is added to the cart. (Refer [A-4](#_Assumptions))   For email receipt changes, there is no impact to EDD/EGS. OPUS shall concatenate the text, for example, “Office 365 share point add Q5” or “Office 365 Exchange remove Q4” in description column of the email.  Note : Receipt Generation logic will not be applicable for NR and LD since no request item will be added to the cart.  Below classes and method shall be enhanced to handle receipt and prtint functionalities   * **LineItemHandler.java** – This is an abstract interface. Need to define abstract methods here for handling AMSProductsRequest objects.   + public void handleObject(AMSProductsRequest object) throws HandlerException; * **LineItemAdapter.java** – This class implements the above interface. So we need to define the above methods here. The implementation is left to the classes that extend this class (such as CartHandler.java).   + public void handleObject(AMSProductsRequest object) throws HandlerException;   public void   * **CartHandler.java** –   + New handleObject Method which will handle the AMSProductsRequest object:     - Use the public void handleObject(AMSProductsRequest request)   throws HandlerException as a guideline in implementing the details for this method. | | 286365\_OPUS\_SR\_365 |

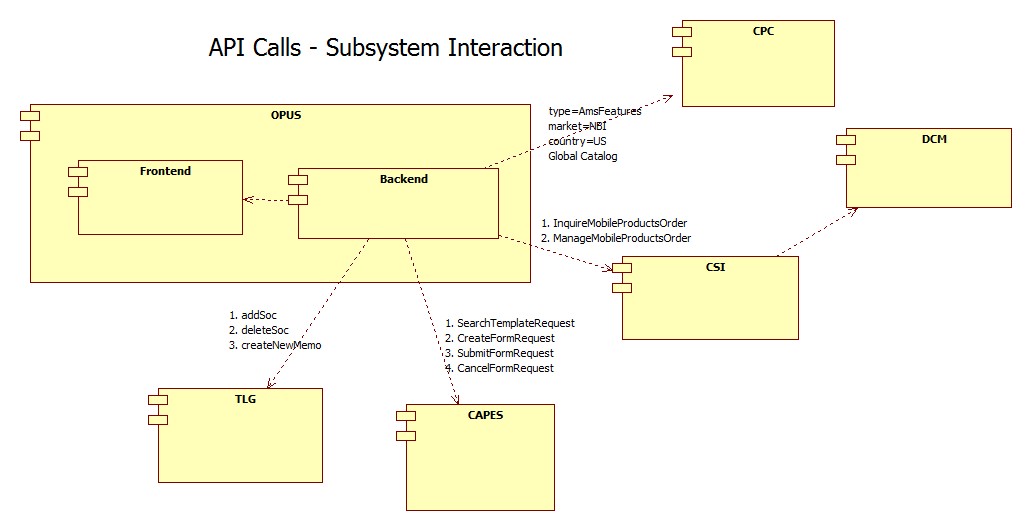
The [SRT\_RM\_ReqPro\_Auto\_Tagging](https://cps.web.att.com/CPSWorkplace/getContent?id=current&vsId=%7B67070CE6-CDED-4932-8F1D-F6073BA40DF8%7D&objectStoreName=IT-Architecture.__.Planning.__.and.__.Integration&objectType=document&guestId=servicesguest) Reference Document provides information on how to use auto-tagging within SRT ReqPro.

If the table is not used to capture requirements, please remove it from document.

## Functional Overview

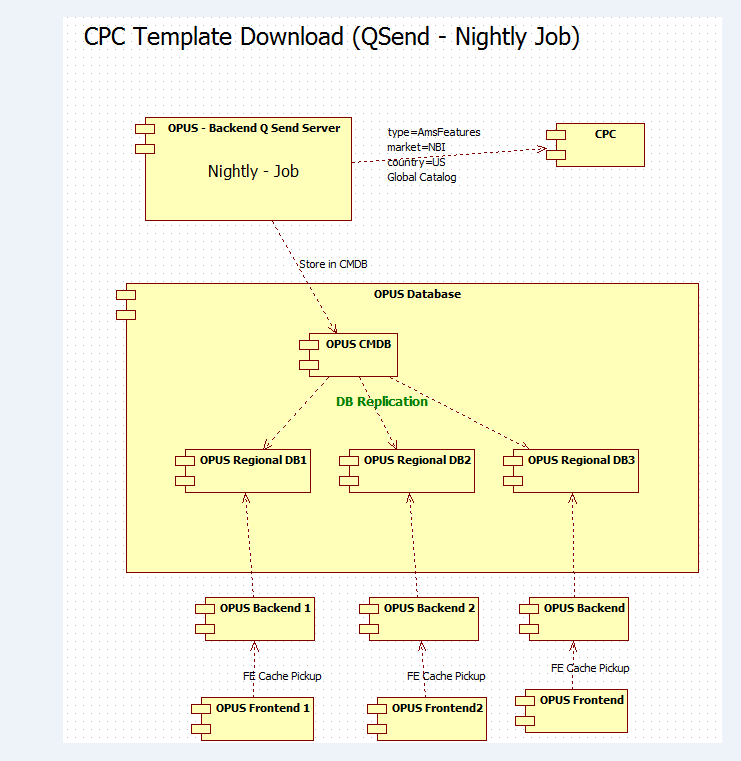
|  |  |  |
| --- | --- | --- |
| **Module Name** | **New/Existing** | **Functional description** |
|  |  |  |

## Middleware Design

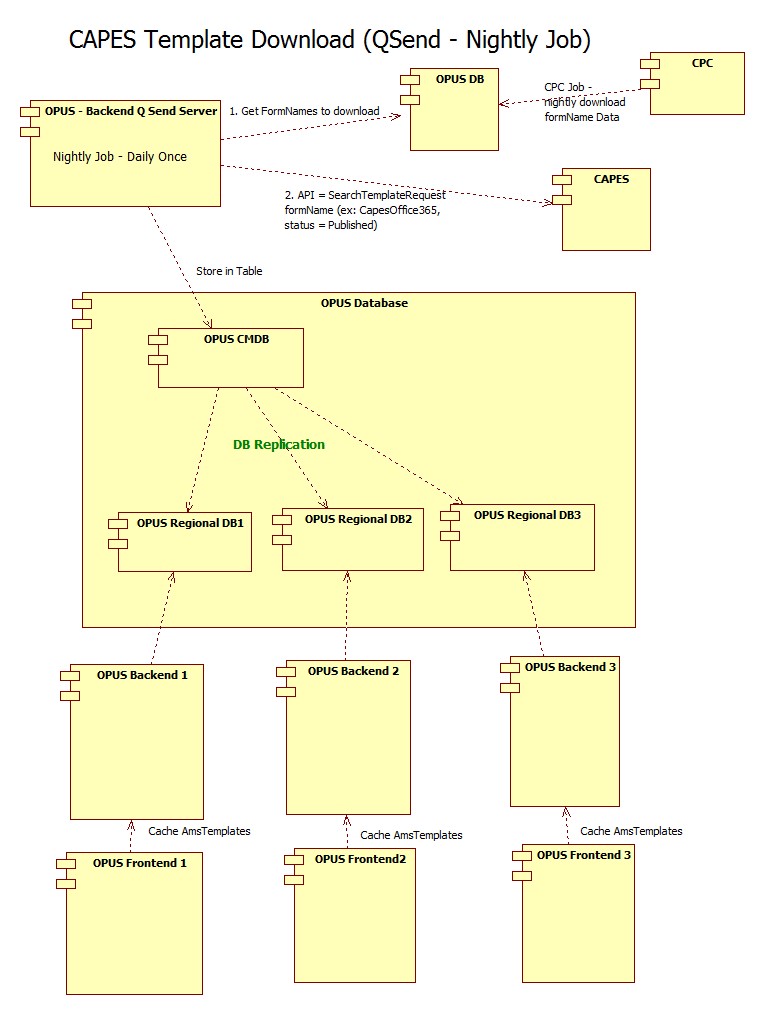


**External Interfaces**

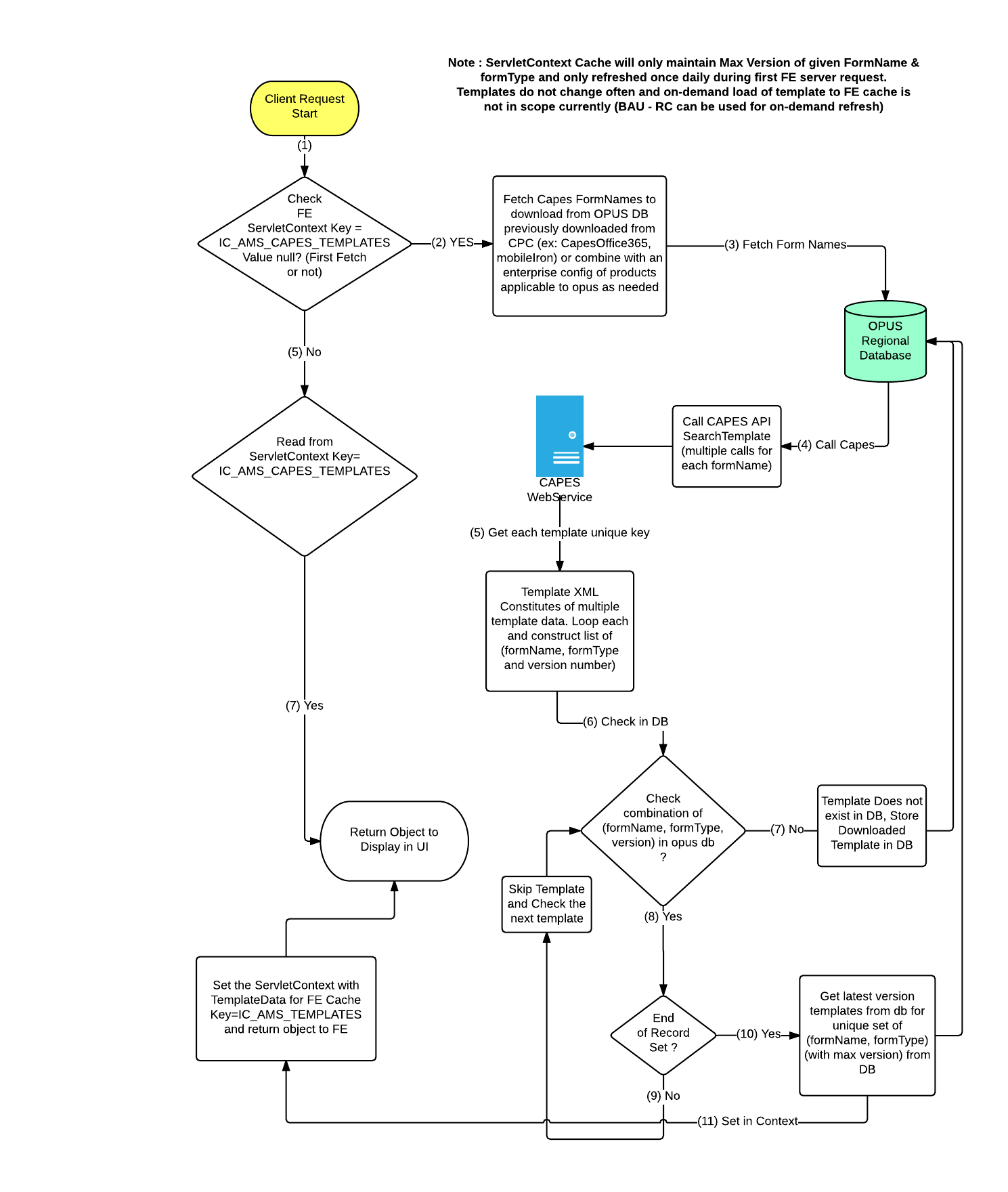
High Level - Component Diagram for CPC Catalog Download (Nightly Job):

****

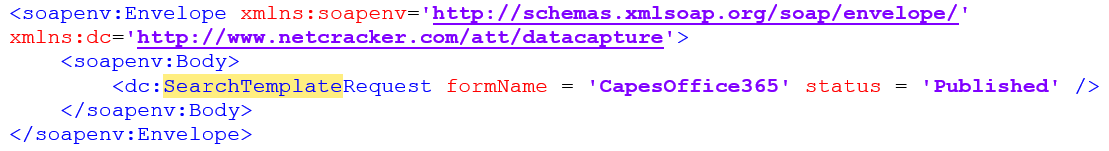
High Level - Component Diagram for CAPES Template Download (Nightly Job):



Process Flow diagram for CAPES – SEARCH TEMPLATE DOWNLOAD



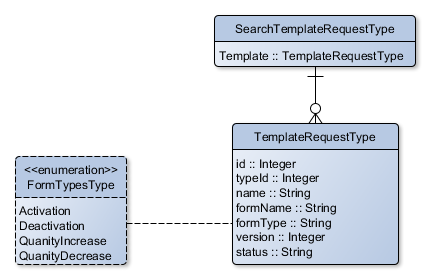
**Sample SearchTemplateRequest:**

****

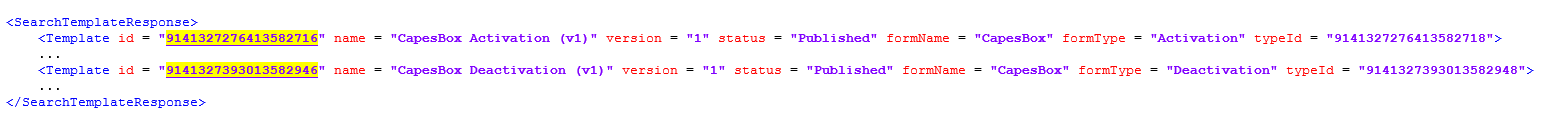
**CAPES will return most latest status=published templates for the formName=”CapesOffice365”**

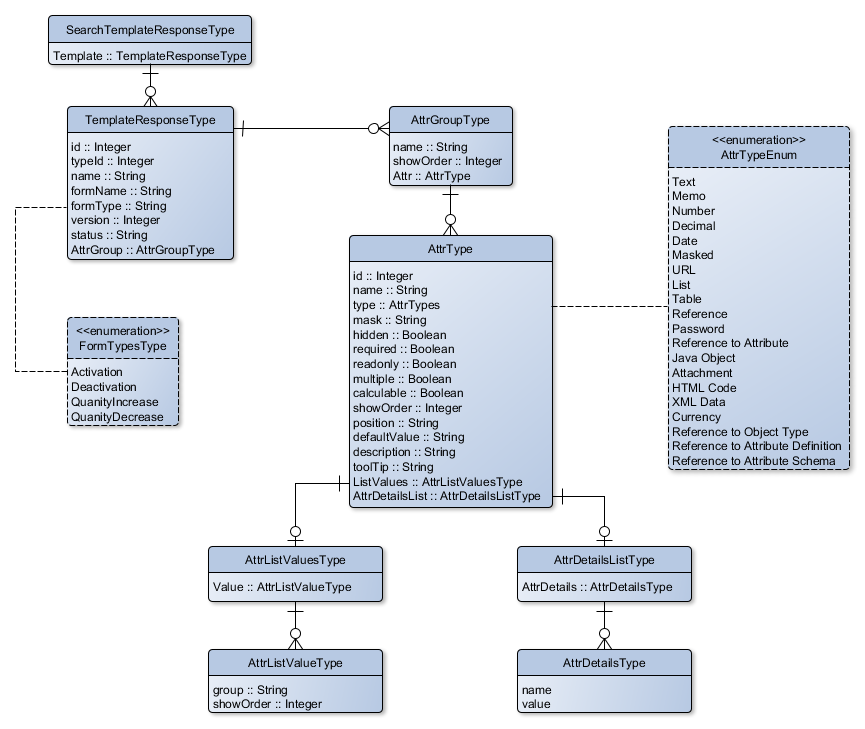
**So lets say 4 formTypes are present, we will get response containing 4 templates but most latest.**

**CAPES will have a Unique Attr Id across all formNames and formTypes. For example: Seller ATT ID field will have a unique Attr Id across all systems (all formNames and formTypes)**



**Sample SearchTemplateResponse :**





Multiple Templates in the same response.

**Sample SOAP Reponse**

<soapenv:Envelope xmlns:soapenv=**"http://schemas.xmlsoap.org/soap/envelope/"** xmlns:dc=**"http://www.netcracker.com/att/datacapture"**>

<soapenv:Body>

<dc:SearchTemplateResponse resultsNumber = **"2"**>

<Template id = **'9142911342313134617'** name = **'CapesOffice365 Activation (v1)'** version = **'1'** status = **'Published'** formName = **'CapesOffice365'** formType = **'Activation'** typeId = **'9142911342313134619'**>

<AttrGroup name = **'Office365 Information'** showOrder = **'0'**>

<Attr id = **'9142911342313134623'** name = **'Admin Name'** type = **'Text'** required = **'false'** showOrder = **'0'** position = **'L'** readonly = **'false'** />

<Attr id = **'9142911342313134628'** name = **'Admin Email Address'** type = **'Text'** required = **'true'** showOrder = **'1'** position = **'L'** readonly = **'false'** readonly = **'false'** />

<Attr id = **'9142911342313134632'** name = **'Billing Address'** type = **'Text'** required = **'false'** showOrder = **'2'** position = **'L'** readonly = **'false'** />

<Attr id = **'9142912365013135068'** name = **'Desired IP Address'** type = **'Text'** required = **'false'** showOrder = **'3'** position = **'L'** readonly = **'false'** />

<Attr id = **'9142912365013135068'** name = **'Desired Domain Name'** type = **'Text'** required = **'false'** showOrder = **'4'** position = **'L'** readonly = **'false'** />

</AttrGroup>

</Template>

<Template id = **'9142911342313134618'** name = **'CapesOffice365 Deactivation (v2)'** version = **'2'** status = **'Published'** formName = **'CapesOffice365'** formType = **'Deactivation'** typeId = **'9142911342313134620'**>

<AttrGroup name = **'Office365 Information'** showOrder = **'0'**>

<Attr id = **'9142911342313134623'** name = **'Admin Name'** type = **'Text'** required = **'false'** showOrder = **'0'** position = **'L'** readonly = **'false'** />

<Attr id = **'9142911342313134628'** name = **'Admin Email Address'** type = **'Text'** required = **'true'** showOrder = **'1'** position = **'L'** readonly = **'false'** readonly = **'false'** />

<Attr id = **'9142911342313134632'** name = **'Billing Address'** type = **'Text'** required = **'false'** showOrder = **'2'** position = **'L'** readonly = **'false'** />

<Attr id = **'9142912365013135068'** name = **'Desired IP Address'** type = **'Text'** required = **'false'** showOrder = **'3'** position = **'L'** readonly = **'false'** />

<Attr id = **'9142912365013135068'** name = **'Desired Domain Name'** type = **'Text'** required = **'false'** showOrder = **'4'** position = **'L'** readonly = **'false'** />

<Attr id = **'9142912365013135068'** name = **'Desired Domain Name (new)'** type = **'Text'** required = **'false'** showOrder = **'5'** position = **'L'** readonly = **'false'** />

</AttrGroup>

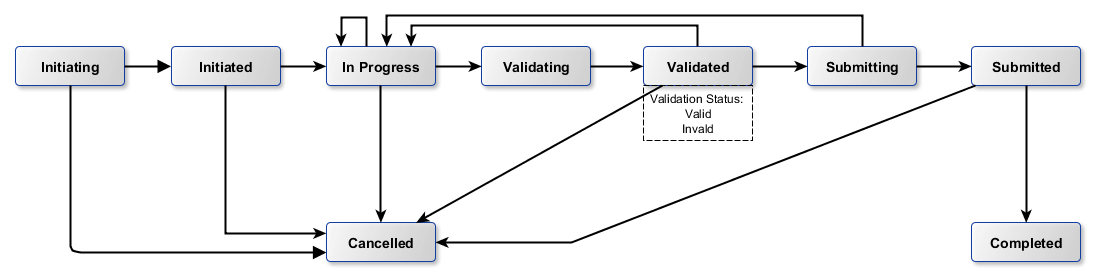
</Template>

</dc:SearchTemplateResponse>

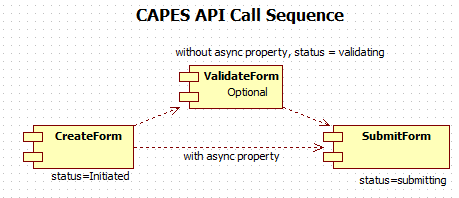
</soapenv:Body>

</soapenv:Envelope>

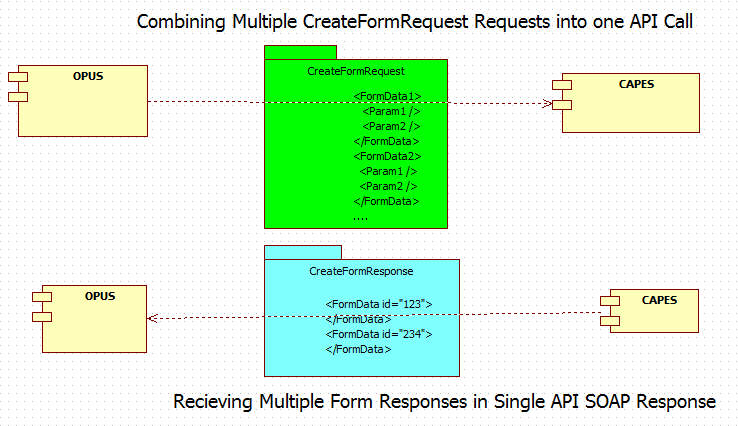
**Internal States of CAPES form flow with-in the CAPES system (Note: OPUS will only use states from Status=Initiated => Status = Validating => status = Submitting (without async property) or Status = Initiated => Status = Submitting (with async property))**



**OPUS – CAPES Calls -**



**OPUS->CAPES Combining multiple requests into one call**

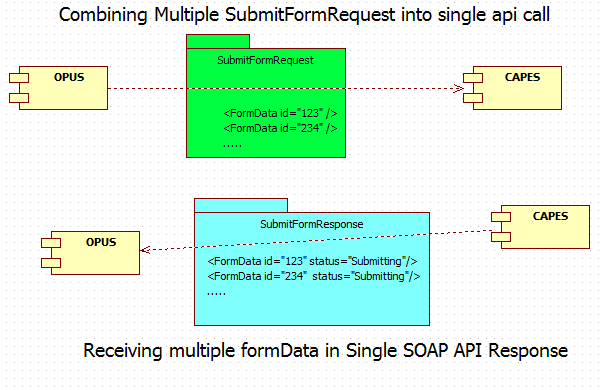


**Sample - Combined SOAP Request for CreateFormRequest API Call**

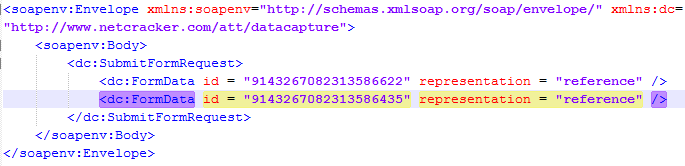


**Sample - Combined SOAP Response - CreateFormResponse**





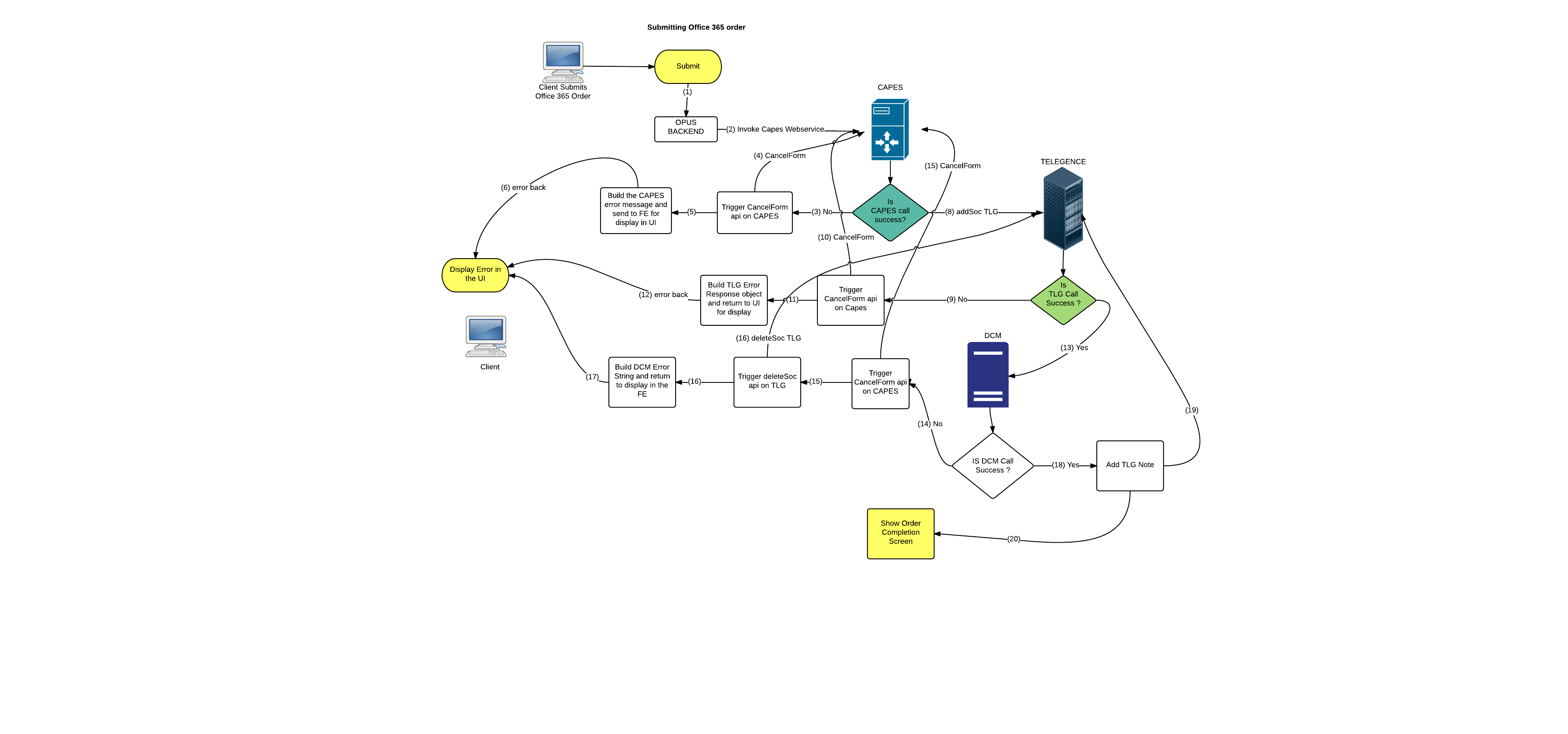
**Sample - Combined SOAP Request for SubmitFormRequest API Call**



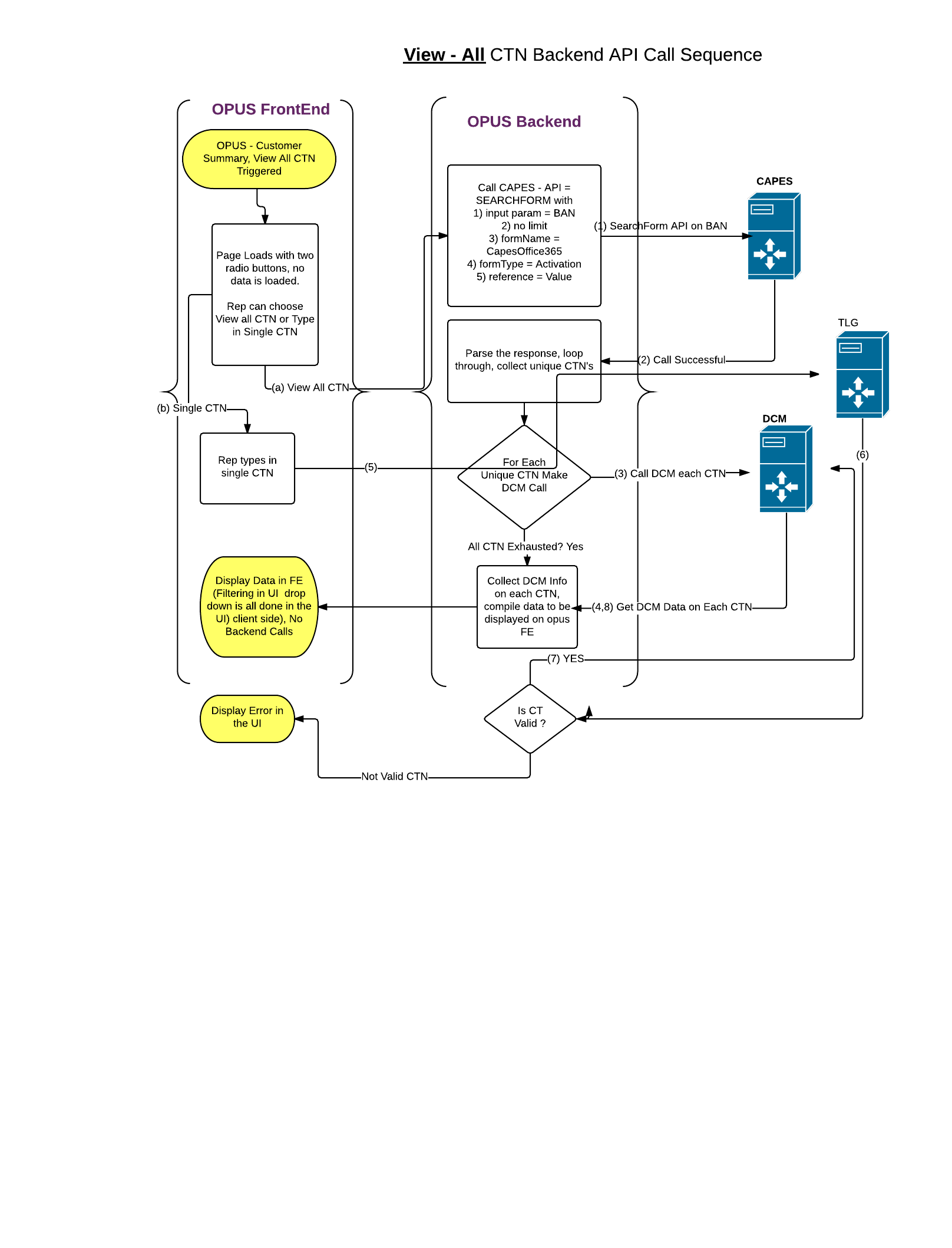
**Sample - Combined SOAP Response - SubmitFormResponse**



**AMS – Office365 Products : Order Submission and Rollback**



**View All CTN – Backend API Call sequence**

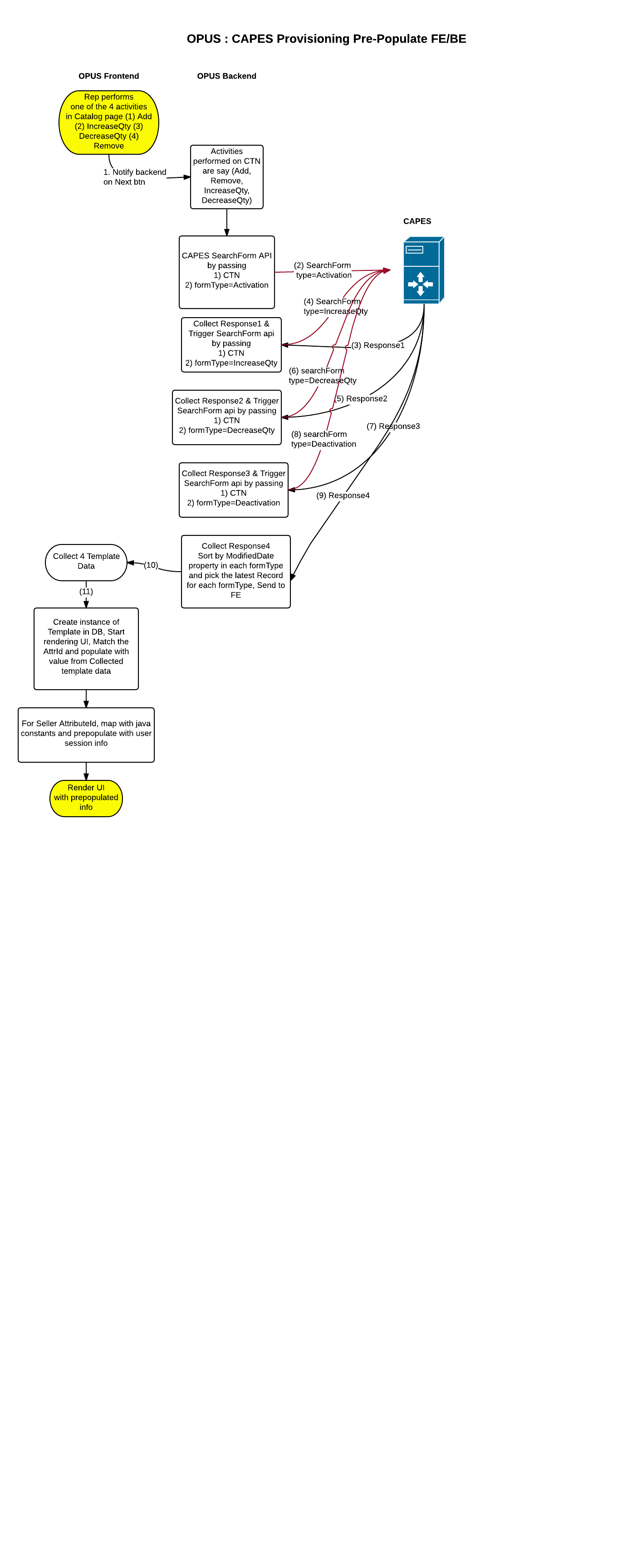


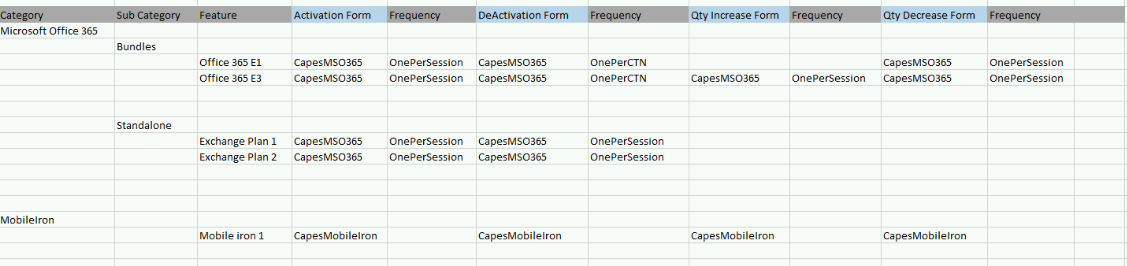
**DESIGN ALERT (AWARENESS DISCLAIMER)**: Since CPC defines which CAPES form is needed for a particular flow, and CPC can make some CAPES forms optional, it is possible to have some CTN’s not having forms filled out and in this case, DCM might have all the data but CAPES may not. This is a possibility and a design alert RISK at this stage.

Plus there is a Performance degredation possibility because we go to CAPES asking for all form data including one field (CTN) across duplicate set of form data and CAPES action given.

Calling DCM on each unique CTN returned might be time consuming as well.

**Prepopulate CAPES Provision Info (without lastModified=”true” and FormData Nesting, Requires 4 OPUS-CAPES api calls.)**



****

**Driving CPC + CAPES Template Integration:**

Note 3: CPC does not have variable formtypes beyond the predefined 4. CPC/CAPES are not dynamic but have 4 predefined formTypes (Activation/Deactivation/QuantityIncrease/QuantityDecrease) today. Any addition of new formType beyond this will be code change/new requirement in all three systems including OPUS.

Note4: OPUS will not use the Frequency values from the CPC (or CPC is enforced to one value always) and will always leverage the form\_name field to determine if a provision form is needed or not based on a SOC activity, When form is configured in CPC, OPUS will always require one/multiple forms to be filled per each order flow.

Note 5: CPC nightly job will be scheduled first prior to CAPES template nightly job.

Note 6: OPUS will hard-code form statuses internally (ex: Status=Initiated, status=Published, status=Submitted etc.). CAPES confirmed that these values will not change. CPC has no data to drive these form statuses today.

Note 7: OPUS will check the user SOC related activity in catalog page and determine one of the 4 actions (add/increase/decrease/remove) and map it against CPC to determine if a provision FORM is required on the user activity. Based on the mapping, opus will determine what data needs to be pulled from CAPES for prepopulation. When no forms are configured in CPC for SOC Activity, CAPES call will not be required for prefill as no provision form is rendered. No form configuration in CPC might also impact View ALL CTN since for option to view all CTN we go to CAPES to determine unique CTNs that have products which DCM might have but CAPES might not.

Note 8: In ordering flow, If rep is adding 3 soc’s and if CPC is configured to require Activation form on the first 2 SOC’s but is configured to not require Activation form (form is optional, blank), then OPUS will determine using an OR condition that we will require a Activation form in that flow. Needing to fill in the provision form takes precedence over not needing to fill a form and filling up a provision form fill up should ideally be not tied to soc but in this case it is.

Note 9 : CAPES Template download job will first look at opus db (previously downloaded CPC data) to pickup unique form names from that list and then go to CAPES to download those uniquely configured forms on the 4 Activities (refer table above).

Note 10: CAPES agreed to send a new attribute/property to indicate “donotcopy”/”donotprepopulate” for each render element in SearchForm API. ETA awaited for confirmation.

Note 11: Code changes are required in the case there are existing templates with new attributes or new templates with new attributes.

Note 12: When new products are onboarded, testing will be required prior to launch in production.

**Performance Design**

NA

### 

### Security Design

*Create an inventory of all SPI/PCI data elements that will require special handling with encryption and watermarking at each architecture layer.*

* **NA**

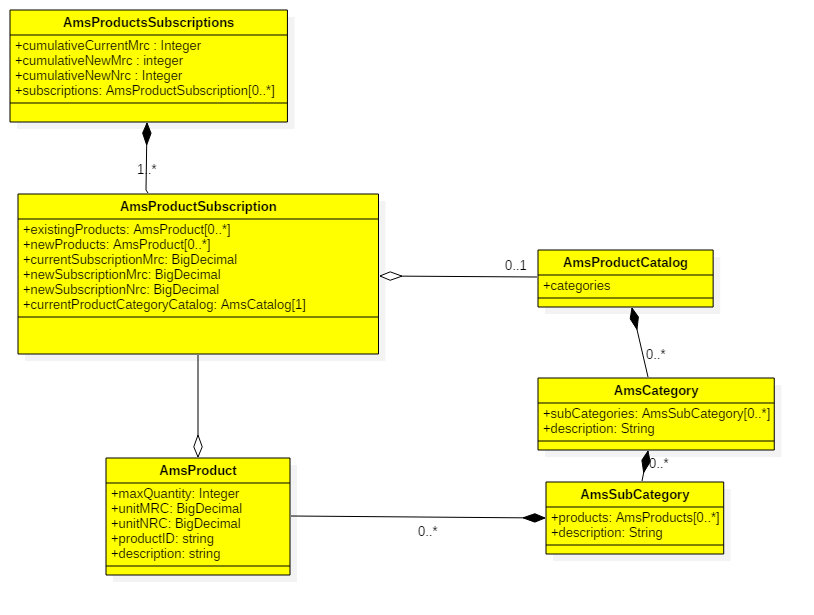
|  |  |  |
| --- | --- | --- |
| **SPI/PCI Data Element Inventory** | | |
| **Screen Elements** | | |
| **#** | **Element Name** | **Handling** |
| *1* | *NA* | *NA* |
|  |  |  |
| **External Interface Elements (Service/EJB calls)** | | |
| **#** | **Element Name** | **Handling** |
| 1 | *NA* | *NA* |
|  |  |  |
| **Database Elements** | | |
| **#** | **Element Name** | **Handling** |
| *1* | *NA* | *NA* |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Watermarking Inventory** | | | |
| **#** | **SPI/PCI Entity/Attribute Name** | **Logging Injection Point** | **Watermarking utility** |
| 1 | NA | NA | NA |

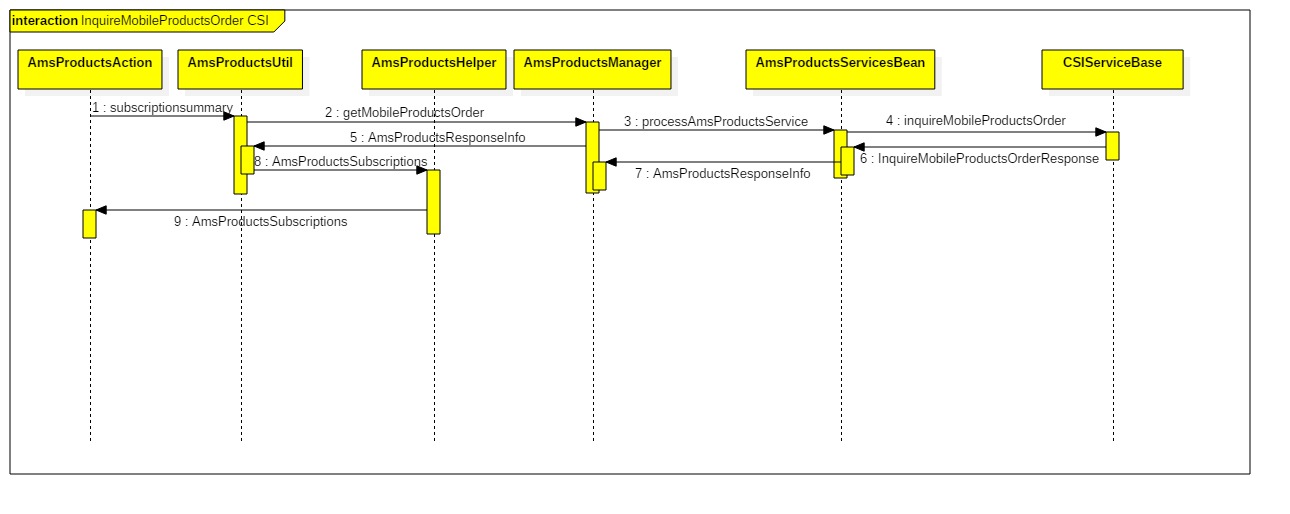
### UI Design

### UI Design - Wireframe

### UI Design - Screen Element Definition and Data Mapping

****

### UI Action Processing - Sequence Diagram

**

### Asynchronous/Batch Processing - Sequence Diagram

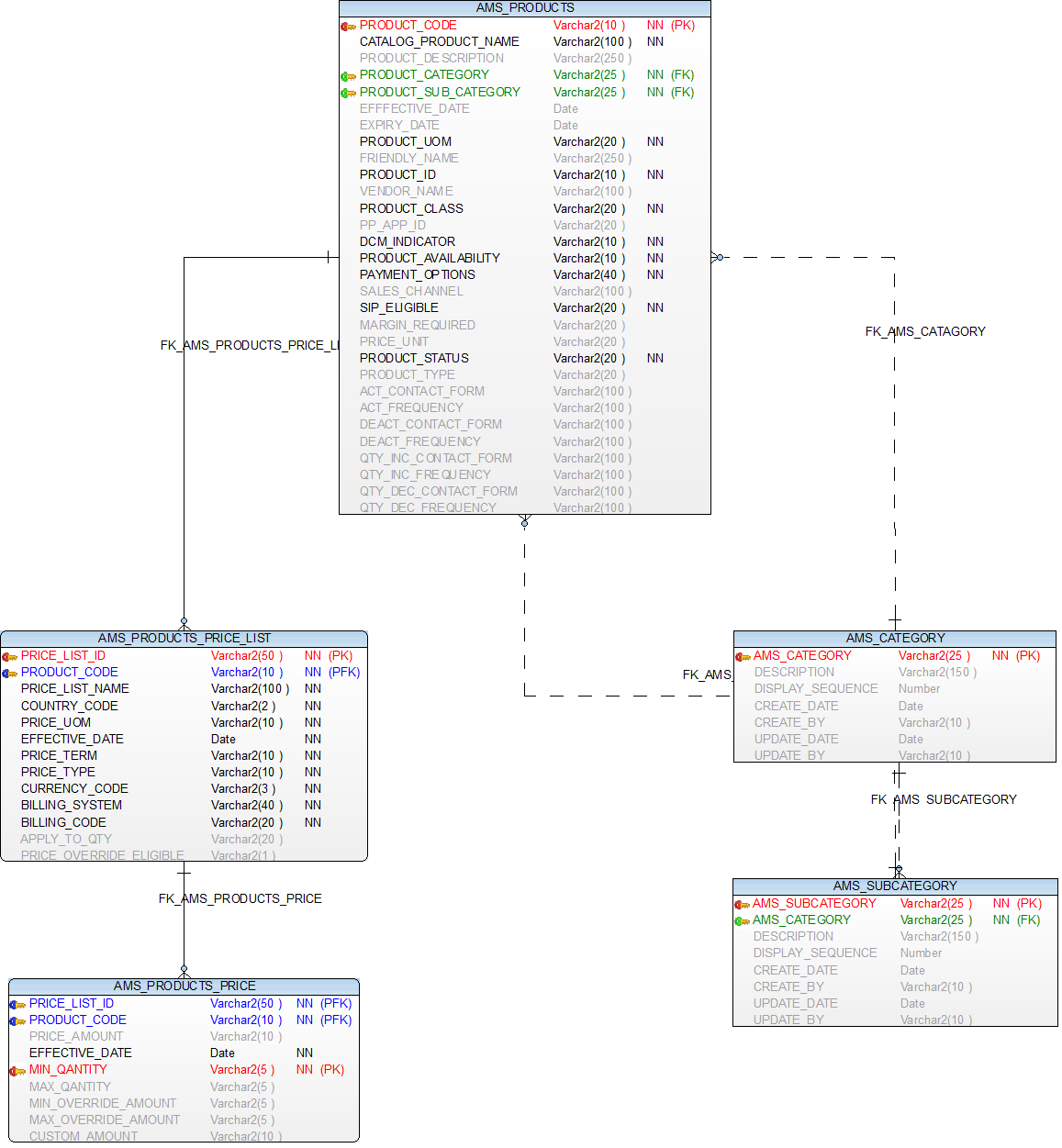
*NA*

**Externalized Configuration**

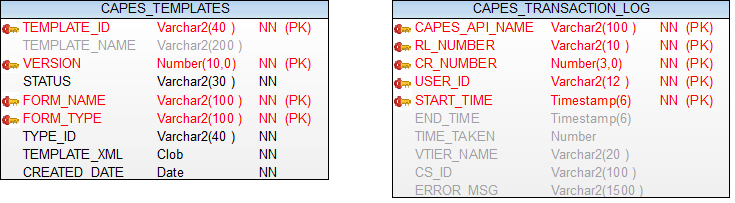
*All application configuration parameters (table filled out with examples).*

|  |  |  |
| --- | --- | --- |
| **Application Configuration Parameters** | | |
| **Parameter Type** | **Parameter name** | **Description** |
| Store Level configuration | 286365\_AMS\_PRODUCTS | This store configurations provide ability to turn on/off project level changes for Office 365 project. **Default value = N** |
| Enterprise Level Configuration | SUPPORTED\_AMS\_PRODUCT\_FAMILY | Applicable product families for AMS products that opus will use. CPC will have lot more product families but OPUS will only use this PIPE separated values to support the product families. Example values are SUPPORTED\_AMS\_PRODUCT\_FAMILY = “Microsoft| other vendor name” when supported. |
| Enterprise Level Configuration |  |  |

## Database Tier Design

**OPUS – AMS : CPC Downloaded Data - Table Structure  
**

**OPUS : CAPES Tables**

****

### Logical Data Model

|  |  |  |
| --- | --- | --- |
| **Change Summary** | | |
| **Entity Name** | **Change type** | **Change Description** |
| AMS\_PRODUCTS | New Table | Below columns will be added |
|  |  |  |
|  |  |  |

product\_code VARCHAR2(10 ) **NOT** **NULL**,

catalog\_product\_name VARCHAR2(100) **NOT** **NULL**,

product\_description VARCHAR2(250) ,

product\_category VARCHAR2(50 ) **NOT** **NULL**,

product\_sub\_category VARCHAR2(50) **NOT** **NULL**,

efffective\_date Date ,

expiry\_date Date ,

product\_uom VARCHAR2(20) **NOT** **NULL**,

friendly\_name VARCHAR2(250),

product\_id VARCHAR2(10 ) **NOT** **NULL**,

vendor\_name VARCHAR2(100 ),

product\_class VARCHAR2(20 ) **NOT** **NULL**,

pp\_app\_id VARCHAR2(20 ),

dcm\_indicator VARCHAR2(10 ) **NOT** **NULL**,

product\_availability VARCHAR2(10 ) **NOT** **NULL**,

payment\_options VARCHAR2(40 ) **NOT** **NULL**,

sales\_channel VARCHAR2(100 ),

sip\_eligible VARCHAR2(20 ) **NOT** **NULL**,

margin\_required VARCHAR2(20 ),

price\_unit VARCHAR2(20 ),

product\_status VARCHAR2(20 ) **NOT** **NULL**,

product\_type VARCHAR2(20 ),

act\_contact\_form VARCHAR2(100 ),

act\_frequency VARCHAR2(100 ),

deact\_contact\_form VARCHAR2(100 ),

deact\_frequency VARCHAR2(100 ),

qty\_inc\_contact\_form VARCHAR2(100 ),

qty\_inc\_frequency VARCHAR2(100 ),

qty\_dec\_contact\_form VARCHAR2(100 ),

qty\_dec\_frequency VARCHAR2(100 )

|  |  |  |
| --- | --- | --- |
| **Change Summary** | | |
| **Entity Name** | **Change type** | **Change Description** |
| AMS\_PRODUCTS\_PRICE\_LIST | New Table | Below columns will be added |
|  |  |  |
|  |  |  |

price\_list\_id VARCHAR2(50 ) **NOT** **NULL**,

product\_code VARCHAR2(50 ) **NOT** **NULL**,

price\_list\_name VARCHAR2(100 )**NOT** **NULL**,

country\_code VARCHAR2(2 ) **NOT** **NULL**,

price\_uom VARCHAR2(10 ) **NOT** **NULL**,

effective\_date **DATE** **NOT** **NULL**,

price\_term VARCHAR2(10 ) **NOT** **NULL**,

price\_type VARCHAR2(10 ) **NOT** **NULL**,

currency\_code VARCHAR2(3 ) **NOT** **NULL**,

billing\_system VARCHAR2(40 ) **NOT** **NULL**,

billing\_code VARCHAR2(20 ) **NOT** **NULL**,

apply\_to\_qty VARCHAR2(20 ),

price\_override\_eligible VARCHAR2(1 )

|  |  |  |
| --- | --- | --- |
| **Change Summary** | | |
| **Entity Name** | **Change type** | **Change Description** |
| AMS\_PRODUCTS\_PRICE | New Table | Below columns will be added |
|  |  |  |
|  |  |  |

price\_list\_id VARCHAR2(10 ) **NOT** **NULL**,

product\_code VARCHAR2(10 ) **NOT** **NULL**,

price\_amount VARCHAR2(10 ),

effective\_date **DATE** **NOT** **NULL**,

min\_qantity VARCHAR2(5 ) **NOT** **NULL**,

max\_qantity VARCHAR2(5 ),

min\_override\_amount VARCHAR2(5 ),

max\_override\_amount VARCHAR2(5 ),

custom\_amount VARCHAR2(10 )

|  |  |  |
| --- | --- | --- |
| **Change Summary** | | |
| **Entity Name** | **Change type** | **Change Description** |
| AMS\_CATEGORY | New Table | Below columns will be added |
|  |  |  |
|  |  |  |

AMS\_CATEGORY VARCHAR2(25) **NOT** **NULL**,

DESCRIPTION VARCHAR2(150),

DISPLAY\_SEQUENCE NUMBER,

CREATE\_DATE **DATE**,

CREATE\_BY VARCHAR2(10) ,

UPDATE\_DATE **DATE**,

UPDATE\_BY VARCHAR2(10)

|  |  |  |
| --- | --- | --- |
| **Change Summary** | | |
| **Entity Name** | **Change type** | **Change Description** |
| AMS\_SUBCATEGORY | New Table | Below columns will be added |
|  |  |  |
|  |  |  |

AMS\_SUBCATEGORY VARCHAR2(25) **NOT** **NULL**,

AMS\_CATEGORY VARCHAR2(25) **NOT** **NULL**,

DESCRIPTION VARCHAR2(150),

DISPLAY\_SEQUENCE NUMBER,

CREATE\_DATE **DATE**,

CREATE\_BY VARCHAR2(10) ,

UPDATE\_DATE **DATE**,

UPDATE\_BY VARCHAR2(10)

***STORE CONFIGURABLE UPDATE:***

**ALTER** **TABLE** STORE\_CONFIGURABLE **ADD** ( PURPOSE **VARCHAR**(100));

COMMENT ON COLUMN STORE\_CONFIGURABLE.PURPOSE

IS 'Possible values are Functional, Deployment Management,null';

**ALTER** **TABLE** STORE\_CONFIGURABLE **ADD** ( TIME\_UNTIL\_ACTIVE **DATE**);

COMMENT ON COLUMN STORE\_CONFIGURABLE.TIME\_UNTIL\_ATIVE

IS ’Date until this configurable is active';

*This table tracks the order data with in opus.*

Table name = ams\_product\_trans\_data

Name           Null     Type

-------------- -------- -------------------------------------

CS\_ID NOT NULL Varchar(20)

CTN            NOT NULL VARCHAR2(10)

BAN            NOT NULL VARCHAR2(30)

FAN            NOT NULL VARCHAR2(10)

TRANSACTION\_ID NOT NULL VARCHAR2(15)

SOC\_CODE                VARCHAR2(15)

QUANTITY                VARCHAR2(4)

FORM\_TYPE      NOT NULL VARCHAR2(12)

ACTIVITY NOT NULL VARCHAR(30)

TRANS\_DATE     NOT NULL DATE

### Database Interfaces

* *NA*

### Batch Interfaces

*NA*

### Replication

*NA*

## Alternative Designs

## Assumptions/Risks

## Some Performance concerns around prepopulation/view all ctn as we don’t have direct APIs available.

## Pre-Production Disaster Recovery Planning

### NA

## Other Plans and References

NA

|  |
| --- |
|  |

## 

## Acceptance & Approvals

Overview

Use this section to capture approvals in the event that electronic approvals via the PRISM Project Workflow Module will not be used.

The Approvers of this work product agree that this document is acceptable and complete to the best of their knowledge and will be used by the project team as an official deliverable for the project. It is further agreed that this document can now be baselined and any changes to these sections from this point forward must follow the Managing Change in the Technology Development Unified Process.

Embed evidence of approval in the review table below, or use the PRISM Approval Functionality in the Project Workflow Module Workflow Template View.

Approvers

|  |  |  |  |
| --- | --- | --- | --- |
| **ATTUID and Name** | **Role** | **Group/Application** | **Version Approved, Approval Date and Approval Evidence** |
| Thakrar, Praveen | PTA | OPUS / OM |  |
| Moturu, Rama | Test Lead | OPUS / OM |  |
| Pabolu, Siva | Retail Functional Architecture | OPUS / OM |  |
| Gunna, Moapal | STD | OPUS / OM |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |