



CX SDD

Version 0.3

December 4, 2017

Private & Confidential

©2017 IBM. All rights reserved. This document is considered private and confidential and is solely intended for internal use only. Copyright materials include, but are not limited to source code, images, icons, text and other material. This material may not be reproduced, displayed, modified, or distributed with the express prior written permission of IBM.

1 Table of Contents

1	Introduction.....	1
1.1	Purpose	1
1.2	Audience.....	1
1.3	Assumption	1
1.4	References	1
1.5	Terms, Acronyms and Definitions.....	2
2	CX Architecture Overview.....	4
2.1	SmapiController	6
2.2	CxService	7
2.3	CleengAdaptor	8
2.4	CxJwt.....	8
2.5	CxDb	8
2.6	CxSecurity.....	8
3	CX Data Domain	9
3.1	Provider Table	10
3.2	Subscriber Table	11
3.3	Subscription Table.....	12
4	CX SMAPI Implementation	13
4.1	SMAPI Information.....	13
4.2	loginSubscriberIdentity	14
4.2.1	Request.....	14
4.2.2	Parameters.....	14
4.2.3	Body	14
4.2.4	Response.....	14
4.2.5	Example	15
4.2.6	Sequential Diagram.....	16
4.3	getSubscriberAccount.....	17
4.3.1	Request.....	17
4.3.2	Parameters.....	17
4.3.3	Body	17

4.3.4	Response.....	17
4.3.5	Example	19
4.3.6	Sequential Diagram.....	21
4.4	getExternalBillingInfo	22
4.4.1	Request.....	22
4.4.2	Parameters.....	22
4.4.3	Body	22
4.4.4	Response.....	22
4.4.5	Example	23
4.4.6	Sequential Diagram.....	24
4.5	updateSubscriberAccount.....	25
4.5.1	Request.....	25
4.5.2	Parameters.....	25
4.5.3	Body	25
4.5.4	Response.....	26
4.5.5	Example	26
4.5.6	Sequential Diagram.....	28
4.6	authorize	29
4.6.1	Request.....	29
4.6.2	Parameters.....	29
4.6.3	Body	29
4.6.4	Response.....	29
4.6.5	Example	30
4.6.6	Sequential Diagram.....	31
4.7	refreshSubscriberStatus.....	32
4.7.1	Parameters.....	32
4.7.2	Returns.....	32
4.7.3	Sequential Diagram.....	32
5	LoginRadius SSO	34
5.1	JWT Payload Format.....	35
6	Development Guidelines.....	37
6.1	IDE and Tools.....	37

6.2	Frameworks.....	37
6.3	Team Workflow	38

Revision History

Version	Date	Modified	Who
0.1	12/1/2017	Initial Release.	Van Nguyen
0.2	12/3/2017	Completed Release.	Van Nguyen

1 Introduction

1.1 Purpose

This software design document describes the blueprint of the CX Application and outlines the steps for building the CX Application.

1.2 Audience

The main intended audiences for the documents are:

- Product Teams
- Engineering Teams
- Testing Teams
- Sale Engineering Teams
- Architect Teams
- Project and Program Managers

1.3 Assumption

The CX Application design is based on the user stories and high-level design from the CBC 3.0 System HLD vo.4.pdf.

1.4 References

This document is depended on these documents:

1. CBC 3.0 System HLD vo.4.pdf
2. Clearleap Subscriber Management API 1.40.pdf
3. Clearleap Client Spec 1.37.6.pdf

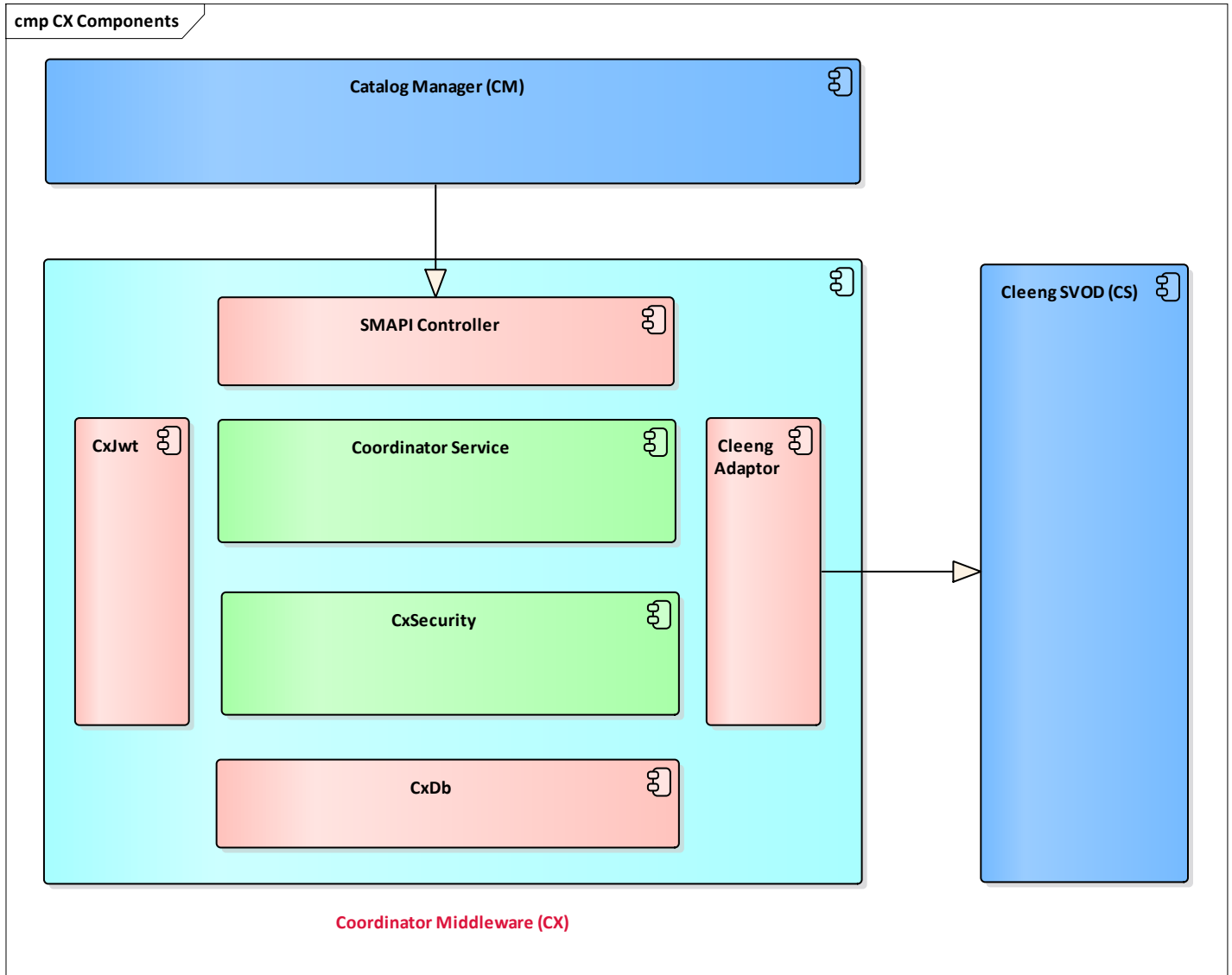
1.5 Terms, Acronyms and Definitions

Name	Description
ICV	IBM Cloud Video Platform
IAP	In-app Purchase
Content Provider	An IBM Cloud Video client who ingests, processes and delivers video content to one or more locations using the IBM Cloud Video SaaS platform.
Content Distributor	An IBM Cloud Video client who receives video content from the IBM Cloud Video SaaS platform and, in turn, distributes that content to subscribers.
Subscriber	A consumer who has signed up and established a perpetual subscription for a video streaming service by successfully creating a Subscriber Account with a Payment Method.
Subscriber Manager (SM)	Responsible for managing subscriber accounts, subscriptions, and recurring billings and payments. Formerly known as TVE.
Catalog Manager (CM)	The online video platform providing OTT streaming service for the CBC solution. Formerly known as Clearplay.
Coordinator Middleware (CX)	A lightweight middleware providing an integration layer between Cleeng SVOD and Catalog Manager.
Cleeng SVOD (CS)	A video e-commerce platform providing a wide range of advanced features for managing users, entitlements and accepting payments.
CBC 1.0 System	The anonymous service launched in 2015 for providing AVOD service for anonymous users.
CBC 2.0 System	The SVOD service planning to launch in Dec 2017 with Apple in-app purchasing and IBM Subscriber Manager.
CBC 3.0 System	The new enhanced SVOD system with Android IAP and web billing using Cleeng.
Standard User	A completely anonymous user.
Member User	A free authenticated user. This user has logged into the service but has not provided a payment method.
Premium User	A paid user via either IAP, credit card, or voucher/coupon.

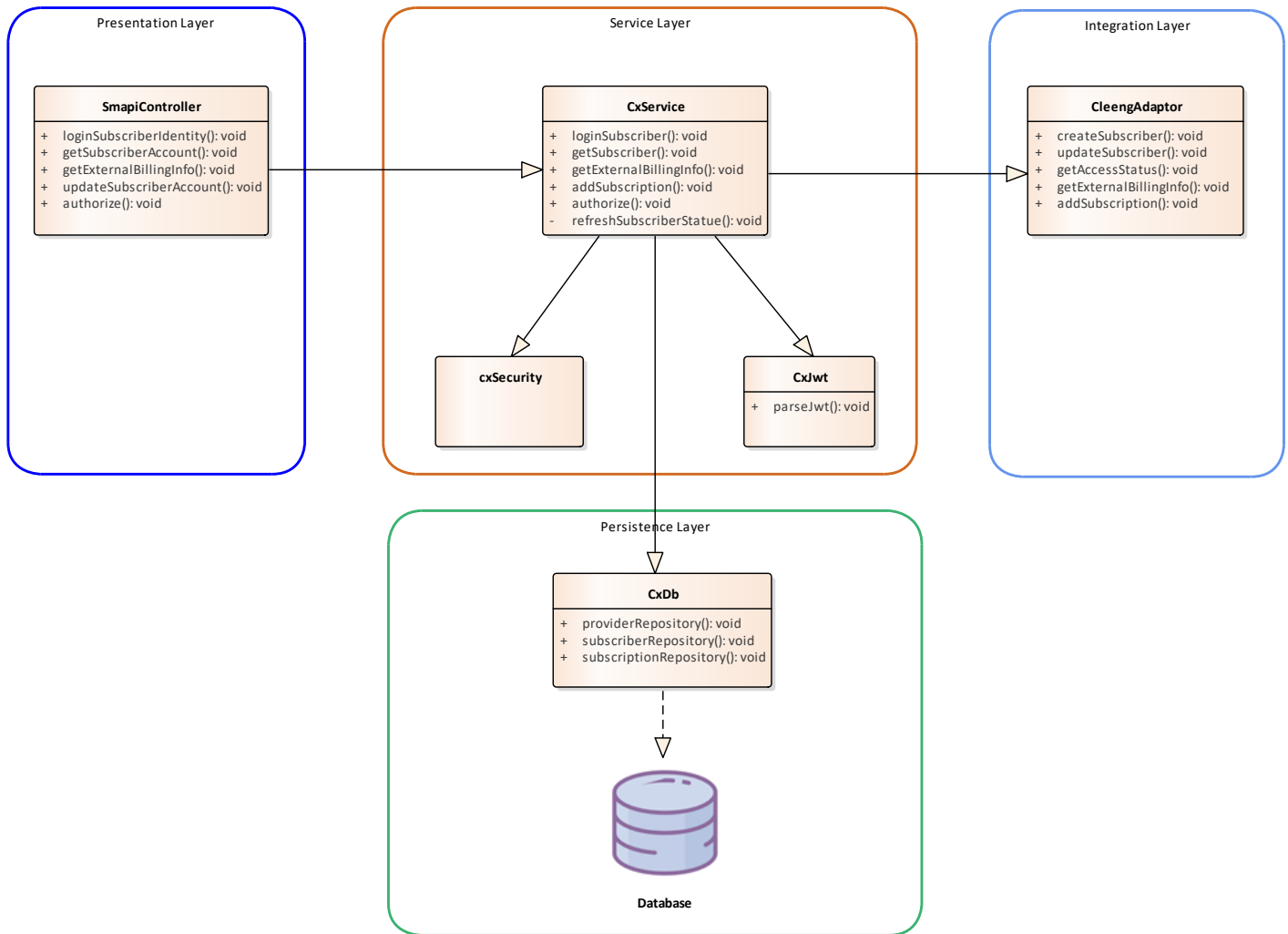
User	Standard, Member, or Premium users.
Subscriber	Member or Premium users.
Client App	iOS, Android, or Web app.

2 CX Architecture Overview

The below Class diagram shows the major classes and their relationships within the CX App. These classes have been designed based on the [CBC 3.0 System HLD vo.4](#). The following sections will discuss these classes and their methods in more detail; however, the [CX Domain Model](#) section shows the working relationships between these classes.



class CX Classes



2.1 SmapiController

The SMAPI controller is responsible for processing the income requests and delegates them to the CxService for performing the actual processing of the data.

API	Parameters	Returns	Description
loginSubscriberIdentity	JWT	identity	Supports SSO by allowing users to log in using the LoginRadius JWT.
getSubscriberAccount	identityGuid	account	Retrieves the subscriber account and the current active subscription pointed by the active_subscription_guid field of the Subscriber table.
getExternalBillingInfo	identityGuid	billingSystem	Retrieves the Cleeng purchase URL used by the client app to redirect the user to the Cleeng checkout website, allowing the user to purchase subscriptions using credit card.
updateSubscriberAccount	identityGuid affiliateCode Subscription	none	Adds a new subscription for the specified user. The current subscription will be set to “canceled” and the new subscription will be set to “active” state.
authorize	identityGuid resources	decisions	Uses this API to check if the user has access to the specified resources.

2.2 CxService

CxService, the main service layer, is responsible for performing the actual business logic.

API	Parameters	Returns	Description
loginSubscriber	JWT	identity	Creates a new subscriber record for new users and updates the user account if the JWT indicated the user data has changed.
getSubscriber	identityGuid	account	Checks the current subscription status with Cleeng system before returning the user account information from the database.
getExternalBillingInfo	identityGuid	billingSystem	Retrieves the Cleeng purchase URL used by the client app for web billing.
add Subscription	identityGuid affiliateCode Subscription	none	Add a new subscription purchased by web billing, Apple IAP, or Android IAP.
authorize	identityGuid resources	decisions	Checks the current subscription status with Cleeng system before checking the user accessibility for the specified resources.
refreshSubscriberStatue	identityGuid	subscriberEntity	Returns the specified user account after the user subscription has been validated with the Cleeng system.

2.3 CleengAdaptor

The CleengAdaptor implementing the BillingAdapter interface is responsible for communicating with external billing system such as Cleeng.

API	Parameters	Returns	Description
createSubscriber	subscriberEntity	response	Registers a new user to Cleeng system.
updateSubscriber	subscriberEntity jwtPayload	response	Synchronizes the user data with the Cleeng system.
getAccessStatus	email password productId	accessGranted	Retrieves the current subscription status with the Cleeng system.
getExternalBillingInfo	email productId	URL	Retrieves the Cleeng purchase URL used by the client app for web billing.
add Subscription	identityGuid affiliateCode Subscription	none	Add a new subscription purchased by web billing, Apple IAP, or Android IAP to the Cleeng system

2.4 CxJwt

The CxJwt has only one static parseJwt(JWT) method which is responsible for decoding and decrypting the JWT.

2.5 CxDb

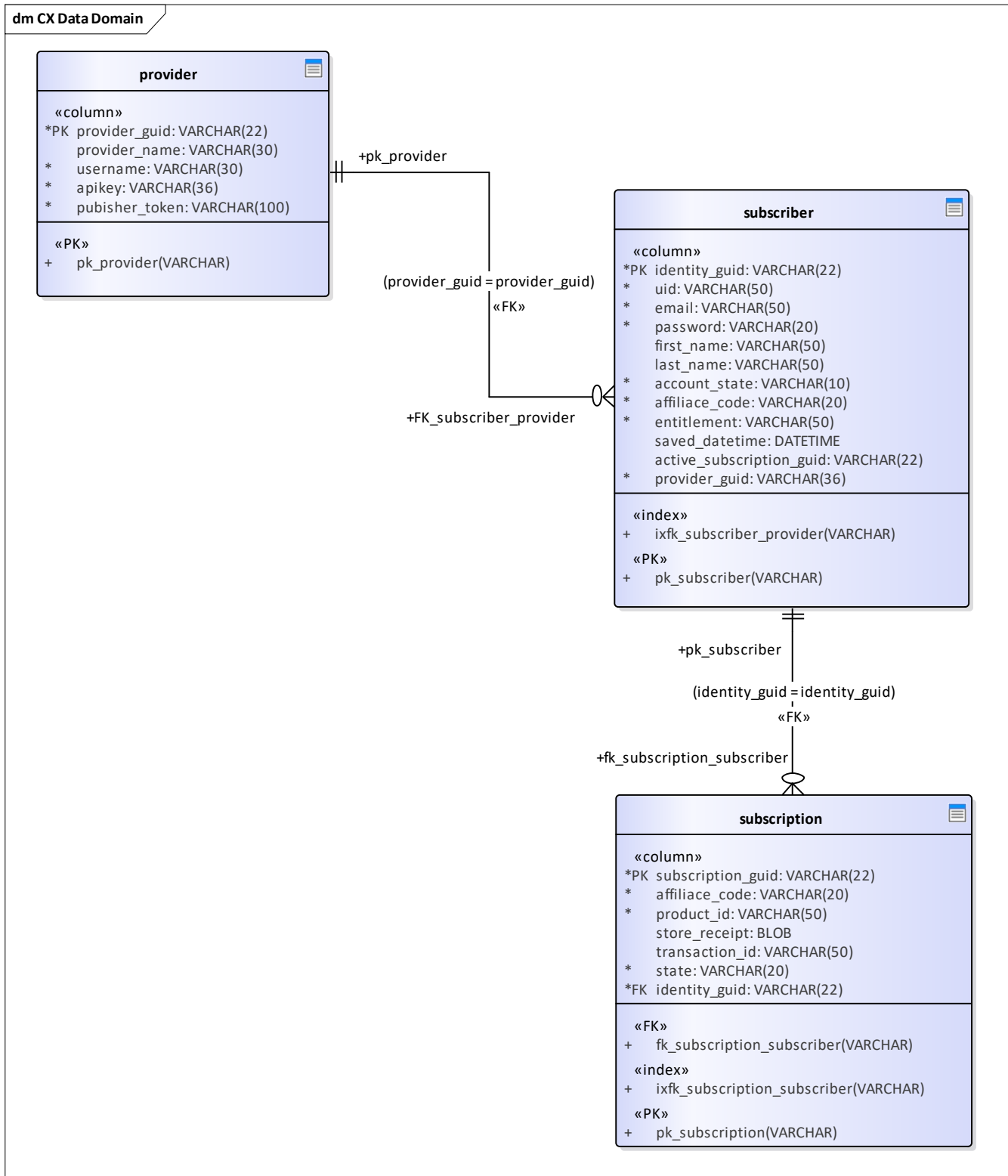
The database module contains the JPA repository classes used for accessing the CX database.

2.6 CxSecurity

The CxSecurity module is responsible for handling the overall security concerns including authentication and authorization.

3 CX Data Domain

The below data model diagram shows the representation of entities as well as relationships between entities within the CX domain. The following sections will describe these entities in more detail.



3.1 Provider Table

The Provider table is used to store the information of an IBM Cloud Video client who used CBC 3.0 to ingest, process and deliver video content to subscribers.

Table Name: **provider**

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
provider_guid	VARCHAR(22)	True	A random Base64 encoded UUID identified the provider.
provider_name	VARCHAR(30)	False	The name of the provider (e.g., “CBC”).
username	VARCHAR(30)	True	The username used as part of the HTTP Basic Authentication (See SMAPI Information for more information). For testing, it will be initialized to “CBC”.
apikey	VARCHAR(36)	True	The apikey used as part of the HTTP Basic Authentication (See SMAPI Information for more information). For testing, it will be initialized to “718838ed-cb6d-4c9e-bdea-705186d89512”.
publisher_token	VARCHAR(100)	True	The Publisher Token can be generated using the Cleeng dashboard. For testing, it will be initialized to “Mvvmu3X1KooTIQk_f_Spw1o4W9Reid4YGOysZTtbA4VNjDM3”.

3.2 Subscriber Table

The Subscriber table is used to store the user account information which is created using the [JWT payload](#) during the [loginSubscriberIdentity](#) use case.

Table Name: **subscriber**

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
identity_guid	VARCHAR(22)	True	A random Base64 encoded UUID identified the user.
provider_guid	VARCHAR(36)	True	The provider GUID that this user belongs to.
uid	VARCHAR(50)	True	The LoginRadius assign a unique ID for this user, and it is populated from the jwtPayload.uid field.
email	VARCHAR(50)	True	The email address of this user and it is populated from the jwtPayload.email field.
password	VARCHAR(20)	True	The generated password by CX used to authenticate with the Cleeng system.
first_name	VARCHAR(50)	False	The first name on the account and it is populated from the jwtPayload.firstName field.
last_name	VARCHAR(50)	False	The last name on the account and it is populated from the jwtPayload.lastName field.
account_state	VARCHAR(10)	True	The state of the account. Valid values are: ok, pending, overdue, unpaid, suspended, closed, and unknown. However, CX always set the accountState to “ok”.
affiliace_code	VARCHAR(20)	True	Identifies the operator that the account belongs to and can be one of the following supported operators: <ul style="list-style-type: none">• CBC_CLEENG: for web billing• CBC_APPLE: for Apple IAP• CBC_ANDROID: for Android IAP It is populated from the affiliateCode request parameter of the updateSubscriberAccount API.
entitlement	VARCHAR(50)	True	The authorization assigned to this user based on the subscription status and it can be the following values: <ul style="list-style-type: none">• CBC_MEMBER: a registered user without buying a subscription and can view videos with ads and has personalization account containing bookmarks, recently watched, and account details page.

			<ul style="list-style-type: none"> CBC_PREMIUM: a subscription user that has the same as CBC_MEMBER but can view online videos limited ads
saved_datetime	DATETIME	False	The last saved UTC datetime, and it is populated from the jwtPayload.changeIndicator field.
active_subscription_guid	VARCHAR(22)	False	The GUID of the current active subscription.

3.3 Subscription Table

Table Name: **subscription**

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
subscription_guid	VARCHAR(22)	True	A random Base64 encoded UUID identified the subscription.
identity_guid	VARCHAR(22)	True	The user GUID that this subscription belongs to.
affiliace_code	VARCHAR(20)	True	<p>Identifies the payment method that used to purchase this subscription and it can be the following values:</p> <ul style="list-style-type: none"> CBC_CLEENG: for web billing CBC_APPLE: for Apple IAP CBC_ANDROID: for Android IAP <p>It is populated from the affiliateCode request parameter of the updateSubscriberAccount API.</p>
product_id	VARCHAR(50)	True	The unique ID identified the subscription offer. The product ID will be assigned by Cleeng system during the creation of the subscription offer. The same Cleeng product ID must be used for Apple and Google stores. Please see Section 5.4 Premium User in the CBC 3.0 HLD for more information on how to link the subscription product from Cleeng system to Apple and Google platform.
store_receipt	BLOB	False	The actual transaction receipt returned from Apple or Android IAP.
transaction_id	VARCHAR(50)	False	The transaction ID from Apple IAP.
state	VARCHAR(20)	True	The state of the current subscription. Possible values: Active, Cancelled and Deleted. It is updated when calling the getSubscriberAccount and authorize APIs.

4 CX SMAPI Implementation

This section focusses on defining the logic flows between the different CX layers and its modules on how to support the five SMAPI. In addition to defining the specification and syntax of each API in detail, a sequential diagram is also included to show the logic that must be implemented by each of the CX classes.

4.1 SMAPI Information

The CX will only support the subset of the SMAPI that are used in the CBC 3.0 system. Furthermore, not all input parameters are supported. The following sections will list the parameters that are needed by the supported APIs. Extra input parameters will not cause any errors; CX just ignores them.

TVE SM allows the user to be identified by `identityGuid`, `accountGuid`, and `username (email)`; however, CX will only support the `identityGuid` because the CM does not use the `accountGuid` and the JWT has replaced the `username`. As a result, the `getExternalBillingInfo` API of the Client Adapter needs to be updated to support the `identityGuid`.

The CX SMAPI has the following design characteristics:

- Supports only secure HTTPS access
- Supports both XML and JSON as request and response formats
- Provides HTTP response codes as well as Service error codes

Security

CX uses the HTTP Basic Authentication for enforcing access control to the SMAPI. The CM constructs the *Authorization* field as follows:

1. The *username* and *apikey* from the Provider table are combined with a single colon. (:))
2. The resulting string is encoded into an octet sequence.
3. The resulting string is encoded using a variant of Base64.
4. The authorization method and a space (e.g., "Basic ") is then prepended to the encoded string, separated with a space.

For example, if the Catalog Manager (CM) uses *CBC* as the username and *718838ed-cb6d-4c9e-bdea-705186d89512* as the apikey, then the *Authorization* header will appear as:

```
Authorization: Basic Q0JDOjcxODgzOGVkbWVhLTcwNTE4NmQ4OTUxMg==
```

4.2 loginSubscriberIdentity

This API allows a user to log in using JWT which generated from a trusted source (i.e., LoginRadius). It will perform the following steps:

1. Verifies and decrypts the JWT.
2. If this is a new user then adds a new record to the Subscription table using the data from the JWT payload.
3. Else updates the user account if the date in the JWT.changeIndicator field is different than the saved_datetime in the Subscriber table.

4.2.1 Request

Method: PUT

URI: https://address/cltve /api/login/

4.2.2 Parameters

None.

4.2.3 Body

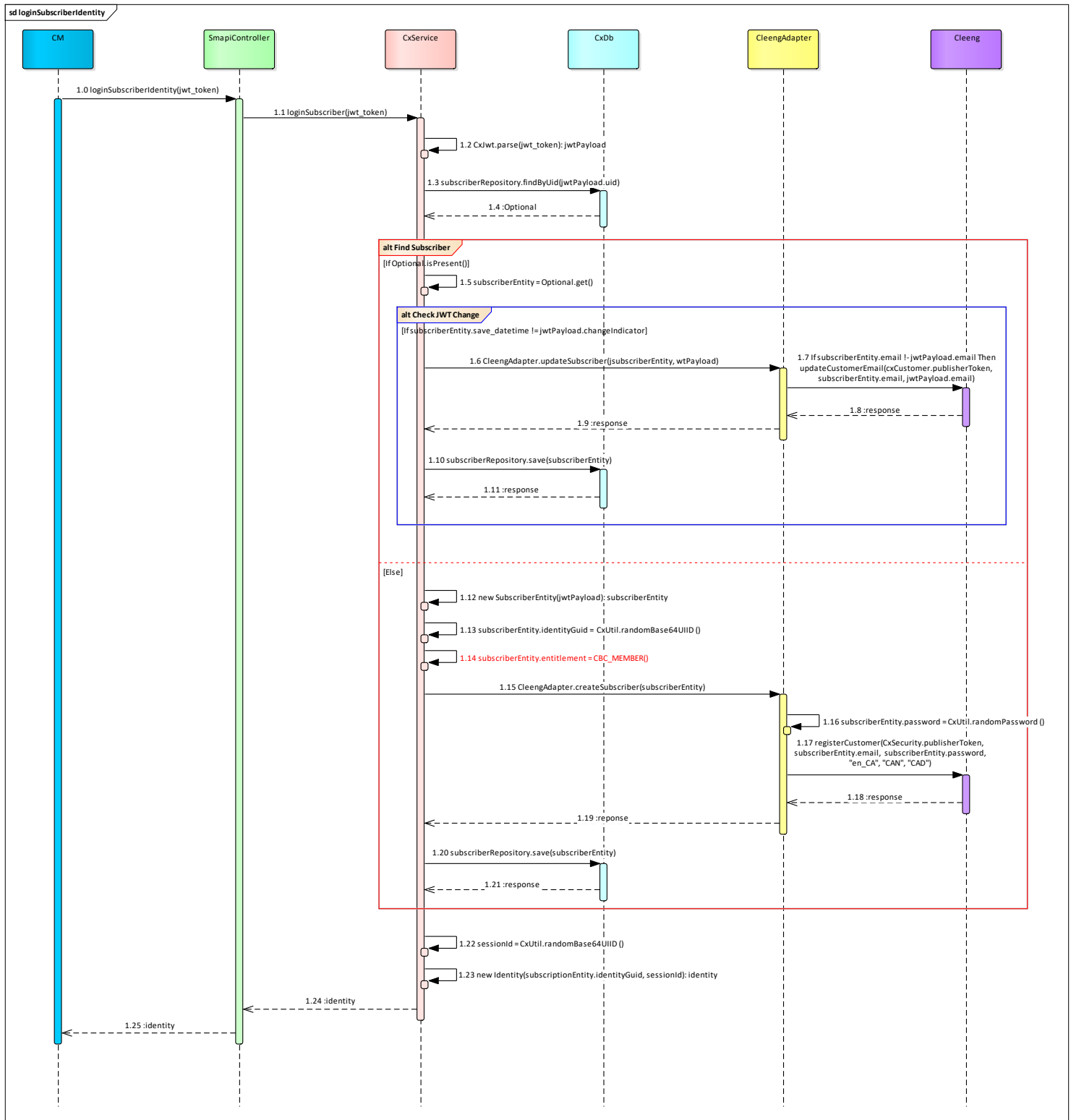
login			
token	Element	Required	The JWT containing the subscriber's profile information.

4.2.4 Response

result			
status	Element	Optional	The status of the call. Valid values are success, failure. Only required if failure. For successes, a 2xx response code may be used to determine success.
errorCode	Element	Optional	An error code, if any.
userMessage	Element	Optional	A message suitable for displaying to the user.
systemMessage	Element	Optional	A message suitable for logging/troubleshooting. Should not be displayed to the user.
identity	Element	Required	Metadata of the identity that was created as result of the action.

4.2.6 Sequential Diagram

The following diagram shows the logic must be performed by each of the CX modules to log in a new or existing LoginRadius user.



4.3 getSubscriberAccount

Uses this API to retrieve the subscriber account and the current active subscription pointed by the active_subscription_guid field of the Subscriber table. The client app will use this API to retrieve the user entitlement to determine which catalog to display to the user.

4.3.1 Request

Method: GET

URI: https://address/cltve /api/accounts/

4.3.2 Parameters

parameters			
identityGuid	Query parameter	Required	Unique identify GUID for this user, and it maps to the identity_guid of Subscription table.

4.3.3 Body

None.

4.3.4 Response

result			
status	Element	Optional	The status of the call. Valid values are success, failure. Only required if failure. For successes, a 2xx response code may be used to determine success.
errorCode	Element	Optional	An error code, if any.
userMessage	Element	Optional	A message suitable for displaying to the user.
systemMessage	Element	Optional	A message suitable for logging/troubleshooting. Should not be displayed to the user.
account	Element	Required	The user account information.

account			
identityGuid	Element	Required	Unique identify GUID for this user, and it maps to the identity_guid of Subscription table.
affiliateCode	Element	Required	Identifies the operator that the account belongs to and can be one of the following supported operators: <ul style="list-style-type: none"> CBC_CLEENG: for web billing CBC_APPLE: for Apple IAP CBC_ANDROID: for Android IAP It maps to the affiliate_code field of the Subscriber and Subscription tables.
accountState	Element	Required	The state of the account. Valid values are: ok, pending, overdue, unpaid, suspended, closed, and unknown. However, CX always set the accountState to “ ok ”.
firstName	Element	Optional	The first name on the account.
lastName	Element	Optional	The last name on the account.
entitlements	Element	Required	A list of entitlements associated with the account.
subscription	Element	Optional	The current subscription signed up by the user.

entitlements			
entitlement	Element	Required	The entitlement assigned to this account and it can be retrieved from the entitlement of the Subscriber table.

subscription			
state	Element	Required	The state of the current subscription. Possible values: Active, Cancelled and Deleted.
productId	Element	Optional	The Apple/Android product ID or Cleeng offer ID.
productRatePlanId	Element	Optional	The transaction Id of the Apple IAP and it maps to the transaction_id of the Subscription table.
paymentMethods	Element	Optional	The payment method used to purchase the subscription.

paymentMethods			
paymentMethodId	Element	Required	The Apple/Android store receipt and it maps to the store_receipt of the Subscription table.

4.3.5 Example

The following shows the example of the request/response of this API.

Request URL:

GET https://tve-cloud.ibm.com/cltve/api/accounts?identityGuid=e3V1AlO2B0WIDVjQ3U8JPQ

Response Body:

The following shows the response for a user **without a subscription**.

```
<result>
  <status>Success</status>
  <account>
    <accountGuid>e3V1AlO2B0WIDVjQ3U8JPQ</accountGuid>
    <affiliateCode>CBC</affiliateCode>
    <accountState>ok</accountState>
    <firstName>Jessica</firstName>
    <lastName>Burton</lastName>
    <entitlements>
      <entitlement>CBC_MEMBER</entitlement>
    </entitlements>
  </account>
</result>
```

The following shows the response for a user with a subscription purchase using **web billing**.

```
<result>
  <status>Success</status>
  <account>
    <accountGuid>e3V1AlO2B0WIDVjQ3U8JPQ</accountGuid>
    <affiliateCode>CBC_CLEENG</affiliateCode>
    <accountState>ok</accountState>
    <firstName>Jessica</firstName>
    <lastName>Burton</lastName>
    <entitlements>
      <entitlement>CBC_PREMIUM</entitlement>
    </entitlements>
    <subscription>
      <state>Active</state>
      <productId>S202885261_CAN</productId>
    </subscription>
  </account>
</result>
```


The following shows the response for a user with a subscription purchase using **Apple IAP**.

```
<result>
  <status>Success</status>
  <account>
    <accountGuid>e3V1AlO2B0WIDVjQ3U8JPQ</accountGuid>
    <affiliateCode>CBC_APPLE</affiliateCode>
    <accountState>ok</accountState>
    <firstName>Jessica</firstName>
    <lastName>Burton</lastName>
    <entitlements>
      <entitlement>CBC_PREMIUM</entitlement>
    </entitlements>
    <subscription>
      <state>Active</state>
      <productId>S202885261_CAN</productId>
      <productRatePlanId>Aacced48801z35Ke</productRatePlanId>
      <paymentMethods>
        <paymentMethod>
          <paymentMethodId>ewoJInNpZ25hdHVyZSIg...</paymentMethodId>
        </paymentMethod>
      </paymentMethods>
    </subscription>
  </account>
</result>
```

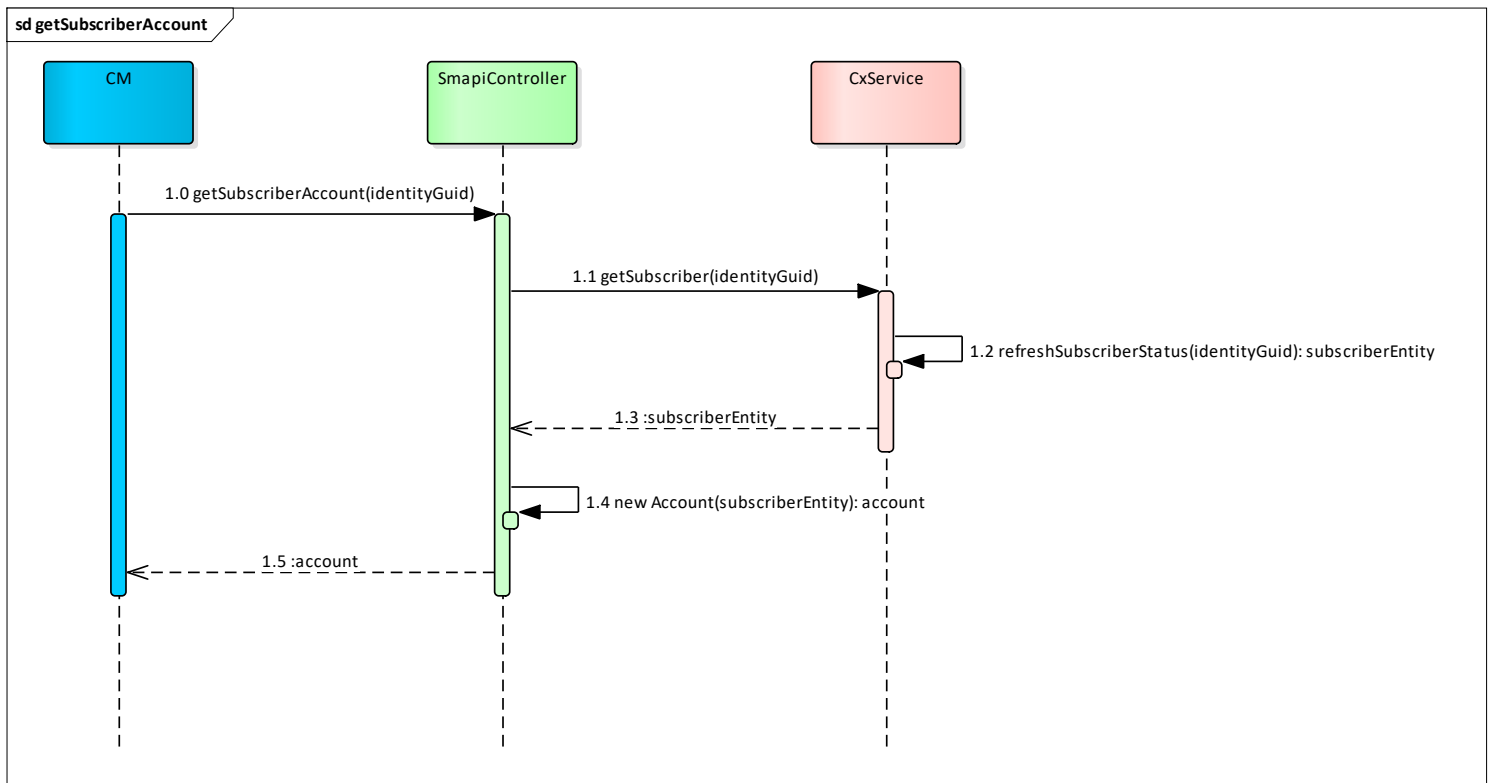
The following shows the response for a user with a subscription purchase using **Android IAP**.

Note: The productId is in lower case for Android IAP.

```
<result>
  <status>Success</status>
  <account>
    <accountGuid>e3V1AlO2B0WIDVjQ3U8JPQ</accountGuid>
    <affiliateCode>CBC_ANDROID</affiliateCode>
    <accountState>ok</accountState>
    <firstName>Jessica</firstName>
    <lastName>Burton</lastName>
    <entitlements>
      <entitlement>CBC_PREMIUM</entitlement>
    </entitlements>
    <subscription>
      <state>Active</state>
      <productId>s202885261_can</productId>
      <paymentMethods>
        <paymentMethod>
          <paymentMethodId>ewoJInNpZ25hdHVyZSIg...</paymentMethodId>
        </paymentMethod>
      </paymentMethods>
    </subscription>
  </account>
</result>
```

4.3.6 Sequential Diagram

The following diagram shows the logic must be performed by each of the CX modules to retrieve the specified user account information.



4.4 getExternalBillingInfo

For the CBC 3.0 release, this API will use to retrieve the Cleeng purchase URL used by the client app to redirect the user to the Cleeng checkout website, allowing the user to purchase subscriptions using credit card.

Also, the CBC 3.0 will only support one subscription offer which is the Monthly SVOD subscription; however, in CBC 3.1, the client app can pass in the “CBC_PRODUCT” as an affiliateCode query parameter to retrieve a list of subscription offers from Cleeng. This will allow CBC to offer multiple subscriptions offers to their users rather than just one monthly SVOD subscription in 3.0 release.

4.4.1 Request

Method: GET

URI: https://address/cltve /api/billing/

4.4.2 Parameters

parameters			
identityGuid	Query parameter	Required	The identity GUID of the Subscriber and it can be retrieved from the identity_guid of the Subscription table.
affiliateCode	Query parameter	Required	For CBC 3.0 release, it must be populated with “CBC_CLEENG”.

4.4.3 Body

None.

4.4.4 Response

result			
status	Element	Optional	The status of the call. Valid values are success, failure. Only required if failure. For successes, a 2xx response code may be used to determine success.
errorCode	Element	Optional	An error code, if any.
userMessage	Element	Optional	A message suitable for displaying to the user.
systemMessage	Element	Optional	A message suitable for logging/troubleshooting. Should not be displayed to the user.
billingSystem	Element	Optional	Information about the billing system.

billingSystem			
type	Element	Required	The same value as the input affiliateCode.
paymentMethodEndPoint	Element	Required	The purchase URL to perform the payment provisioning.

4.4.5 Example

The following shows the example of the request/response of this API.

Request URL:

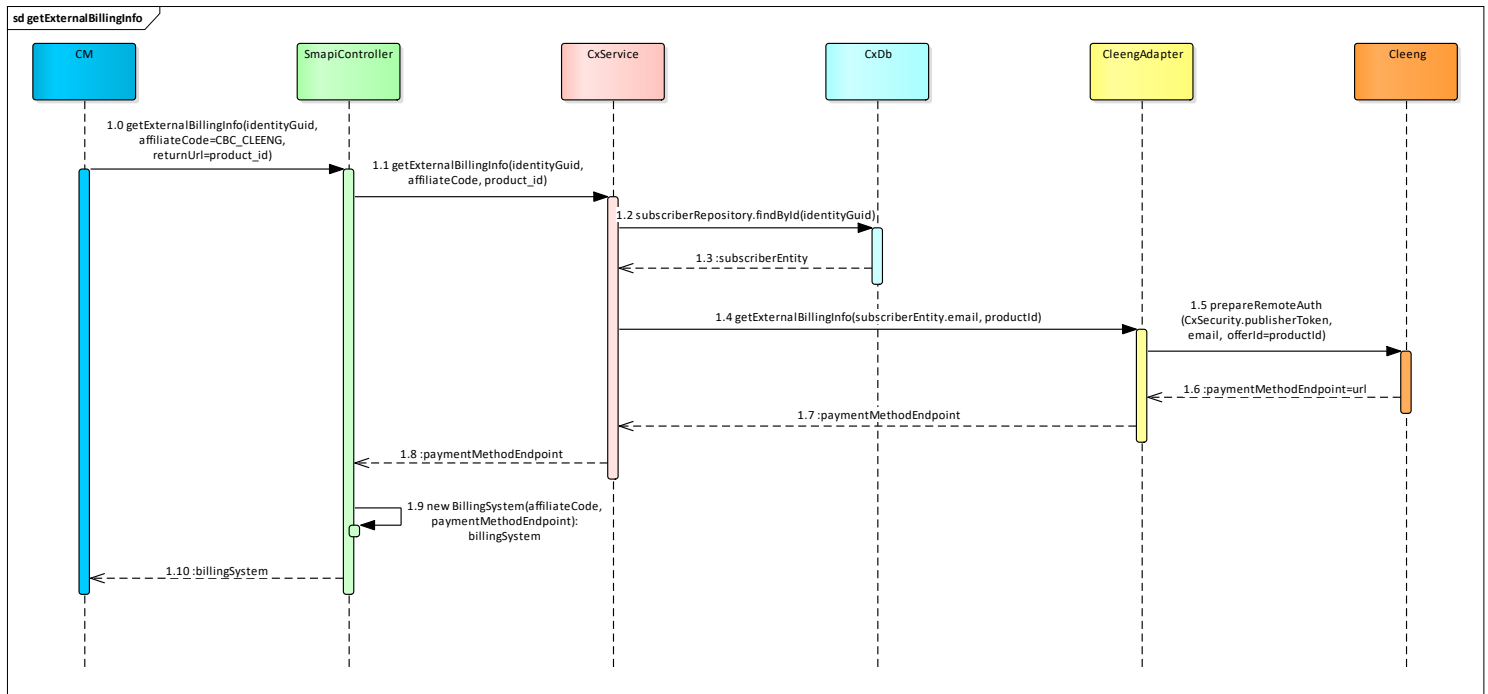
```
GET https://tve-cloud.ibm.com/cltve/api/billing?identityGuid=e3VlA1O2B0WIDVjQ3U8JPQ
&affiliateCode=CBC_CLEENG
```

Response Body:

```
<result>
  <status>Success</status>
  <billingSystem>
    <type>Cleeng</type>
    <paymentMethodEndpoint>
      https://cleeng.com/auth/2/purchase?v=2.1&popup=1&remote_token=4a05f0bd07870f4cec92456a7a38452059
      a4a394:6484554b9bfd9b1609a51f3dcb0b3077d5e11ab6&redirectUri=https%3A%2F%2Fdev.cleeng.com%2Fjs-
      api%2F2.0%2Fdefault-channel.html&offerId=S202885261_CAN
    </paymentMethodEndpoint>
  </billingSystem>
</result>
```

4.4.6 Sequential Diagram

The following diagram shows the logics must be performed by each of the CX modules to retrieve the purchase URL from Cleeng.



4.5 updateSubscriberAccount

Uses this API to add a new subscription for the specified user. This new subscription state will be set to “active” and the previous subscription if existed, its state will be set to “cancelled”.

4.5.1 Request

Method: POST

URI: <https://address/cltve/api/accounts/>

4.5.2 Parameters

parameters			
identityGuid	Query parameter	Required	Unique identify GUID for this user and it maps to the identity_guid of Subscription table.
affiliateCode	Query parameter	Required	Identifies the operator that the account belongs to and it can be one of the following supported operators: <ul style="list-style-type: none">• CBC_CLEENG: for web billing• CBC_APPLE: for Apple IAP• CBC_ANDROID: for Android IAP It maps to the affiliate_code field of the Subscriber and Subscription tables.

4.5.3 Body

subscription			
state	Element	Required	The state of the current subscription. Possible values: Active, Cancelled, and Deleted.
productId	Element	Optional	The Apple/Android product ID or Cleeng offer ID.
productRatePlanId	Element	Optional	The transaction Id of the Apple IAP and it maps to the transaction_id of the Subscription table.
paymentMethods	Element	Optional	The payment method used to purchase the subscription.

paymentMethods			
paymentMethodId	Element	Required	The Apple/Android store receipt and it maps to the store_receipt of the Subscription table.

4.5.4 Response

result			
status	Element	Optional	The status of the call. Valid values are success, failure. Only required if failure. For successes, a 2xx response code may be used to determine success.
errorCode	Element	Optional	An error code, if any.
userMessage	Element	Optional	A message suitable for displaying to the user.
systemMessage	Element	Optional	A message suitable for logging/troubleshooting. Should not be displayed to the user.

4.5.5 Example

The following shows the example of the request/response for subscription purchase using **web billing**.

Request URL:

POST <https://tve-cloud.ibm.com/cltve/api/accounts>

Request Body:

```
identityGuid=e3V1AlO2B0WIDVjQ3U8JPQ&affiliateCode=CBC_CLEENG
<account>
  <subscription>
    <state>Active</state>
    <productId>S202885261_CAN</productId>
  </subscription>
</account>
```

Response Body:

```
<result>
  <status>Success</status>
</result>
```

The following shows the example of the request/response for subscription purchase using **Apple IAP**.

Request URL:

POST <https://tve-cloud.ibm.com/cltve/api/accounts>

Request Body:

```
identityGuid=e3V1AlO2B0WIDVjQ3U8JPQ&affiliateCode=CBC_APPLE
<account>
  <subscription>
    <state>Active</state>
    <productId>S202885261_CAN</productId>
    <productRatePlanId>Aacced48801z35Ke</productRatePlanId>
    <paymentMethods>
      <paymentMethod>
        <paymentMethodId>ewoJInNpZ25hdHVyZSIg...</paymentMethodId>
      </paymentMethod>
    </paymentMethods>
  </subscription>
</account>
```

Response Body:

```
<result>
  <status>Success</status>
</result>
```

The following shows the example of the request/response for subscription purchase using **Android IAP**.

Request URL:

POST <https://tve-cloud.ibm.com/cltve/api/accounts>

Request Body:

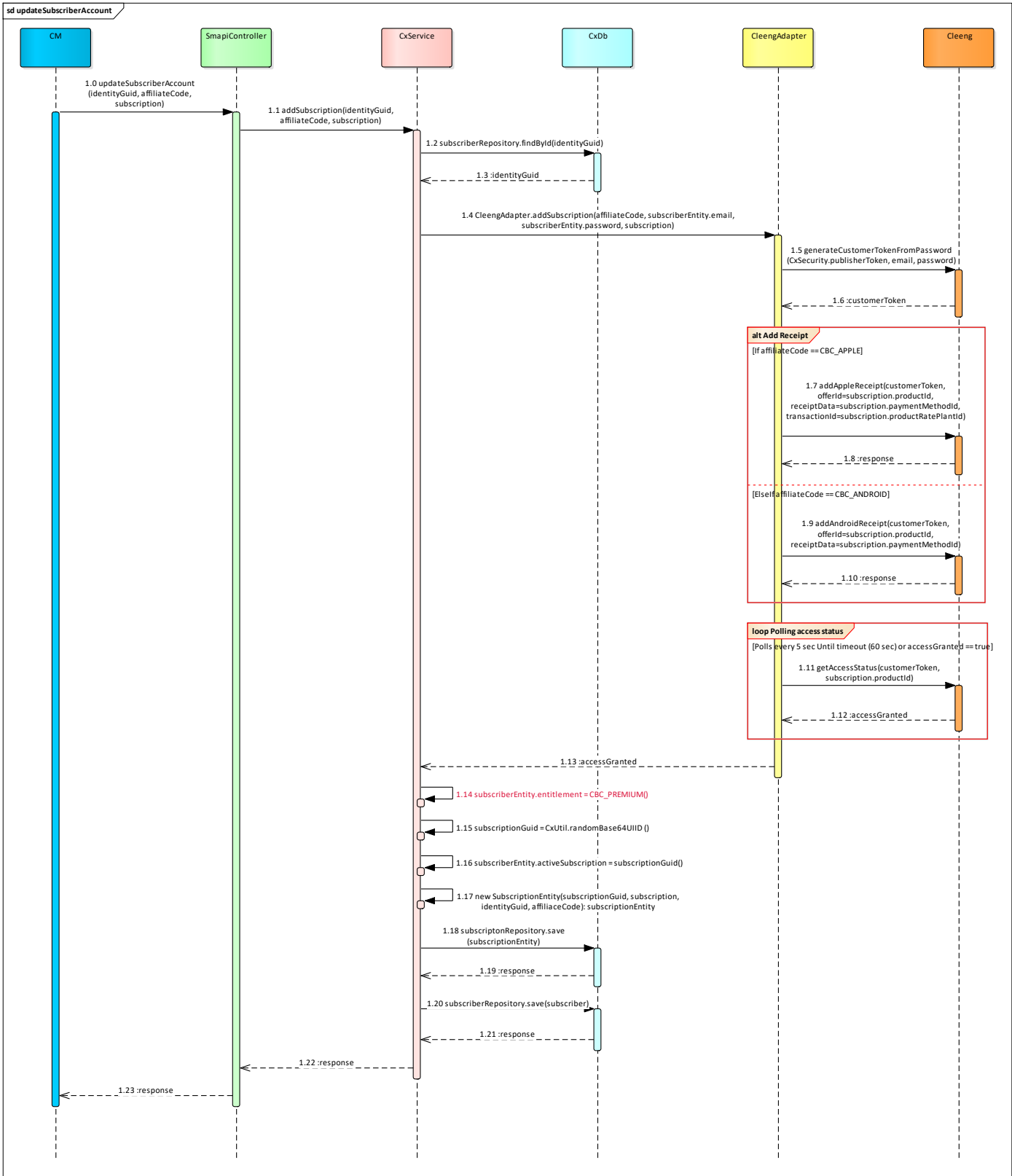
```
identityGuid=e3V1AlO2B0WIDVjQ3U8JPQ&affiliateCode=CBC_ANDROID
<account>
  <subscription>
    <state>Active</state>
    <productId>s202885261_can</productId>
    <paymentMethods>
      <paymentMethod>
        <paymentMethodId>ewoJInNpZ25hdHVyZSIg...</paymentMethodId>
      </paymentMethod>
    </paymentMethods>
  </subscription>
</account>
```

Response Body:

```
<result>
  <status>Success</status>
</result>
```


4.5.6 Sequential Diagram

The following diagram shows the logics must be performed by each of the CX modules to add a new subscription purchased using web billing or Apple/Android IAP.



4.6 authorize

Uses this API to check if the user has access to the specified resources.

4.6.1 Request

Method: POST

URI: https://address/cltve /api/ authorize/<identityGuid>

4.6.2 Parameters

parameters			
identityGuid	URI parameter	Required	Unique identify GUID for this user and it maps to the identity_guid of Subscription table.

4.6.3 Body

resources			
resourceId	Element	Required	The resource needs to be checked.

4.6.4 Response

result			
status	Element	Optional	The status of the call. Valid values are: success, failure. Only required if failure. For successes, a 2xx response code may be used to determine success.
errorCode	Element	Optional	An error code, if any.
userMessage	Element	Optional	A message suitable for displaying to the user.
systemMessage	Element	Optional	A message suitable for logging/troubleshooting. Should not be displayed to the user.
decisions	Element	Optional	Results of the authorization.

decisions			
decision	Element	Required	The result of the authorization for the specified resource.

decision			
resourceId	Attribute	Required	Identifies the requested resource.
decision	Element	Required	The authorization decision. Possible values: Deny, Permit

4.6.5 Example

The following shows the request/response using JWT token.

Request URL:

<https://tve-cloud.ibm.com/cltve/api/authorize/73f3a786-2cf2-48df-b2af-d06fbb22aa4f>

Request Body:

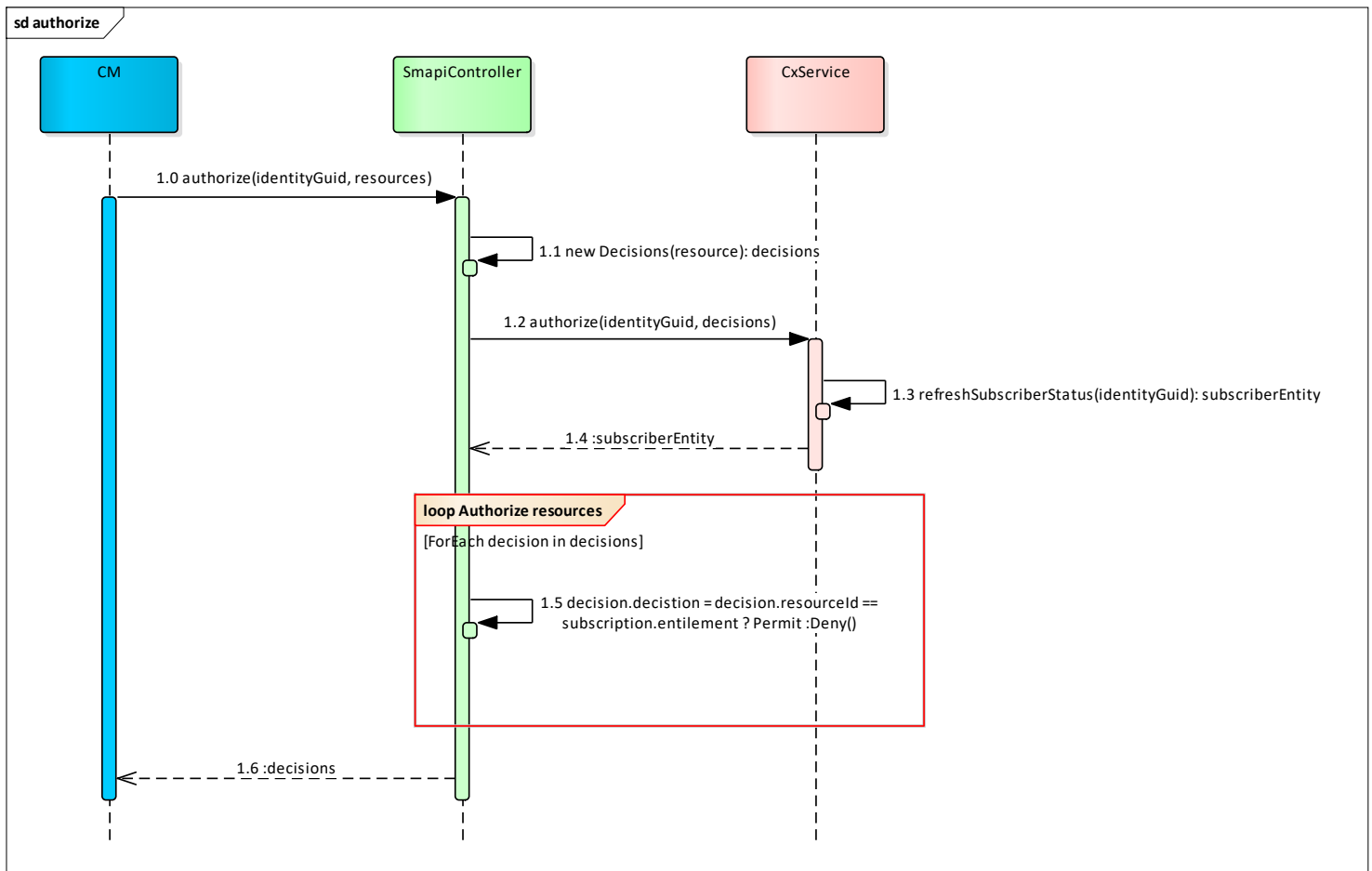
```
<resources>
  <resourceId>CBC_MEMBER</resourceId>
  <resourceId>CBC_PREMIUM</resourceId>
</resources>
```

Response Body:

```
<result>
  <status>Success</status>
  <decisions>
    <decision resourceId="CBC_MEMBER">Permit</decision >
    <decision resourceId="CBC_PREMIUM">Deny</decision >
  </decisions>
</result>
```

4.6.6 Sequential Diagram

The following diagram shows the logics must be performed by each of the CX modules to retrieve the purchase URL from Cleeng.



4.7 refreshSubscriberStatus

The caller calls this CxService private method to check the status of the current active subscription and update the user entitlement to CBC_PREMIUM if the subscription is still active or to CBC_MEMBER if the subscription is cancelled.

4.7.1 Parameters

parameters			
identityGuid	String	Required	Unique identify GUID for this user and it maps to the identity_guid of Subscription table.

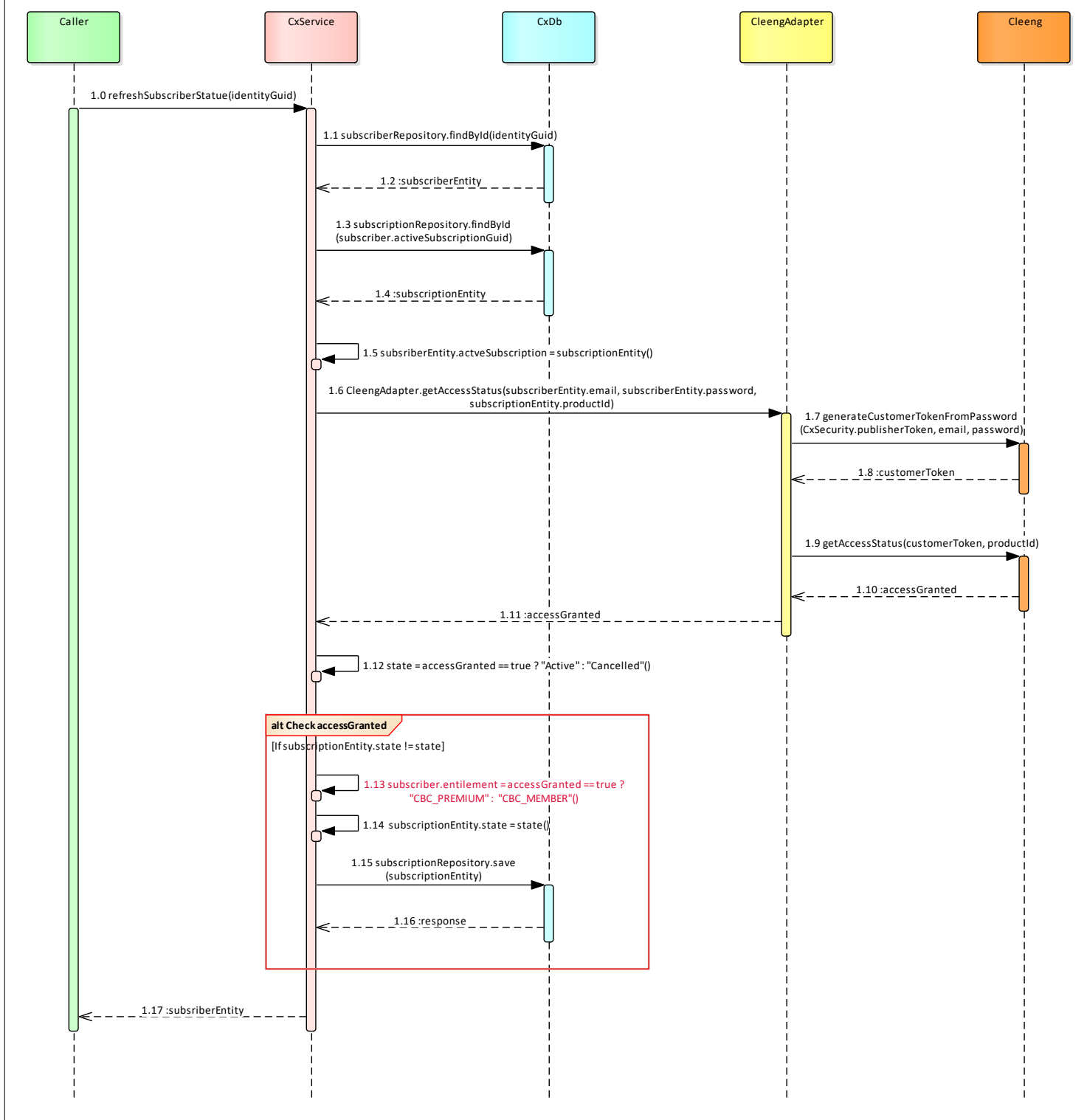
4.7.2 Returns

returns			
subscriberEntity	SubscriberEntity	Required	The subscriber entity contains the user account information.

4.7.3 Sequential Diagram

The following diagram shows the logics must be performed by each of the CX modules to check the status of the current active subscription.

sd refreshSubscriberStatus



5 LoginRadius SSO

In CBC 3.0 system, the LoginRadius is continued to function as an SSO provider storing the user's credential and performing the authentication and authorization. However, CX is now responsible for extracting the users' credential from the JSON Web Token (JWT) and passing them to the Cleeng platform. Acting as a proxy layer, CX automatically converts JWT to Cleeng authentication API allowing a seamless integration with the Cleeng Identity Manager.

JWT is an open, industry standard RFC 7519 method for sharing the user's authentication and authorization between systems. CX uses the JWT Identity Adapter (JIA) for authenticating/authorizing by decoding/decrypting the JWT. The major benefit using the JWT is that there is no integration between CX and LoginRadius because JWT already includes the user data. JIA can be furthermore used to support other external SSO systems that support JWT format. Because there is no communication between CX and LoginRadius, the Client App is responsible for passing the JWT to CX after it has successfully logged in the user with LoginRadius.

5.1 JWT Payload Format

The following table lists the required JWT payload fields.

Note: The JIA needs to use the `changeIndicator` field to determine when to sync the user data with the Cleeng platform.

Field Name	Description
uid	The unique ID for the user.
email	The email address of the user.
username	The username used to login to LoginRadius.
firstName	The first name on the account.
middleName	The middle name/initial on the account.
lastName	The last name on the account.
address1	The first line of the address on the account.
address2	The second line of the address on the account.
city	The city on the account.
state	The state on the account.
zip	The zip code on the account.
country	The country on the account. 2-letter country code supplied by ISO.
homePhone	The home phone on the account.
changeIndicator	This field contains the UTC datetime of the last user profile updated.

The following is a sample of the LoginRadius JWT:

```
{
  "iss": "https://cbc-login-dev.hub.loginradius.com",
  "sub": "890e0f2d19a2442ead480eel39caa505",
  "jti": "0429c165-cea2-4410-a966-8eeb7e77c3c9",
  "iat": 1489172945,
  "nbf": 1489172945,
  "exp": 1489173545,

  "uid": "5535b544-0bb7-4c5a-bf63-d6d0dd01191d",
  "email": "vandar1123@gmail.com",
  "username": "vandar1",
  "firstName": "Van",
  "middleName": "C",
  "lastName": "Nguyen",
  "address1": "3100 Breckinridge Blvd",
  "address2": null,
  "city": "Duluth",
  "state": "GA",
  "zip": "30096",
  "country": "USA",
  "homePhone": "7704182800",
  "changeIndicator": "03/15/2017 16:51:22"
}
```

6 Development Guidelines

This section lists the IDE, tools, and frameworks used in the CX project.

6.1 IDE and Tools

The following table lists the minimum required IDE and tools that need to be installed by the team members.

Tool Name	Version	Download Link
IntelliJ	2017.3	https://www.jetbrains.com/idea/
Java 8 JDK	8u151	http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
Gradle	4.3.1	https://gradle.org/install/

6.2 Frameworks

The following table lists the frameworks that will be used to build the CX App and its modules.

Tool Name	Version	Download Link
Spring Boot	2.0.0 M7	http://projects.spring.io/spring-boot/
Spring Framework	5.0.2	http://projects.spring.io/spring-framework/
Spring Data JPA	2.0.2	http://projects.spring.io/spring-data-jpa/
Spring Security	5.0.0	http://projects.spring.io/spring-security/
Junit	5	http://junit.org/junit5/
jjwt	0.9.0	https://github.com/jwtkt/jjwt
Log4j	1.7.25	https://www.slf4j.org/download.html

6.3 Team Workflow

The CX team will be using GitHub Flow for team collaboration:

1. The Project Manager and Tech Lead are responsible for using GitHub issues to create stories, tasks, bugs, etc.
2. A Milestone is created for each sprint and after during sprint planning, backlog tasks will be assigned to this Milestone.
3. When an issue is assigned to a team member, the team member needs to create a topic branch and then developing and committing using the created branch.
4. The team member needs to open a Pull Request for code review and QA testing before merging the new code to the Master branch.
5. The QA engineer needs to validate the Pull Request by running the unit tests.
6. Tech Lead will perform final review and the working branch can deploy to the Master branch.