|  |  |  |  |
| --- | --- | --- | --- |
| gsma_logo_colour_rgb_sm | | RCSJTA TT BB CR\_030  Video Share API | |
| Meeting Information | | | |
| Meeting Name and Number | |  | |
| Meeting Date | |  | |
| Meeting Location | | Conference Call | |
| Document Information | | | |
| Document Author(s) | | Heri Parid Pardian, Samsung | |
| Document Creation Date | | June 10th 2014 | |
|  | | Approval | X |
| This document is for: *(mark X as appropriate)* | | Discussion |  |
|  | | Information only |  |
| Security Classification – Non Confidential / Confidential GSMA Material (Delete as appropriate and delete the rows below that do not apply) | | | Can be distributed to: (mark X as appropriate or specify group |
| Non Confidential | | Public |  |
| Confidential | | Project Team or Group | X |
| Confidential | | GSMA HQ Staff | X |
| Confidential | | GSMA Full Members | X |
| Confidential | | GSMA Associate Members | X |
| Confidential | | GSMA Rapporteur Members | X |
| Document Summary | | | |
| Video Share API proposal for hiding complexity to render video stream | | | |
| Document History | | | |
| Date | Version | Author / Comments and for updates description of changes | |
| 06/10/2014 | 0.1 | Initial doucment | |
|  |  |  | |
|  |  |  | |
|  |  |  | |

© GSMA 2010. The GSM Association (“Association”) makes no representation, warranty or undertaking (express or implied) with respect to and does not accept any responsibility for, and disclaims liability for the accuracy or completeness or timeliness of the information contained in this document. The information contained in this document may be subject to change without prior notice. This document has been classified according to the GSMA [Document Confidentiality Policy](https://infocentre.gsm.org/cgi-bin/prddets.cgi?274175). GSMA meetings are conducted in full compliance with the GSMA [Antitrust Policy](https://infocentre.gsm.org/cgi-bin/docdisp.cgi?275305).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| RCC TF Change Request Form | | | | |
| Title: | 4.4.8 Video Share API | | | |
| Type of Change Request: Mark the appropriate box by “X” | New Feature |  | Major   |  | | --- | | X |   Change Errata |  |
| Targeted Document | joyn Terminal API Specification | | | |
| Impact on Other Official Documents: | None | | | |

======================= Summary of Changes =============================

06/10/2014 – Samsung : propose simplicity for Video Share API

======================= End of Summary =================================

======================= 1st Change ==================================

# Video Share API

## Background

GSMA API provides Video Share API for 3rd party developers which let 3rd party developers handle video stream to be rendered by 3rd party apps using 3rd party codec. The API is not relatively easy to be used by common 3rd party developers, 3rd party developers must consider about rendering video stream to be able to be played into video canvas.

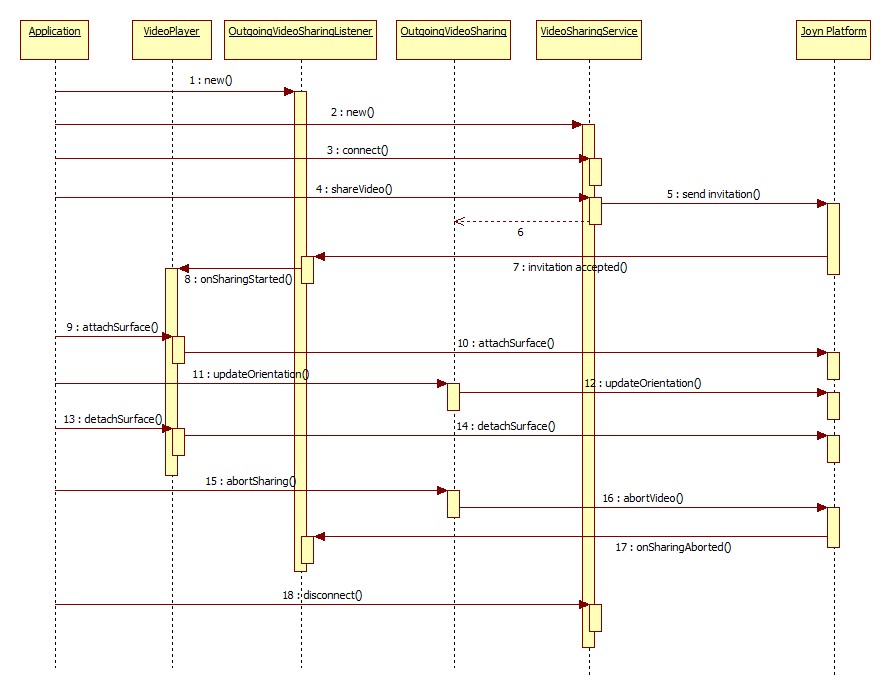
This document propose the Video Share API which hiding the complexity especially when 3rd party developers handle video stream. This API proposal offers simplicity for 3rd party developers, so that 3rd party developers don’t have to worry about the complexity of handling encoding/decoding video stream. 3rd party developers just need to attach Android Surface to the API and let core side to do so.

This API proposal exposes all functionality related to sharing live video stream via the Video Share Service. It covers all of the existing GSMA API Video Sharing API usecases.

## Video Share API proposal calling flow

The Video Share API proposal service provides the API through which the user can share or and receive the video stream. The following figures in this section contains basic call flows of the Video Share API.

Figure 1 is an example that shows how to share video (outgoing share).



1. : Share the video stream to a certain contact
2. The RCS client instantiates a listener instance of the OutgoingVIdeoSharingListener. At this time, it also specifies a contact to share video stream with.
3. The RCS client instantiates a service instance of the VideoSharingService.
4. The RCS client establishes a connection with the instance of VideoSharingService.
5. The RCS client shares a video stream with given contact and the instance of OutgoingVideoSharingListener. It will return instance of OutgoingVideoSharing and send the invitation to a given contact.
6. When the invititation is accepted by remote end, the listener function (onSharingStarted step 8) is invoked to inform the RCS client that sharing session has been established. The onSharingStarted provides an instance of VideoRenderer which will be used to attach/detach Android Surface instance.
7. As depicted in Figure 1 step 9, The RCS client attaches Surface instance to be used by JoynPlatform, JoynPlatform will render the video stream to the underlying Surface.
8. While video share session is in progress, The RCS client can update the orientation of camera, and let JoynFlatform inform the other end by calling updateOrientation as depicted on step 11.
9. Finally, the RCS client terminates the video sharing session by calling abortSharing method and disconnects from the VideoSharingService. At this time, the VideoSharingService discards all listeners associated with this client.

The incoming video sharing flows is shown at the following Figure 2.

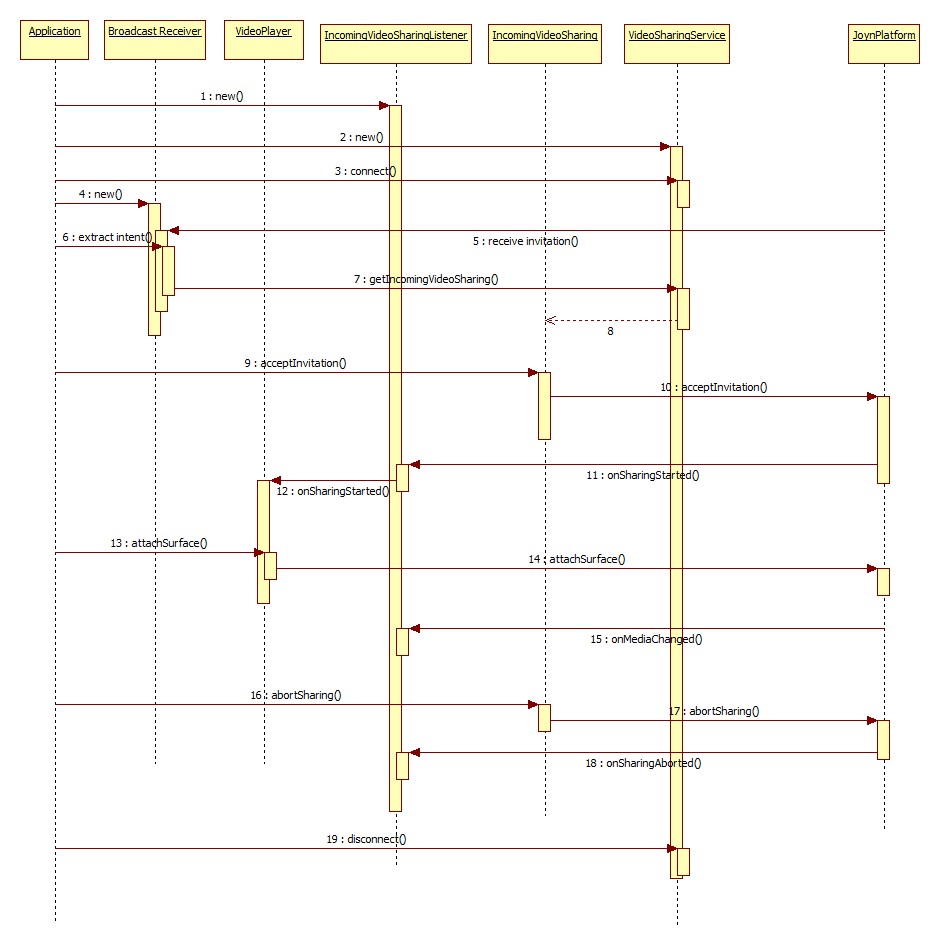


Figure 2: Get share video stream for a certain contact.

1. The RCS client instantiates a listener instance of the IncomingVideoSharingListener.
2. The RCS client instantiates a service instance of the VideoSharingService.
3. The RCS client establishes a connection with the instance of VideoSharingService.
4. The RCS client instantiates an instance of the Android Broadcast Receiver with a specific intent filter to get Invitation event from remote end.
5. When JoynPlatfom receives an invitation from the remote end, it will notify the RCS client’s broadcast receiver that the new invitation for sharing video is waiting for the confirmation, whether it will be accepted or rejected by the RCS client.
6. The RCS client accepts the invitation by calling acceptInvitation method.
7. When the invititation is accepted by the RCS client, the listener function (onSharingStarted step 12) is invoked by JoynPlatform to inform the RCS client that sharing session has been established. The onSharingStarted provides an instance of VideoRenderer which will be used to attach/detach Android Surface instance.
8. As depicted in Figure 2 step 13, The RCS client attaches Surface instance to be used by JoynPlatform, JoynPlatform will render the video stream to the underlying Surface.
9. While video share session is in progress, The RCS client can receive the orientation event update which performed by remote end as depicted on step 15.
10. Finally, the RCS client terminates the video sharing session by calling abortSharing method and disconnects from the VideoSharingService. At this time, the VideoSharingService discards all listeners associated with this client.

## API List

#### Package

Package name **com.gsma.services.rcs.vsh**

#### Methods and Callbacks

Class **VideoSharingService**:

This class offers the main entry point to share a live video during a CS call, when the call hangs up the sharing is automatically stopped. Only one application may initiate and/or accept video sharing.

* Method: connects to the API.

void connect()

* Method: disconnects from the API.

void disconnect()

* Method: starts outgoing sharing live video share with a contact. The parameter contact supports the following formats: MSISDN in national or international format, SIP address, SIP-URI or Tel-URI.

OutgoingVideoSharing shareVideo(String contact, OutgoingVideoSharingListener listener) throws JoynServiceException, JoynContactFormatException

* Method: Starts incoming live video share with a given sharingId from the invitation event.

IncomingVideoSharing getIncomingVideoSharing(String sharingId) throws JoynServiceException

* Method: returns an active outgoing live video share

OutgoingVideoSharing getOutgoingActiveVideoSharing(OutgoingActiveVideoSharingListener listener) throws JoynServiceException

* Method: returns an active incoming live video share.

IncomingVideoSharing getIncomingActiveVideoSharing(IncomingActiveVideoSharingListener listener) throws JoynServiceException

* Method: returns the configuration of video sharing service.

VideoSharingServiceConfiguration getConfiguration() throws JoynServiceException

Class **VideoSharingListener**:

This is a base class for listening to the incoming/outgoing video sharing event.

* Method: callback called when the sharing has been aborted or terminated.

void onSharingAborted()

* Method: callback called when the sharing has failed. The available value are:   
  0 – Sharing failed  
  1 – Invitation declined

void onSharingError(int error)

Class **OutgoingVideoSharingListener**:

This class which extends VideoSharingListener is used to listen to the outgoing video sharing event.

* Method: callback called when the sharing has been established.

void onSharingStarted(VideoPlayer player)

Class **IncomingVideoSharingListener**:

This class which extends VideoSharingListener is used to listen to the incoming video sharing event.

* Method: callback called when the sharing has been established.

void onSharingStarted(VideoPlayer player)

* Method: callback called when user changed the camera orientation.

void onCameraChanged(VideoDescriptor descriptor)

Class **OutgoingActiveVideoSharingListener**:

This class which extends VideoSharingListener is used to listen to an active outgoing video sharing event.

* Method: handles video player while video sharing is in progress.

void handleVideoPlayer(VideoPlayer player)

Class **IncomingActiveVideoSharingListener**:

This class which extends VideoSharingListener is used to listen to an active incoming video sharing event.

* Method: handles video player while video sharing is in progress.

void handleVideoPlayer(VideoPlayer player, VideoDescriptor descriptor)

* Method: callback called when user changed the camera orientation.

void onCameraChanged(VideoDescriptor descriptor)

Class **VideoSharing**:

This class represents video sharing session.

* Method: returns the sharing ID of the video sharing.

String getSharingId() throws JoynServiceException

* Method: returns the remote contact.

String getRemoteContact() throws JoynServiceException

* Method: returns the state of the sharing. The available value are:   
  INVITED, INITIATED, STARTED, TERMINATED, ABORTED, and FAILED

int getState() throws JoynServiceException

* Method: aborts the sharing.

void abortSharing() throws JoynServiceException

Class **OutgoingVideoSharing**:

Class which extends VideoSharing represents outgoing video sharing session.

* Method: updates the orientation of video. The available parameter orientation are:

1 – LANDSCAPE  
2 – PORTRAIT

void updateOrientation(int orientation) throws JoynServiceException

* Method: toggles the source of camera.

void switchCamera() throws JoynServiceException

Class **IncomingVideoSharing**:

Class which extends VideoSharing represents incoming video sharing session.

* Method: accepts an incoming video sharing invitation.

void acceptInvitation(IncomingVideoSharingListener listener) throws JoynServiceException

* Method: rejects an incoming video sharing invitation.

void rejectInvitation() throws JoynServiceException

Class **VideoPlayer**:

Media for processing video stream to Android Surface.

* Method: attaches video surface to be proceed by core implementation.

void attachSurface(Surface surface, VideoDescriptor descriptor) throws JoynServiceException

* Method: detaches video surface from core implementation.

void detachSurface() throws JoynServiceException

Class **VideoDescriptor**:

Class represents an object for video share parameters.

* Method: returns the orientation of the video stream.

int getOrientation()

* Method: returns the width of video frame.

int getWidth()

* Method: returns the height of video frame.

int getHeight()

Class **VideoSharingServiceConfiguration**:

Class represents video sharing service configuration.

* Method: returns the maximum authorized duration of the video sharing.

long getMaxTime()

#### Intent

The following intent is used by 3rd party application for receiving an incoming invitation.

**Action**:

com.gsma.services.rcs.vsh.action.NEW\_VIDEO\_SHARING

**Extra:**

contact (String)-MSISDN of the contact sending invitation.

contactDisplayname (String) - display name of the contact sending the invitation

sharingId (long) – The identity of incoming video sharing session.

======================= End of changes================================