COMPASS Client Integration Guide

Version 3.1 SR10

It's smarter, it's safer. It's VO.

Copyright

The contents of this documentation are strictly confidential and the receiver is obliged to use them exclusively for his or her own purposes as defined in the contractual relationship. No part of Viaccess-Orca applications or this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without permission in writing from Viaccess S.A and/or Viaccess-Orca Israel Ltd.

The information in this document is subject to change without notice. Neither Viaccess S.A nor Viaccess-Orca Israel Ltd warrants that this document is error free. If you find any error in this documentation or wish to make any comment, please report them to Viaccess-Orca in writing at documentation@viaccess-orca.com.

Trademarks

Viaccess-Orca is a trademark of Viaccess S.A ® in France and/or other countries. All other products and company names mentioned herein are the trademarks of their respective owner.

Viaccess S.A and or Viaccess-Orca Israel Ltd may hold patents, patent applications, trademarks, copyrights or other intellectual property rights over the products described in this document. Unless expressly specified otherwise in a written license agreement, the delivery of this document does not imply the concession of any license over these patents, trademarks, copyrights or other intellectual property rights.

Table of Contents

1 Introduction

COMPASS is mainly a backend solution, but it was designed with the user experience in mind. COMPASS APIs enable the creation of a user interface that will be intuitive, with fast and easy content navigation, naturally embedded into the viewing experience.

COMPASS open and flexible APIs provide the freedom to design any viewer interface on any device. COMPASS's smart business rules, along with its blending and filtration option, allow marketing teams to plan viewer interactions that fulfill business goals.

This guide describes COMPASS best practices for the integration of recommendations and Social TV into the TV service user interface.

Note

COMPASS Social features are not supported for EU based customers.

1.1 COMPASS Interfaces

A standard COMPASS implementation includes two integration points between COMPASS and the service middleware: the backend and the client, as illustrated in the following figure.

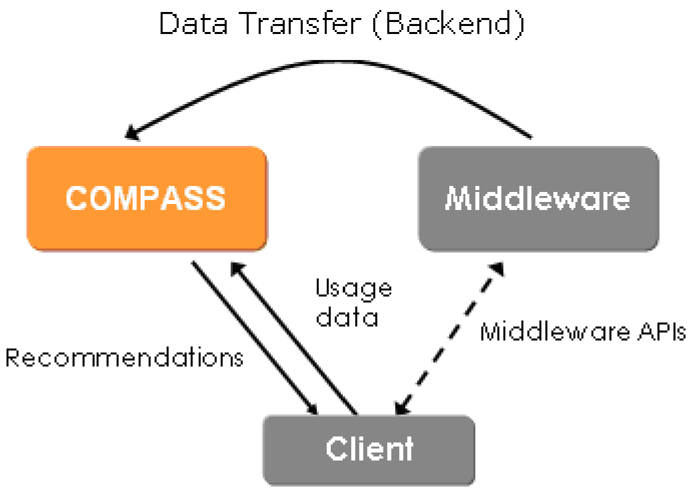


Figure 1: A standard COMPASS implementation

Note

The details of the backend integration processes are specific to the IPTV middleware deployed.

The COMPASS backend interface is described in the COMPASS Backend Integration Guide.

1.2 In this Guide

This guide includes the following chapters:

Topic	Contents
2 Version Information	Describes any changes made in this document since the previous version.
3 Households and Profiles	Describes how to enable service subscribers to receive recommendations based on a household profile.
4 Usage Data Collection	Describes Usage Data Collection within COMPASS for on-demand, and live content, and the search function.
5 Recommendations	Explains the various types of content Recommendations in COMPASS.
6 Social TV	Explains how to use the Social TV features within COMPASS. **Note** COMPASS Social features are not supported for EU based customers.
7 Search8 Error Handling	Describes error handling in COMPASS Client Integration.

2 Version Information

This topic describes changes made to this document since the previous release.

This chapter includes the following sections:

2.1 What's New

The following changes were made in recent versions.

2.1.1 3.1 SR7

- Search on VOD content can return a representative season of a series or a season.
- Search on Live content can return the next matching episode (in the future) or all matching episodes.
- The wish list can be configured to keep items that have already been watched.

2.2 Deprecation

No deprecations were performed since the previous version.

3 Households and Profiles

This topic provides guidelines for enabling service subscribers to enjoy recommendations.

A profile is the basis for COMPASS generated personalized recommendations. The profile is linked to an account, which is represented by the household entity. This entity can be either imported from the service backend system or created on the fly upon self-registration.

This chapter includes the following sections:

3.1 Authentication

Personalized recommendations are provided by COMPASS in accordance with COMPASS's ability to identify the profile requesting the recommendations. The following categories exist:

Users that are authenticated in the system – users that are authenticated in COMPASS enjoy all the personal services that COMPASS can provide.

Users that are not authenticated in the system – also referred to as anonymous users. These users enjoy a subset of the full COMPASS-related functionality, since they cannot be identified and therefore, no personal data is available for COMPASS. Refer to the COMPASS Overview document for more information.

Before recommendation requests are sent to COMPASS, the client implementation should define whether the request requires authentication of the user or not. This depends on the required user experience. If the user is authenticated by COMPASS, then COMPASS can provide personal recommendations based on the user profile and order history, alongside statistical recommendations (e.g., most popular recommendations).

If the user does not go through the authentication process, COMPASS does not receive profile identification for the user. COMPASS can however provide recommendations for users that are not authenticated to the service yet, but where there is some general information available on their viewing preferences. Statistical recommendations can also be provided.

3.2 Household

The household entity in COMPASS holds the live and VOD subscriptions for the household. COMPASS recommendations are filtered by household subscriptions by default, in order to provide viewers with the most relevant recommendations for them.

A household can be created in COMPASS in one of two modes:

- **Offline** – This is the default method. Using this method the households are imported from the backoffice systems. For more information refer to the COMPASS Backend Integration Guide.

- **On the fly** – This mode is relevant for services in which the subscription is created online. The household needs to be created on the fly in COMPASS for the viewer to enjoy VOD recommendations immediately after first access to the service. The services assigned to a household created on the fly will be services specified when creating the household on the fly or, in case no services are specified, those defined to be auto-subscription services, which are available to all subscribers. Linear recommendations are also provided at this stage. For a household that is subscribed to channel packages, the profiles of this household will be able to receive recommendations on linear content within the household packages. For more information refer to the COMPASS RT API Developers Guide, to the CreateHousehold API.

3.3 Profile

Since profiles are the basis for COMPASS to generate sets of personalized recommendations, an active profile needs to be set either by the viewer, or automatically by the client.

A profile is linked to a specific household and needs to be given a unique name, enabling the identification of viewer profiles and assignment to a client device within the household. A default profile name can be defined in the system and multiple profiles can be created for a specific household. The viewer can then fine tune his profile by actively providing personal information, including his name, an icon that distinguishes his profile from other profiles, and also other information like favorite genres and channels.

3.3.1 Creating a Profile

A profile needs to be defined in the client in order for COMPASS to be able to provide recommendations.

3.3.1.1 Profile Image

The profile can be identified also by an image selected by the viewer. The source of this image can be a list of avatars from which to choose, provided by the operator, or an image loaded by the viewer.

The user interface may require that the image be displayed in a variety of sizes for display on different screens. The images are identified by a specified name known by the user interface (i.e., profile_images_big; profile_image_small).

> **Note**
>
> When a user joins Social TV, then the profile image is used by default and then the application enables using dedicated images relevant to Social TV, for example images taken from the Facebook account which are linked to the TV profile.
>
> > **Note**
> >
> > COMPASS Social features are not supported for EU based customers.

Use the CreateProfileImage API to create the profile image. For more information on the APIs for managing the profile images, refer to the COMPASS RT API Developers Guide.

3.3.1.2 Profile PIN

COMPASS enables the definition of a four-digit PIN per profile. This personal code can be used as part of the profile authentication in COMPASS. This PIN, if defined, is used also when accessing the Social TV section in the SUI, in order to prevent other household members from entering this private area.

The operator can use a proprietary authentication mechanism if one exists.

3.3.1.3 Profile Additional Information

COMPASS provides the ability to add entity information to a profile that is not in the structured data mode, but is needed for functionality other than the COMPASS logic. To do this, the profile can have properties that are defined in the COMPASS Admin, and then added to the profile entity using the updateProfile API.

3.3.2 Providing Profile Explicit Preferences

COMPASS enables the viewer to explicitly provide some information on his personal preferences, as described below.

3.3.2.1 Explicit Genres and Channels

The viewer can provide information on his preferred genres and channels (UpdateProfileChannelList and UpdateProfileGenreList).

3.3.2.2 Profile's Exclude List

COMPASS enables the viewer to exclude specific content and genres from the recommendations provided to him (AddToProfileExcludeList).

The viewer can change this indication by accessing the list of content marked to be excluded and either select specific content and change the action (RemoveFromProfileExcludeList) or clear the entire list (ClearProfileExcludeList).

3.3.2.3 VOD Wishlist

COMPASS enables the viewer to manage a list of items that may be of interest to him at a future time, and not at the moment when he browses the service catalog.

COMPASS provides APIs for adding (AddToWishList), removing (RemoveFromWishList) and displaying content items in a wishlist (GetProfileWishList).

3.3.3 Profile Creation and Authentication Flow

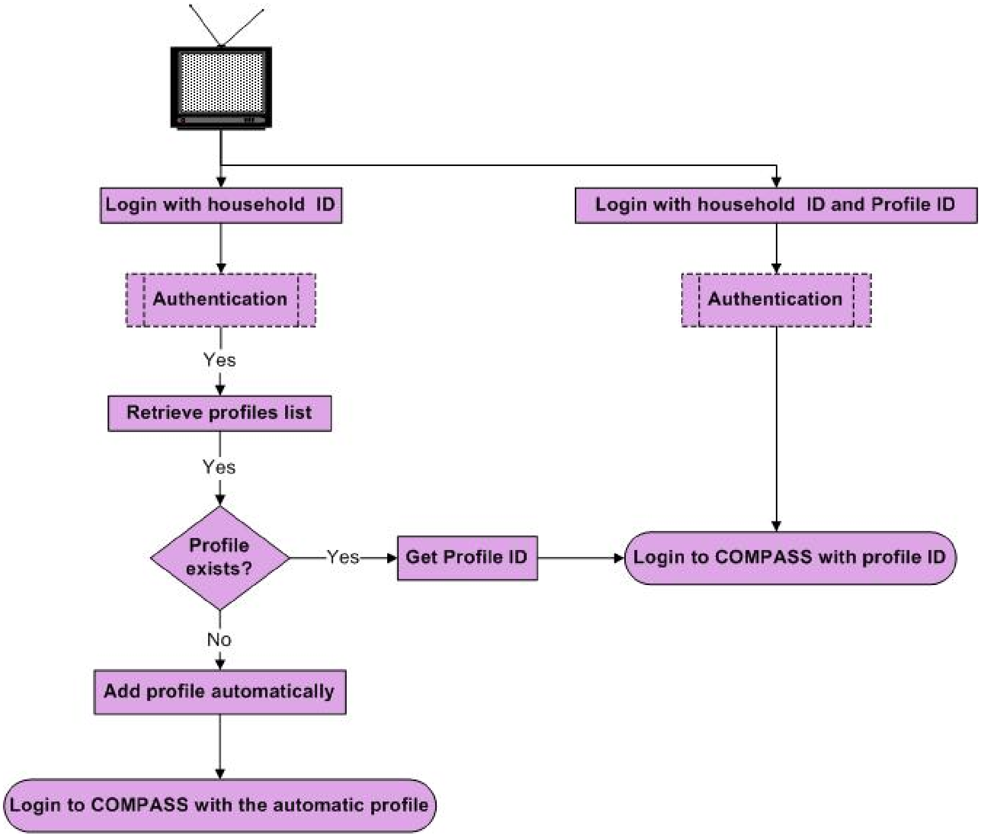

Login with household ID
Login with household ID and Profile ID
Authentication
Authentication
Yes
Retrieve profiles list
Yes
Profile exists?
Yes
Get Profile ID
Login to COMPASS with profile ID
No
Add profile automatically
Login to COMPASS with the automatic profile

Figure 2: Profile Creation flow diagram

COMPASS supports the authentication of either a specific household or a specific profile (within the household).

If the authentication is performed based on the household, then the profile list for that household must be retrieved and authentication based on one of the profiles in the list is performed. When no profile exists for the household provided, a profile can be automatically created for the household. This option is usually effective when the profile is not exposed in the user interface.

If both the household and profile identifiers are provided, then profile authentication by COMPASS can be performed in a single step.

The authentication process should be customized to integrate with the middleware needs and security requirements. The default implementation provided by COMPASS requires the viewer to login to RiGHTv, and then uses the RiGHTv cookie in order to securely identify the household ID.

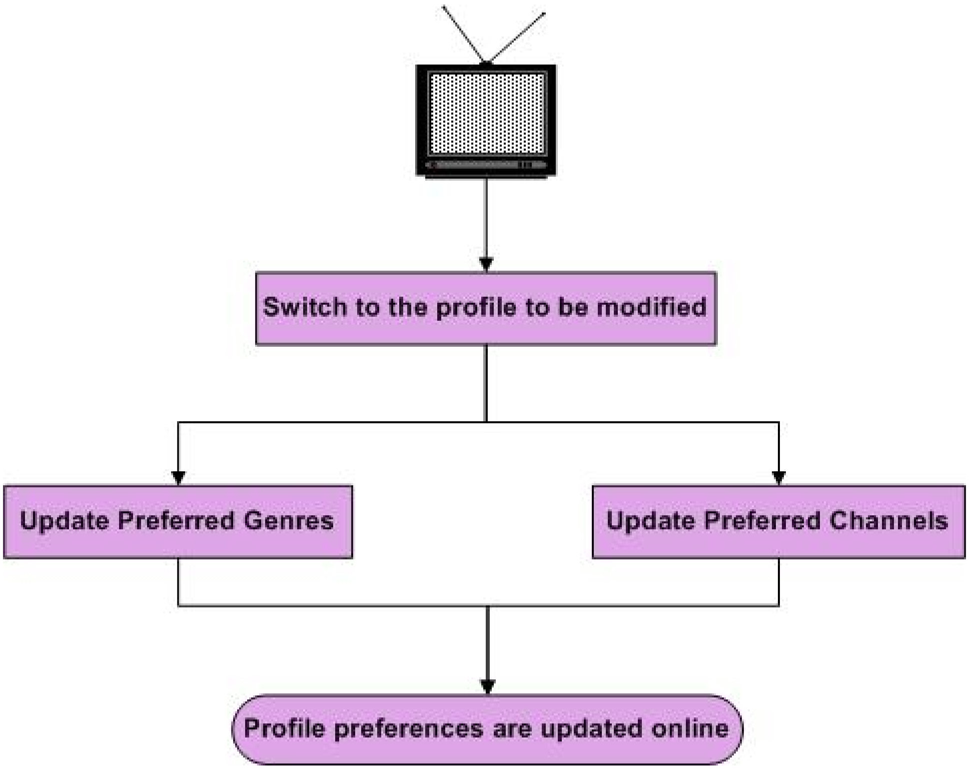

Switch to the profile to be modified
Update Preferred Genres
Update Preferred Channels
Profile preferences are updated online

Figure 3: Update Profile explicit preferences flow diagram

The COMPASS profile represents the profile viewing habits deduced from the usage data, if data exists, and from information the profile explicitly submitted into the system. The explicit information provided by the viewer is updated online and therefore applied automatically, the next time the profile is used to retrieve recommendations.

Note

For a suspended profile, the only data used for personal recommendations is information the profile explicitly submitted to the system.

3.4 Opt-in/Opt-out Functionality

COMPASS recommendations are based on learning the profile's behavior by collecting usage data from the client. As soon as the profile is defined (as described above), usage data representing the viewer's behavior is collected and assigned to that profile.

Although the main purpose of collecting usage data is to personalize the user experience for the viewers, anonymized usage data is still very valuable, since it enables identifying viewing patterns that can be then taken into account when generating recommendations.

A profile can be created with either Active or Suspended status:

- **Active status** means that personal usage data is automatically collected for that profile, and there may be a need for an Opt-out option in the client user interface for viewers who do not want to have their personal usage data registered.
- **Suspended status** means that no personal usage data is registered for that profile, and there is a need for an Opt-in option in the client to enable personal usage data collection whereby the viewer can get personal recommendations.

The usage data can be either anonymized for suspended users (i.e., treat the events as if they arrived from an anonymous user) or discarded. Anonymization refers to order, rating, and zap events. Refer to the relevant section of the COMPASS Administration Guide for more information.

If the usage data of opted out users is to be discarded it is recommended not to send it to the server for the suspended profiles.

Active users can choose to delete any personal data that was collected in the service. In this case, COMPASS will anonymize the data without it. This action is enabled while the profile is still in Active status.

In case the user changes their status from Active to Suspended, or if an Active user deletes the usage data collected but remains in Active status, the following actions are performed in COMPASS:

- The profile channels, genres, viewed/favorite seasons, and wishlist are deleted.
- Order and rate history are deleted.
- Collaborative Filtering processed usage data (the user interests) are anonymized.

This can be achieved by using the DeleteProfileHistory API. For more information, refer to the COMPASS RT API Developer’s Guide.

Note

If COMPASS is displayed for suspended profiles, these profiles get non-personal recommendations based on statistics and the default configuration of the system. The opt-in/opt-out options need to be described in a simple and precise way so that the viewer understands the value of the personal recommendations generated based on the usage data.

4 Usage Data Collection

This topic describes Usage Data Collection within COMPASS for on-demand and live content, and the search function.

In order to record the viewer's behavior and be able to monitor COMPASS effectiveness, it is important to properly record all relevant actions performed by the viewer in the client.

Each of the events sent to the COMPASS server needs to include the information relevant for further analysis; for example: an indication whether the action was performed from COMPASS (as a result of a recommendation) or from the regular interface. These properties are described separately in the API documentation.

Recording these actions requires the client to send usage data to COMPASS per profile, using the RegisterEvent API, as described in the next topic.

This chapter includes the following sections:

4.1 Usage Data for On-Demand Content

This topic refers to traditional on demand content and also to catch up TV content offered as on demand.

The simple definition for **on demand** content is content that is viewed by the subscribers upon request. **Catch up** content in the context of COMPASS refers to television shows originally delivered as linear TV, that are then offered to viewers as VOD for a limited time period, soon after it was aired.

COMPASS handles Catch up TV in a very similar way to regular VOD, except for the provisioning. Since Catch up TV can be provisioned much more frequently than regular VOD, COMPASS provisioning now captures VOD content that changes frequently during the day. COMPASS also handles usage data registered for this content with a delay, to make sure that all usage data is registered for catch up content, even if the provisioning process has not ended.

For on-demand and catch up content, the usage data registered for a specific profile are:

Action	RegisterEvent API – Event Type
Order placement	Order
Rating entry	Rate
Viewing the content until the end	Content_end
Adding content to the profile's personal wish list	Update_wishlist

These actions provide an indication that the viewer showed interest in the movie.

4.1.1 Order Placement

The order action is handled differently depending on the business model in which the movie is offered. The following are some examples of different business models for purchasing on-demand, and a short description of the way this must be implemented in the COMPASS client.

4.1.1.1 Regular On-Demand Rental

This business model provides subscribers unlimited access to a specific movie for a limited time, paying on a per transaction basis.

For every order performed by the subscriber, the client must send an event to the COMPASS server.

4.1.1.2 Prepaid VOD

This model allows subscribers to purchase a pre-defined number of rented movies for a specific amount.

The implementation of this in terms of COMPASS is the same as for regular on-demand movie rental, as described previously.

4.1.1.3 Subscription Video-On-Demand

This model provides subscribers unlimited access to a variable set of movies for a flat fee. This model is particularly appropriate for Catch up content.

Assuming that in this model the viewers have access to the set of movies, but still need to order each movie separately, the implementation of this in terms of COMPASS is the same as for regular on-demand movie rental, as described previously.

4.1.1.4 VOD Package

This model provides viewers unlimited access to a few specific movies for a limited time, paying on a per package basis.

For the implementation of this order in terms of COMPASS, for every movie within the package, the client must send an event to the COMPASS server.

4.1.2 Rating a Movie

When a viewer rates a movie, an event is sent to COMPASS. Assuming that viewers convey their true feelings about a movie, this is very useful data for COMPASS.

4.1.3 Preview

When a preview request to see a movie is made, an event is sent to COMPASS. This provides some idea as to whether or not the viewer is interested in the content.

4.1.4 Content End

A content end event is sent to COMPASS when a profile stops viewing an on-demand video, within a configurable time period before the video ends, or at the video end.

Implementing this event is valuable for improving recommendations from the collaborative filtering engine for cases in which the only usage data available is on-demand content orders. The content end event indicates that the viewer reached the end of the content. In this case COMPASS concludes that the viewer liked the content. The alternative is to have only order events, which are less informative as to the viewer's taste.

4.1.5 Add/Remove from Wish List

When there is a request to add a movie to a wish list or favorites list, an event should be sent to COMPASS. This provides an indication as to whether the viewer is interested or not in the content.

If the profile wish list is managed using COMPASS APIs, there is no need to implement this event explicitly. When wish list functionality is implemented outside of COMPASS, registering the wish list information in COMPASS can be accomplished by using the RegisterEvent API and setting the following properties as:

```
type = update_wish_list
add_to_list=true
```

When removing a content item from the profile's wish list, the action should be registered with the *add_to_list* property, set to *false*.

By default watched content items are removed from the wish list automatically. To keep content items on the wish list even after they are watched, set the parameter *remove-from-wishlist-on-order* in the Configuration to *false*. In this case content items that were watched are not automatically removed from the wish list and can be removed manually using the RemoveFromWishList API. Contact VO Professional Service for information on changing the Configuration.

4.2 Usage Data for Live Content

For live programs, COMPASS refers to the channel zapping, set reminder, and PVR actions performed by the profile. These actions provide an indication that the subscriber shows interest in the program.

In addition, COMPASS handles specific usage data which is used to capture the preferences of the viewer in regard to series content offered in linear TV.

For linear content, the usage data registered for a specific profile are:

Action	RegisterEvent API – Event Type	Comments
Channel zapping	Zap	
Set Reminder	Reminder	
PVR	Record	
Record	Record_series	The program ID refers to a single episode. add_to_series_record
Play	Play	

These actions provide an indication that the viewer showed interest in the movie.

4.2.1 Channel Zapping

A channel zap event is sent to COMPASS when a profile switches to a new channel and remains in it for a configurable time period. This time period represents a reasonable time period from which it can be recognized that the viewer showed interest in the program.

4.2.2 Set Reminder

A set reminder event is sent to COMPASS when a viewer sets a reminder to view a program in the future. This action indicates that the viewer is interested in the program.

4.2.3 PVR

4.2.3.1 Record

A PVR event is sent to COMPASS when a profile requests to record a program (which is not categorized as series content – refer to 4.2.3.3 Play below) scheduled to be broadcast in the future. This action indicates that the viewer is interested in the program.

For the calculation of the series preferred by the viewer, COMPASS requires the registration of two additional events: series recording and the playback of series content already recorded. Refer to the topics that follow, 4.2.3.2 Series Recording and 4.2.3.3 Play.

4.2.3.2 Series Recording

The events of recording an entire series or cancelling a series recording request are sent to COMPASS. These actions provide a significant indication of the interest level of a viewer for a series.

4.2.3.3 Play

The play action of a previously recorded episode, which is part of a series, is sent to COMPASS. This action provides a significant indication of the interest of a viewer for the series.

4.3 Usage Data for Monitoring and Analytics

The following is a description of the usage data collected by COMPASS for providing statistics on COMPASS performance.

In order to effectively monitor the recommendation system, it is important to send the most accurate usage data, when possible. You should also provide as much information as possible with this event. One of the interesting types of information that can be sent is whether the action was performed from a recommendation.

4.3.1 Impressions

The list of recommendations displayed in a single screen are sent to COMPASS. In the case when the viewer scrolls to see more recommendations, the new recommendations displayed should be registered in COMPASS by sending a new impressions event.

This information is significant for understanding the effectiveness of COMPASS. Any actions performed on recommendations can be measured in comparison to the impressions events for a specified time frame.

4.3.2 Clicks

The clicks on content are sent to COMPASS. This information is significant for understanding the effectiveness of COMPASS.

4.4 Usage Data for Search

The following is a description of the usage data collected by COMPASS Search for providing auto-suggest capabilities.

4.4.1 Search Conversion

The content item that was eventually selected by the viewer, among the search results he received, is used by COMPASS Search for calculating the popularity of the search results. The most popular results are considered the most likely to answer what the viewer is seeking. The results are ordered by these criteria in the Auto-Suggest feature.

Each time the viewer selects a search result for further navigation, an action should be sent to the COMPASS Search server. In this way, viewers can be presented immediately with the results most likely to match what they are seeking.

4.5 Usage Data Optimization

In order to optimize the usage event collection, it is advisable not to send events to the COMPASS server for content that is not provisioned in COMPASS. For example: If COMPASS is configured not to provision adult content, the client needs to validate the parental control level of content when sending events to the COMPASS server. This ensures that the traffic is not loaded unnecessarily.

5 Recommendations

This topic explains the various types of content Recommendations in COMPASS.

This chapter includes the following sections:

5.1 Recommendations for Different Content Types

COMPASS identifies three different content types: season, program, and movie. In some cases these content types require different handling when generating recommendations, due to their different nature. For example, popularity calculations for an individual movie are different than the popularity calculations for a season.

COMPASS entities contain a contentType property with this indication. Refer to the COMPASS RT API documentation for more information.

For a detailed description of these types refer to the COMPASS Overview document.

5.2 Requesting Recommendations

COMPASS generates recommendations through a variety of built-in engines, providing the most effective recommendations for the content available in the service. COMPASS APIs enable retrieving recommendations from a single specific engine, for example, to display a list of most popular content in the past week in a general location in the client, or from a mix of engines using a blend that can be configured in the COMPASS in Voyage TVE Console GUI.

The Blends screen enables configuring multiple blends in order to match the desired user experience in each screen. For more information refer to the COMPASS Administration Guide.

5.2.1 Personal Recommendations (for a profile)

The request for recommendations from COMPASS is done per content type: On-Demand and/or Live. These recommendations can be provided out of a mix from the different engines, as defined in the blends defined in the COMPASS in Voyage TVE Console GUI.

It is recommended to retrieve more recommendations for each type than the number of recommendations that are displayed in COMPASS, in order to minimize the requests to the server and enable the subscriber to request more recommendations, with minimal response time. In the client implementation, the request for more recommendations can be either for both content types or per each content type separately.

The number of recommendations to be retrieved for a specific profile needs to take into account the following:

- The number of recommendations to be retrieved for live content takes into account the fact that live programs change rapidly in comparison to on-demand content, which is more static.
- The recommendation list can be filtered, based on the viewer's viewed on-demand content.

5.2.2 Recommendation Reason

Recommendations that are displayed with supporting information on why the recommendations were provided create trust and can speed up the user's purchasing decision.

For this, as part of a VideoRecommendation object returned for a GetLiveRecommendationList request, COMPASS provides an explanation, if one exists, for the recommendation.

The Recommendation Reason response format provides an easy way to develop the client side app to clearly identify the type of reason that is provided per content.

The VideoRecommendation object contains a RecommendationReason object, which contains a liked content object. A new property called "reasonTemplateCode" was added.
This property contains a value that indicates the type of reason returned for the content.

The following table contains some examples.

ReasonTemplateCode	**Description**	**Example for recommendation Reason text**
1000	Most Popular Movies recommendations will indicate the reference time period that is the basis of the popularity calculation. The values can be: Last week, Last month, All Times.	"The Shawshank Redemption (1994) is the most popular movie this month.
2000	External rating	Top IMDB
3000	Promotions	"Skyfall (2012) can be purchased at 50% off during this month.

ReasonTemplateCode	Description	Example for recommendation Reason text
4000	Personal – When there is no single content in the user’s viewing history that can be identified as the most significant influence on the specific recommendation, the recommendation is based on several contents with the same contribution, and only the engine name will be provided as part of the recommendation information. This can be seen above as “type=cf”. This means that the recommendations were generated by the Collaborative Filtering engine.	“Just for you – watch Suits”
5000	This code indicates that the personal reason includes information on a liked content. See below.	
10000	When there is one content in the user’s viewing history that has the most significant influence on the specific recommendation, it will be provided.	Because you liked “The Good Wife”
10001	When there are shared genres between the two contents, they will be returned.	Because you like “Medical Drama”
10010	When there are shared contributors between the two contents, they will be returned.	Because you like “Julliana Margolis”
11000	When the percentage of users who viewed a content in the user’s viewing history and also viewed the recommended content is significant, it will be provided too.	75% of people who like “The Good Wife” also liked “Suits”

The example below refers to the reason template code = 11011. This is when there is a liked content, and usage data facts and shared genres and contributors:

Example

```
<VideoRecommendation id="51" externalContentId="51"
name="Reservoir dogs" rating="4.0" ratersCount="1" prLevel="0"
prName="0" contentType="Movie" type="cf">
```

```
      <genres>
         <Genre id="3170" name="Drama"
externalId="3170"></Genre>
              <Genre id="3186" name="Crime"
externalId="3186"></Genre>
             </genres>
      <availabilities>
         <ServiceAvailability videoId="33" serviceId="1"
startTime="1447585497000"
endTime="1457595497000"></ServiceAvailability>
      </availabilities>
      <recommendationReasons>
         <RecommendationReason type="likedContent”
reasonTemplateCode = 5000>
<LikedContents>
               <LikedContent contentExternalId="31"
contentName="Pulp Fiction" similarityPercentage="72.0"
“reasonTemplateCode = 11011>
                  <genres>
                     <Genre id="3170" name="Drama"
externalId="3170"></Genre>
                          <Genre id="3186" name="Crime"
externalId="3186"></Genre>
                  </genres>
                  <contributors>
                     <CastPerson castId="2" personName="Martin
Scorsese"></CastPerson>
                  </contributors>
               </LikedContent>
            </LikedContents>
         </RecommendationReason>
      </recommendationReasons>
   </VideoRecommendation>
```

5.2.3 Recommendations for Unauthenticated Users

Recommendations can be generated for users that are in the service for the first time. In this case, the user can be prompted to select a small number of preferred movies from a list. Then the application can immediately provide the user with a list of personal recommendations, based on the selected items.

For this, the GetVideoRecommendationsList can be used with either the related content engine or the collaborative filtering engine, by providing the list of IDs of the relevant entities in the params property.

5.2.4 Recommendations from a Specific Engine

COMPASS APIs enable retrieving recommendations from a specific engine separately.

This can be used, for example, to display the list of most popular content in the past week in a general location in the client.

5.2.5 Recommendations from Specific Services

COMPASS APIs enable retrieving recommendations for a specific list of services.

This can be used, for example, to display the most popular recommendations for a specific VOD service available to the subscribers.

5.2.6 Recommendations from Specific Source Channels

COMPASS APIs enable retrieving recommendations for specific source channels. Such a request will return catch up content that was originally aired on at least one of the channels specified in the request.

5.2.7 Recommendations per On Demand Category

When the viewer browses the catalog for different categories, recommendations can be requested in the context of the specific category. Recommendations per category are enabled when requesting recommendations from all engines. The category set in the API should be the category ID.

Below are some example use cases that are relevant for recommendations per category:

1. Personal recommendations per category – the target user experience is to provide a personal user experience as much as possible. For this, per each category a personal blend can be defined for the recommendations per category.
   The result will be that for some of the categories that the engines can provide personal recommendations, they will be displayed. For all other categories, a list of the most recent content in the category, with the relevant context defined in the recommendations request, will be displayed. This is the blend inherent fallback mechanism, in order provide good recommendations and avoid an empty response.
2. VOD landing page - the target user experience is to provide different lists of recommendations for a pre-defined list of categories.
   The GetRecommendationList API enables setting multiple category IDs to retrieve recommendations from and also define that the result should be grouped by category.

5.2.8 Live Recommendations - Catch Up Recommendations

Recommendations are referred to as ‘catch up only’ for programs whose broadcast ended and are now available as catch up. Viewer series and related content engines return content available as catch up, in addition to linear content that is currently scheduled to be broadcasted in the future. All filters applied on recommendations (for example, channel, genre,and device) are applied on catch up recommendations.
The type of linear content to be returned in the response can be set in the GetLiveRecommendationList and it can include linear programs, catch up programs, or both - depending on the target user experience defined.

5.2.9 Filtering Recommendations by Subscription

COMPASS APIs enable retrieving recommendations for a profile filtered by the profile's subscriptions. This provides the most relevant recommendations. This avoids making recommendations for content that is not accessible to the subscriber.

COMPASS filters recommendations by subscription by default, to provide the most relevant recommendations to the viewer. Promotions provided by COMPASS are an exception, they are configured by default to be provided out of the user's subscription. This configuration can be overridden by the actual recommendations request sent by the client application as follows:

		Recommendations request from the client app	
		Within subscriptions	**Out of subscriptions**
Promotions engine Configuration	**Within subscriptions**	Within subscriptions	Out of subscriptions
	Out of subscriptions	Out of subscriptions (other recommendations are within subscriptions)	Out of subscriptions

5.2.10 Filtering Recommendations by Technical Properties

COMPASS APIs enable retrieving recommendations for a profile, filtered by the actual device type requesting recommendations or by specific technical specifications of the device from which the recommendations are requested. This provides the most relevant recommendations. This avoids making recommendations for content that cannot be viewed by this device.

If information about the device type's content can be viewed and it is available in the service and provisioned by COMPASS, then recommendations can be filtered by the device type. Another option would be to refer to specific characteristics relevant to the technical capabilities of the device, like encoding, delivery mode (download or streaming) and quality class, which relate to the definition of the content (HD/SD/3D).

5.2.11 Filter Viewed On-Demand Content

To improve the effectiveness of the recommendations provided by COMPASS, and provide the viewer with recommendations on new content, the content already viewed by the viewer should be filtered out of the recommendations provided to him.

By default COMPASS is configured to filter out content viewed by the viewer from any recommendations request. This configuration is set in the COMPASS in Voyage TVE Console GUI, and can be overridden in any specific recommendations request.

In addition COMPASS enables filtering out the viewed content in the client site. The list of viewed content can be retrieved for the current profile for a pre-defined period of time. This value is part of the COMPASS configuration, with a default value of 6 months back.

It is not mandatory to retrieve the list of viewed content with every recommendations request. However, this list should then be updated in the client periodically, specifically when new on-demand content is purchased by the viewer.

For more information refer to the COMPASS Administration Guide and the COMPASS RT API Developers Guide.

5.2.12 Filter by Genres per Dayparts

COMPASS enables defining a list of genres to be excluded in recommendations requests during specific segments of the day. This list is defined per COMPASS platform. This parameter can optimize the recommendations in deployments where family profiles are more common than personal profiles and therefore specific genres can flood statistical recommendations. For example: children's content can be very popular in a service, and therefore appear in the most popular content at all hours, even at night. For more information on this parameter, contact Orca professional services.

When requesting recommendations, the request can override these settings, if it was defined.

5.2.13 Recommendations for a Content Item

The collaborative filtering engine and the related content engine can provide recommendations on a specific item, in addition to their ability to provide personal recommendations for a given profile. The Blends screen enables configuring a blend relevant for recommendations on a specific content item. For more information refer to the COMPASS in Voyage TVE Console GUI.

Recommendations can also be enabled on most recent contents produced, based on the year of production as defined by the creation date in the service platform. This allows for retrieving VOD recommendations for a specific year or years of production timeframe; for example, 'Provide recommendations for content from 2014-2016'.

5.2.14 Recommendations Retrieval Flow

COMPASS RT APIs are provided for retrieving recommendations from the COMPASS server. There are separate APIs for on-demand recommendations and for live recommendations. These APIs also enable different filters for providing different sets of recommendations for various client-side requirements. For more information, refer to the COMPASS RT API Developer Guide.

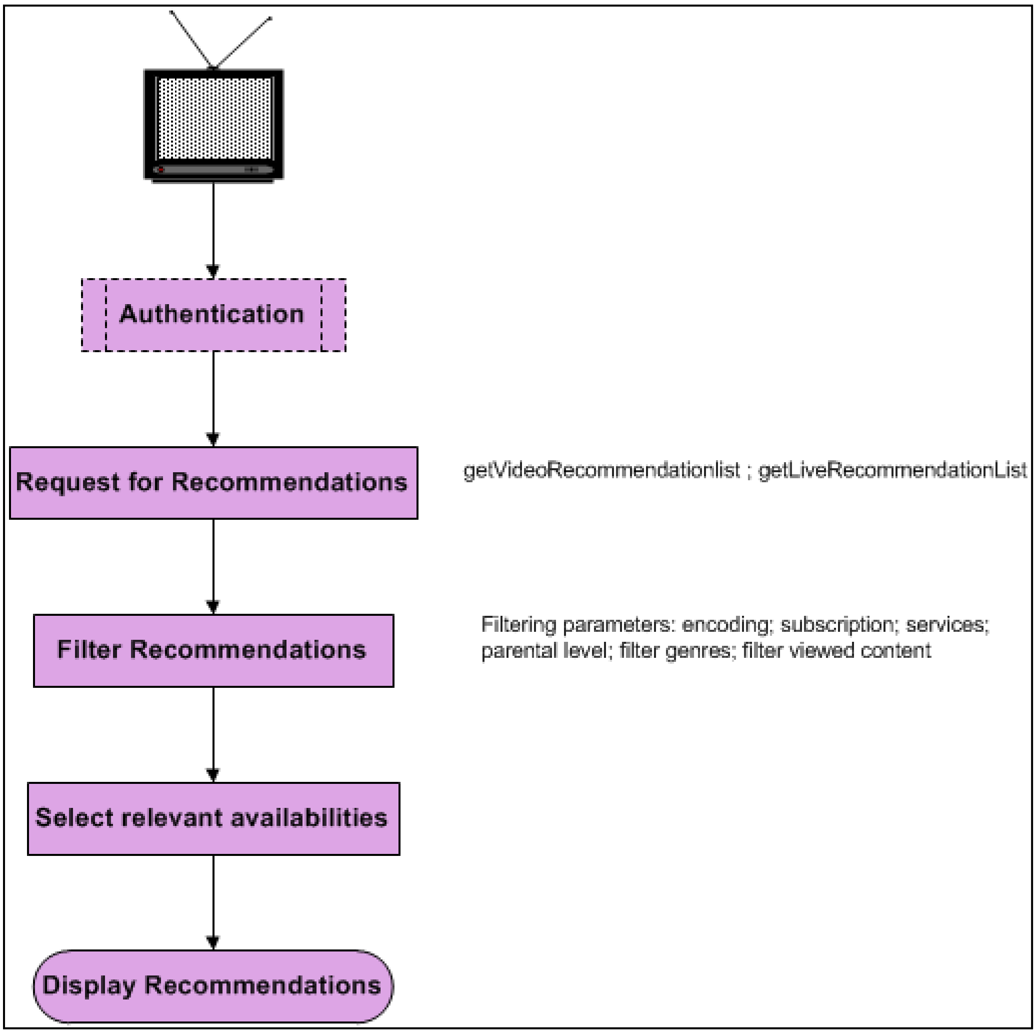

Authentication
Request for Recommendations
getVideoRecommendationlist ; getLiveRecommendationList
Filter Recommendations
Filtering parameters: encoding; subscription; services; parental level; filter genres; filter viewed content
Select relevant availabilities
Display Recommendations

Figure 7: Recommendation Retrieval flow diagram

The response of the recommendation retrieval APIs is a set of recommended content items with the relevant availabilities: the relevant instances in which the content is available in the service. The availabilities are usually used in the client implementation in order to enable the user to access the recommended content (e.g., place an order) directly from the displayed recommendation.

5.2.15 Recommendations on Series Content

Some COMPASS engines provide recommendations on series content.

The recommendations vary by engine and can be of type series, season or episode entity. This section provides some guidelines on how to handle these types of recommendations when developing the user interface.

The relevant engines are as follows:

Content	Engine	Recommendation
	Collaborative Filtering	Series
VOD	Related Content	Series
	Most Popular Series	Series
	Top Rated Series	Series
	Viewer Series	Episode (either from the currently viewed season or the first episode for the next season).
	Promotions	Season or episode
	External Rating	Season or episode
Live	Promotions	Episode
	Viewer Series	Episode
	Related content	Episode

The basis of the recommendations models remains a season. However, in order to provide a recommendation on a series, COMPASS marks a single season to represent a series in the specific recommendation model. Therefore, the recommendations' lists consist of a single season per series with series metadata for display.

For example, before COMPASS 3.0, the most popular series list would consist of seasons. Therefore, the list could consist of more than one season for a series. The new concept defines the unit of reference to be a series and not a season. Therefore there would be a single instance for a series within the list. This is clearer to the user and of higher benefit to the service provider since it enables exposing more contents from the service in the different models.

The different engines were modified to recommend by series as described above; however, the GetVideoRecommendations API was only slightly modified to return a season object with additional series information, in order to display the relevant series in the client application.

Note

The engines that handle series content, especially the viewer series engine for live and VOD content, will not provide optimal recommendations if the series metadata (i.e., series name, season sequence, episode sequence) is not provided or not consistent. The most popular engine can provide recommendations for episodes independently of the season/series they belong to. This is enabled when calling the engine with the value for the "content_type" parameter set to "programs".

5.2.16 Displaying Series Recommendations

The following is a description of the main scenarios in which series content is recommended and how to implement them in the application.

COMPASS provides a set of APIs that enable referring from one series entity to another, to achieve a consistent user experience regarding series content, as described below.

The GetSeriesSeasonList API should be used when the application displays a recommendation on a VOD season and you want to display other seasons to the viewer, from the same series.

The GetVideoSeasonContentList API should be used when the application displays a recommendation for a VOD season and you want to enable the viewer to drill down and view the items related to this season.

The GetVideoSeasonContentList API should be called using the season ID from the original episode, for a case when the application displays a recommendation on a single VOD episode and you want to display to the viewer other episodes to view, from the same season as the episode displayed.

The GetLiveSeriesAvailabilityList API should be used in a case when the application displays a recommendation on a live episode and you want to display other episodes to the viewer, from the same series.

To learn more about the APIs mentioned in this section, please refer to the COMPASS RT API Developers Guide.

5.3 Explore

The Explore capability is used in order to provide a more active means of discovering content to watch. In this case the viewer launches the Explore functionality from specific content. Then he should specify the exploration method, either by actors, directors or genres. The response consists of the most popular content of the relevant criterion.

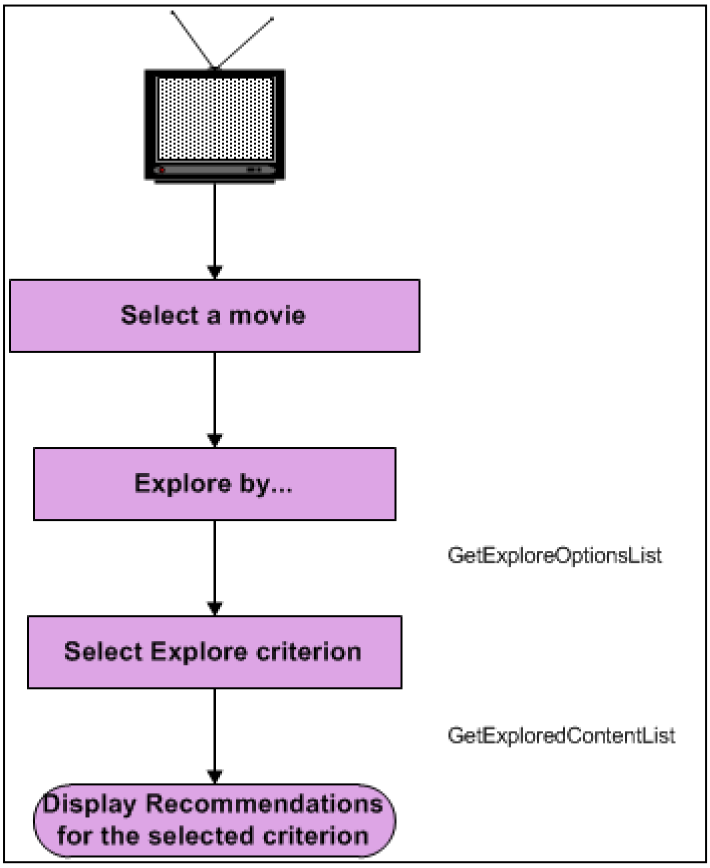


Figure 8: Explore flow diagram

After the viewer selected a movie for reference when looking for something to see, he should select the exploration method for example: actors. This means that the viewer wants to find more content based on similar actors.

The GetExploreOptionsList should then be called in order to display the list of actors that are part of the cast of the movie, in order for the user to select the specific actor.

After the viewer selects a specific name, the GetExploreContentList should be called with the selected actor, in order to retrieve the list of content for the selected actor.

5.3.1 Retrieve Content by Editorial Inputs

The popularity of content can be determined by more than one criterion. The obvious criterion is to count the consumption of the content in the service. This can be achieved in COMPASS by requesting recommendations from the most popular engine. Another type of popularity can be defined by the service editorial teams.

COMPASS provides an interface to ingest the popular cast and other contributors (such as director or writer). This can be then used to retrieve a list of content items ordered by the number of popular contributors.

To retrieve the list of content as described above use the GetRecommendationListByPopularContributor API. For more information on this API refer to the COMPASS RT Reference Guide.

5.4 TV Planner

TV Planner refers to providing recommendations for live programs for a future time period. For example: recommendations for programs this evening.

The TV Planner recommendations are generated by the live recommendations engines in COMPASS, but for a future time frame.

Note

The maximum time frame for future live recommendations should be synchronized with the time frame defined for provisioning live content into COMPASS.

Note

The live recommendations provided by COMPASS are based on personal preferences. However, since the recommendations are ordered by the profile's preferences (and not necessarily by time) it is advised to request recommendations for a small time window, when possible, in order to provide optimized recommendations.

5.5 Displaying Rating Data

COMPASS offers rating capabilities for VOD content. COMPASS saves the rating of a viewer (profile) for each content item it has rated and also the average rating for all viewers.

COMPASS provides this data for the following scenarios:

- When retrieving recommendations
- When displaying this information for a list of content items that is not a list of recommendations (GetContentList).

6 Social TV

Note

COMPASS Social features are not supported for EU based customers.

This implementation of Social TV refers to a direct connection between the COMPASS profile and the user's Facebook account to enable the user to interact with all of his Facebook friends by sharing a recommendation for content within the service catalog to Facebook, where all his friends can see it.

In addition personal recommendations and VOD gifts can be set within the service to a specific friend within the service.

The operator is required to configure the service to apply this Social TV implementation. For more information on Social TV configurations and implementation, refer to the COMPASS Administration Guide.

The following section describes how to implement Social TV by connecting directly a profile in the service with a defined Facebook account:

- Connecting a Profile to a Facebook Account
- Building a Social TV Network Explicitly

6.1 Connecting a Profile to a Facebook Account

This implementation of Social TV refers to a direct connection between the COMPASS profile and the user's Facebook account to enable the user to interact with all of his Facebook friends by sharing a recommendation for content within the service catalog to Facebook, where all his friends can see it.

In addition personal recommendations and VOD gifts can be set within the service to a specific friend within the service.

The operator is required to configure the service to apply this Social TV implementation. For more information on Social TV configurations and implementation, refer to the COMPASS Administration Guide.

The following section describes how to implement Social TV by connecting directly a profile in the service with a defined Facebook account.

This chapter includes the following sections:

6.1.1 Linking the Service to Facebook

A custom application is required for connecting to Facebook. This allows integration between the social network site and the COMPASS' Social network management component. Synchronization of Facebook for the purpose of retrieving someone’s social network or publishing a comment on a wall, are managed via this special Facebook application.

For more information on how to build a Facebook application, refer to Facebook developers' documentation.

To ensure that this process is secure, a signature parameter is sent by the application when performing login. The sender application must generate the signature (using a simple DES encryption algorithm) that should be shared with the external application, as described in the following section.

6.1.1.1 Generating a Secure Request from Facebook

The signature to be used for securing the actions performed from the Facebook applications on Social TV should be generated as follows:

1. The parameters to be sent in the request must be sorted alphabetically by key.
2. The key-value pairs must be concatenated.
3. Use DES encryption on the concatenated string, using a secret key (the secret key is set in the COMPASS in Voyage TVE Console GUI and must be known to the Facebook application. Refer to the COMPASS Administration Guide for more information).
4. Append the signature parameter to the original request.
   For example: the following request is to be sent to COMPASS:
   `http://<server>:<port>/compass/<Api>?network_type=facebook&external_id=1001`
5. The secret key is: *12345678* (binary: [*B@356f144c*)
6. The parameters are sorted and concatenated: *external_id=1001network_type=facebook*
7. Signature (String):
   RXlaakpaVHJvd2d2S2N1SkFmM1VYeHdyTFIwZGZ3UkdyamtONCtzeUEwbjJubkpXcjRzT29RPT0=
8. The Request: `http://<server>:<port>/compass/<Api>?network_type=facebook&external_id=1001&RXlaakpaVHJvd2d2S2N1SkFmM1VYeHdyTFIwZGZ3UkdyamtONCtzeUEwbjJubkpXcjRzT29RPT0=`

If the API cannot verify the signature, an exception is raised.

6.1.2 Joining Social TV

The process of linking the COMPASS profile to a personal Facebook account should is identical to the custom Login with a Facebook function that is common in many sites today.

The client should use the standard Facebook Connect functionality in which the user should be prompt to enter his Facebook credentials (e-mail and password) and the application should perform the following actions:

1. Connect to Facebook.
2. Create the Social identity and use the first name and last name from the Facebook account as the values for the mandatory parameters.

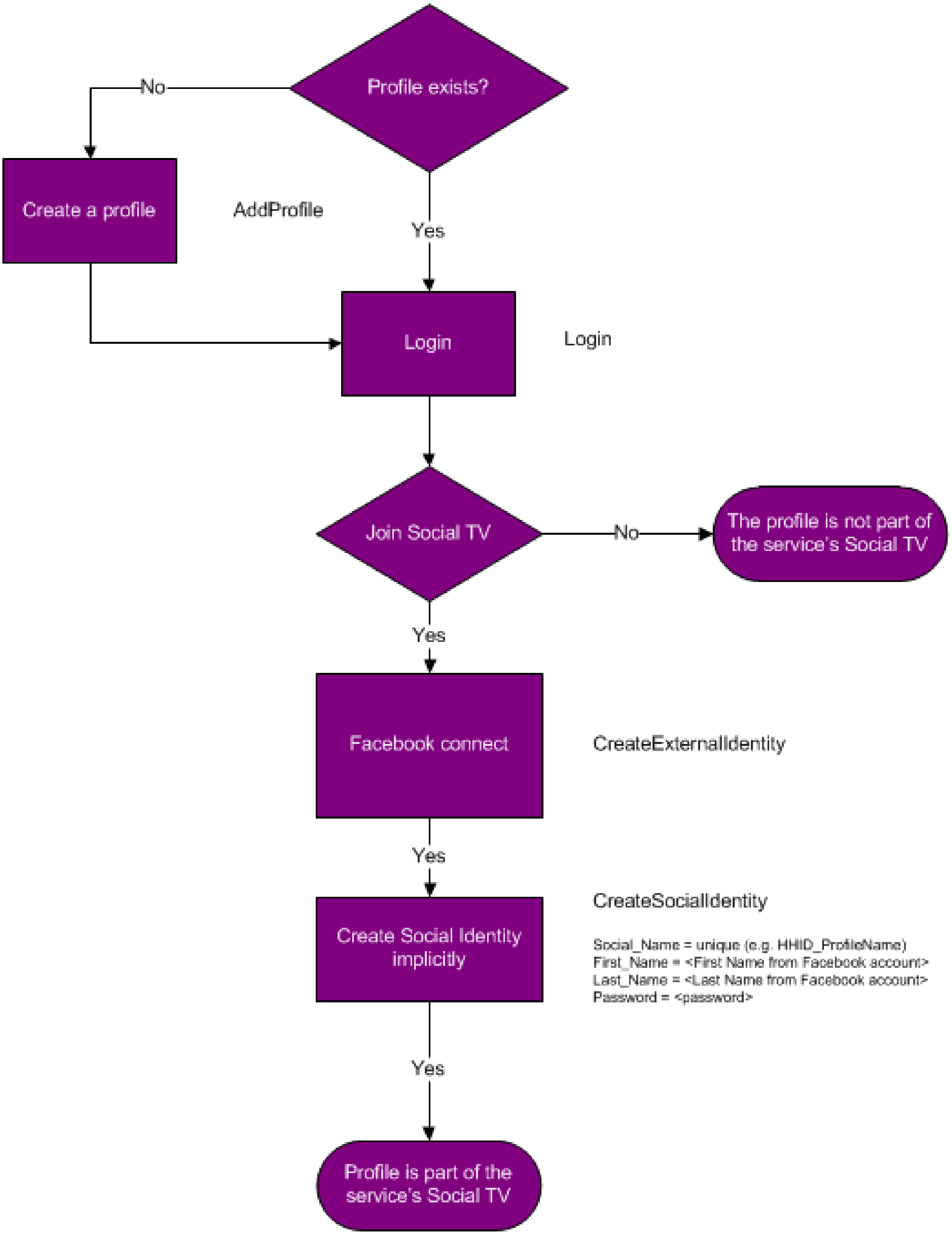
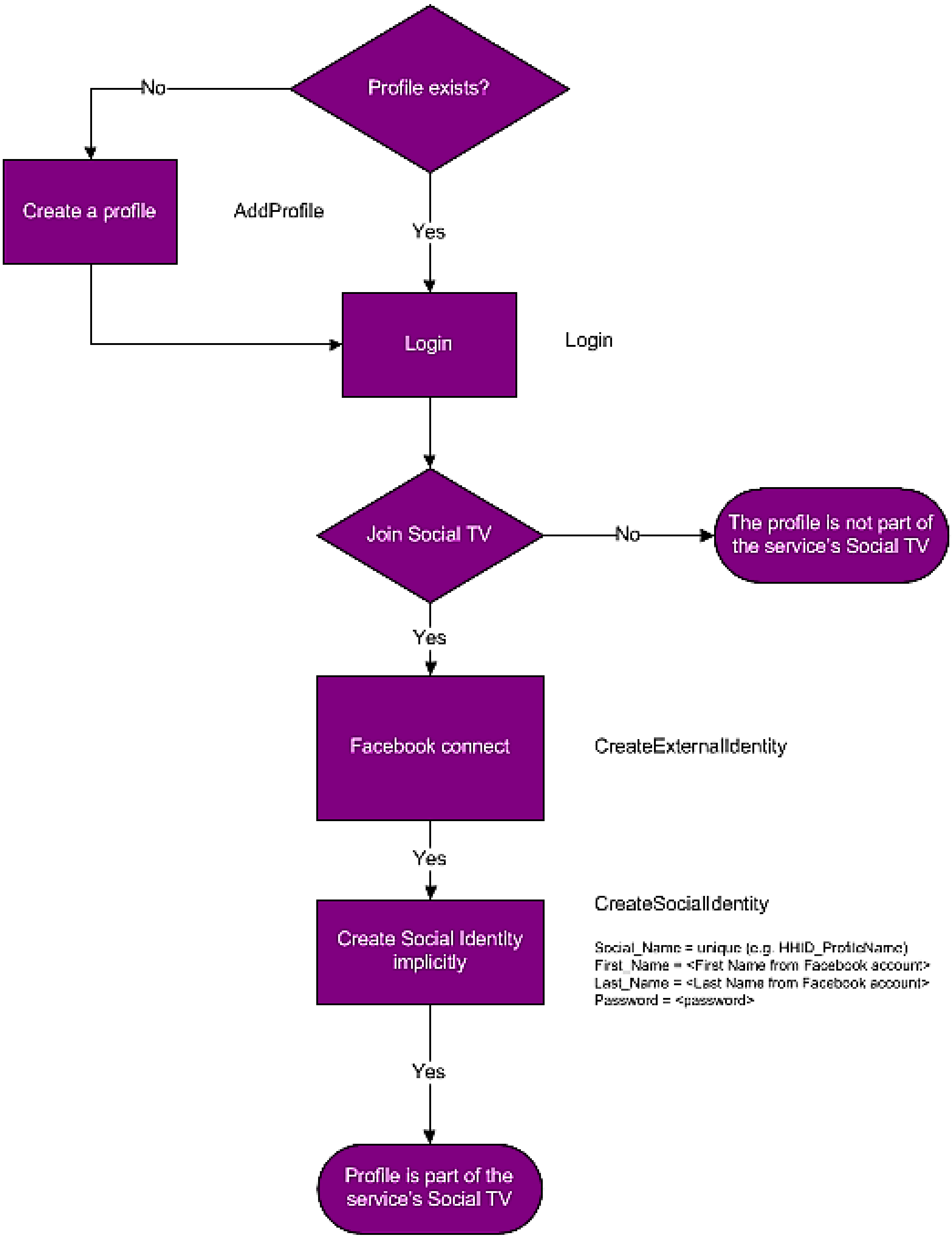

Profile exists?
No
Create a profile
AddProfile
Yes
Login
Login
Join Social TV
No
The profile is not part of the service's Social TV
Yes
Facebook connect
CreateExternalIdentity
Yes
CreateSocialIdentity
Create Social Identity implicitly
Social_Name = unique (e.g. HHID_ProfileName)
First_Name = <First Name from Facebook account>
Last_Name = <Last Name from Facebook account>
Password = <password>
Yes
Profile is part of the service's Social TV

Figure 9: Login to Social TV flow diagram


Profile exists?
No
Create a profile
AddProfile
Yes
Login
Login
Join Social TV
No
The profile is not part of the service's Social TV
Yes
Facebook connect
CreateExternalIdentity
Yes
CreateSocialIdentity
Create Social Identity implicitly
Social_Name = unique (e.g. HHID_ProfileName)
First_Name = <First Name from Facebook account>
Last_Name = <Last Name from Facebook account>
Password = <password>
Yes
Profile is part of the service's Social TV

6.1.2.1 Social Image

When a profile joins Social TV the profile image is used by default and then the application enables using dedicated images relevant to Social TV, for example, images taken from the Facebook account linked to the TV profile.

The application may require the image to be displayed in a variety of sizes for display in different screens. The images are identified by a specified name known by the user interface (i.e., profile_ images_big, profile_images_small).

To create the social image, use the CreateSocialImage API. For more information on the APIs for managing the social images, refer to the COMPASS RT API Developers Guide.

6.1.3 Leaving Social TV

In this implementation the user can define to break the link between the COMPASS profile and the Facebook profile.

In this case the following actions should be performed:

1. Disconnect from Facebook by deleting the external identity - run `DeleteExternalIdentity` API.
2. De-activate the Social ID so the user will not be able to perform any Social TV action, but his name will still exist in sent messages or gifts to friends - Run the `DeleteSocialIdentity` API.

Note

When the social identity is deleted, the images are synchronized back to the profile in order to have a consistent personal image.

6.1.4 Creating a List of Friends

Once the service is configured to automatically synchronize the COMPASS profile and the Facebook account, COMPASS periodically updates the list of friends added or deleted by the user in his Facebook account.

The synch for a specific profile is performed upon using any Social TV functionality.

The synch is not performed online. There is a minimal elapse time between synchronizations that is configurable in the COMPASS Administration Guide. The default is defined to be 24 hours. This is the minimal elapse time since it will be done after the last synch and upon a social activity performed by the user.

6.1.5 Publishing Messages on the Facebook Wall

In addition to sending personal recommendations to friends, a social TV user can publish his recommendation on his wall directly from the service portal (on TV or any other device) or from the Facebook application.

The flow for this action is based on the description of the 6.1.7 Sending Recommendations to Friends flow described previously, with the following specifications:

1. Define **Facebook** as the Social Network in the network_type_list.
2. Provide the additional information needed in order to properly build the message on the wall, as listed below:
    - Define **Facebook** as the Social Network in the network_type_list.
    - Define **Facebook** as the Social Network in the network_type_list.
    - Define **Facebook** as the Social Network in the network_type_list.
    - Define **Facebook** as the Social Network in the network_type_list.

Note

There is no verification that the action ended successfully. Moreover, if the user deleted the Facebook application of the service (which handles the recommendations), the message will not be posted on the user's wall.

For more information, refer to the description of the SendSocialMessage API in the COMPASS RT API Developer Guide.

6.1.6 Publishing the Personal Wishlist

COMPASS supports a new use case in which a social user can enable publishing his wishlist on his Facebook wall, to inform his friends about content that he marked that he wishes to view.

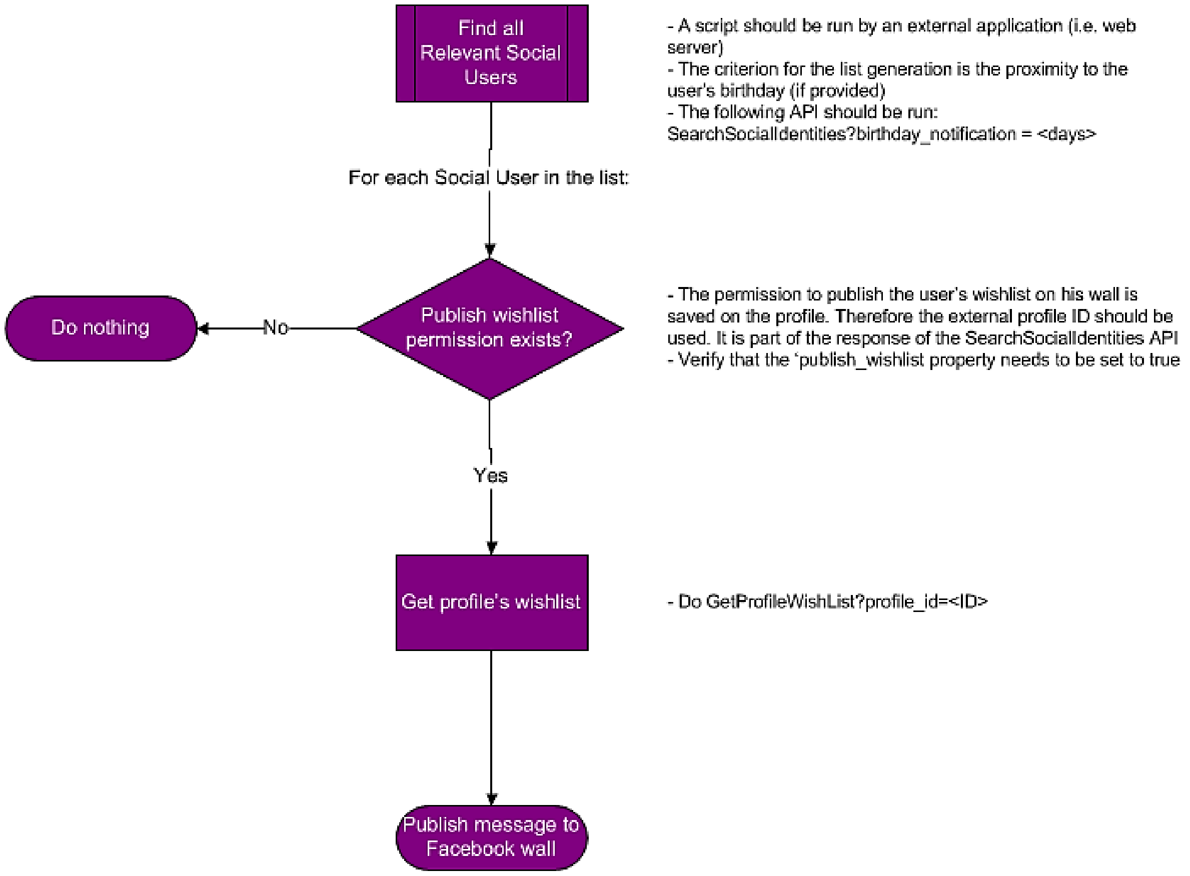
This then can be combined with the VOD gift use cases, by enabling friends to buy one of the items in the wishlist for a friend.

For this the social user must give the system permission to publish his wishlist on his wall.

Note

When publishing the user's wishlist on the wall, it is recommended to publish a link to the user's wishlist in the operator's portal instead of publishing a list of the content in a Facebook post. This results in a nicer user experience, which also will enable performing easily the relevant actions on the items within the wishlist.

Figure 10: Publishing Personal Wishlist flow diagram


Find all Relevant Social Users
For each Social User in the list:
Publish wishlist permission exists?
No
Do nothing
Yes
Get profile's wishlist
Publish message to Facebook wall

- A script should be run by an external application (i.e. web server)
- The criterion for the list generation is the proximity to the user's birthday (if provided)
- The following API should be run:
SearchSocialIdentities?birthday_notification = <days>

- The permission to publish the user's wishlist on his wall is saved on the profile. Therefore the external profile ID should be used. It is part of the response of the SearchSocialIdentities API
- Verify that the 'publish_wishlist property needs to be set to true

- Do GetProfileWishList?profile_id=<ID>

6.1.7 Sending Recommendations to Friends

The recommendations sent by Social TV users are implemented as any other message, with the exception that in this case the content ID or availability ID for the recommended content needs to be sent within the message.

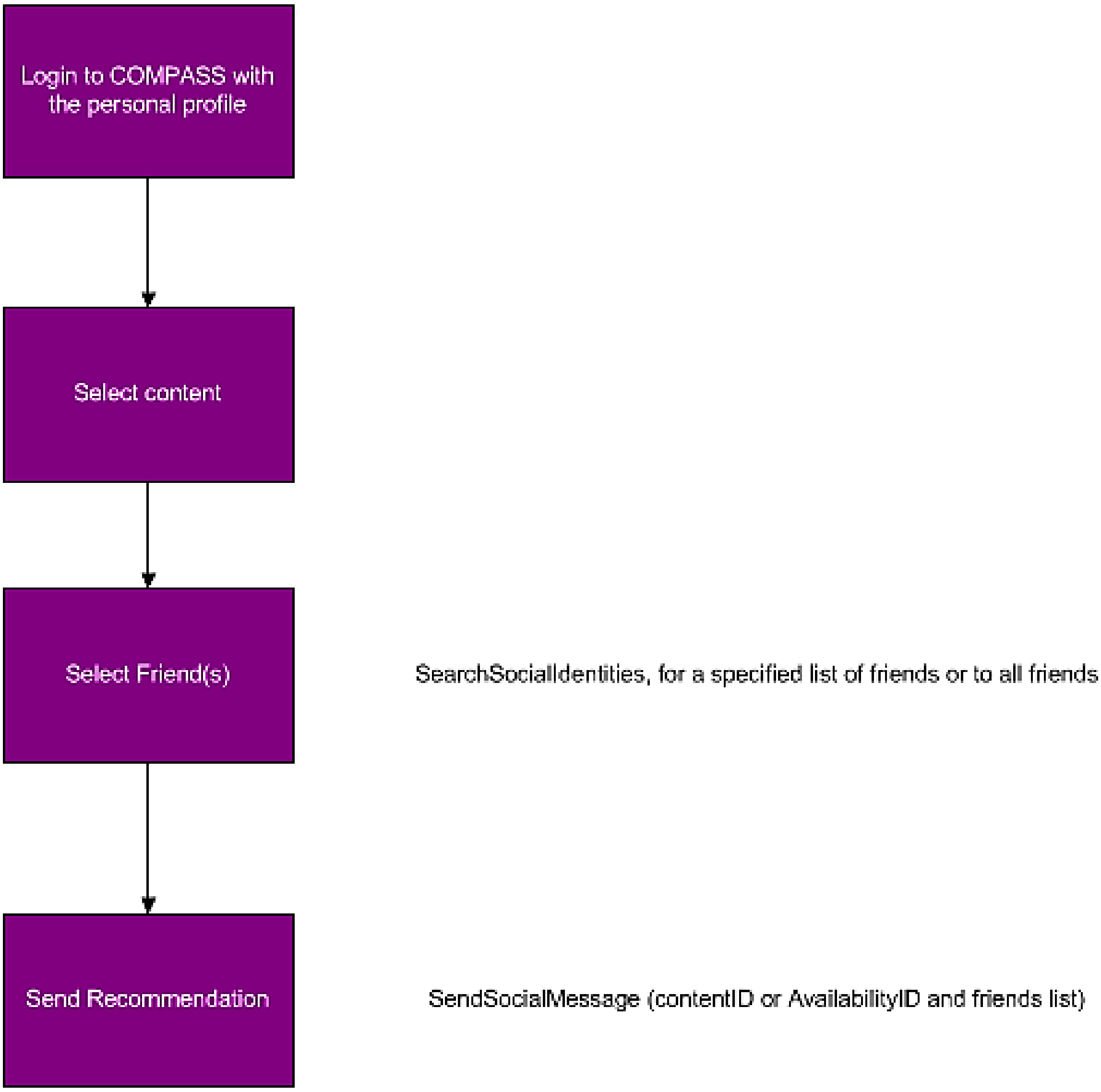

Login to COMPASS with the personal profile
Select content
Select Friend(s)
SearchSocialIdentities, for a specified list of friends or to all friends
Send Recommendation
SendSocialMessage (contentID or AvailabilityID and friends list)

Figure 11: Sending Recommendations to Friends flow diagram

6.1.8 Sending a VOD Gift

Social TV VOD gift messages, enable adding a personal layer to the ability provided in a service to send a VOD movie, as a gift to a different subscriber of the service.

This flow can be implemented assuming that the VOD gift functionality is enabled in the service and executes all the required steps, such as billing the sender and enabling the recipient to consume the content.

The following flow describes the flow of implementing VOD gift in a service that has Social TV users that will enjoy personalized gifts which are displayed only to them, due to the link between Social TV and COMPASS, which adds the personalization level. Users that are not part of Social TV will probably enjoy the VOD gift functionality at the household level.

When sending messages of type 'vodgift', vod gift information should be supplied in a property parameter. The name of the property should be 'vod_gift_id' and the value should be the external VOD gift ID. Additional properties can be set for VOD gift, such as the external content ID, content name, etc.

Note

A VOD gift message can be sent only to a single active friend.

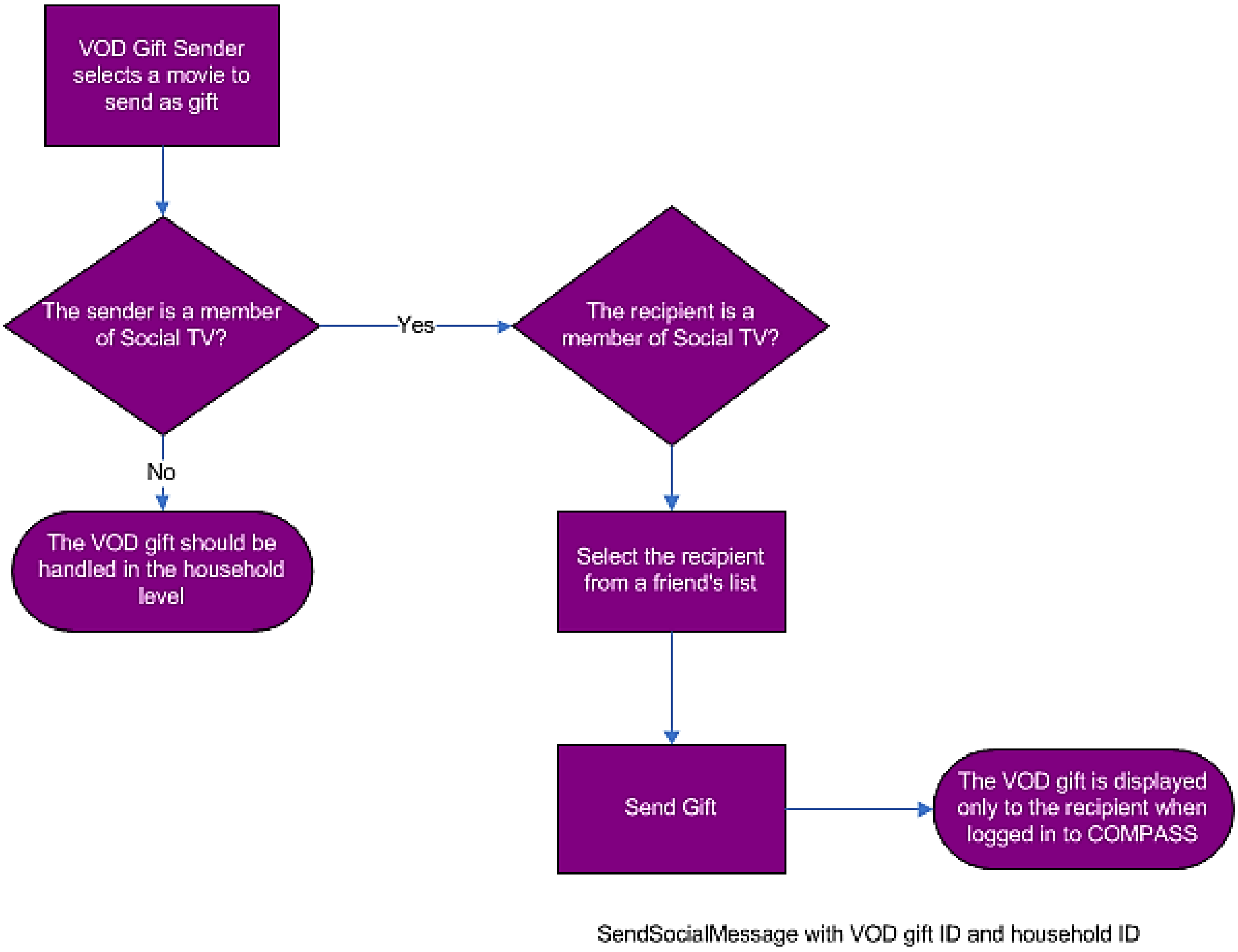

VOD Gift Sender selects a movie to send as gift
The sender is a member of Social TV?
Yes
The recipient is a member of Social TV?
No
The VOD gift should be handled in the household level
Select the recipient from a friend's list
Send Gift
The VOD gift is displayed only to the recipient when logged in to COMPASS
SendSocialMessage with VOD gift ID and household ID

Figure 12: Sending a VOD gift

6.1.9 Displaying Social Messages

Social TV APIs enable a variety of display options for social messages so that the viewer can see his friends' opinions on the specific content item, to help him decide whether the content is relevant for him and simplify the purchase decision.

Another option would be to display social recommendations in a list similar to an inbox, referred to in Social TV as the storyboard. The following is more information on ways to use Social TV APIs to display social messages.

Note

To optimize the performance of the social messages retrieval, social messages older than one year are archived. To retrieve these messages the dedicated API needs to be used.
For more information, refer to the COMPASS RT API Developer Guide.

6.1.9.1 Display Recommendations per Content

When a viewer is in the process of deciding whether to view specific content offered to him by the service, one of most important factors that will affect his decision, are his friend's opinion about the content.

In order to implement this viewer experience as part of the content information displayed to the viewer, the messages sent to this viewer need to be filtered by a specific content ID. (GetReceivedSocialMessages and use the property parameter to define the specific content with which to refer.)

For more information, refer to the GetReceivedSocialMessages API description of the API in the COMPASS RT API Developer Guide.

6.1.9.2 Displaying Content Data in Social Recommendations

A friend’s recommendation for specific content can be displayed, regardless of the recipient friend's subscriptions or the availability of the content within the service.

More specifically, a recommendation on a live program should still appear in the storyboard even if the program broadcast time passed. Another example would be a VOD recommendation for a subscription VOD package to which the recipient has not yet subscribed.

In addition, the design of the storyboard may require displaying metadata and images for the recommended content.

For these purposes the GetContentList API should be used, by providing the list of IDs of the content items that need to be displayed, as part of the storyboard.

6.1.9.3 Storyboard

The storyboard is the location in the portal in which the Social TV member messages appear. Social messages can be displayed by a variety of filters, based on the user experience that is required. The following are the displays that Social TV enables, using a variety of run time APIs.

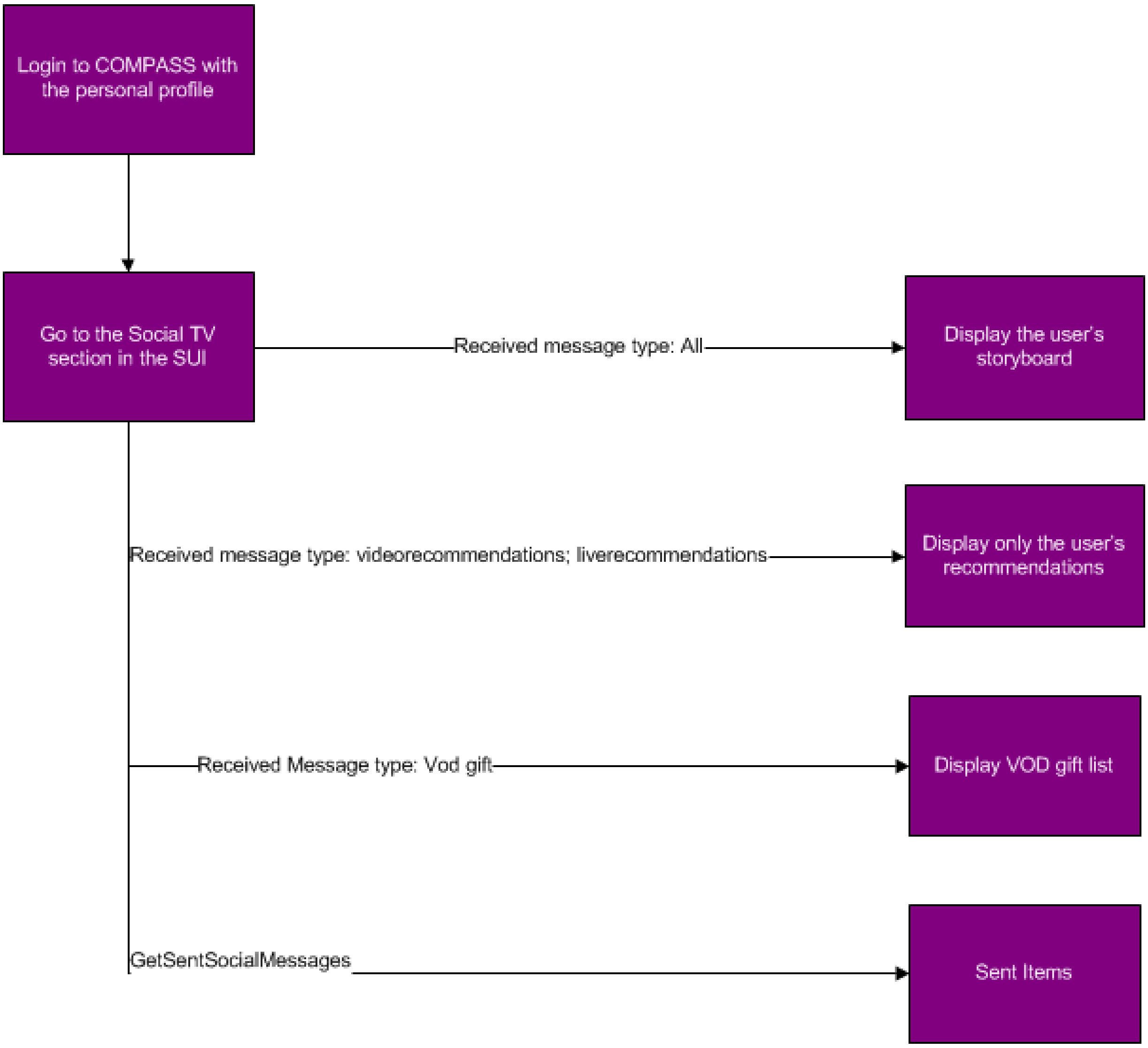

Login to COMPASS with the personal profile
Go to the Social TV section in the SUI
Received message type: All
Display the user's storyboard
Received message type: videorecommendations; liverecommendations
Display only the user's recommendations
Received Message type: Vod gift
Display VOD gift list
GetSentSocialMessages
Sent Items

Figure 13: Storyboard flow diagram

> **Note**
>
> Messages that contain special characters should be saved after being escaped, in order to be properly displayed in the user interface. This of course requires un-escaping the text for display.

6.1.9.4 Managing Social TV Messages

The following is a detailed description of the different statuses of the Social TV messages.

This version of Social TV does not handle any automatic action on messages. Message management must be handled on the client side when implementing the specific actions.

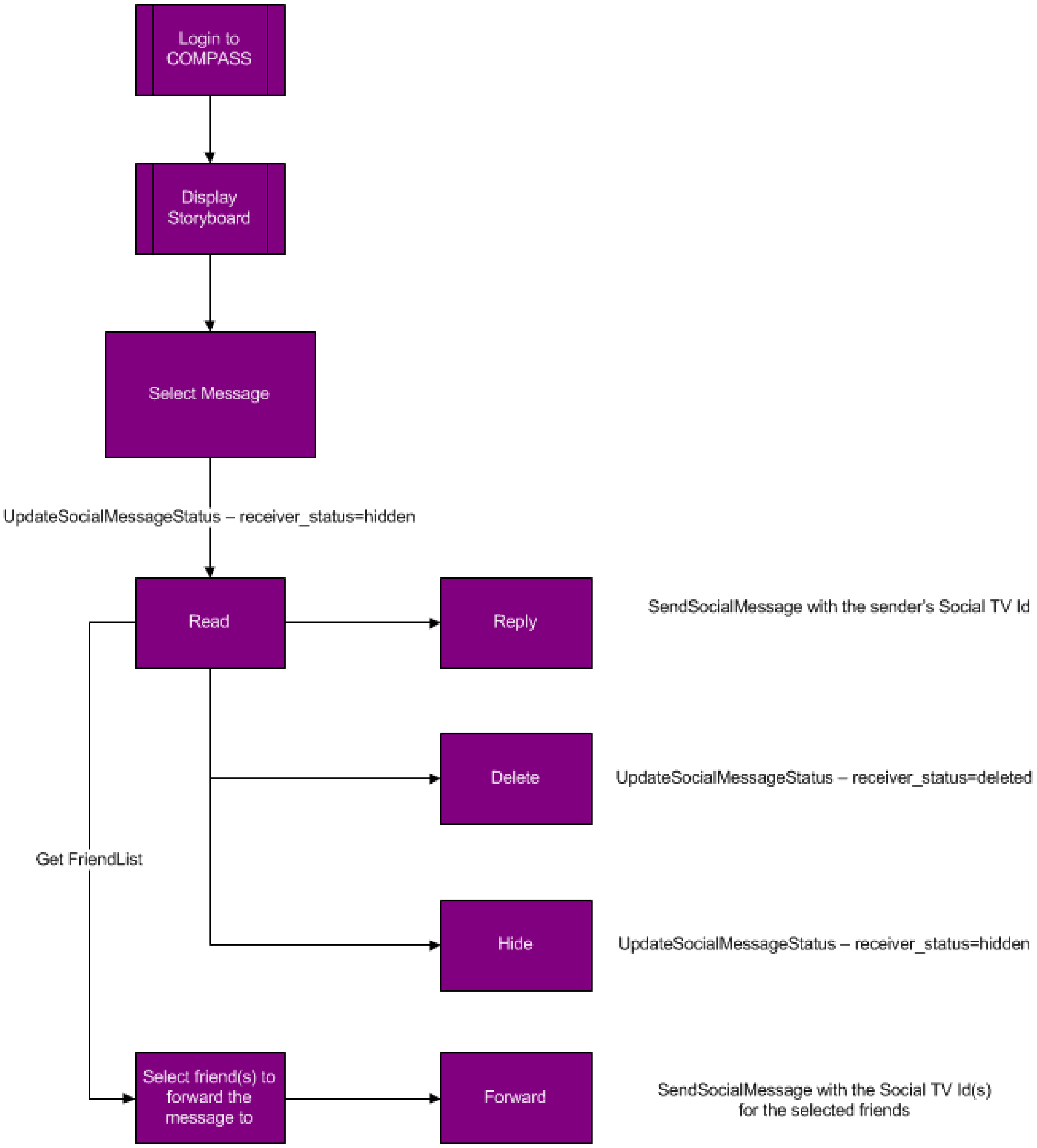

Login to COMPASS
Display Storyboard
Select Message
UpdateSocialMessageStatus – receiver_status=hidden
Read
Reply
SendSocialMessage with the sender's Social TV Id
Delete
UpdateSocialMessageStatus – receiver_status=deleted
Get FriendList
Hide
UpdateSocialMessageStatus – receiver_status=hidden
Select friend(s) to forward the message to
Forward
SendSocialMessage with the Social TV Id(s) for the selected friends

Figure 14: Manage Social Message flow diagram

6.2 Building a Social TV Network Explicitly

This implementation refers to having the user build a friends list (out of the service subscribers) to be focused on discussions about content.

The social network is then created by the actions of users who search for, locate, and invite other users to become their Social TV friends. Users can easily control and manage their list of friends, via the SUI.

The following section describes how to implement Social TV, by connecting directly a profile in the service with a defined Facebook account.

This chapter includes the following sections:

6.2.1 Joining Social TV

Upon registration to the service, viewers are requested to provide several identification keys which can be used to search for them.

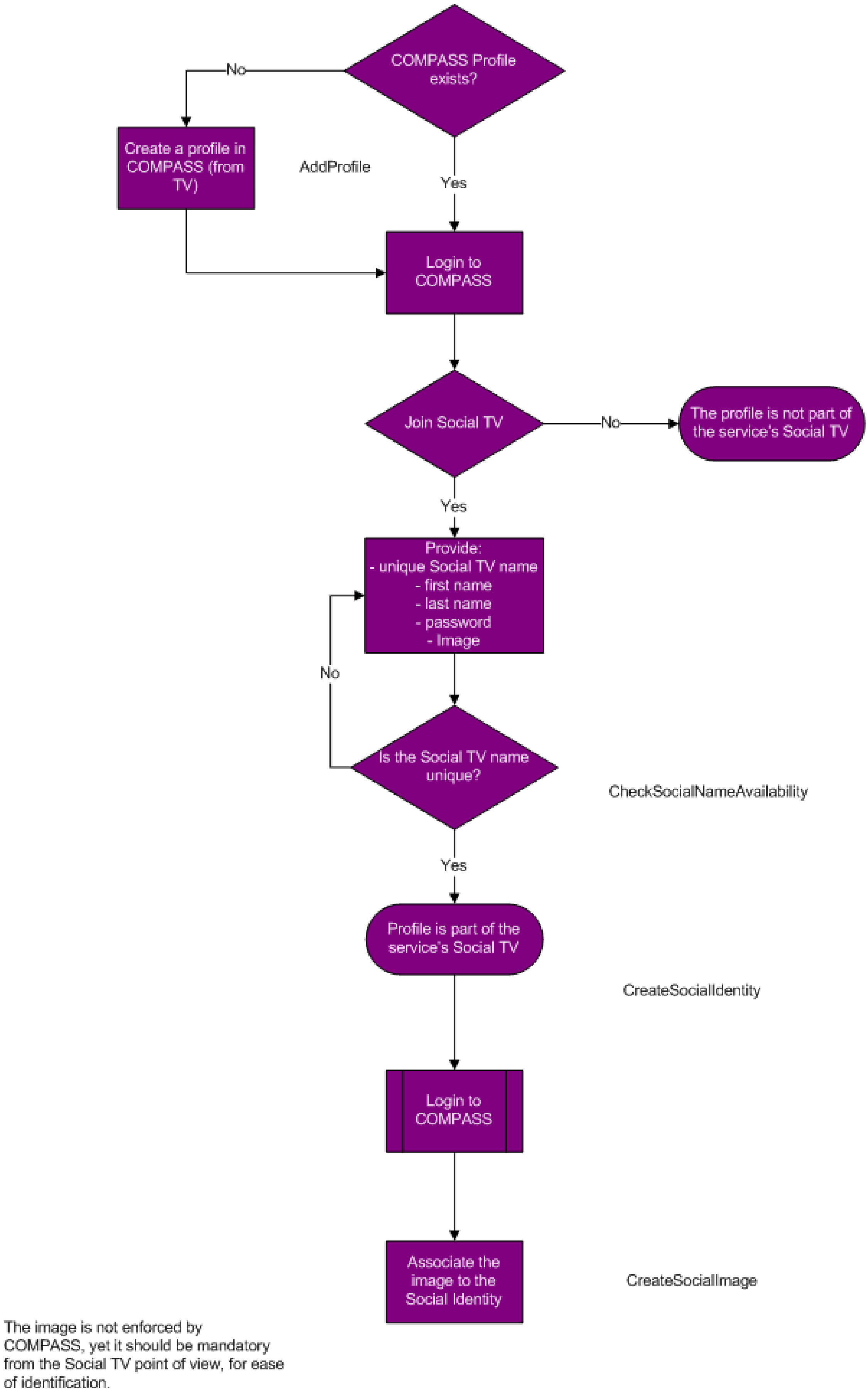

COMPASS Profile exists?
No
Create a profile in COMPASS (from TV)
AddProfile
Yes
Login to COMPASS
Join Social TV
No
The profile is not part of the service's Social TV
Yes
Provide:
- unique Social TV name
- first name
- last name
- password
- Image
No
Is the Social TV name unique?
CheckSocialNameAvailability
Yes
Profile is part of the service's Social TV
CreateSocialIdentity
Login to COMPASS
Associate the image to the Social Identity
CreateSocialImage
The image is not enforced by COMPASS, yet it should be mandatory from the Social TV point of view, for ease of identification.

Figure 15: Join Social TV flow diagram

6.2.2 Leaving Social TV

There are two actions that a Social TV member can perform in order to stop being part of the service's Social TV: deactivate his account or delete his account.

6.2.3 Deactivating the Social TV Identity

This action enables the Social TV member to leave Social TV with an option to return at a later time. The DeactivateSocialIdentity API is to be used in this case, and it switches the status of the Social TV Identity to Inactive. To return to Social TV, the ActivateSocialIdentity API should be used, which switches the Social TV Identity to Active.

A deactivated user does not appear in any list of friends or search results.

6.2.4 Deleting the Social TV Identity

This action enables the Social TV member to leave Social TV with no option to return at a later time, unless a new identity is created. The DeleteSocialIdentity API is used in this case and deletes the user from the lists of friends in which he appears.

Note

When the social identity is deleted, the images are synchronized back to the profile in order to have a consistent personal image.

6.2.5 Creating a List of Friends

As soon as the viewer acquires a Social TV identity, the services' social network can be created by finding and inviting other subscribers to become Social TV friends. The following flow describes how to implement the ability to search for Social TV friends, either on TV or on the Social Network portal.

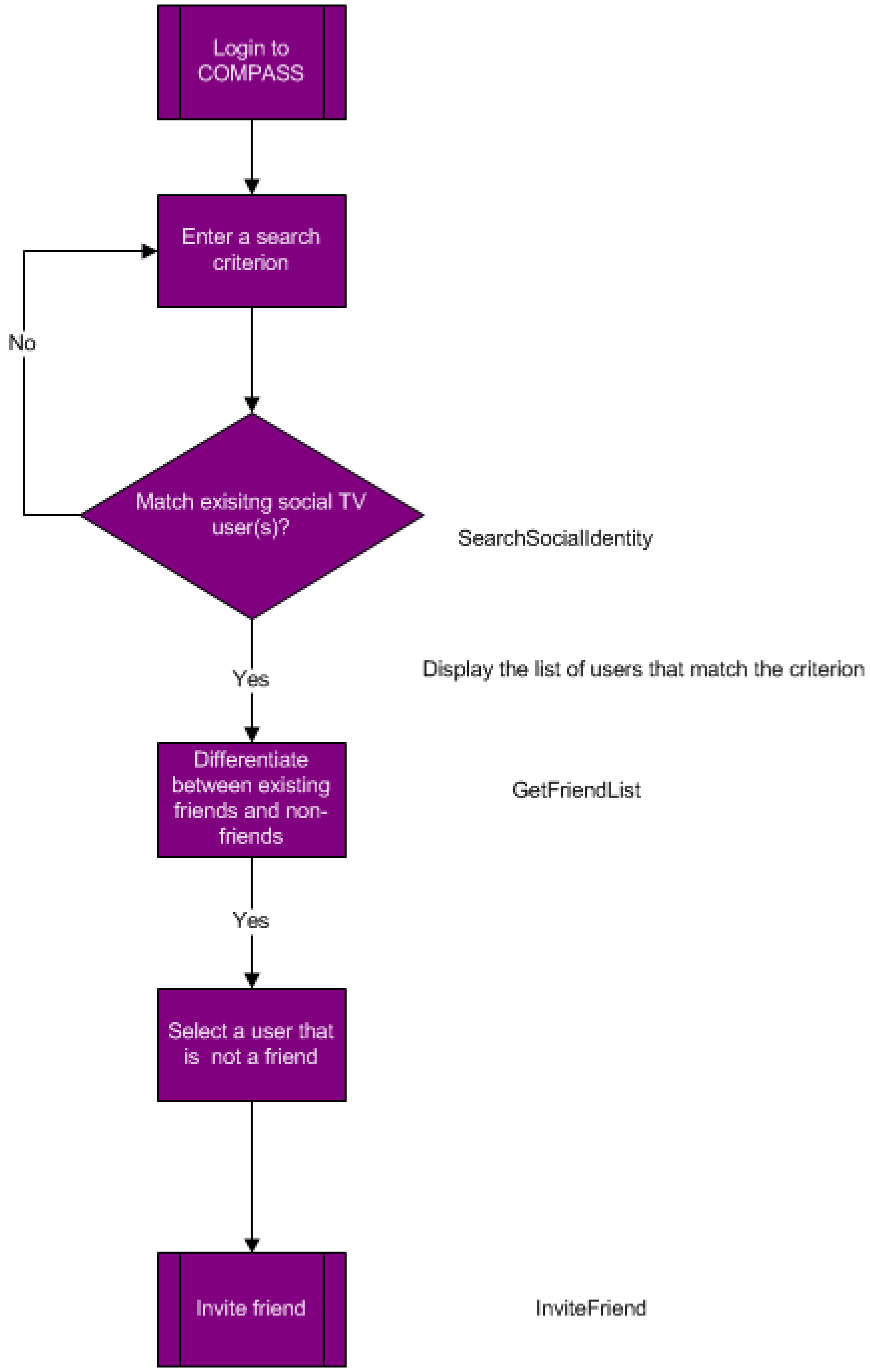

Login to COMPASS
Enter a search criterion
No
Match exisitng social TV user(s)?
SearchSocialIdentity
Yes
Display the list of users that match the criterion
Differentiate between existing friends and non-friends
GetFriendList
Yes
Select a user that is not a friend
Invite friend
InviteFriend

Figure 16: List of Friends Creation flow diagram

6.2.6 Handling Friends' Invitations

Once the social TV member starts defining his social network, he needs to be able to manage his relationships. The following flow describes how to implement the different screens that display lists that reflect the status of the network of a Social TV user, either on TV or on the Social Network portal.

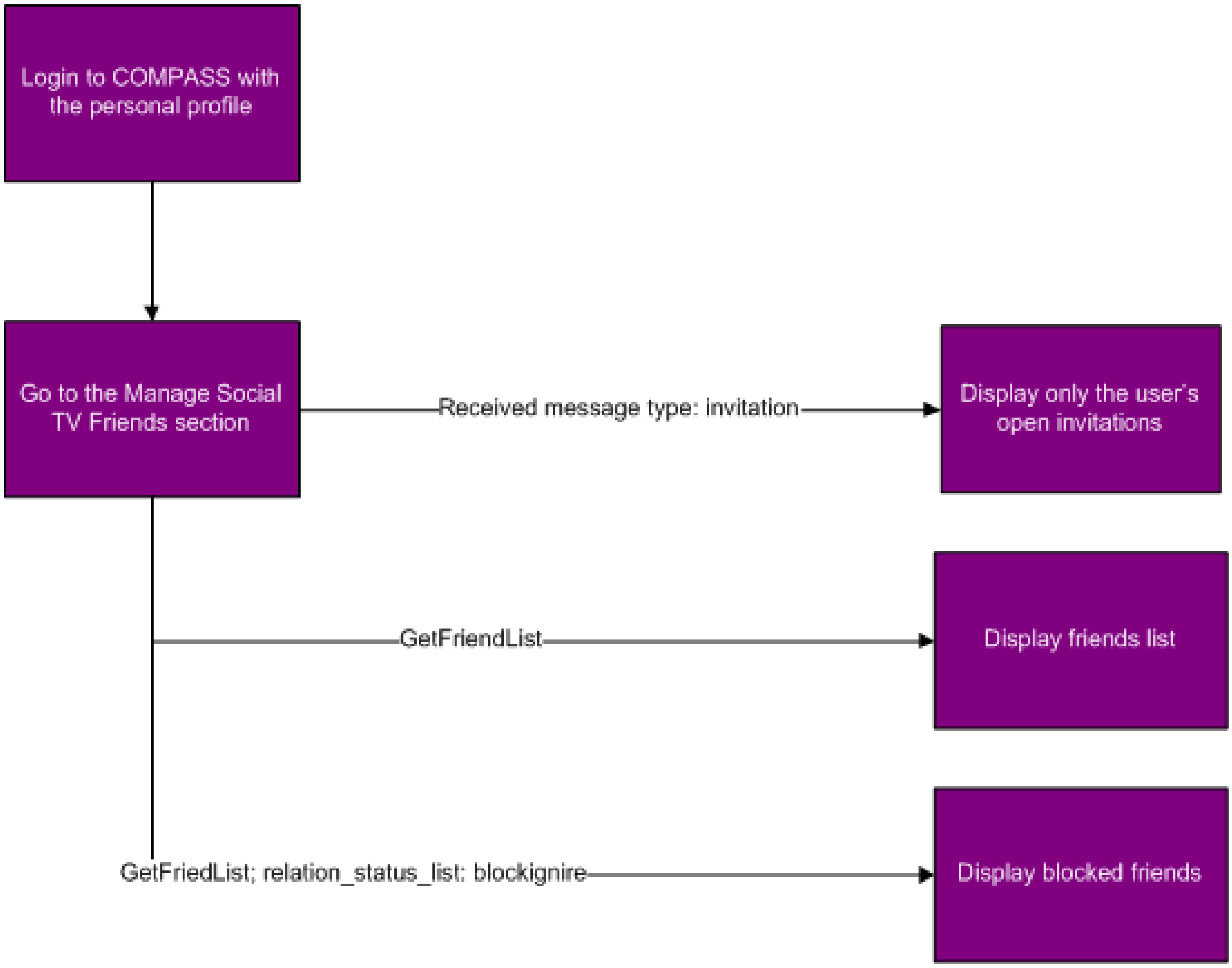

Login to COMPASS with the personal profile
Go to the Manage Social TV Friends section
Received message type: invitation
Display only the user's open invitations
GetFriendList
Display friends list
GetFriedList; relation_status_list: blockignire
Display blocked friends

Figure 17: Managing Friends Invitations flow diagram

6.2.7 Sending Recommendations to Friends

Refer to Figure 18: Sending Recommendations to Friends flow diagram in 6 Social TV.

6.2.8 Sending VOD Gift

Refer to the section 6 Social TV.

6.2.9 Displaying Social Messages

Refer to the section 6 Social TV.

7 Search

This topic explains how to search for, and filter content within COMPASS.

COMPASS Search is a fast and effective means for content discovery, when viewers know exactly what they want. It allows subscribers to locate easily the content they are seeking, via the SUI.

Search results are displayed, enabling the viewer to act upon them. This includes: ordering/watching movies, viewing info pages, recommending a content item to a friend, finding related content or further exploring content within the service.

COMPASS Search supports searching for movies, series, seasons, episodes, live programs, catch up content, actors, directors or writers. The auto-suggestion option predicts the items that are most likely to match the viewer's search criterion, based on content popularity in search results.

Each of these is described in a dedicated section below.

For more information about the implementation, refer to the COMPASS Search RT API Reference Guide.

This topic includes the following sections:

- Searching for Content
- Auto-Suggestion

7.1 Searching for Content

COMPASS Search enables finding various types of content:

- Movies
- Live programs and catch up content
- Series, Seasons & Episodes
- Actors, Directors or Writers

The results can be filtered by genres or parental level. For specific types of content there may be additional relevant filters, for example: live or catch-up content can be filtered by channel ID.

The results are sorted by relevance, an internal score indicating the extent to which there is a match between the searched criteria and the results returned. Other sorting options are also available, for example: sorting the results alphabetically or (for live content) by program start time.

7.1.1 Filtering Search Results by Technical Properties

COMPASS Search enables searching for content items, filtered by the actual device model requesting the results or by specific technical specifications of the device from which the recommendations are requested. Another option would be to refer to specific characteristics relevant to the technical capabilities of the device, like encoding, delivery mode (download or streaming) and quality class, which relate to the definition of the content (HD/SD/3D).

These provide the most relevant results for the viewer, avoiding results for content that cannot be viewed using a certain device.

7.1.2 Filtering Search Results by Service

COMPASS Search also enables searching for content items filtered by the given services.

Search results can also be filtered by the viewer's subscriptions, by explicitly providing the list of services to which the household is subscribed.

This provides the most relevant results for the viewer, avoiding results for content that is not accessible to the subscriber. However, in some cases, the operator may want to present such results in order to encourage the viewer to subscribe to new offerings.

7.1.3 Configuring Season Search

When searching on VOD content:

- To return a series in the search results:
  - SearchAll: set the parameter *series_content_type* to *series* and the parameter *group_entities* to *true*. A representative season in the series is returned. When sorting the results by content name, the representative season in the series is sorted by the series name.
  - SearchVideos and SearchSuggest: set the parameter *content_type* to *series*. A representative season in the series is returned. When sorting the results by content name, the representative season in the series is sorted by the series name.
  - To assign the same popularity to all seasons in a series set the parameter *vod.season.popularity.equalized* in the Configuration to *true*. Contact VO Professional Service for information on changing the Configuration.
- To return all seasons matching the search string, set the parameter *series_content_type* to season.

7.1.4 Configuring Episode Search

When searching on Live content:

- To return the next matching episode (in the future), set the parameter *live.search.episode.returns* in the Configuration to next. Only the next matching episode that will be broadcasted is returned. An episode that is currently being broadcasted will not be returned.
- To return all live programs matching the search string, set the parameter *live.search.episode.returns* in the Configuration to all.

Note

Contact VO Professional Service for information on changing the Configuration.

7.2 Auto-Suggestion

Auto-suggestion intends to save time for the viewer by predicting the desired results before he finishes entering the search criteria. The results are returned according to their popularity among the returned search results.

The typical implementation for this feature is similar to an auto-completion interface: the viewer enters characters one by one and after several characters begins to receive immediate feedback for his actions.

The following steps are an example of a usage flow:

1. The viewer enters 3 characters in the search input box.
2. The SearchSuggest API is called to receive the results most likely to match the characters entered by the viewer. Ideally, most viewers will find the desired result among those returned here.
3. The viewer enters a fourth character.
4. The SearchSuggest API is called again, refining the previous results.
5. Still the viewer does not find the content he is looking for, so he activates the search (by pressing a button, or clicking ‘Enter’).
6. The relevant search API is called, to present all results matching the given criteria.

8 Error Handling

This topic describes error handling in COMPASS Client Integration.

Error handling for COMPASS is implemented in the same way as any other service in the system. In addition, since the COMPASS database contains data that is retrieved from the middleware's database in an offline process run periodically, there may be some error scenarios that are caused by information that was not yet refreshed and thus needs to be handled properly in the client. For example: a movie item was unexpectedly removed from the middleware, and is not available to subscribers anymore. Until the next synchronization process between COMPASS and the middleware, COMPASS may still recommend this content item.