CBC 3.0 System HLD

Version 0.5

December 3, 2017

Private & Confidential

1 Table of Contents

Revision History

Version	Date	Modified	Who
0.1	10/31/2017	Initial Release	Van Nguyen
0.2	11/6/2017	The first completed draft	Van Nguyen
0.3	11/6/2017	Completed the Section 4: Coordinator Middleware	Van Nguyen
0.4	11/7/2017	Minor updated to the Section 5.	Van Nguyen
0.5	12/3/2017	Updated the Section 4.3 Coordinator Middleware to sync with the CX SDD v0.2	Van Nguyen

1 Introduction

1.1 Purpose

This document describes the architecture of the CBC 3.0 system in detail including the replacement of the IBM Subscriber Manager with Cleeng SVOD system for supporting IAP and web billing.

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

This design is based on using Cleeng SVOD system rather than Subscriber Manager.

1.4 References

This document is depended on these documents:

1. CBC SVOD Requirements for Web Billing Partner (https://wiki.clearleap.com/pages/viewpage.action?title=CBC+SVOD+Requirements+for+Web+Billing+Partner&spaceKey=CS)
2. CBC Content Tiers and Availability Windows (https://wiki.clearleap.com/display/CS/CBC+Content+Tiers+and+Availability+Windows)

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 ecommerce 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. 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 CBC Requirements

The following table lists the requirements that will be supported by the CBC 3.0 system. The CBC 2.0 requirements are included because the Subscriber Manager will be replaced by Cleeng SVOD in version 3.0. Please see Ref 1 for more information.

ID	Description	Notes
CBC 2.0 Requirements		
C2-001	Supported devices: iOS, tvOS, Android, Web App.	
C2-002	User tiers: • Standard (anonymous) • Member (free authenticated) • Premium (paid subscriber)	
C2-003	LoginRadius SSO.	
C2-004	Apple in-app purchasing.	
C2-005	Reduced ads for Premium users.	
C2-006	Recently watched: will be supported for Member & Premium users.	
C2-007	Account settings: Changes to LoginRadius account will be propagated to the rest of the system.	
C2-008	Premium experience across all apps.	
C2-009	Account States and Entitlements.	N/A for CBC 3.0 System
C2-010	Reporting: Supports events reporting defined in CBC-20171006.csv	
C2-011	Language/localization: English Canada.	
C2-012	Creating a new account or logging in with an account that exists in LR, Facebook, or Google but does NOT exist in IBM Subscriber Manager.	
C2-013	"Play" entitlements: The Client apps call getAccount API to verify the user’s entitlement.	
C2-014	Marketing emails: Send target email to Member and Premium users manually by CBC but want it automated in 3.0	

CBC 3.0 Requirements		
C3-001	Supported same devices as CBC 2.0.	See C2-001
C3-002	Feature parity with CBC 2.0.	
C3-003	Content tiers and multiple license windows.	
C3-004	User entitlements: change CBC_SVOD to CBC_MEMEBER and CBC_PREMIUM entitlements.	See C2-002
C3-005	Web billing.	
C3-006	Switch billing platforms.	
C3-007	Security: Credit card data is protected.	
C3-008	Geo-restrictions.	
C3-009	Email: supports Upaknee email provider.	See C2-014
C3-010	Accessibility: web UI must meet Canadian accessibility standards.	
C3-011	PCI Compliance.	
C3-012	Reporting: expanded reporting.	
C3-013	The login API needs to automatically create a new account for new user, thus eliminating the Client from calling the createAccount API.	See C2-012
C3-014	The Client app needs to validate user entitlements by call the getAccount API.	
C3-015	"Play" entitlements: CM will call the “authorize” SMAPI to validate the user’s entitlements.	
CBC 3.1 Requirements (can be moved to CBC 3.0)		
C31-001	Android IAP	
C31-002	Free Premium accounts	

3 CBC 3.0 System User Stories

The following table lists the user stories for the System 3.0 system. These stories were created based on the CBC Requirements.

ID	Description	Notes
US-001	As a user, I want to view online videos from my iOS, tvOS, Android or Web App.	– Covers C2-001, C3-001
US-002	As a user, I want to view online videos with ads as an anonymous user (**Standard** user.)	– Covers C2-002
US-003	As a user, I want to view online videos with ads, but I can personalize my account such as bookmarks, recently watched, and account details page if I become a **Member** user — a registered user without buying a subscription.	– Covers C2-002
US-004	As a user, I want to view online videos with limited ads, and I can personalize my account such as bookmarks, recently watched, and account details page if I become a **Premium** user, a registered user with a subscription purchased using Apple/Google IAP or Web App with my credit card.	– Covers C2-002
US-005	As a user, I want to upgrade from Standard to Member by creating a free account and upgrade from Member to Premium by purchasing a subscription.	– Covers C2-002
US-006	As a user, I want to use LoginRadius SSO to access my services from watch.cbc.ca, client apps, as well as across all cbc.ca sub-domains.	– Covers C2-003
US-007	As an admin, I want to create the following subscription product with the following attributes for all three platforms: Apple Store, Google Play, and Cleeng so my users can purchase them using Apple/Google IAP or Web App with my credit card: Title: CBC SVOD Monthly Subscription product Period: Monthly Trial Period: 1 month Price: $4.99 Tax Rate: CAN Tax Value Currency: CAN	
US-008	As a user, I want to use Apple IAP to purchase the "CBC SVOD Monthly Subscription product."	– Covers C2-004 – See US-007

US-009	As a Premium user, I want no ads on content (with the exception of <10% of content that must always show a house ad.)	– Covers C2-005
US-010	As a subscriber, I want to receive a recently watched shelf that is maintained server-side and updated across all devices.	– Covers C2-006
US-011	As a subscriber, I want to update my account using LoginRadius, and the changes should be propagated to the rest of the system.	– Covers C2-007, C2-012
US-012	As a Premium user, I want to purchase the "CBC SVOD Monthly Subscription product" once using either Apple IAP, Android IAP, or Web App and be able to view the premium content videos from any of these apps: iOS, tvOS, Android, or Web.	– Covers C2-008 – See US-007
US-013	As an admin, I want to receive the events report with the following fields: • Timestamp: UTC format (2017-09-21 18:14:14.487) • Event: can be one of the following values o REGISTER - new account created with Pending status o SUBSCRIBE - account moved from Pending to OK status o CLOSE - account moved from OK to CLOSED status o RENEW - account moved from Closed to OK status o ACCESS VOUCHER - account created in an OK state with operator ID CBCAV • Account_Number: LoginRadius UID • Operator: can be one of the following values: o CBC: o CBC_APPLE: o CBC_ANDROID: o CBC_CLEENG: • Email: LoginRadius Email	– Covers C2-010
US-014	As a user/admin, I want Canada English localization (en_CA) for all apps and web portals.	– Covers C2-011
US-015	As a CBC, I want CBC 3.0 system to support features that are parity with CBC 2.0 system.	– Covers C3-002

US-016	As a user, I want to view videos based on the content tiers and multiple license windows. Please see the Ref 2 for more information.	– Covers C3-003
US-017	As a CM, I want CX to tell me if the current user has the following entitlements: • CBC_MEMBER • CBC_PREMIUM	– Covers C2-014, C2-015, C3-004
US-018	As a user, I want to use my credit card to purchase the "CBC SVOD Monthly Subscription product."	– Covers C3-005 – See US-007
US-019	As a user, I want to be able to switch between the three different billing platforms: Apple Store, Google Play, and Cleeng.	– Covers C3-006
US-020	As a user, I want my credit card information to be stored in a PCI compliance system.	– Covers C3-007, C3-011 – Cleeng requirement
US-021	As a CBC, I want Geo-blocking for Canada geolocation to ensure that the purchaser uses Canadian IP address and credit cards issued by Canadian banks.	– Covers C3-008 – Cleeng requirement
US-022	As a CBC, I want to send the following emails to account holders as follow: • Send targeted email newsletters to users in free trial only (remind them of the Member benefit) • Receipt for subscription sign up ($0 because they are in free trial) • When free trial is about to expire • Receipt for first paid month • Receipt for all subsequent paid month • Credit card expiry notification email • Cancellation of service confirmation email	– Covers C2-014, C3-009
US-023	As a CBC, I want the notification email described in US-022 to be sent out by Upaknee system.	– Covers C3-009 – Recommend for CBC 3.1
US-024	As a CBC, I want the system to support Canadian accessibility standards.	– Covers C3-010 – Cleeng requirement
US-025	As an Admin, I want the following reports from all three billing platforms: Apple Store, Google Play, and Cleeng. • Who is in a free trial? • How much time is left in a free trial? • The timestamp of when a user canceled his/her subscription	– Covers C3-012 – Check if Cleeng support these reports

US-026	As a Client app, I want to just call the login API for both new and existing users.	– Covers C2-012, C3-013
US-027	As a user, I want to use Google IAP to purchase the “CBC SVOD Monthly Subscription product.”	– Covers C31-001 – See US-007
US-028	As a VIP user, I want to access Premium service without purchasing a subscription	– Covers C31-002

4 Architecture Overview

The CBC 3.0 system is the next version of CBC SVOD based OTT service. It will extend the CBC 2.0 system by expanding the SVOD service with support for web billing and Android in-app purchase. Furthermore, the CBC 3.0 system will add the “Content Tiers and Availability Windows” feature that will help CBC to upsell the Premium subscriptions and easier content management for the CBC content team. The following contents will provide the background of the System 2.0 system and then explant more in-depth of the CBC 3.0 system architecture.

4.1 CBC 2.0 System

The CBC 2.0 system (see Figure 4-1) is designed to provide a SVOD service in addition to the existing AVOD service and give CBC users an opportunity to buy a limited ad-free OTT experience with deeper content archives and exclusive offerings (not available in the AVOD product). This experience is going to be consistent across all content (owned and licensed). This would include access to 16 live linear streams, a TVOS application along with Apple IAP.

To support the mentioned features, the IBM Subscriber Manager is integrated with the existing system to provide LoginRadius SSO support via JWT, user account management, and Apple in-app purchase.

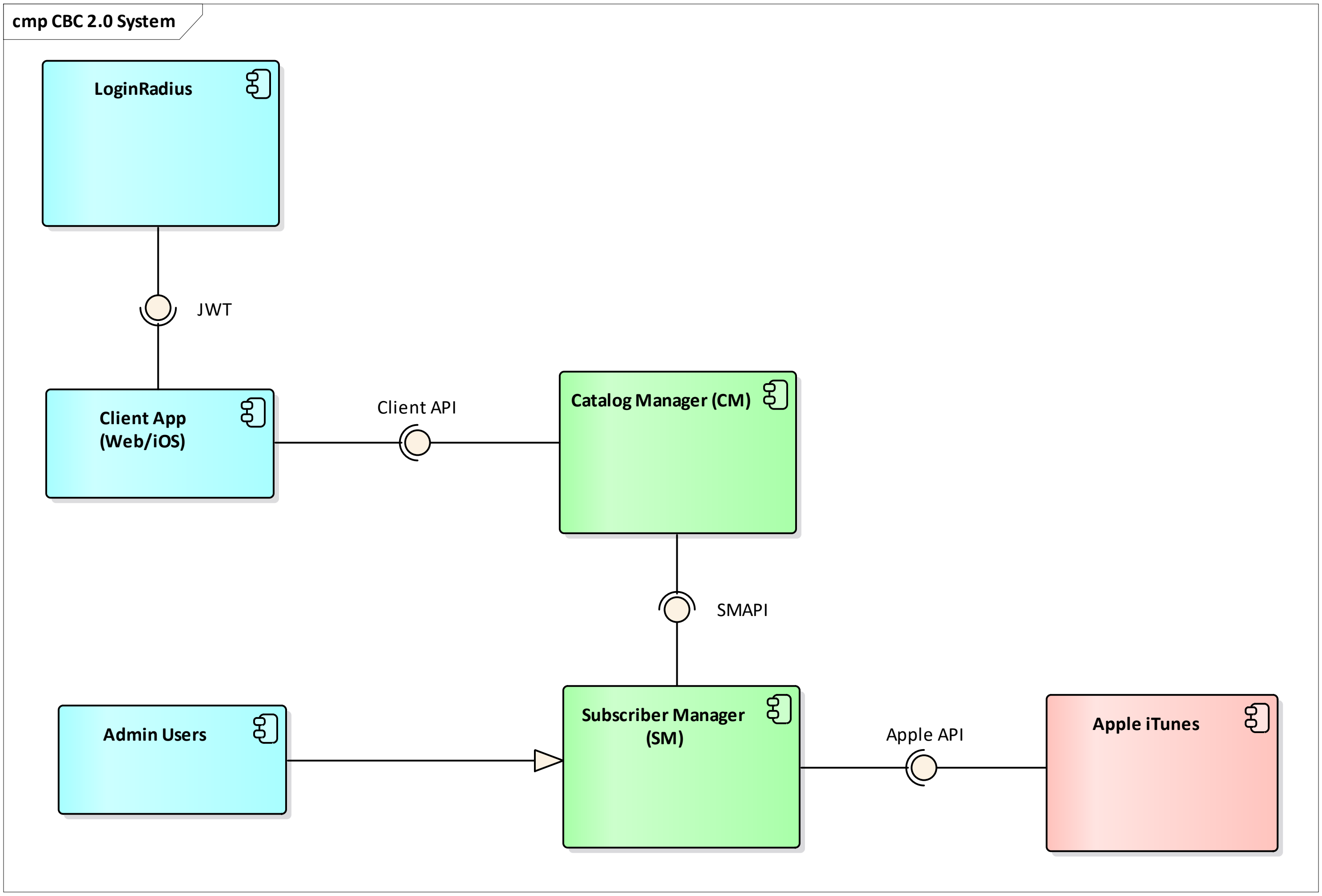
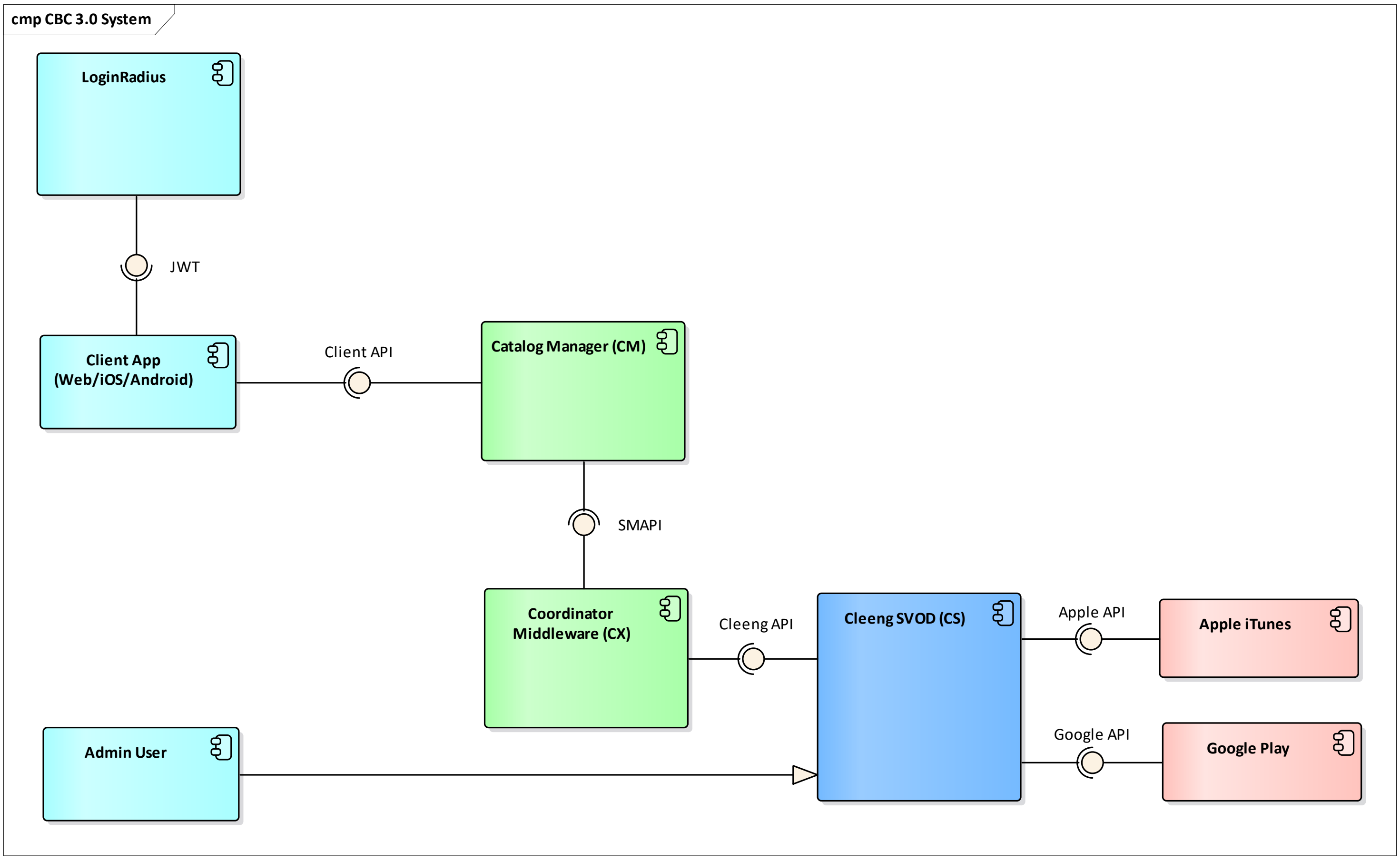


Figure 4-1: CBC 2.0 system

4.2 CBC 3.0 System

Because of the new sophisticated CBC requirements, it has decided that the new CBC 3.0 system will replace the IBM Subscriber Manager with Cleeng SVOD platform. The new IBM Coordinator Middleware (CX) is used to provide drop-in replacement of the Subscriber Manager with Cleeng SVOD platform and backward compatibility with the previous CBC 2.0 system. With the addition of the Cleeng platform, the CBC 3.0 system will provide the following new features and functionalities:

- Continues to use LoginRadius SSO and seamlessly integrates with Cleeng Identity Manager.
- Changing to the user account in LoginRadius will automatically propagate to other components in the system.
- Users can purchase subscriptions using Google IAP and Cleeng credit card payments.
- Client apps can play videos based on "Content Tiers and Availability Windows."
- Users can switch between the three different billing platforms: Apple Store, Google Play, and Cleeng.
- Provides must more comprehensive reports including transactional, subscriber, customers, and offer.



4.3 Coordinator Middleware

This section will delve further into the working of the Coordinator Middleware (CX). The CX is a lightweight middleware providing an integration layer between the Catalog Manager (CM) and the Cleeng SVOD (CS). Also, it provides a backward compatibility layer by supporting the Subscriber Manager (SM) API.

The Figure 4-2 illustrates the following major components of the CX:

- **SMAPI Controller**: provides the SMAPI compatibility layer by mapping the incoming SMAPIs to the Coordinator Service commands.
- **Coordinator Service**: is the brain of the system. It is responsible for processing incoming commands and supervising other components executing the required logic.
- **CxJwt**: is the identity manager component – responsible for handling the JWT from LoginRadius, authenticating user identities, and authorizing user access rights (entitlements).
- **Cleeng Adaptor**: provides standard open API allowing straightforward integration with Subscriber Manager platform such as Cleeng video e-commerce platform.
- **CxDb**: The database module contains the data access objects used to accessing the CX database.

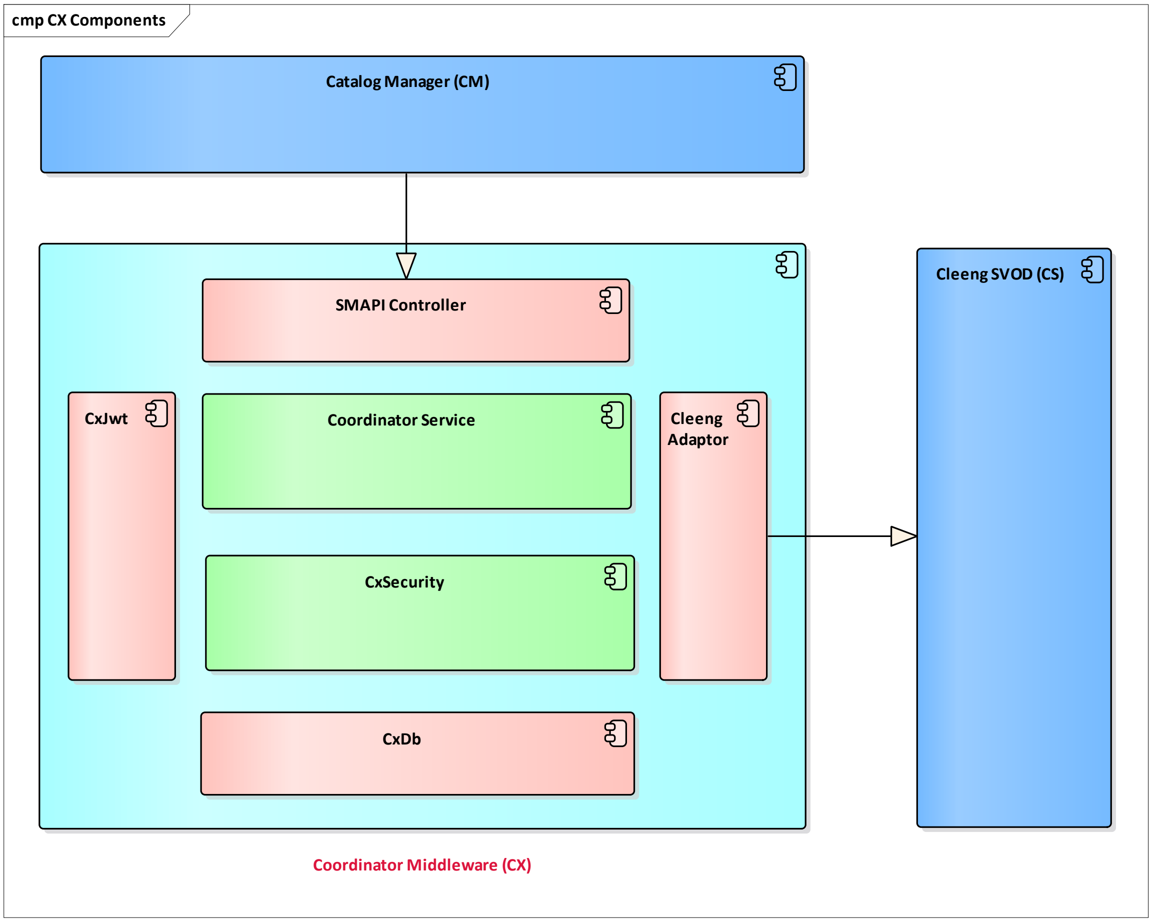


Figure 4-2: Coordinator Middleware

4.4 Entitlements

Note: In CX, the account state is always set to "**ok**" because the user will have access to the contents regardless if he/she purchases a subscription. Thus, it is recommended that the client should use the entitlement to determine which catalog to display to the user.

When the getAccount and authorize APIs is called, CX will determine the user entitlement by first calling the Cleeng's getAccessStatus API to check the current status of the subscription (Cleeng is responsible for monitoring the Apple/Google receipts.)

If the getAccessStatus API return *true*, CX will set the user entitlement to "CBC_PREMIUM" else to "CBC_MEMBER".

Because CX automatically updates the user's entitlement, the Client app just needs to check the entitlement only and can ignore the account state.

4.5 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.

4.5.1 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:

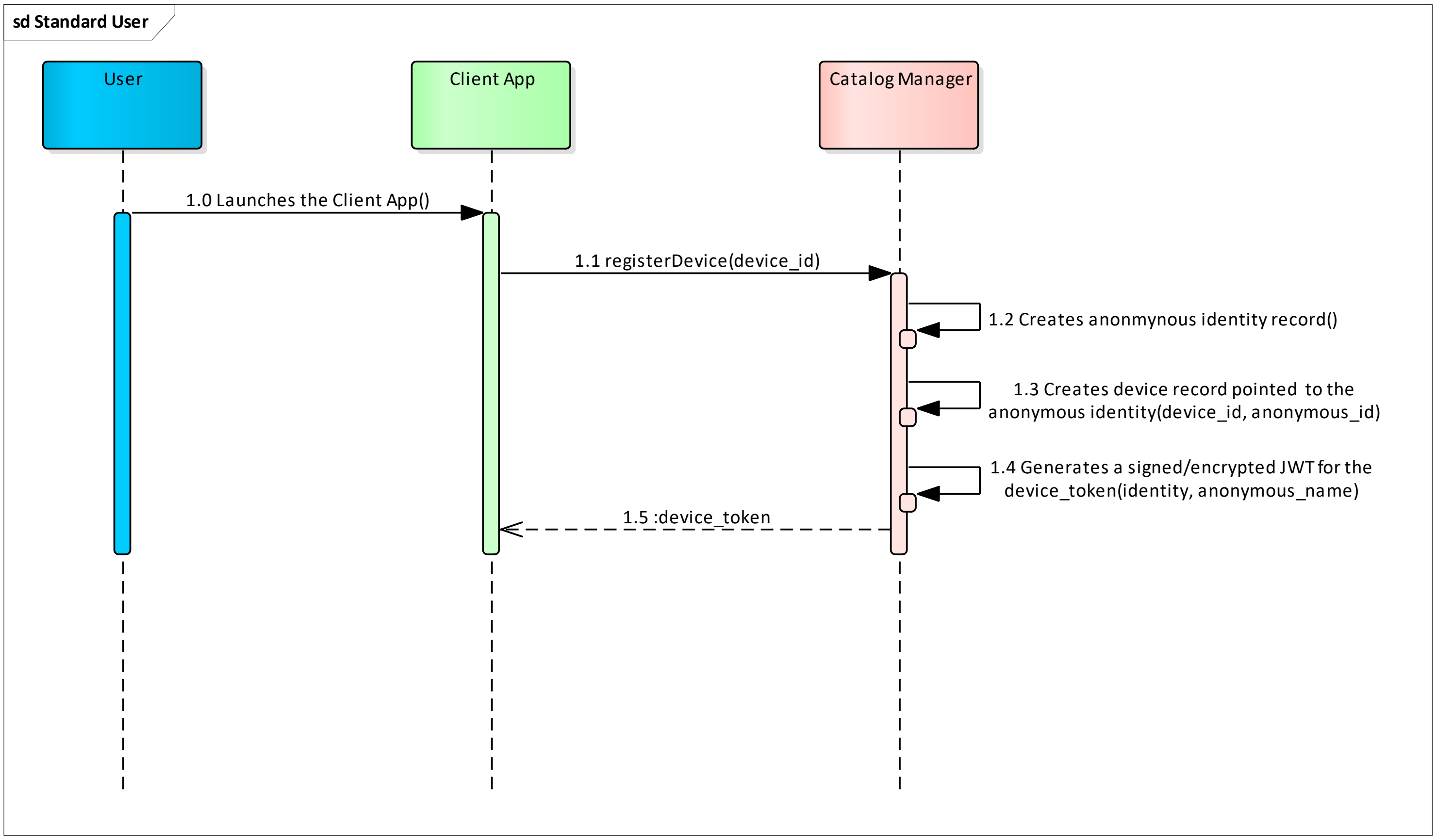
```
{
   "iss": "https://cbc-login-dev.hub.loginradius.com",
   "sub": "890e0f2d19a2442ead480ee139caa505",
   "jti": "0429c165-cea2-4410-a966-8eeb7e77c3c9",
   "iat": 1489172945,
   "nbf": 1489172945,
   "exp": 1489173545,

   "uid": "5535b544-0bb7-4c5a-bf63-d6d0dd01191d",
   "email": "vandarl123@gmail.com",
   "username": "vandarl",
   "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"
}
```

5 CBC 3.0 System Use Cases

This section outlines the different use cases showing how the CBC 3.0 system supports the user stories described in CBC 3.0 System User Stories. Please note that these use cases apply to iOS, Android, and Web apps.

5.1 Standard User

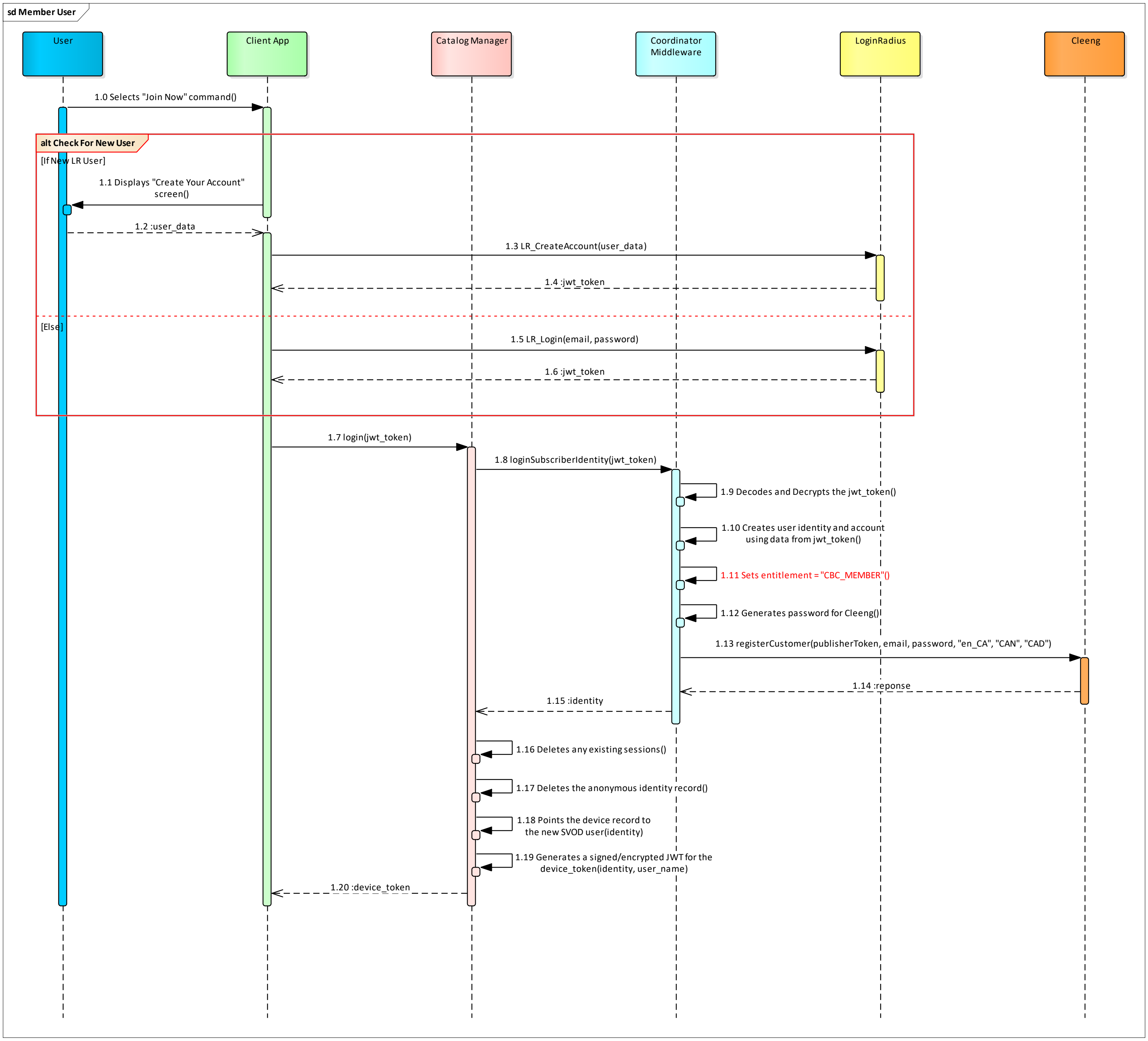
The below sequential flow shows that the Client app needs to call the registerDevice API to identify this user as a Standard user allowing this user to access the AVOD service – an ad-supported experience for current CBC shows and documentaries across multiple platforms and devices such as iOS, Android, Web, Chromecast, etc.


sd Standard User
User
Client App
Catalog Manager
1.0 Launches the Client App()
1.1 registerDevice(device_id)
1.2 Creates anonmynous identity record()
1.3 Creates device record pointed to the anonymous identity(device_id, anonymous_id)
1.4 Generates a signed/encrypted JWT for the device_token(identity, anonymous_name)
1.5 :device_token

5.2 Member User

The Client app needs to perform the following steps to upgrade a Standard user to a Member user:

- **Step 1.0**: The Standard user selects the “Join Now” command to initiate the sign-up process.
- **Steps 1.1 – 1.4**: If the user does not yet register with LoginRadius (LR), the app needs to display the "Create Your Account" screen allowing the user to enter their data then forwards them to LR.
- **Steps 1.5 – 1.6**: However, if the user already registered with LR, the app just needs to log in with LR.
- **Step 1.7**: Next, the app needs to pass the jwt_token to CX allowing it to create the user account with Cleeng platform.
- **Steps 1.8 – 1.11**: After CX updating its database using the data extracted from the decoded jwt_token, it also sets the entitlement to “CBC_MEMBER” which will be used later by CM for creating the “Content Tiers and Availability Windows” (see US-016 user story.)
- **Steps 1.12 – 1.15**: Before CX registers this user with the Cleeng platform, it generates a password and then calls the registerCustomer API.
- **Steps 1.16 – 1.20**: Before returning the device_token to the Client app, CM needs to remove the user previous anonymous data because the user data now resides with CX.

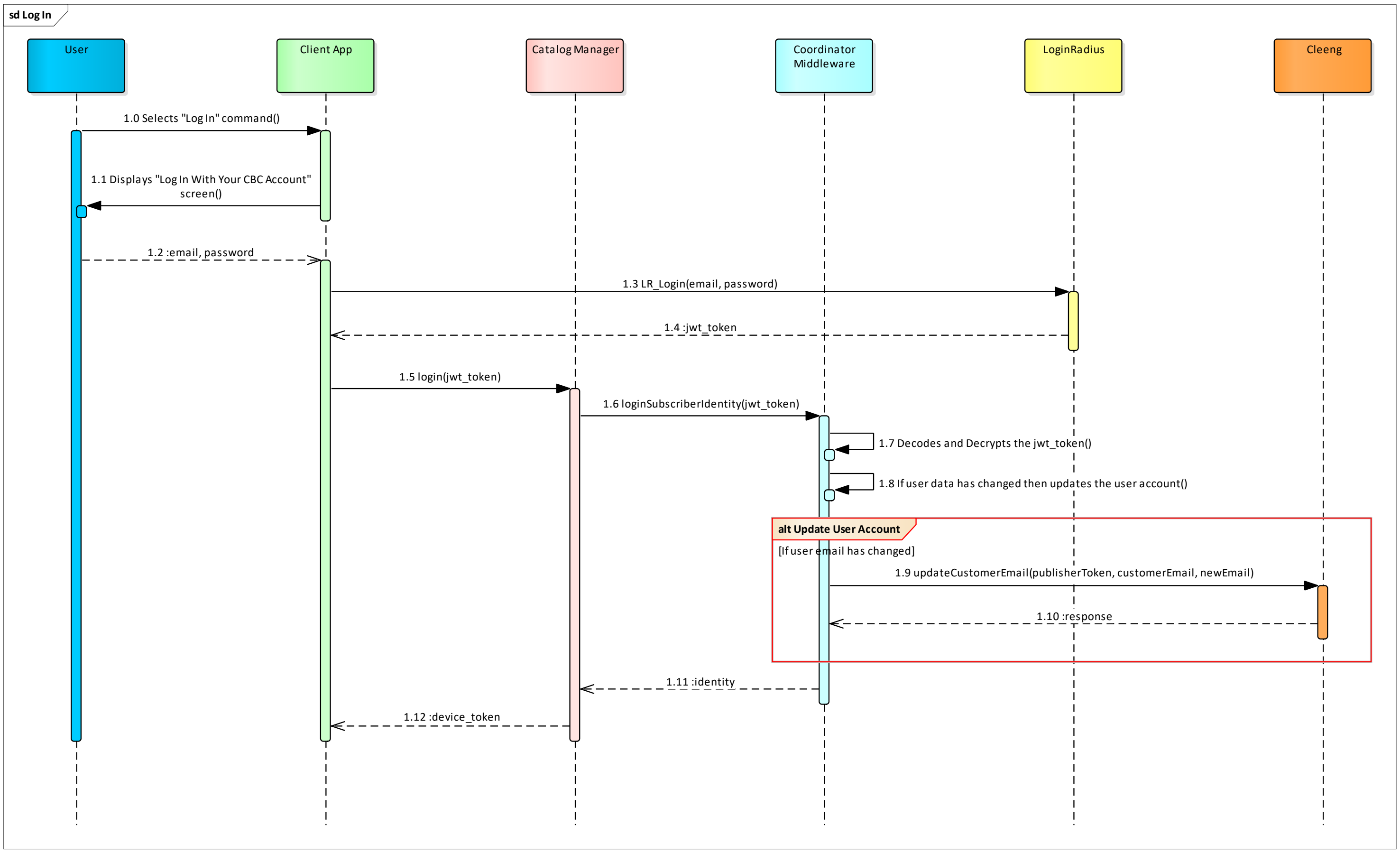

sd Member User
User
Client App
Catalog Manager
Coordinator Middleware
LoginRadius
Cleeng
1.0 Selects "Join Now" command()
alt Check For New User
[If New LR User]
1.1 Displays "Create Your Account" screen()
1.2 :user_data
1.3 LR_CreateAccount(user_data)
1.4 :jwt_token
[Else]
1.5 LR_Login(email, password)
1.6 :jwt_token
1.7 login(jwt_token)
1.8 loginSubscriberIdentity(jwt_token)
1.9 Decodes and Decrypts the jwt_token()
1.10 Creates user identity and account using data from jwt_token()
1.11 Sets entitlement = "CBC_MEMBER"()
1.12 Generates password for Cleeng()
1.13 registerCustomer(publisherToken, email, password, "en_CA", "CAN", "CAD")
1.14 :reponse
1.15 :identity
1.16 Deletes any existing sessions()
1.17 Deletes the anonymous identity record()
1.18 Points the device record to the new SVOD user(identity)
1.19 Generates a signed/encrypted JWT for the device_token(identity, user_name)
1.20 :device_token

5.3 Log In

The Client app needs to perform the following steps to log in a Subscriber (Member or Premium user):

Note: If the user account has been changed in LR then the changes will be automatically propagated to the Cleeng platform by CX (see US- US-006 and US-011 user stories.)

- **Step 1.0**: The Subscriber selects the “Log In” command to initiate the Log In process.
- **Steps 1.1 – 1.2**: In turn, the app displays the Log In screen allowing the user to enter their credential.
- **Steps 1.3 – 1.4**: Next, the app then logs in the user with LR.
- **Steps 1.5**: After successfully logged in and received a valid jwt_token, the app calls the login API to log in with CX. Please note that the login API is also used to create an account with CX (see Member User use case.)
- **Steps 1.6 – 1.12**: If the *changeIndicator* field indicating the user data has changed, CX updates its database and notifies Cleeng platform of the changes.


sd Log In
User
Client App
Catalog Manager
Coordinator Middleware
LoginRadius
Cleeng
1.0 Selects "Log In" command()
1.1 Displays "Log In With Your CBC Account" screen()
1.2 :email, password
1.3 LR_Login(email, password)
1.4 :jwt_token
1.5 login(jwt_token)
1.6 loginSubscriberIdentity(jwt_token)
1.7 Decodes and Decrypts the jwt_token()
1.8 If user data has changed then updates the user account()
alt Update User Account
[If user email has changed]
1.9 updateCustomerEmail(publisherToken, customerEmail, newEmail)
1.10 :response
1.11 :identity
1.12 :device_token

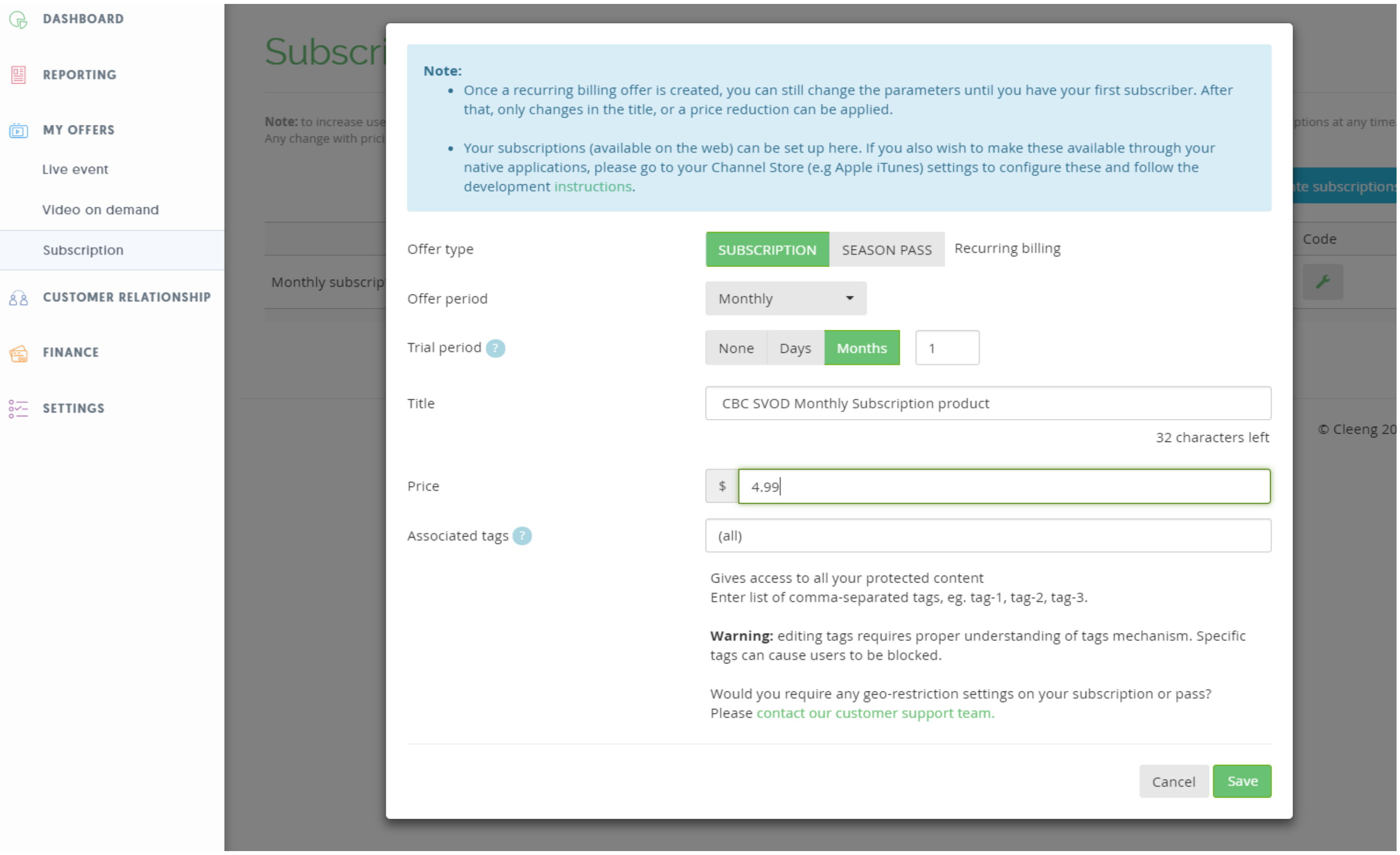
5.4 Premium User

This section shows how to link the subscription products from Apple Store/Google Play to the Cleeng platform by set the Apple Store/Google Play **product ID** to be the same as the Cleeng **offer ID**, allowing the user to purchase the same subscription product from all three billing platforms (see US-012 user story.)

The subscription product for Cleeng platform must be created first before creating the same subscription product for the other two App Stores. The following steps show how to create the "CBC SVOD Monthly Subscription product" (see US-007 user story) for all three billing platforms:

STEP 1. Create the "CBC SVOD Monthly Subscription product" for the Cleeng platform.

1. Go to https://cleeng.com/us/my-account and **log in** to the CBC account.
2. Click on '**Subscription**' tab under the "My Offers' section on the left menu of the Cleeng dashboard. From here new subscriptions can be created.
3. After click on the blue button 'Create subscriptions,' enter the following data and then click on the "Save" button:
   - Offer type: SUBSCRIPTION
   - Offer period: Monthly
   - First period free: Yes
   - Title: CBC SVOD Monthly Subscription product
   - Price: 4.99
   - Associated tags: (all)



Note:
- Once a recurring billing offer is created, you can still change the parameters until you have your first subscriber. After that, only changes in the title, or a price reduction can be applied.
- Your subscriptions (available on the web) can be set up here. If you also wish to make these available through your native applications, please go to your Channel Store (e.g Apple iTunes) settings to configure these and follow the development instructions.

Offer type: SUBSCRIPTION | SEASON PASS | Recurring billing

Offer period: Monthly

Trial period ?: None | Days | Months | 1

Title: CBC SVOD Monthly Subscription product

32 characters left

Price: $ 4.99

Associated tags ?: (all)

Gives access to all your protected content
Enter list of comma-separated tags, eg. tag-1, tag-2, tag-3.

Warning: editing tags requires proper understanding of tags mechanism. Specific tags can cause users to be blocked.

Would you require any geo-restriction settings on your subscription or pass?
Please contact our customer support team.

Cancel | Save

Step 2: Create the “CBC SVOD Monthly Subscription product” for the Apple Store

1. Create an auto-renewable subscription with the Product ID to be the same as the Cleeng’s Offer ID (e.g., S123123123_CAN.) which was created in STEP 1.
2. Enter the following data to match the Cleeng subscription product:
    - Subscription Duration: 1 Month
    - Free Trial: 1 Month
    - Subscription Prices: 4.99 CAN
    - Subscription Display Name: CBC SVOD Monthly Subscription product

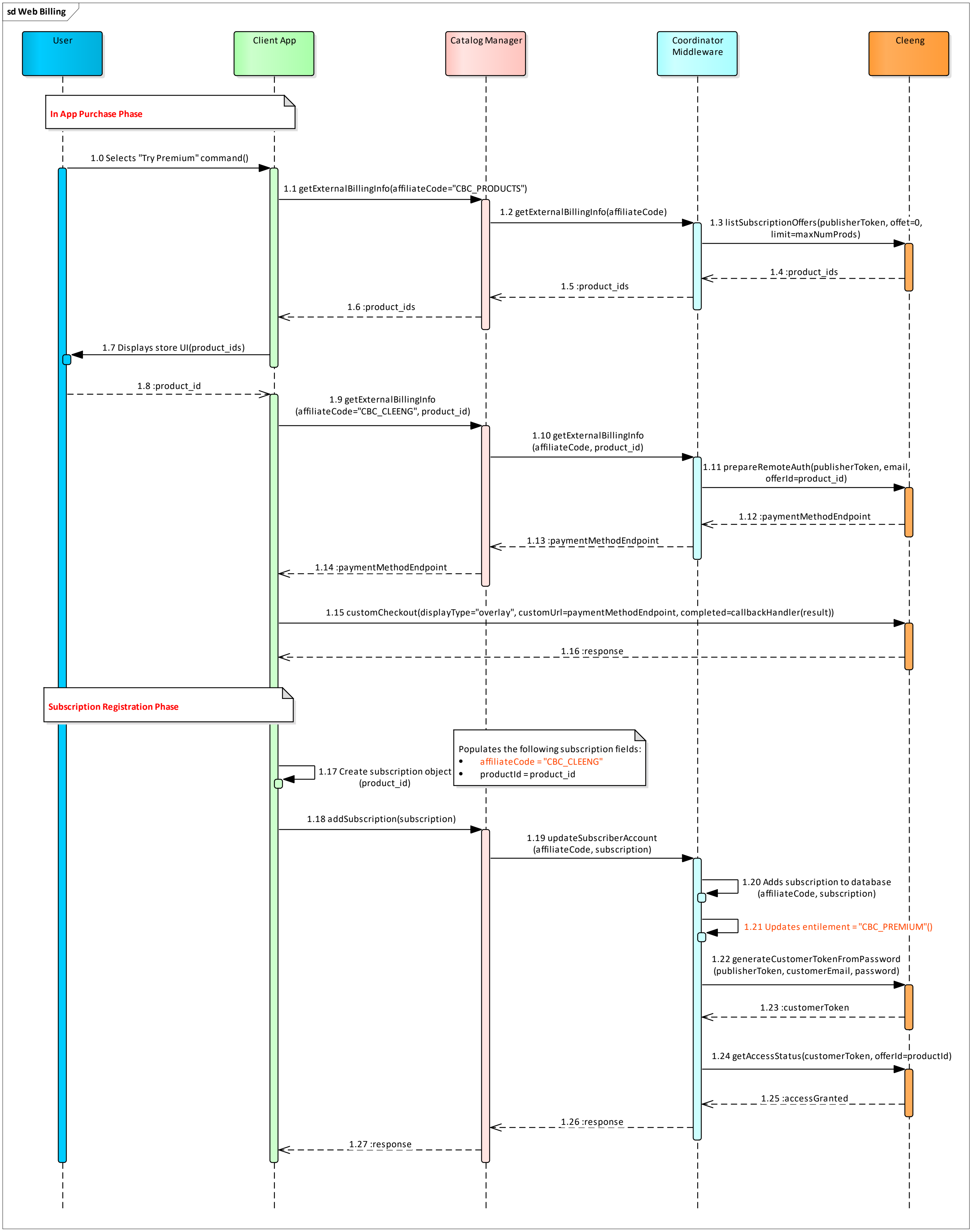
Step 3: Create the “CBC SVOD Monthly Subscription product” for the Google Play

1. Create an auto-renewable subscription with the Product ID to be the same as the Cleeng’s Offer ID but must be in lowercase letters (e.g., s123123123_can.) which was created in STEP 1.
2. Enter the following data to match the Cleeng subscription product:
    - Billing period: Monthly
    - Free trial period: 30 days
    - Prices: 4.99 CAN
    - Title: CBC SVOD Monthly Subscription product

5.5 Web Billing

This use case outlines how a Member user can upgrade to the Premium user by purchasing a subscription using web billing. Before performing the web purchase flow, the subscription products must already be created (see Premium User use case) and the user already logged in (see Log In use case).

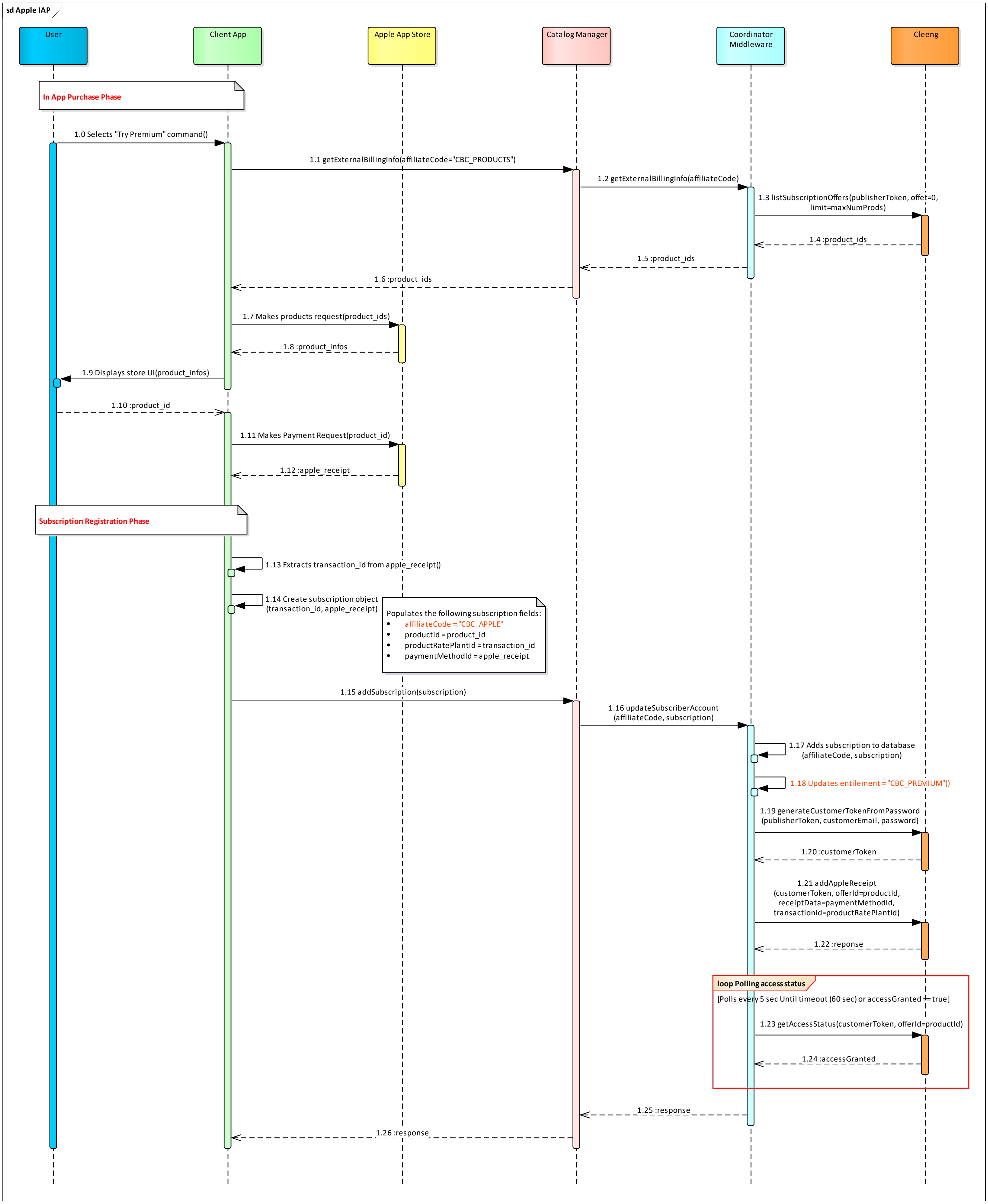
- **Step 1.0**: The Member user selects the “Try Premium” command to initiate the web purchase process.
- **Step 1.1 – 1.6**: The app then calls the getExternalBillingInfo API with the “CBC_PRODUCTS" affiliate code to retrieve the saved subscription product IDs.
- **Step 1.7 – 1.8**: Next, the app displays the store UI screen allowing the user to select the desired product ID.
- **Step 1.9 – 1.14**: The app then calls the getExternalBillingInfo API with the selected product ID, allowing the Cleeng platform to construct the paymentMethodEndpoint.
- **Step 1.15 – 1.16**: The app then calls the customCheckout API with the returned paymentMethodEndpoint, resulting in the user redirects to the Cleeng Checkout for the user to enter their credit card data.
- **Step 1.17 – 1.19:** After returning from Cleeng Checkout, the app calls the addSubscription API to register the user’s subscription with CX.
- **Step 1.20 – 1.27:** After saving the subscription to the database, CX upgrades the user entitlement to “CBC_PREMIUM” which will be used later by CM for creating the “Content Tiers and Availability Windows” (see US-016 user story.) It then calls Cleeng to validate the user’s subscription.


sd Web Billing
User
Client App
Catalog Manager
Coordinator Middleware
Cleeng
In App Purchase Phase
1.0 Selects "Try Premium" command()
1.1 getExternalBillingInfo(affiliateCode="CBC_PRODUCTS")
1.2 getExternalBillingInfo(affiliateCode)
1.3 listSubscriptionOffers(publisherToken, offet=0, limit=maxNumProds)
1.4 :product_ids
1.5 :product_ids
1.6 :product_ids
1.7 Displays store UI(product_ids)
1.8 :product_id
1.9 getExternalBillingInfo (affiliateCode="CBC_CLEENG", product_id)
1.10 getExternalBillingInfo (affiliateCode, product_id)
1.11 prepareRemoteAuth(publisherToken, email, offerId=product_id)
1.12 :paymentMethodEndpoint
1.13 :paymentMethodEndpoint
1.14 :paymentMethodEndpoint
1.15 customCheckout(displayType="overlay", customUrl=paymentMethodEndpoint, completed=callbackHandler(result))
1.16 :response
Subscription Registration Phase
Populates the following subscription fields:
• affiliateCode = "CBC_CLEENG"
• productId = product_id
1.17 Create subscription object (product_id)
1.18 addSubscription(subscription)
1.19 updateSubscriberAccount (affiliateCode, subscription)
1.20 Adds subscription to database (affiliateCode, subscription)
1.21 Updates entilement = "CBC_PREMIUM"()
1.22 generateCustomerTokenFromPassword (publisherToken, customerEmail, password)
1.23 :customerToken
1.24 getAccessStatus(customerToken, offerId=productId)
1.25 :accessGranted
1.26 :response
1.27 :response

5.6 Apple IAP

This use case outlines how a Member user can upgrade to the Premium user by purchasing a subscription using Apple in-app purchase. Before performing the in-app purchase flow, the subscription products must already be created (see Premium User use case) and the user already logged in (see Log In use case).

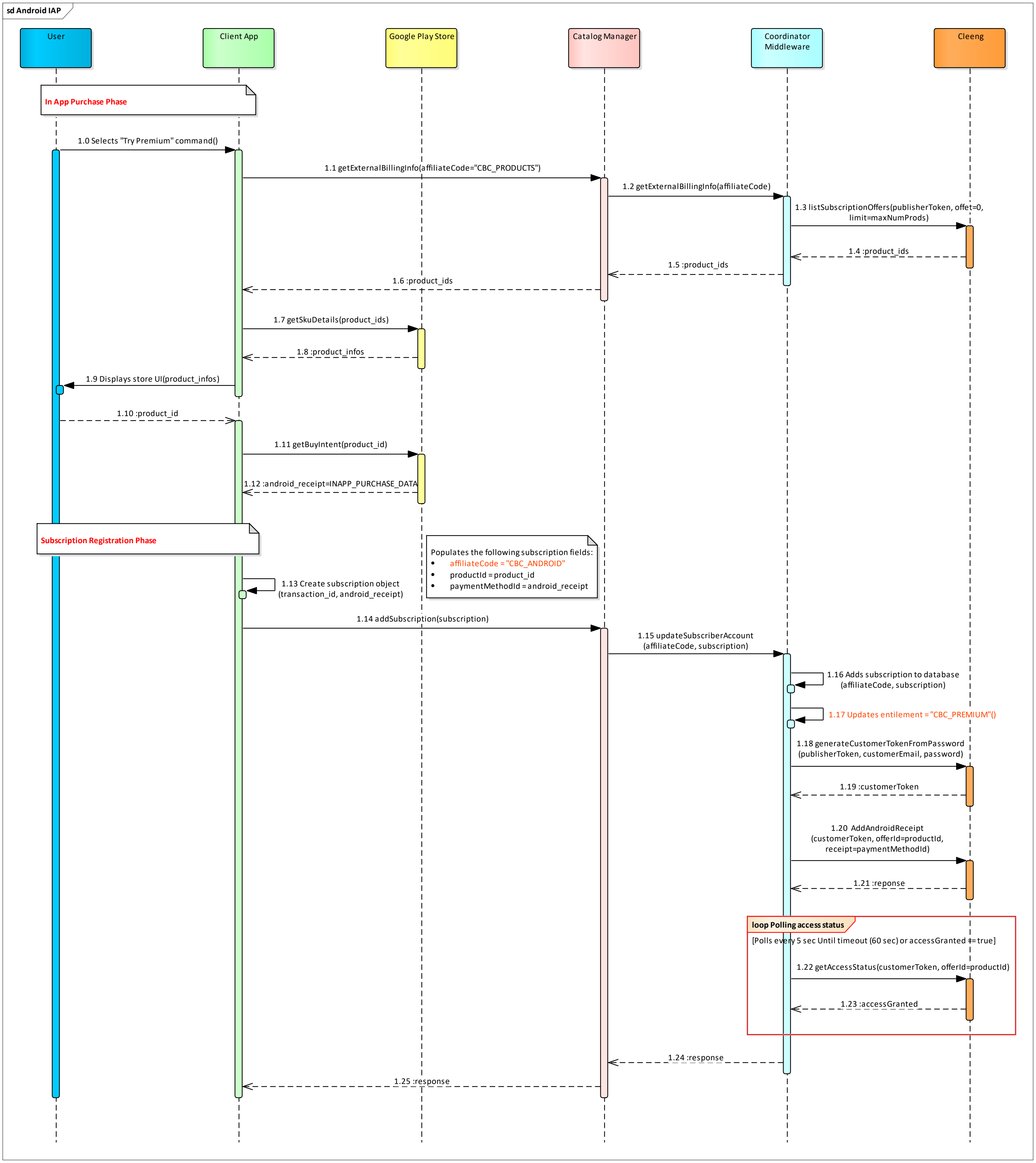
- **Step 1.0**: The Member user selects the "Try Premium" command to initiate the in-app purchase process.
- **Step 1.1 – 1.6**: The app then calls the getExternalBillingInfo API with the "CBC_PRODUCTS" affiliate code to retrieve the saved subscription product IDs.
- **Step 1.7 – 1.8:** The app sends a request to the Apple Store to get the product information.
- **Step 1.9 – 1.10**: The app displays the product information, allowing the user to select the desired product ID.
- **Step 1.11 – 1.12**: The app sends a payment request with the selected product to the Apple Store, which processes payment and returns the receipt for completed transaction.
- **Step 1.13 – 1.16:** After returning from Apple Checkout, the app calls the addSubscription API to register the user's subscription with CX.
- **Step 1.17 – 1.27:** After saving the subscription to the database, CX upgrades the user entitlement to "CBC_PREMIUM" which will be used later by CM for creating the "Content Tiers and Availability Windows" (see US-016 user story.) It then calls Cleeng to validate the user's subscription.


sd Apple IAP
User
Client App
Apple App Store
Catalog Manager
Coordinator Middleware
Cleeng
In App Purchase Phase
1.0 Selects "Try Premium" command()
1.1 getExternalBillingInfo(affiliateCode="CBC_PRODUCTS")
1.2 getExternalBillingInfo(affiliateCode)
1.3 listSubscriptionOffers(publisherToken, offet=0, limit=maxNumProds)
1.4 :product_ids
1.5 :product_ids
1.6 :product_ids
1.7 Makes products request(product_ids)
1.8 :product_infos
1.9 Displays store UI(product_infos)
1.10 :product_id
1.11 Makes Payment Request(product_id)
1.12 :apple_receipt
Subscription Registration Phase
1.13 Extracts transaction_id from apple_receipt()
1.14 Create subscription object (transaction_id, apple_receipt)
Populates the following subscription fields:
• affiliateCode = "CBC_APPLE"
• productId = product_id
• productRatePlantId = transaction_id
• paymentMethodId = apple_receipt
1.15 addSubscription(subscription)
1.16 updateSubscriberAccount (affiliateCode, subscription)
1.17 Adds subscription to database (affiliateCode, subscription)
1.18 Updates entilement = "CBC_PREMIUM"()
1.19 generateCustomerTokenFromPassword (publisherToken, customerEmail, password)
1.20 :customerToken
1.21 addAppleReceipt (customerToken, offerId=productId, receiptData=paymentMethodId, transactionId=productRatePlantId)
1.22 :reponse
loop Polling access status
[Polls every 5 sec Until timeout (60 sec) or accessGranted == true]
1.23 getAccessStatus(customerToken, offerId=productId)
1.24 :accessGranted
1.25 :response
1.26 :response

5.7 Android IAP

This use case outlines how a Member user can upgrade to the Premium user by purchasing a subscription using Android in-app purchase. Before performing the in-app purchase flow, the subscription products must already be created (see Premium User use case) and the user already logged in (see Log In use case).

- **Step 1.0**: The Member user selects the "Try Premium" command to initiate the in-app purchase process.
- **Step 1.1 – 1.6**: The app then calls the getExternalBillingInfo API with the "CBC_PRODUCTS" affiliate code to retrieve the saved subscription product IDs.
- **Step 1.7 – 1.8:** The app sends a request to the Google Play to get the product information.
- **Step 1.9 – 1.10**: The app displays the product information, allowing the user to select the desired product ID.
- **Step 1.11 – 1.12**: The app sends a payment request with the selected product to the Google Play, which processes payment and returns the receipt for completed transaction.
- **Step 1.13 – 1.15:** After returning from Apple Checkout, the app calls the addSubscription API to register the user's subscription with CX.
- **Step 1.16 – 1.27:** After saving the subscription to the database, CX upgrades the user entitlement to "CBC_PREMIUM" which will be used later by CM for creating the "Content Tiers and Availability Windows" (see US-016 user story.) It then calls Cleeng to validate the user's subscription.


sd Android IAP
User
Client App
Google Play Store
Catalog Manager
Coordinator Middleware
Cleeng
In App Purchase Phase
1.0 Selects "Try Premium" command()
1.1 getExternalBillingInfo(affiliateCode="CBC_PRODUCTS")
1.2 getExternalBillingInfo(affiliateCode)
1.3 listSubscriptionOffers(publisherToken, offet=0, limit=maxNumProds)
1.4 :product_ids
1.5 :product_ids
1.6 :product_ids
1.7 getSkuDetails(product_ids)
1.8 :product_infos
1.9 Displays store UI(product_infos)
1.10 :product_id
1.11 getBuyIntent(product_id)
1.12 :android_receipt=INAPP_PURCHASE_DATA
Subscription Registration Phase
1.13 Create subscription object (transaction_id, android_receipt)
Populates the following subscription fields:
• affiliateCode = "CBC_ANDROID"
• productId = product_id
• paymentMethodId = android_receipt
1.14 addSubscription(subscription)
1.15 updateSubscriberAccount (affiliateCode, subscription)
1.16 Adds subscription to database (affiliateCode, subscription)
1.17 Updates entilement = "CBC_PREMIUM"()
1.18 generateCustomerTokenFromPassword (publisherToken, customerEmail, password)
1.19 :customerToken
1.20 AddAndroidReceipt (customerToken, offerId=productId, receipt=paymentMethodId)
1.21 :reponse
loop Polling access status
[Polls every 5 sec Until timeout (60 sec) or accessGranted = true]
1.22 getAccessStatus(customerToken, offerId=productId)
1.23 :accessGranted
1.24 :response
1.25 :response