[Skip to end of metadata](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson" \l "page-metadata-end)

* Created by [Wang, Chong](https://chalk.charter.com/display/~P2738300), last modified by [Gunduboina, Pravallika](https://chalk.charter.com/display/~P2763890" \o ") on [Apr 16, 2018](https://chalk.charter.com/pages/diffpagesbyversion.action?pageId=390824416&selectedPageVersions=30&selectedPageVersions=31)

[Go to start of metadata](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#page-metadata-start)

* [1. Requirements](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-Requirements)
* [2. Use Cases](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-UseCases)
  + [2.1. Subscriber and Recording Migration](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-SubscriberandRecordingMigration)
  + [2.2. Recording Discard](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-RecordingDiscard)
* [3. Data Migration](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-DataMigration)
  + [3.1. Subscriber Migration](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-SubscriberMigration)
    - [3.1.1. CDVR Subscriber](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-CDVRSubscriber)
    - [3.1.2. Ericsson Subscriber](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-EricssonSubscriber)
    - [3.1.3. Rules for Subscriber Migration](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-RulesforSubscriberMigration)
    - [3.1.4. Channel Entitlement Checking](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-ChannelEntitlementChecking)
  + [3.2. Recording Migration](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-RecordingMigration)
    - [3.2.1. Recording Structure in CDVR](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-RecordingStructureinCDVR)
      * [3.2.1.1. RecordingReference](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-RecordingReference)
      * [3.2.1.2. Recording](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-Recording)
      * [3.2.1.3. Program](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-Program)
    - [3.2.2. Recording Records in Ericsson](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-RecordingRecordsinEricsson)
    - [3.2.3. Rules for Recording Migration](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-RulesforRecordingMigration)
      * [3.2.3.1. Program Mapping](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-ProgramMapping)
      * [3.2.3.2. Recording Mapping](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-RecordingMapping)
      * [3.2.3.3. RecordingReference Mapping](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-RecordingReferenceMapping)
* [4. Details of Components](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-DetailsofComponents)
  + [4.1. Ericsson-to-CDVR Migration Tool](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-Ericsson-to-CDVRMigrationTool)
    - [4.1.1. Settings](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-Settings)
    - [4.1.2. Migration Steps](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-MigrationSteps)
    - [4.1.3. Migration Report](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-MigrationReport)
  + [4.2. Changes Needed in CS](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-ChangesNeededinCS)
    - [4.2.1. New boolean "fromEricsson" in Recording](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-Newboolean%22fromEricsson%22inRecording)
    - [4.2.2. New RESP APIs](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-NewRESPAPIs)
      * [4.2.2.1. API for Entitlement Update](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-APIforEntitlementUpdate)
      * [4.2.2.2. API for Event Recording Migration](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-APIforEventRecordingMigration)
  + [4.3. Changes Needed in BG](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-ChangesNeededinBG)
    - [4.3.1. Delete Recording from Ericsson System](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-DeleteRecordingfromEricssonSystem)
  + [4.4. Changes Needed in SMS](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-ChangesNeededinSMS)
* [5. References](https://chalk.charter.com/display/KUMO/Design+-+Migrating+Subscribers+and+Recordings+in+Ericsson#Design-MigratingSubscribersandRecordingsinEricsson-References)

1. Requirements

Ericsson nDVR system will phase out by end of the year. The existing subscribers and their recordings in Ericsson system should be migrated into the new CDVR cloud scheduler system. All the future recordings done by the subscribers imported from Ericsson system after migration will be handled and managed by CDVR system. A deadline will be set, for example, the end of year 2018. After the deadline, recordings done on Ericsson system will not be available in the CDVR system.

Ericsson system used recording time in "MINUTES" to count against quota. CDVR uses the number of recordings as the unit of quota. The quota number in Ericsson system is ignored. The used quota for migrated recordings will be counted directly on the number of migrated recordings.

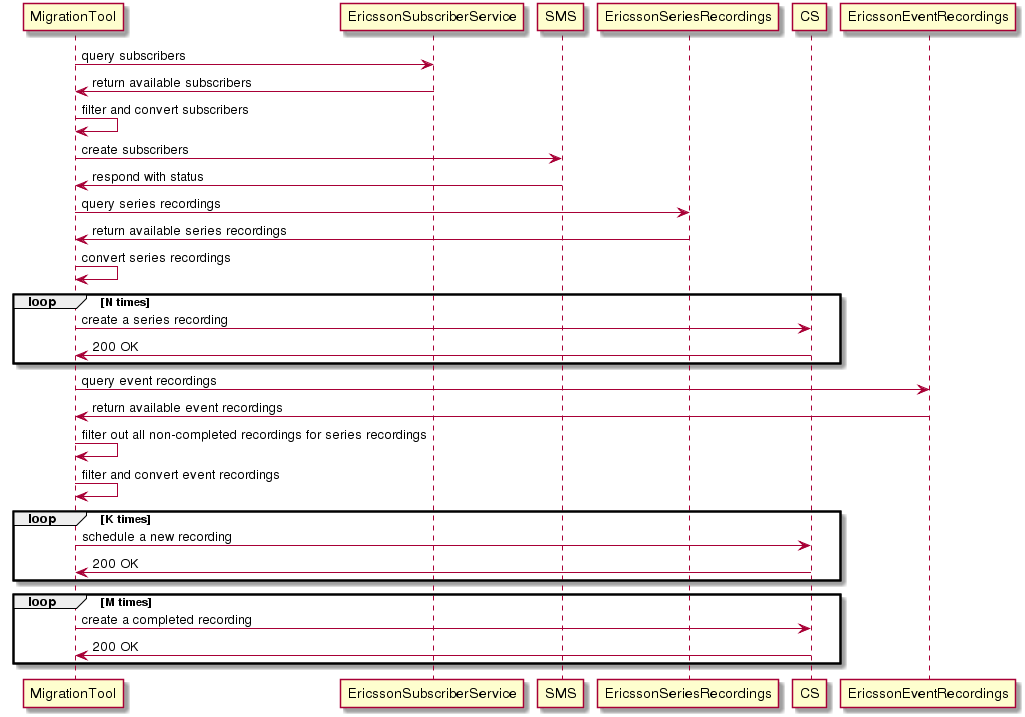
The following lists the requirements in items easy to read.

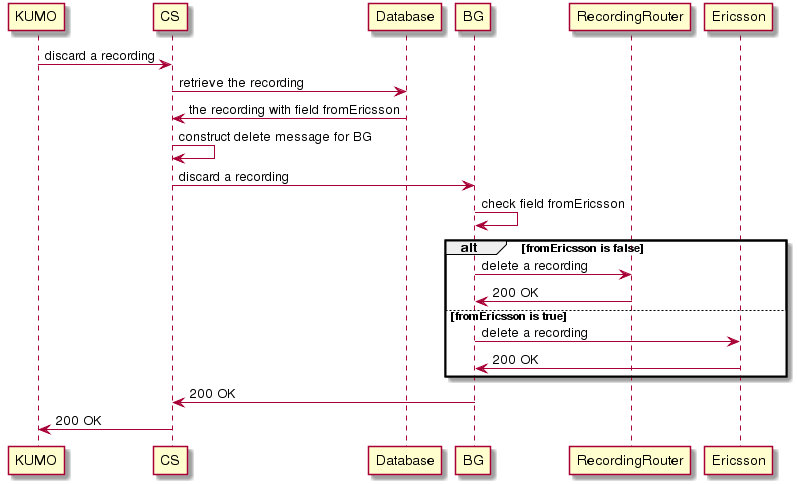
* Subscribers in Ericsson system are migrated into SMS (Subscriber Management Service) in CDVR.
* Recordings in Ericsson system are migrated into CS (Cloud Scheduler) in CDVR. Only recording records are imported into CS. The recording data are still kept by Ericsson system.
* Scheduled event and series recordings which happen in future will be re-scheduled in CDVR's way. Scheduled recordings in Ericsson system will be deleted after migration.
* CS needs to make a difference between migrated recordings and native recordings.
* BG needs to delete all migrated recordings after a set deadline.
* Playback of a completed recording on Ericsson system will be handled by KUMO, which means playback requests to FM only target recordings done in CDVR system.

A new tool should be created to migrate subscribers and recordings from Ericsson system to CDVR. The tool just runs to complete. It's not needed to deploy the tool to AWS cluster. CS, BG and SMS need to be modified to support the migrated and native recordings at the same time.

2. Use Cases

2.1. Subscriber and Recording Migration



2.2. Recording Discard  


3. Data Migration

3.1. Subscriber Migration

**3.1.1. CDVR Subscriber**

The subscriber information in CDVR contains the following critical data.

* ID (unique identifier)
* status
* service package (quota assignment)
* NCS service list (permitted NCS services)
* Geo tag list (ordered geo fencing list)

Sample Subscriber:

|  |
| --- |
| {    "id": "charteruser1",    "status": "ACTIVE",    "name": "Demo User 1",    "servicePackage": {      "id": "large"    },    "ncsServiceIds": [      109,      307    ],    "geoTags": [      "RTX-152",      "LNC",      "CSC"    ]  } |

Sample Service Package:

|  |
| --- |
| {    "id": "large",    "name": "Large Service",    "quotaType": "SHOWS",    "assignedQuota": 200,    "quotaConflictResolutionPolicy": "OLDEST\_CONTENT\_FIRST\_EXPIRE\_POLICY",    "assignedTuners": 5,    "features": [      "NPVR"    ],    "activeSessionQuota": 50  } |

In service package, most of the settings are quota, including "assignedQuota", "assignedTuners" and "activeSessionQuota". Charter doesn't use tuner quota, which can be ignored. Currently only one conflict resolution policy is supported. Feature "NPVR" is needed for event recordings.

**3.1.2. Ericsson Subscriber**

Based on Chris' document at [~~SVCDVR-589~~](https://jira.charter.com/browse/KUMOGA-540?src=confmacro) - Document Ericsson's Subscriber Data Model **CLOSED**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| $ curljson -i "[http://69.76.117.195:5927/v2/subscribers/search?Offset=0&Limit=1"](http://69.76.117.195:5927/v2/subscribers/search?Offset=0&Limit=1)  {      "TotalResults": 936,      "Subscribers": {          "Subscriber": [              {                  "HomeID": "715ad612fafa832383ada28d6ea44d2d",                  "GeoID": "geo\_1",                  "Type": 2,                  "Class": "SystemDefault",                  "State": 1,                  "StateChangeTime": "1481559663",                  "Members": {                      "Member": [                          ""                      ]                  }              }          ]      }  } | | | | | |
| **Prop** | **Type** | **Enum** | **Comment** |
| GeoID | String |  | Ignored. |
| Type | Integer | 1=CBR, 2=ABR, 3=Both | Everything created by Kumo is '2'. |
| Class | String | "basic\_[25|50|100]\_h" | "SystemDefault" translates to "basic\_100\_h". |
| State | Integer | 1=Enabled, 2=Disabled |  |
| Members |  |  | Ignored. |

Geo tags for the subscriber should be retrieved from another service "Ericsson's GeoFencing Service".

|  |
| --- |
| $curlxml -i "[http://69.76.122.22:5928/v1/home/{SubId}/"](http://69.76.122.22:5928/v1/home/%7BSubId%7D/)  <?xml version='1.0' encoding='utf8'?>  <GetHomeResponse>    <SchedulerContainer class="basic\_100\_h" GeoID="1" HomeID="715ad612fafa832383ada28d6ea44d2d" State="1" StateChangeTime="2017-11-21T10:00:40Z" Type="2">      <Metadata />    <Members />    <BoxStatus Occupied="0" QuotaType="minutes" Scheduled="0" Total="6000" />    </SchedulerContainer>    <ad\_tags>      <ad\_tag>68506</ad\_tag>      <ad\_tag>LNC-99</ad\_tag>      <ad\_tag>LNC</ad\_tag>    </ad\_tags>  </GetHomeResponse> |

<ad\_tag> is a geo tag. This list is ordered.  
<BoxStatus> is quota usage.

**3.1.3. Rules for Subscriber Migration**

The following rules should be followed when converting a subscriber in Ericsson system to CDVR.

* Subscriber in "disabled" state is ignored.
* Subscriber in "enabled" state will be converted to an active subscriber in CDVR.
* Each migrated subscriber will be assigned with default recording quota 100 shows.
* Tuner and active session quota are not used in CDVR. Set a large number 9999 to tuner and active session quota to avoid limitation by these two quota.

Here is the mapping table which maps the attributes in both system.

| **Attribute in CDVR** | **Entity in CDVR** | **Attribute in Ericsson** | **Entity in Ericsson** | **Comment** |
| --- | --- | --- | --- | --- |
| id | Subscriber | HomeID | Subscriber |  |
| name | Subscriber | HomeID | Subscriber |  |
| status | Subscriber | State | Subscriber | State 1 - Active |
| geoTags | Subscriber | ad\_tags | Ericsson's GeoFencing Service | Ordered values of ad\_tags |
| ncsServiceIds | Subscriber | n/a | n/a | No NCS service list available in Ericsson system. Since KUMO layer has the mechanism to do entitlement checks,  the NCS list here can be ignored. |
| assignedQuota | Service Package | n/a | n/a | Default - 100 |
| assignedTuners | Service Package | n/a | n/a | Fixed - 9999 |
| activeSessionQuota | Service Package | n/a | n/a | Fixed - 9999 |

By following the rules above, the sample Ericsson subscriber in previous section will be migrated into the following subscriber in CDVR.

|  |
| --- |
| {    "id": "715ad612fafa832383ada28d6ea44d2d",    "status": "ACTIVE",    "name": "715ad612fafa832383ada28d6ea44d2d",    "servicePackage": {      "id": "default\_ericsson\_svcpackage"    },    "ncsServiceIds": [],    "geoTags": [      "68506",      "LNC-99",      "LNC"    ]  }    {    "id": "default\_ericsson\_svcpackage",    "name": "Default Service Package for Migrated Ericsson Subscribers",    "quotaType": "SHOWS",    "assignedQuota": 100,    "quotaConflictResolutionPolicy": "OLDEST\_CONTENT\_FIRST\_EXPIRE\_POLICY",    "assignedTuners": 9999,    "features": [      "NPVR"    ],    "activeSessionQuota": 9999  } |

**3.1.4. Channel Entitlement Checking**

In Ericsson system, there is no channel entitlement information which can be mapped to NCS service list in SMS' subscriber. Luckily KUMO layer has mechanism to deal with entitlement checks and updates.

When a subscriber schedules a recording, KUMO will do entitlement check to make sure the subscriber is entitled to record the program in the channel. Any recording scheduling recevied by CS has already passed KUMO's entitlement check.

KUMO also has an ECP listener to listen on entitlement changes. If some channels are removed from a subscriber's entitlement, KUMO could notify CS of the entitlement changes so that CS can cancel all scheduled recordings on these channels.

So the entitlement checks against the NCS service list in SMS can be removed from CS and BG, which also means the NCS service list in subscriber is useless and can be ignored or removed.

To complete the entitlement checks, the following tasks should be done.

* Remove entitlement checks in CS and BG.
* Support a REST API in CS to cancel all recordings on channels without entitlement for a subscriber.
* KUMO's ECP listener will notify CS of the channel entitlement removal.

3.2. Recording Migration

**3.2.1. Recording Structure in CDVR**

In CDVR, a recording is organized in two levels. A "recordingreference" represents the recording belonging to a subscriber. The "recordingreference" points to a "recording" which represents the real recording on the recording router. In "Unique Copy" mode which Charter uses, "recordingreference" and "recording" are one to one paired. A "recording" points to the EPG event called program that is recorded. In the "program", the associated channel is referenced.

**3.2.1.1. RecordingReference**

The following fields are important in a recording reference when doing migration

* id: unique ID to identify the recording reference
* status: configured status of the recording. It will be combined with recording's status to decide the final state of the recording.
* recording: pointing to the recording record associate with recording router.
* keepForever: if the recording is protected or not. If protected, the recording won't be squeezed out automatically.
* seriesId: ID of the series the recording belongs to.
* recordingType: SubscriberEventRecording or SubscriberSeriesRecording.
* subscriber: id of the subscriber who recorded the recording.
* channelId: id of the channel on which the recording was done.
* bookmarkOffset: the offset of the bookmark set by playback.
* lastWatched: timestamp when the recording was watched last time.
* seriesEpisodeAcceptPolicy: REPEATS or FIRST\_RUN\_ONLY. Used for series recording.
* parent: pointing to parent series recording reference if it's one episode of a series. Used for series recording.
* children: the set of episodes in a series recording. Used for series recording.

The following lists all the available fields and their mappings to the columns of database table for RecordingReference.

|  |
| --- |
| RecordingReference {        @Id      @Column(name = "id")      private String id;        @Column(name = "status")      private String status;        @Column(name = "laststatusmodified")      private Date lastStatusModified;        @ManyToOne(fetch = FetchType.EAGER)      @JoinColumn(name = "recording")      private Recording recording;        @Column(name = "referencestart")      private Date referenceStart;        @Column(name = "referenceend")      private Date referenceEnd;        @NotNull      @Column(name = "keepforever", nullable = false)      private boolean keepForever;        @Column(name = "expireundelete", nullable = true)      private Date expireundelete;        @Column(name = "size", nullable = true)      private Float size;        @Column(name = "duration", nullable = true)      private Float duration;        @Column(name = "seriesid")      private String seriesId;        @Column(name = "recordingtype")      private String recordingType;        @Column(name = "seriesepisodeacceptpolicy")      private String seriesEpisodeAcceptPolicy;        @Column(name = "lastWatched", nullable = true)      private Date lastWatched;        @Column(name = "title", nullable = true)      private String title;        @Column(name = "subscriber")      private String subscriber;        @Column(name = "subscriberuser")      private String user;        @Column(name = "createdon")      private Date createdOn;        @Column(name = "lastmodified")      private Date lastModified;        @Column(name = "errorcode")      private String errorCode;        @Column(name = "errormessage")      private String errorMessage;        @ManyToOne      @JoinColumn(name = "parentid")      private RecordingReference parent;        @OneToMany(mappedBy = "parent")      private Set<RecordingReference> children;        @Column(name = "channelid")      private String channelId;        @Column(name = "startoffset")      private Float startOffset;        @Column(name = "endoffset")      private Float endOffset;        @Column(name = "episodelimit")      private Integer episodeLimit;        @Column(name = "parentset")      private String parentSetId;        @Column(name = "retentionhours")      private Integer retentionHours;        @Column(name = "recordingscope")      private String scope;        @Column(name = "bookmarkoffset")      private Long bookmarkOffset;  } |

**3.2.1.2. Recording**

The following fields are important in a recording when doing migration.

* uniqueId: the ID used in recording router to identify the recording. It's called "schedule ID" in Cisco's recording router.
* recordingStart: recording start time in UTC.
* recordingEnd: recording end time in UTC.
* status: status of the recording.
* copyType: always "UNIQUE\_COPY" for Charter.
* program: pointing to the EPG event being recorded.
* retentionPeriod: the deadline the recording can be kept. It can be used to meet the deadline requirement for Ericsson's recordings.
* channelId: id of the channel.
* airingId: id of the EPG event.
* seriesId: id of the series this recording belongs to.
* programId: id of the EPG program recorded.
* title: the original title of the program.
* episodeTitle: the episode title of the program.
* duration: the duration of the recording in minutes.
* contentId: the content ID assgined by recording router for the recording which is used in playback.

The following lists all the available fields and their mappings to the columns of database table for Recording.

|  |
| --- |
| Recording {        @Id      @GeneratedValue(strategy = GenerationType.AUTO)      @Column(name = "id")      private Long id;        @NotNull      @Column(name = "status", nullable = false)      @XmlElement(name = "status")      private String status;        @NotNull      @Column(name = "recordingstart", nullable = false)      private Date recordingStart;        @NotNull      @Column(name = "recordingend", nullable = false)      private Date recordingEnd;        @NotNull      @Column(name = "uniqueid", nullable = false)      private String uniqueId;        @NotNull      @Enumerated(EnumType.STRING)      @Column(name = "copytype", nullable = false)      private CopyType copyType;        @Column(name = "copytypekey", nullable = true)      private String copyTypeKey;        @NotNull      @ManyToOne(fetch = FetchType.EAGER)      @JoinColumn(name = PROGRAM\_FIELD)      private Program program;        @Column(name = PROGRAM\_FIELD, insertable = false, updatable = false)      private Long dbProgramId;        @Column(name = "failoffset", nullable = true)      private Date failOffset;        @Column(name = "retentionPeriod", nullable = true)      private Date retentionPeriod;        @Column(name = "channelid")      private String channelId;        @Column(name = "airingid")      private String airingId;        @Column(name = "title")      private String title;        @Column(name = "seriesid")      private String seriesId;        @Column(name = "episodetitle")      private String episodeTitle;        @Column(name = "programid")      private String programId;        @Column(name = "lastmodified")      private Date lastModified;        @Column(name = "errorcode")      private String errorCode;        @Column(name = "errormessage")      private String errorMessage;        @Column(name = "duration")      private Float duration;        @Column(name = "startoffset", nullable = true)      private Float startOffset;        @Column(name = "endoffset", nullable = true)      private Float endOffset;        @Version      private Long version;        @Column(name = "createdon")      private Date createdOn;        @Column(name = "laststatusmodified")      private Date lastStatusModified;        @Column(name = "contentid", nullable = true)      private String contentId;  } |

**3.2.1.3. Program**

The following fields are important in a program when doing migration.

* airingId: constructed by channel ID, program ID and event start time in UTC.
* programId: id to uniquely identify the program.
* channelEntity: pointing to channel entity the program belongs to.
* originalTitle: the original title for the program.
* episodeTitle: the title for the episode.
* startTime: the airing start time of the EPG event.
* duration: the duration of the program in minutes.
* seriesId: the id of the series the program belongs to.
* rating: the rating of the program for parental control.

The following lists all the available fields and their mappings to the columns of database table for Program.

|  |
| --- |
| Program {        @Id      @GeneratedValue(strategy = GenerationType.AUTO)      @Column(name = "id")      private Long id;        @Column(name = "airingid", nullable = true)      private String airingId;        @Column(name = "programid", nullable = true)      private String programId;        @ManyToOne(fetch = FetchType.LAZY)      @JoinColumn(name = "channel", nullable = true)      private Channel channelEntity;        @Column(name = "originaltitle", nullable = true)      private String originalTitle;        @Column(name = "starttime", nullable = true)      private Date startTime;        @Column(name = "duration", nullable = true)      private Integer duration;        @Column(name = "description", nullable = true)      private String description;        @Column(name = "seriesid", nullable = true)      private String seriesId;        @Column(name = "seasonid", nullable = true)      private String seasonId;        @Column(name = "episodetitle", nullable = true)      private String episodeTitle;        @Column(name = "episodedescription", nullable = true)      private String episodeDescription;        @Column(name = "episodenumber", nullable = true)      private String episodeNumber;        @Column(name = "seasonnumber", nullable = true)      private String seasonNumber;        @Column(name = "rating", nullable = true)      private String rating;        @Column(name = "productioncountry", nullable = true)      private String productionCountry;        @Column(name = "missinginepg")      private Boolean missinginepg;        @Column(name = "restrictions")      @Type(type = "JsonObject")      private JSONObject restrictions;        @Column(name = "metadata")      @Type(type = "JsonObject")      private JSONObject jsonMetadata;        @Column(name = "lastrecordingdeletedon")      private Date lastRecordingDeletedOn;  } |

**3.2.2. Recording Records in Ericsson**

Based on Chris' documents in [~~SVCDVR-245~~](https://jira.charter.com/browse/KUMOGA-449?src=confmacro) - Support Migration from Ericsson to GA **CLOSED** .

Serires recordings and event recordings in Ericsson system need to be queried on different endpoints. The series recording record doesn't contain the references to the episode recordings. So the relationshiop between event recordings and series recordings need to be figured out by the "SeriesID" in event recordings and series recordings.

The following is a sample series recording in Ericsson system.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| $ curljson -i "[http://69.76.110.4:5927/v2/series\_recordings/{SubId}"](http://69.76.110.4:5927/v2/series_recordings/%7BSubId%7D)  {      "TotalResults": 1,      "SeriesDataList": {          "SeriesData": [              {                  "SeriesID": "9232976",                  "Name": "The Eric Andre Show",                  "Channel": "46153",                  "Size": 4136000000,                  "IsProtected": true,                  "RecordCriteria": 2,                  "BookingTime": "1521738544",                  "StartTimeOffset": "0",                  "EndTimeOffset": "0",                  "KeepMaxEpisodes": 0,                  "MultichannelRecording": 0,                  "RecordingMetadata": {                      "Data": []                  },                  "ChannelList": {                      "Channel": []                  }              }          ]      }  } | | | | | |
| **Prop** | **Type** | **Enum** | **Comment** |
| Size | long |  | disk space required, in bytes, for recorded and scheduled events |
| IsProtected | boolean |  | Protect rec from being deleted to reclaim space (SpaceConflictPolicy=forever) |
| RecordCriteria | int | 1=FIRST\_RUN\_ONLY, 2=REPEATS |  |
| BookingTime | timestamp |  | Instant the series was scheduled, in seconds |
| StartTimeOffset | int |  | Start the recording early, in seconds |
| EndTimeOffset | int |  | Stop the recording late, in seconds |
| KeepMaxEpisodes | int |  | Ignored |
| MultichannelRecording | int |  | Ignored |
| RecordingMetadata |  |  | Ignored |
| ChannelList |  |  | Ignored |

The following is a sample event recording in Ericsson system.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| $ curljson -i "[http://peakview-fe-prod.timewarnercable.com:5927/v2/subscribers/recordings/{SubId}?Offset=0&Limit=1"](http://peakview-fe-prod.timewarnercable.com:5927/v2/subscribers/recordings/%7BSubId%7D?Offset=0&Limit=1)  {      "TotalResults": 97,      "Offset": 0,      "Limit": 1,      "Recording": [          {              "ShowingID": "AUTO\_60048EP01412293087214846904000001483535731302",              "Channel": "60048",              "ChannelCallLetter": "TOONHD",              "ShowStartTime": "1484690400",              "ShowEndTime": "1484691300",              "StartTime": "1484690894",              "EndTime": "1484691300",              "BookingTime": "1484690894",              "IsProtected": false,              "MaxAgeRating": "",              "AlreadyWatched": false,              "BeingWatched": true,              "Bookmark": "0",              "AbsoluteBookmark": "0",              "BookmarkTimestamp": "0",              "Progress": 100,              "RecordingMetadata": {                  "Data": []              },              "CopyrightExpirationTime": "0",              "ProgramInfo": {                  "ProgramID": "EP014122930872",                  "Name": "The Amazing World of Gumball",                  "SubTitle": "",                  "SeasonID": "12630964",                  "SeasonName": "",                  "SeriesID": "8648428",                  "SeriesName": "The Amazing World of Gumball",                  "Rating": "0",                  "Description": "No, this animated series is not about anthropomorphic chewing gum, it's about a young cat named Gumball Watterson. Gumball has a penchant for getting into trouble, often resulting from schemes he comes up with, but he never seems to learn his lesson. Gumball's best friend is a fish named Darwin, who used to be the family pet until he grew legs and became part of the family. Gumball resents his younger sister, Anais, because she is the smartest member of the family. Gumball's mother is the breadwinner in the family, working long hours at the rainbow factory, and his father stays at home watching TV and playing video games.",                  "Year": 0,                  "EpisodeNumber": 39,                  "EpisodeTitle": "The Stories",                  "Genre": "",                  "Type": "Program"              },              "CBRDetails": {                  "Size": 0,                  "Duration": "0",                  "Bitrate": 0,                  "State": -1,                  "Error": ""              },              "ABRDetails": {                  "Size": 577000000,                  "Duration": "404",                  "Bitrate": 0,                  "State": 4,                  "Error": ""              }          }      ]  } | | | | | |
| **Prop** | **Type** | **Enum** | **Comment** |
| ShowStartTime | timestamp |  | EPG event's start time, in seconds |
| ShowEndTime | timestamp |  | EPG event's end time, in seconds |
| StartTime | timestamp |  | Recording's start time, in seconds |
| EndTime | timestamp |  | Recording's end time, in seconds |
| isProtected | boolean |  | Protect rec from being deleted to reclaim space (SpaceConflictPolicy=forever) |
| Bookmark | duration |  | Seconds into the recording |
| AbsoluteBookmark | timestamp |  | StartTime <= AbsoluteBookmark <= EndTime (except it's '0' when not set) |
| BookmarkTimestamp | timestamp |  | Instant of the most recent update to Bookmark, in seconds |
| Progress | int |  | Percentage recorded of in-progress recording |
| Size | long |  | Bytes |
| Duration | int |  | Seconds, expected duration for in-progress recs, actual for completed |
| State | int | 1=scheduled, 2=ongoing, 3=processing, 4=completed, 5=failed, 6=deleted, 7=soft delete |  |
| Error | string |  | Only set when State=5 |

**3.2.3. Rules for Recording Migration**

The following lists the rules for migrating recordings from Ericsson to CDVR.

* Only recordings in states "scheduled", "ongoing", "processing" and "completed" are migrated. Recordings in states "failed", "deleted" and "soft delete" will be ignored.
* Recordings in states "scheduled", "ongoing" and "processing" which were done on a channel that doesn't exist in CDVR will be rejected during migration.
* Recordings in state "completed" which were done on a channel that doesn't exist in CDVR will be migrated with null channel and empty channel ID.
* Recordings in states "scheduled", "ongoing" and "processing" with valid channel will be migrated as new recordings in CDVR.
* Recordings in state "completed" will be migrated as "COMPLETED" recordings in CDVR which only allow playback and deletion.
* Recordings migrated with state "completed" will be deleted from CDVR after a set deadline.

**3.2.3.1. Program Mapping**

Here is the mapping table which maps the attributes in both system.

| **Attribute in CDVR Program** | **Attribute in Ericsson Recording** | **Comment** |
| --- | --- | --- |
| airingId | Channel  ProgramInfo->ProgramID  ShowStartTime | Constructed in the following format:  Channel\_ShowStartTime\_ProgramInfo->ProgramID |
| programId | ProgramInfo->ProgramID |  |
| channelEntity | Channel | Channel entity in CDVR which maps the channel ID will be used. |
| originalTitle | ProgramInfo->Name |  |
| episodeTitle | ProgramInfo->EpisodeTitle |  |
| startTime | StartTime |  |
| duration | ShowStartTime  ShowEndTime | (ShowEndTime - ShowStartTime) / 60 |
| seriesId | ProgramInfo->SeriesID |  |
| rating | ProgramInfo->Rating |  |

**3.2.3.2. Recording Mapping**

| **Attribute in CDVR Recording** | **Attribute in Ericsson Recording** | **Comment** |
| --- | --- | --- |
| uniqueId | ShowingID |  |
| recordingStart | StartTime |  |
| recordingEnd | EndTime |  |
| status | ABRDetails->State | state 1 → SCHEDULED  state 2 and 3 → CAPTURING  state 4 → COMPLETED |
| copyType | n/a | UNIQUE\_COPY |
| program | ProgramInfo | Point to the program created in previous section. |
| retentionPeriod | n/a | Recording in state 4 will be assigned the deadline. Other recordings will be decided by CDVR. |
| channelId | Channel |  |
| airingId | Channel  ProgramInfo->ProgramID  ShowStartTime | Same as in program. |
| seriesId | ProgramInfo->SeriesID |  |
| programId | ProgramInfo->ProgramID |  |
| title | ProgramInfo->Name |  |
| episodeTitle | ProgramInfo->EpisodeTitle |  |
| duration | StartTime  EndTime | (EndTime - StartTime) / 60 |

**3.2.3.3. RecordingReference Mapping**

| **Attribute in CDVR RecordingReference** | **Attribute in Ericsson Recording** | **Comment** |
| --- | --- | --- |
| id | ShowingID | Use the same ID used in recording. |
| status | ABRDetails->State | Recordings in states 1,2,3 will be migrated as new recordings in CDVR. |
| recording | n/a | Points to Recording object created in previous section. |
| keepForever | IsProtected | boolean true or false |
| seriesId | ProgramInfo->SeriesID |  |
| recordingType | Based on query endpoint |  |
| subscriber | n/a | subscriber ID in the query path. |
| channelId | Channel |  |
| bookmarkOffset | Bookmark |  |
| lastWatched | BookmarkTimestamp |  |
| seriesEpisodeAcceptPolicy | RecordCriteria | 1=FIRST\_RUN\_ONLY, 2=REPEATS |
| parent | SeriesID | Construct the relationship based on "SeriesID" in recording. |
| children | SeriesID | Construct the relationship based on "SeriesID" in recording. |

4. Details of Components

4.1. Ericsson-to-CDVR Migration Tool

The migration tool will be built as a docker image. Configuration can be done by passing environment variables. The logs can be retrieved from docker logs.

The tool will talk with Ericsson system to retrieve subscribers and recordings, convert them into the formats accepted by CDVR system, and migrate them into CDVR system. The scheduled event and series recordings can be cancelled on demand or cancelled after the migration. The recorded data will be kept inside Ericsson system until a set deadline. CDVR will instruct Ericsson system to clean the completed recordings when the deadline comes.

**4.1.1. Settings**

The following properties are needed for the migration tool to work.

|  |
| --- |
| ericssonSubscriberUrl: '<http://69.76.117.195:5927/v2/subscribers/search'>  ericssonSubscriberAdTagUrl: '<http://69.76.122.22:5928/v1/home'>  ericssonSubscriberSeriesUrl: '<http://69.76.110.4:5927/v2/series_recordings'>  ericssonSubscriberRecordingUrl: '<http://peakview-fe-prod.timewarnercable.com:5927/v2/subscribers/recordings'>  ericssonDeleteRecordingUrl: '<http://69.76.110.4:5927/v2/recordings/delete'>  ericssonDeleteSeriesUrl: '<http://69.76.110.4:5927/v2/series_recordings'>  cdvrCsBaseUrl: '<http://localhost:25100/cs/v3'>  cdvrSmsBaseUrl: '<http://localhost:25130/sms/v3'> |

**4.1.2. Migration Steps**

The migration tool should follow the following ordered steps to do proper migration.

* Retrieve active subscriber data from Ericsson.
* Retrieve AD tags for subscribers from Ericsson.
* Combine subscriber data and AD tags, and create mapping subscriber in SMS.
* Retrieve series recordings from Ericsson.
* Convert and re-schedule the series recording in CS.
* Retrieve scheduled event recordings from Ericsson.
* Convert and re-schedule the event recordings in CS.
* Retrieve completed recordings from Ericsson.
* Convert and inject completed recordings to CS. Recordings belongs to an existing series recording will be attached to the series recording. Set the retention expire time at the deadline set for Ericsson recordings.
* (Optional) Cancel series recordings and scheduled event recordings in Ericsson system.

**4.1.3. Migration Report**

At the end of the log, the migration tool should output a summary of the migration.

|  |
| --- |
| ==================================Migration Summary Report==============================  Total Number of Subscribers in Ericsson: 6873  The Number of Subscribers Migrated Successfully: 6629  The List of Subscribers Migrated Successfully:  1 - subscriber00001  2 - subscriber00002  ...  6629 - subscriber06629  The Number of Subscribers Migrated with Failures: 244  The List of Subscribers Migrated with Failures:  1 - subscribererror00001  2 - subscribererror00002  ...  244 - subscribererror00244  -----------------------------------------------------------  Total Number of Series in Ericsson: 56  The Number of Series Migrated Successfully: 50  The List of Series Migrated Successfully:  1 - series00001  2 - series00002  ...  50 - series00050  The Number of Series Migrated with Failures: 6  The List of Series Migrated with Failures:    1 - series00001  2 - series00002  ...  6 - series00006  -----------------------------------------------------------  Total Number of Scheduled Recordings in Ericsson: 56  The Number of Scheduled Recordings Migrated Successfully: 50  The List of Scheduled Recordings Migrated Successfully:  1 - schedulerecording00001  2 - schedulerecording00002  ...  50 - schedulerecording00050  The Number of Scheduled Recordings Migrated with Failures: 6  The List of Scheduled Recordings Migrated with Failures:    1 - schedulerecording00001  2 - schedulerecording00002  ...  6 - schedulerecording00006  -----------------------------------------------------------  Total Number of Completed Recordings in Ericsson: 56  The Number of Completed Recordings Migrated Successfully: 50  The List of Completed Recordings Migrated Successfully:  1 - recording00001  2 - recording00002  ...  50 - recording00050  The Number of Completed Recordings Migrated with Failures: 6  The List of Completed Recordings Migrated with Failures:    1 - recording00001  2 - recording00002  ...  6 - recording00006 |

4.2. Changes Needed in CS

* Add new boolean "fromEricsson" to Recording to indicate if a recording is migrated from Ericsson. Default is false. Database and query response should be modified to support it.
* Pass "fromEricsson" to BG when deleting a completed recording, so that BG can delete recording on Cisco recording router or Ericsson depending on this field.
* Remove channel entitlement check when scheduling recordings.
* Add REST API to support entitlement removal for a subscriber.
* Add REST API to migrate completed event and series recordings from Ericsson.

**4.2.1. New boolean "fromEricsson" in Recording**

Dabase table "recording" should be inserted with another column "fromericsson".

|  |
| --- |
| Column       |           Type           |                       Modifiers  --------------------+--------------------------+-------------------------------------------------------   id                 | bigint                   | not null default nextval('seq\_recordingid'::regclass)   uniqueid           | character varying(255)   | not null   recordingstart     | timestamp with time zone |   recordingend       | timestamp with time zone |   status             | character varying(255)   | not null   copytype           | character varying(255)   | not null   copytypekey        | character varying(255)   |   program            | bigint                   |   failoffset         | timestamp with time zone |   retentionperiod    | timestamp with time zone |   channelid          | character varying(255)   |   airingid           | character varying(255)   |   programid          | character varying(255)   |   episodetitle       | character varying(255)   |   seriesid           | character varying(255)   |   title              | character varying(255)   |   lastmodified       | timestamp with time zone |   laststatusmodified | timestamp with time zone |   createdon          | timestamp with time zone |   errorcode          | character varying(255)   |   errormessage       | character varying(255)   |   duration           | double precision         |   startoffset        | double precision         |   endoffset          | double precision         |   version            | bigint                   |   contentid          | character varying(255)   |   fromericsson       | boolean                  | default false |

Database migration script should be created to handle the addition of "fromericsson" column.

If "fromericsson" is true, the field "contentid" could be used to carry Ericsson's data for constructing playback URL in Ericsson system.

**4.2.2. New RESP APIs**

**4.2.2.1. API for Entitlement Update**

**Request**

Method: POST

Path: /cs/v3/subscribers/{subID}/entitlements

Body: a list of channel IDs (Station IDs)

|  |
| --- |
| {      "removedEntitlements": [          "58515",          "19566",          "59627"      ]  } |

**Response**

200 OK

400 Bad Request

**4.2.2.2. API for Event Recording Migration**

**Request**

Method: POST

Path: /cs/v3/subscribers/{subID}/recordingmigration

Body: it contains all information needed to create program, recording and recordingreference.

|  |
| --- |
| {    "Recording": [          {              "ShowingID": "AUTO\_60048EP01412293087214846904000001483535731302",              "Channel": "60048",              "ChannelCallLetter": "TOONHD",              "ShowStartTime": "1484690400",              "ShowEndTime": "1484691300",              "StartTime": "1484690894",              "EndTime": "1484691300",              "IsProtected": false,              "Bookmark": "0",              "ProgramInfo": {                  "ProgramID": "EP014122930872",                  "Name": "The Amazing World of Gumball",                  "SubTitle": "",                  "SeasonID": "12630964",                  "SeasonName": "",                  "SeriesID": "8648428",                  "SeriesName": "The Amazing World of Gumball",                  "Rating": "0",                  "Description": "About Gumball",                  "Year": 0,                  "EpisodeNumber": 39,                  "EpisodeTitle": "The Stories",                  "Genre": "",                  "Type": "Program"              }          }      ]  } |

**Response**

200 OK

400 Bad Request

|  |
| --- |
| {    "results": [      {          "status": 0,          "message": "success or error message"      }    ]  } |

Each result inside the list maps to one recording migration. Status 0 is success, while non-zero status means error.

4.3. Changes Needed in BG

* Support boolean "fromEricsson" in recording for playback query from FM.
* Support deleting recordings from Ericsson if "fromEricsson" is true.
* Remove channel entitlement check when scheduling recording on recording router.
* Make sure all migrated completed recordings from Ericsson be deleted after the set deadline.

**4.3.1. Delete Recording from Ericsson System**

Ericsson uses a batch-method of deleting recordings, where you specify one Recording Id (ShowingID) and a list of Subs (HomeID). Since unique copy is used in Charter, one recording should only belong to one subscriber. So only one home ID is used when deleting a recording in CDVR.

|  |
| --- |
| $ postjson -i -d '{      "ShowingID": "AUTO\_19548EP01859571004515222078000001521117067267",      "Homes": {          "Home": [              {                  "HomeID": "CSteinhoff"              }          ]      }  }' 'http://69.76.110.4:5927/v2/recordings/delete'    HTTP/1.1 200 OK  Server: Ericsson VSPP Scheduler (Version: 4.2.7.0 SVN Revision: 84906)  Content-Length: 102  Content-Type: application/json; charset=utf-8  Date: Tue, 27 Mar 2018 18:27:01 GMT  Pragma: no-cache  Connection: keep-alive  Keep-Alive: timeout=19  Vary: accept-encoding    {      "RecordingStatus": [          {              "HomeID": "CSteinhoff",              "Status": 200,              "VacatedRecordingForecast": {                  "Recording": []              }          }      ]  } |

Possible "Status" codes:

* 200 - OK
* 400 - Bad Request
* 404 - Recording not found
* 471 - Unknown Subscriber Identifier
* 472 - Unknown ShowingID
* 500 - Internal Server Error

4.4. Changes Needed in SMS

* Support Empty NcsServiceIds array while creating a Erricson Migrated subscriber

5. References

* Ericsson's Subscriber Data Mode: [~~SVCDVR-589~~](https://jira.charter.com/browse/SVCDVR-589) - Document Ericsson's Subscriber Data Model **CLOSED**
* Ericsson's Recording Data Model: [~~SVCDVR-590~~](https://jira.charter.com/browse/SVCDVR-590) - Document Ericsson's Recording Data Model **CLOSED**
* Ericsson's Series Data Model: [~~SVCDVR-591~~](https://jira.charter.com/browse/SVCDVR-591) - Document Ericsson's Series Data Model **CLOSED**
* How to Cancel Scheduled Recordings in Ericsson: [~~SVCDVR-640~~](https://jira.charter.com/browse/SVCDVR-640) - Document How to Cancel Scheduled Recordings in Ericsson **CLOSED**
* How to Cancel Series in Ericsson: [~~SVCDVR-641~~](https://jira.charter.com/browse/SVCDVR-641) - Document How to Cancel Series in Ericsson **CLOSED**
* Kumo/FM interaction for Ericsson Playback: [~~SVCDVR-592~~](https://jira.charter.com/browse/SVCDVR-592) - Document Ericsson's Playout **CLOSED**