AT&T Service Specification

Category: Call management

Service: WebRTC

Revision: 1 1 3

Revision Date: date 1/20/2014

Author: Developer Realization Team

Legal Disclaimer

This document and the information contained herein (collectively, the "Information") is provided to you (both the individual receiving this document and any legal entity on behalf of which such individual is acting) ("You" and "Your") by AT&T, on behalf of itself and its affiliates ("AT&T") for informational purposes only. AT&T is providing the Information to You because AT&T believes the Information may be useful to You. The Information is provided to You solely on the basis that You will be responsible for making Your own assessments of the Information and are advised to verify all representations, statements and information before using or relying upon any of the Information. Although AT&T has exercised reasonable care in providing the Information to You, AT&T does not warrant the accuracy of the Information and is not responsible for any damages arising from Your use of or reliance upon the Information. You further understand and agree that AT&T in no way represents, and You in no way rely on a belief, that AT&T is providing the Information in accordance with any standard or service (routine, customary or otherwise) related to the consulting, services, hardware or software industries.

AT&T DOES NOT WARRANT THAT THE INFORMATION IS ERROR-FREE. AT&T IS PROVIDING THE INFORMATION TO YOU "AS IS" AND "WITH ALL FAULTS." AT&T DOES NOT WARRANT, BY VIRTUE OF THIS DOCUMENT, OR BY ANY COURSE OF PERFORMANCE, COURSE OF DEALING, USAGE OF TRADE OR ANY COLLATERAL DOCUMENT HEREUNDER OR OTHERWISE, AND HEREBY EXPRESSLY DISCLAIMS, ANY REPRESENTATION OR WARRANTY OF ANY KIND WITH RESPECT TO THE INFORMATION, INCLUDING, WITHOUT LIMITATION, ANY REPRESENTATION OR WARRANTY OF DESIGN, PERFORMANCE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, OR ANY REPRESENTATION OR WARRANTY THAT THE INFORMATION IS APPLICABLE TO OR INTEROPERABLE WITH ANY SYSTEM, DATA, HARDWARE OR SOFTWARE OF ANY KIND. AT&T DISCLAIMS AND IN NO EVENT SHALL BE LIABLE FOR ANY LOSSES OR DAMAGES OF ANY KIND, WHETHER DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL, PUNITIVE, SPECIAL OR EXEMPLARY, INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, LOSS OF GOODWILL, COVER, TORTIOUS CONDUCT OR OTHER PECUNIARY LOSS, ARISING OUT OF OR IN ANY WAY RELATED TO THE PROVISION, NON-PROVISION, USE OR NON-USE OF THE INFORMATION, EVEN IF AT&T HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH LOSSES OR DAMAGES.

Revision History

Date	Revision	Description
12/19/201	30 0 1	rs662h/sf045m - DPS Internal
12/23/201	30 0 4	rs662h/sf045m - Updated the operation - Still part of Initial Draft
12/26/201	30 0 5	rs662h/sf045m - Added Events and ongoing file transfer handling
1/03/2014	1.1.2	rs662h/sf045m - Initial Draft
1/20/2014	1.1.3	rs662h/sf045m - 1. Incorporated review comments from E2E team (Jeff and Frank),Joe and other teams. 2. Service Errors added in sections wherever applicable. 3. Overall Call flow section added in appendix. 4. Assumption and dependencies section updated with stickiness and CORS.

Contents

1 Introduction

This WebRTC API augmented with In-App Calling for Mobile Numbers (ICMN) provisioning provides AT&T WebRTC-enabled IP voice telephony between 3G, 4G, LTE, PSTN, and internet-connected users. Additionaly this API provides video , 1-1 chat, group chat and file transfer functionality for the users with WebRTC endpoints.

1.1 Audience

General audience for API service specifications are:

• Gateway team (development team)

• Architect and Feature owner

• North Bound Governance team

• Sample App and SDK/IDE teams

1.2 Assumptions and Dependencies

- **Geo-Redundancy / stickiness**: OAuth Authorize , Assign VTN / User Name, require stickiness due to the expected latency for data replication between the 3 active BlackFlag Data Centers. This solution TBD will be leveraged for WebRTC GA as required once it is available.

  Note : We will likely need first AT usage stickiness rules to ensure that once an Access Token is generated, provisioned to API GW, and returned to client calling application, that there could be a race condition between the time the AT/RT and associated Metadata is replicated to other BF DCs, therefore stickiness with first time AT usage should stick to BF DC where the AT was created. This is because for WebRTC a Developer Hosted Server makes the get Token request and its first usage is from a specific client Browser/User Agent which could be in a different geo-location and therefore may result in a request failure if DC replication has not completed.

- **CALEA requirements** : To meet the CALEA requirement WebRTC endpoint Client IP needs to be captured.

- **CORS support** : For browser based libraries Gateway must have the support for CORS.

2 Provisioned Data

2.1 Oauth Scope

The value for the OAuth scope parameter used with this API is as shown below.

Model	OAuth Scope Value	Brief Description
Authorization Code	RTC	For ICMN case
Client Credentials	RTC	For VTN and noTN case

2.2 Provisioning: *WebRTC API*

The Developer shall select the service name “WebRTC” in the app provisioning form (of Matrix), The domain must have been previously registered by the OPA(Organization Profile Adminstrator). The VTNs may be selected by area code or not selected (no TN case).

The provisioning form for this API may be similar to that shown in the mock-up below.

WebRTC [X]
App Domain: my-org.com

Select an area code for your app VTNs	206	<VTN is populated upon save>

App account profile information is provisioned once by the Developer when the app account is created. This profile data may later be changed. The table below shows API-specific, app account profile provisioning fields, and some notes on the rules constraining the values stored in these fields.

Parameter	***Data Type***	***Value Rules***
Scope	String	Ability to use the “RTC” scope value is added to the app profile when the Developer selects “RTC“ in the App provisioning form.
App VTNs	Phone number list	An app may have zero, one, or many VTNs allocated to it based on the Developer preference at app provisioning time. The format of VTN is 10 digit phone number. At runtime, the Developer may decide whether to offer a VTN or no VTN experience (username only).
OPA provided domain	Unique org domain	This is a unique domain per Org. This is defined and provisioned by the OPA. This provisioning must be done before WebRTC API provisioning to apps may be done. This domain is used in both the VTN and the noTN external ID creation cases. The third case is ICMN which uses the AT&T provided domain “mon.api.att.com”.

Figure 1 – Key provisioning elements in the App Provisioning form

At app runtime, when the system is assigning external identities to call sessions, the system constructs an external ID of the app instance to use. In cases where external identity is to be a VTN or username, this run-time provisioning uses the OPA-provided domain (that is unique across Portal Developer Orgs) plus the user name or VTN.

- <user_name>@OPA_provided_domain
- <VTN>@OPA_provided_domain

In the ICMN case, a special domain called mon.api.att.com is used to construct the external identity in the IMS network. This type of external identity is given in the following format:

- <ICMN MSISDN>@mon.api.att.com

3 E911: *Requirements*

Enhanced 911, E-911, or E911 is a system used in North America that links emergency callers with the appropriate public resources. E911 system provides both caller location and identification.This location may be a physical address or other geographic reference information such as X/Y map coordinates. The caller's telephone number is used in various ways to derive a location that can be used to dispatch police, fire, emergency medical, and other response resources.

In North America the incoming 9-1-1 call is typically answered at the Public Safety Answering Point of the governmental agency that has jurisdiction over the caller's location. The U.S. Federal Communications Commission (FCC) has several requirements applicable to wireless or mobile telephones and network operator's in-service phones must be E911-compliant. As a result WebRTC endpoint needs to pass a valid E911 identifier as a part of session registration. The E911 ID is not required in NoTN case. To get a valid E911 identifier, an existing API may be used.

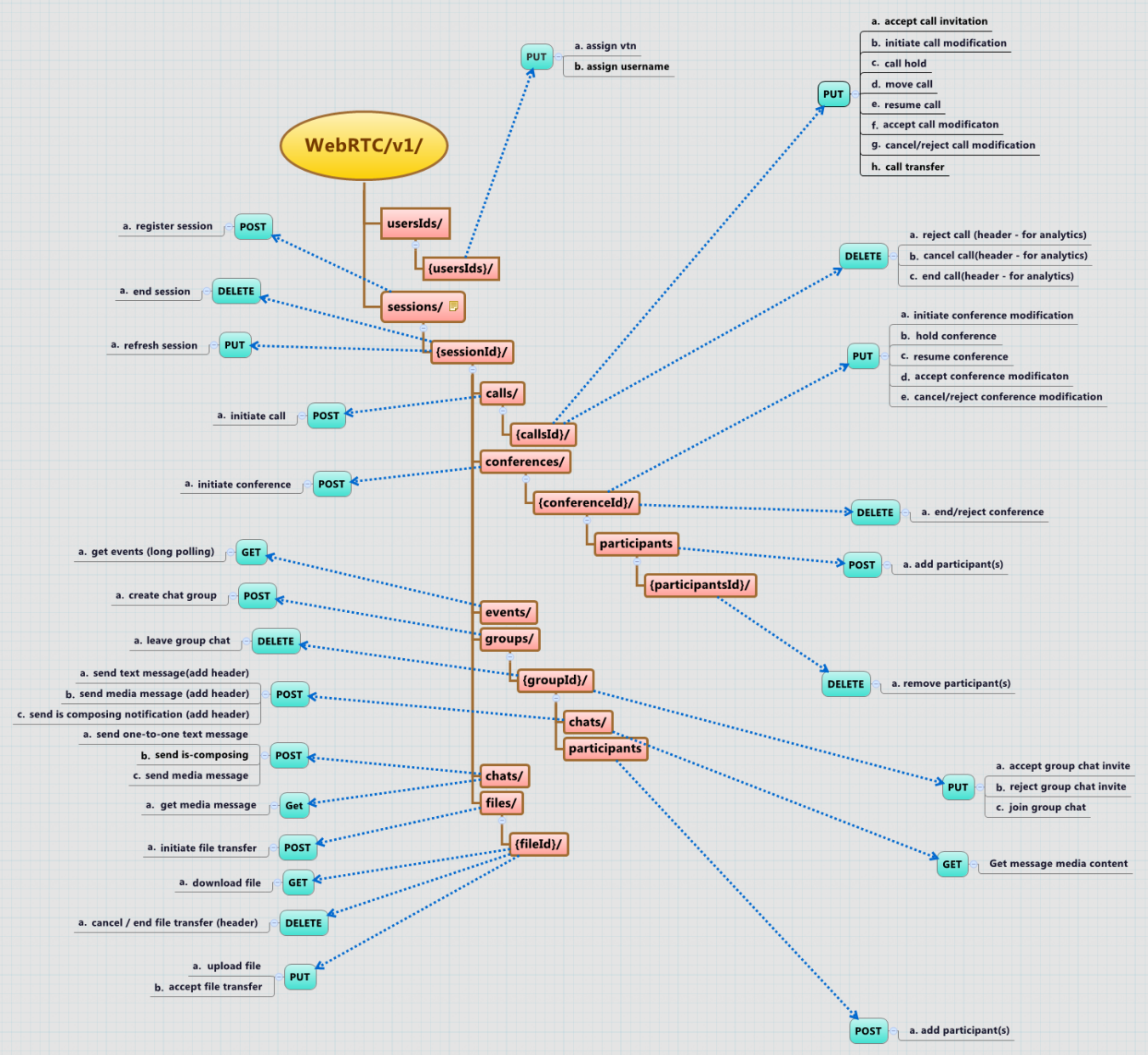
4 RESTful Web Services Definition

4.1 REST Operation Summary

Operation Name	Resource URI relative to https://api.att.com/webRTC	HTTP Verb
User Management		
Associate userId with token	/v1/userIds/{userId}	POST
Session Management		
Create Session	/v1/sessions/	POST
Refresh Session	/v1/sessions/{sessionId}	PUT
End Session	/v1/sessions/{sessionId}	DELETE
Events		
Get Events	/v1/events/	GET
Calls		
Start Call	/v1/calls/	POST
Modify Call	/v1/calls/{callId}	PUT
End Call	/v1/calls/{callId}	DELETE
Conference		
Create Conference	/v1/conferences	POST
Add Participant to Conference	/v1/conferences/{conferenceId}/participants/{participantId}	PUT
Remove Participant from Conference	/v1/conferences/{conferenceId}/participants/{participantId}	DELETE
Modify Conference	/v1/conferences/{conferenceId}/	PUT
Cancel End Reject Conference	/v1/conferences/{conferenceId}/	DELETE
Group Chat		
Create Chat Group	/v1/groups/	POST
Modify Group Chat	/v1/groups/{groupId}	PUT
Send Message to Group Chat	/v1/groups/{groupId}/chats/	POST
Get Media From Group Chat	/v1/groups/{groupId}/chats/?filerefnum	GET
Add Participants To Group Chat	/v1/groups/{groupId}/participants/	POST

Operation Name	Resource URI relative to https://api.att.com/webRTC	HTTP Verb
Exit Group Chat	/v1/groups/{groupId}/	DELETE
One-to-One Chat		
Send Message To One-to-One Chat	/v1/chats/	POST
Get Media From One-to-One Chat	/v1/chats/?filerefnum	GET
File Transfer		
Initiate File Transfer	/v1/files/	POST
Get File Data	/v1/files/{fileId}	GET
Modify File Transfer	/v1/files/{fileId}	PUT
Cancel End File Transfer	/v1/files/{fileId}	DELETE

Resource Tree Structure ::



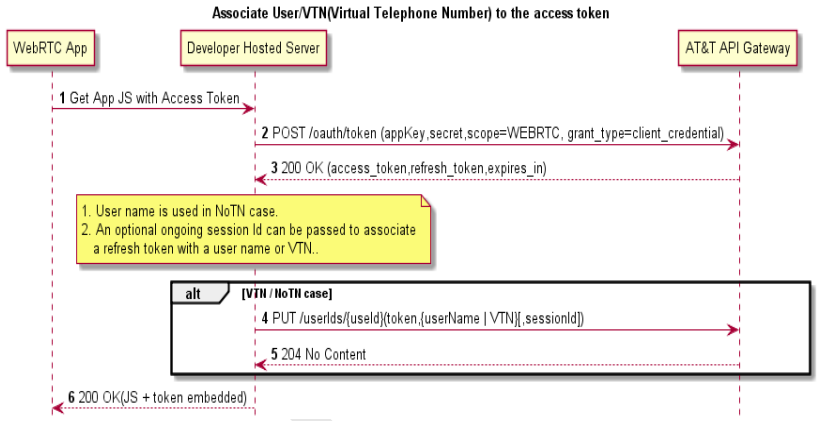
4.2 Operation: Associate userId with token

4.2.1 Functional Behavior

The Associate UserId With Token method is used to associate a user name or VTN (Virtual Telephone Number) with an OAuth access token.

Note: This method is not applicable for the ICMN case.

4.2.2 Call flow



4.2.3 Version Impact Summary

Service Version	Major or Minor Impact	Changes Introduced by this Version
1	Major	Initial Release

4.2.4 Authentication and Authorization

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description
authorization_code	Yes	RTC	1. Obtain the authorization code by passing App Key and App Secret along with scope. This redirects the user to AT&T authorization page to capture user consent. 2. Obtain OAuth access token using the OAuth authorization code along with the App Key and App Secret by making a Get Access Token request with grant_type=authorization_code. • Supports ICMN case
client_credentials	No	RTC	1. Obtain OAuth access token with the App Key and App Secret along with the scope by making a Get Access Token request with grant_type=client_credentials • Supports VTN case • Supports no-TN case

4.2.5 Representation Formats

4.2.6 Representation Formats

Direction	Supported Respresentation Formats
Request	XML, JSON,URLENCODED
Response	XML, JSON,URLENCODED

4.2.7 Input Parameters

Parameter	Data Type	Req?	Brief description	Location
accept	String	No	Specifies the format of the body of the response. Valid values are: • application/json • application/xml • application/x-www-form-urlencoded The default value is application/json. Per rfc2616: ”If no Accept header field is present, then it is assumed that the client accepts all media types.” By default our services return application/json. Normal Accept header processing rules shall be followed according to rfc2616. *Note*: If there is no entity body in a normal successful response, this parameter is still needed to specify the format in the case of an error response message.	Header
authorization	String	Yes	Specifies the authorization type and token. The acceptable format for this parameter is the phrase ”Bearer OAuth Token” followed by a space ( ) and an OAuth access token. If this parameter value is missing from the header, then the API Gateway returns a message with an HTTP status code of 400 Invalid Request. If the OAuth access token is not valid, then the API Gateway returns an HTTP status code of 401 Unauthorized with a WWW-Authenticate HTTP header.	Header
userId	String	Yes	Specifies the user name or **VTN** which needs to be associated with the OAuth access token. This parameter should be used for **VTN** or **NoTN** cases only. • Format for **VTN** is vtn:<10 digit number> • Format for **NoTN** is alphanumeric.	URI Path
sessionId	String	No	Specifies the ongoing WebRTC session identifier to enable the client app associate a refresh token with a user name or VTN.	Header

4.2.7.1 Request – Example (Associate token with user name - JSON)

This operation is used to associate user name with an OAuth access token.

```
PUT /RTC/v1/userIds/alice HTTP/1.1
Host: api.att.com
Authorization: Bearer abcdef12345678
Content-Type: application/json
Accept: application/json
```

4.2.7.2 Request – Example (Associate token with user name - XML) This operation is used to associate user name with an OAuth access token. The response format is XML.

```
PUT /RTC/v1/userIds/alice HTTP/1.1
Host: api.att.com
Authorization: Bearer abcdef12345678
Content-Type: application/xml
Accept: application/xml
```

4.2.7.3 Request – Example (Associate token with VTN - JSON) This operation is used to associate VTN with an OAuth access token. The response format is JSON.

```
PUT /RTC/v1/userIds/vtn:345695959 HTTP/1.1
Host: api.att.com
Authorization: Bearer abcdef12345678
Content-Type: application/json
Accept: application/json
```

4.2.7.4 Request – Example (Associate token with VTN - XML) This operation is used to associate VTN with an OAuth access token. The response format is XML.

```
PUT /RTC/v1/userIds/vtn:345695959 HTTP/1.1
Host: api.att.com
Authorization: Bearer abcdef12345678
Content-Type: application/xml
Accept: application/xml
```

4.2.7.5 Request – Example (Associate token with VTN in an ongoing WebRTC session-JSON) This operation is used to associate VTN with an OAuth access token in an on-going WebRTC session. The response format is JSON.

```
PUT /RTC/v1/userIds/vtn:345695959 HTTP/1.1
Host: api.att.com
Authorization: Bearer abcdef12345678
Content-Type: application/json
Accept: application/json
sessionId: 4ba569b5-290d-4f1f-b3af-255731383204
```

4.2.7.6 Request – Example (Associate token with VTN in an ongoing WebRTC session - XML)

This operation is used to associate VTN with an OAuth access token in an ongoing WebRTC session. The response format is XML.

```
PUT /RTC/v1/userIds/vtn:345695959 HTTP/1.1
Host: api.att.com
Authorization: Bearer abcdef12345678
Content-Type: application/xml
Accept: application/xml
sessionId: 4ba569b5-290d-4f1f-b3af-255731383204
```

4.2.8 Output Parameters

Parameter	Data Type	Req?	Brief description	Location

4.2.8.1 Response – Example (Associate token with user name / VTN success response)

This shows the response to associate VTN / user name with an OAuth access token.

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2010 02:51:59 GMT
```

4.2.9 Service Exceptions

MessageId	Text	Variables	Parent HTTP Code
SVC0001	A service error has occurred. Error code is <error_explanation>	error_explanation : <content_here>	400
SVC0002	Invalid input value for Message part <part_name>	part_name : name of the input parameter that resulted in the error.	400
SVC0003	Invalid input value for Message part <part_name>, valid values are <part_values>	part_name : name of the input parameter that resulted in the error. part_value : value of input parameter that was found to be in error.	400
SVC0004	No valid addresses provided in the Message part <part_name>	part_name : name of the input parameter that resulted in the error.	400

4.2.10 Policy Exceptions

MessageId	Text	Variables	Parent HTTP Code
POL0001	A policy error occurred. For example, rate limit error, authentication and authorization error.	N/A	401, 403
POL0002	Privacy verification failed for address <address>, request is refused	address : <content_here>	403
POL0003	Too many addresses specified in Message part	N/A	403
POL1009	User has not been provisioned for %1	%1 : System that has not been provisioned.	403

4.3 **Operation:** Create Session

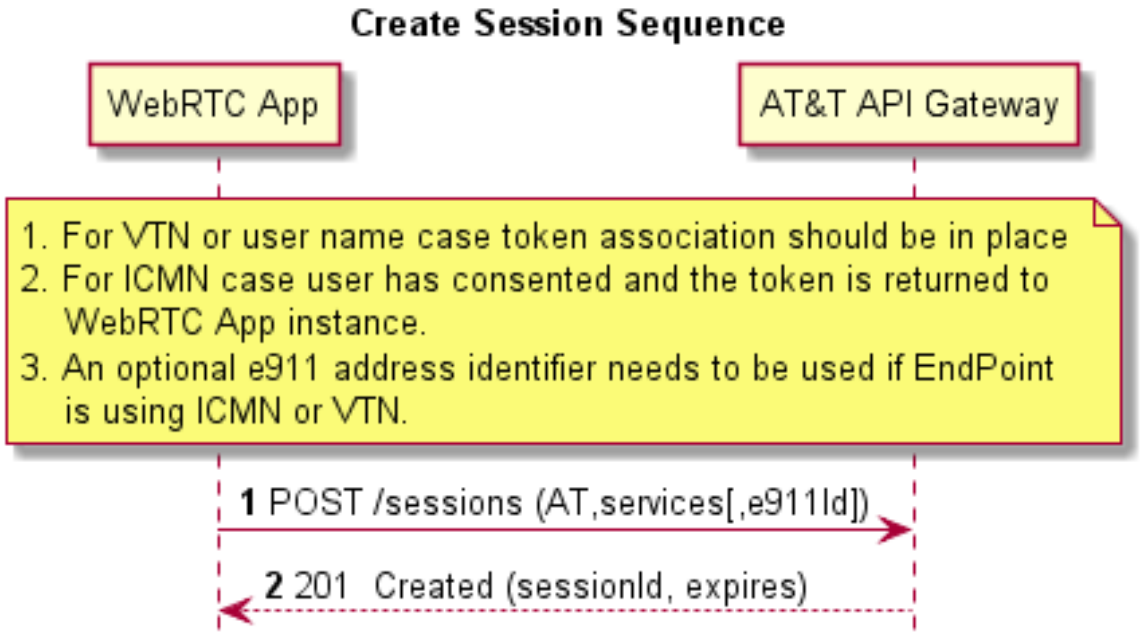
4.3.1 Functional Behavior

The Create Session method creates the WebRTC session. The session identifier is a globally unique value in the system and it represents a single instance of a WebRTC app. The OAuth acccess token used in the request is then logically associated to the newly created session identifier. Optionally, an e911 identifier is used in the request in ICMN and VTN case.

Once a session is created, a valid session identifier is returned to the user. The session expiration time is also returned with the session identifier. The user must refresh the session before session expiration time in order to keep the endpoint active.

The user also must establish an event channel in order to receive notification of various run-time events in which endpoint might be interested and want to respond appropriately.

4.3.2 Call flow



4.3.3 Version Impact Summary

Service Version	Major or Minor Impact	Changes Introduced by this Version
1	Major	Initial Release

4.3.4 Authentication and Authorization

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description
authorization_code	Yes	RTC	1. Obtain the authorization code by passing App Key and App Secret along with scope. This redirects the user to AT&T authorization page to capture user consent. 2. Obtain OAuth access token using the OAuth authorization code along with the App Key and App Secret by making a Get Access Token request with grant_type=authorization_code. • Supports ICMN case
client_credentials	No	RTC	1. Obtain OAuth access token with the App Key and App Secret along with the scope by making a Get Access Token request with grant_type=client_credentials • Supports VTN case • Supports no-TN case

4.3.5 Representation Formats

4.3.6 Representation Formats

Direction	Supported Respresentation Formats
Request	XML, JSON,URLENCODED
Response	XML, JSON,URLENCODED

4.3.7 Input Parameters

Parameter	Data Type	Req?	Brief description	Location
accept	String	No	Specifies the format of the body of the response. Valid values are: • application/json • application/xml • application/x-www-form-urlencoded The default value is application/json. Per rfc2616: ”If no Accept header field is present, then it is assumed that the client accepts all media types.” By default our services return application/json. Normal Accept header processing rules shall be followed according to rfc2616. *Note*: If there is no entity body in a normal successful response, this parameter is still needed to specify the format in the case of an error response message.	Header

Parameter	Data Type	Req?	Brief description	Location
authorization	String	Yes	Specifies the authorization type and token. The acceptable format for this parameter is the phrase ”Bearer OAuth Token” followed by a space ( ) and an OAuth access token. If this parameter value is missing from the header, then the API Gateway returns a message with an HTTP status code of 400 Invalid Request. If the OAuth access token is not valid, then the API Gateway returns an HTTP status code of 401 Unauthorized with a WWW-Authenticate HTTP header.	Header
content-length	Integer	No	Specifies the length of the content in octets. This parameter is only required for a non-streaming request.	Header
content-type	String	Yes	Specifies the type of content of the body. The acceptable values for this parameter are: • application/xml • application/json	Header
x-e911Id	String	Conditional	Specifies the e911 Id associated with the user location which has been retreived through e911 API call. This is required in case of ICMN/VTN scenario. This is not required for no TN/username scenario.	Header
x-Arg	String	No	This parameter will indicate the script, language and version of the code. ClientSDK=<ScriptName><Lang><Version>	Header
session	session Object	Conditional	Contains the session details for creating a session. This parameter is not required for refreshing session using HTTP PUT.	Body

Structure of Session Object

Parameter	Data Type	Required?	Brief description

services	List of Strings	Yes	Contains the list of underlying services required for this session. The acceptable values for this parameter are: • im_chat • file_transfer • ip_voice_call • ip_video_call
mediaType	String	Yes	Specifies the mediaType for audio and video call. This parameter specifies the type of media protocol used for AudioVideo and MediaConference. Only one type is able to be included at registration. Supports **rtmp** and **rtp** values. **Note :** Transparent JSON SDP is not supported in the current implementation.

4.3.7.1 Request – Example (Create Session - JSON)

This example demonstrates how to create a WebRTC session resource that supports underlying services including chat, file transfer, voice, and video telephony. The response format is JSON.

```
POST /WebRTC/v1/sessions HTTP/1.1
Host: api.att.com
Authorization: Bearer abcdef12345678
Content-Type: application/json
Accept: application/json
x-e911Id: f81d4fae-7dec-11d0-a765-00a0c91e6bf6
x-Arg: ClientSDK=<ScriptName><Lang><Version>
{
    "session": {
        "services": [
            "im_chat",
            "file_transfer",
            "ip_voice_call",
            "ip_video_call"
        ]
        "mediaType":"rtp"
    }
}
```

4.3.7.2 Request – Example (Create Session - XML)

This example demonstrates how to create a WebRTC session resource that supports underlying services including Chat, File transfer, voice, and video telephony. The response format is XML.

```
POST /WebRTC/v1/sessions HTTP/1.1
Host: api.att.com
Authorization: Bearer abcdef12345678
Content-Type: application/xml
Accept: application/xml
x-e911Id: f81d4fae-7dec-11d0-a765-00a0c91e6bf6
x-Arg: ClientSDK=<ScriptName><Lang><Version>
<?xml version="1.0" encoding="UTF-8"?>
<session>
   <services>
      <element>im_chat</element>
      <element>file_transfer</element>
      <element>ip_voice_call</element>
      <element>ip_video_call</element>
   </services>
   <mediaType>rtp</mediaType>
</session>
```

4.3.8 Output Parameters

Parameter	Data Type	Req?	Brief description	Location
location	String	Yes	Specifies the URI of the newly created session object. It contains the **session Id** which would be used by other methods during consumption.	Header
x-expires	String	Yes	Specifies the session expiration time in seconds. Note : Refer user_session_timeout in the appendix (parameters)	Header

4.3.8.1 Response – Example (Create Session Response - JSON)

This shows the response to the Create Session request.

```
HTTP/1.1 201 Created
Content-Type: application/json
Date: Thu, 04 Jun 2010 02:51:59 GMT
```

```
location: /WebRTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204
x-expires: 3600
```

4.3.8.2 Response – Example (Create Session Response - XML

This example shows the response to a Create Session request in XML format.

```
HTTP/1.1 201 Created
Content-Type: application/xml
Date: Thu, 04 Jun 2010 02:51:59 GMT
location: /WebRTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204
x-expires: 3600
```

4.3.9 Service Exceptions

MessageId	Text	Variables	Parent HTTP Code
SVC0001	A service error has occurred. Error code is <error_explanation>	error_explanation : <content_here>	400
SVC0002	Invalid input value for Message part <part_name>	part_name : name of the input parameter that resulted in the error.	400
SVC0003	Invalid input value for Message part <part_name>, valid values are <part_values>	part_name : name of the input parameter that resulted in the error. part_value : value of input parameter that was found to be in error.	400
SVC0004	No valid addresses provided in the Message part <part_name>	part_name : name of the input parameter that resulted in the error.	400

4.3.10 Policy Exceptions

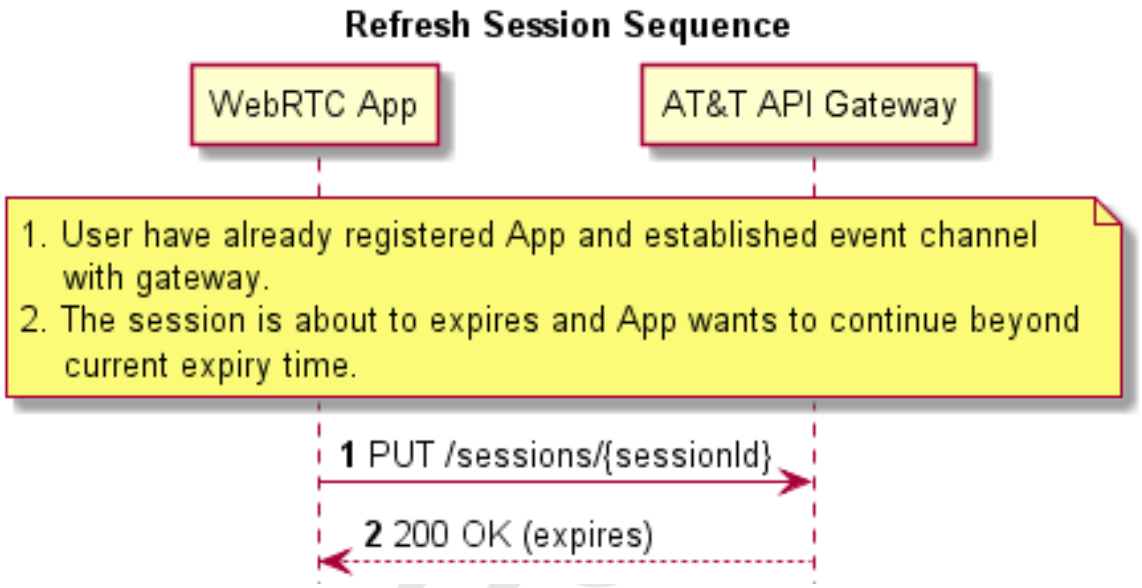
MessageId	Text	Variables	Parent HTTP Code
POL0001	A policy error occurred. For example, rate limit error, authentication and authorization error.	N/A	401, 403
POL0002	Privacy verification failed for address <address>, request is refused	address : <content_here>	403
POL0003	Too many addresses specified in Message part	N/A	403
POL1009	User has not been provisioned for %1	1% : System that has not been provisioned.	403
POL1100	Max number of session exceeded allowed limit %1.	1% : Number of session allowed.	403

4.4 Operation: Refresh Session

4.4.1 Functional Behavior

Refresh session method is used to extend the current session. Session have preconfigured system wide timeout expiry value (refer parameters in appendix for current expiry value). Endpont must refresh the session before expiry.

4.4.2 Call flow



4.4.3 Version Impact Summary

Service Version	Major or Minor Impact	Changes Introduced by this Version
1	Major	Initial Release

4.4.4 Authentication and Authorization

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description
authorization_code	Yes	RTC	1. Obtain the authorization code by passing App Key and App Secret along with scope. This redirects the user to AT&T authorization page to capture user consent. 2. Obtain OAuth access token using the OAuth authorization code along with the App Key and App Secret by making a Get Access Token request with grant_type=authorization_code. • Supports ICMN case
client_credentials	No	RTC	1. Obtain OAuth access token with the App Key and App Secret along with the scope by making a Get Access Token request with grant_type=client_credentials • Supports VTN case • Supports no-TN case

4.4.5 Representation Formats

4.4.6 Representation Formats

Direction	Supported Respresentation Formats
Request	XML, JSON,URLENCODED
Response	XML, JSON,URLENCODED

4.4.7 Input Parameters

Parameter	Data Type	Req?	Brief description	Location
accept	String	No	Specifies the format of the body of the response. The acceptable values for this parameter are: • application/json • application/xml • application/x-www-form-urlencoded The default value is application/json. Per rfc2616: ”If no Accept header field is present, then it is assumed that the client accepts all media types.” By default our services return application/json. The normal Accept header processing rules shall be followed according to rfc2616. *Note*: If there is no entity body in a normal successful response, this parameter is still needed to specify the format in the case of an error response message.	Header
authorization	String	Yes	Specifies the authorization type and token. The acceptable format for this parameter is the phrase ”Bearer OAuth Token” followed by a space ( ) and an OAuth access token. If this parameter value is missing from the header, then the API Gateway returns a message with an HTTP status code of 400 Invalid Request. If the OAuth access token is not valid, then the API Gateway returns an HTTP status code of 401 Unauthorized with a WWW-Authenticate HTTP header.	Header
sessionId	String	Yes	Specifies the session identifier being refreshed.	URI Path

4.4.7.1 Request – Example (Refresh Session - JSON) This example demonstrates how to refresh an existing session before expiry.

```
PUT /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204 HTTP/1.1
Host: api.att.com
Authorization: Bearer abcdef12345678
Content-Type: application/json
Accept: application/json
```

4.4.7.2 Request – Example (Refresh Session - XML)

This example demonstrates how to refresh an existing session before expiration.

```
PUT /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204 HTTP/1.1
Host: api.att.com
Authorization: Bearer abcdef12345678
Content-Type: application/xml
Accept: application/xml
```

4.4.8 Output Parameters

Parameter	Data Type	Req?	Brief description	Location
content-type	String	Yes	Specifies the type of content of the body. The acceptable values for this parameter are: • application/xml • application/json	Header
x-expires	String	Yes	Specifies the session expiration time in seconds.	Header

4.4.8.1 Response – Example (Refresh session) JSON

This shows the response to the Refresh Session request. The output format is JSON.

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 34
Date: Thu, 04 Jun 2010 02:51:59 GMT
x-expires: 3600
```

4.4.8.2 Response – Example (Refresh session) XML

This shows the response to the Refresh Session request. The output format is XML.

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 34
Date: Thu, 04 Jun 2010 02:51:59 GMT
x-expires: 3600
```

4.4.9 Service Exceptions

MessageId	Text	Variables	HTTP Status Code
SVC0001	A service error has occurred. Error code is <error_explanation>	error_explanation : <content_here>	400
SVC0002	Invalid input value for Message part <part_name>	part_name : name of the input parameter that resulted in the error.	400
SVC0003	Invalid input value for Message part <part_name>, valid values are <part_values>	part_name : name of the input parameter that resulted in the error. part_value : value of input parameter that was found to be in error.	400
SVC0004	No valid addresses provided in the Message part <part_name>	part_name : name of the input parameter that resulted in the error.	400

4.4.10 Policy Exceptions

MessageId	Text	Variables	HTTP Status Code
POL0001	A policy error occurred. For example, rate limit error, authentication and authorization error.	N/A	401, 403

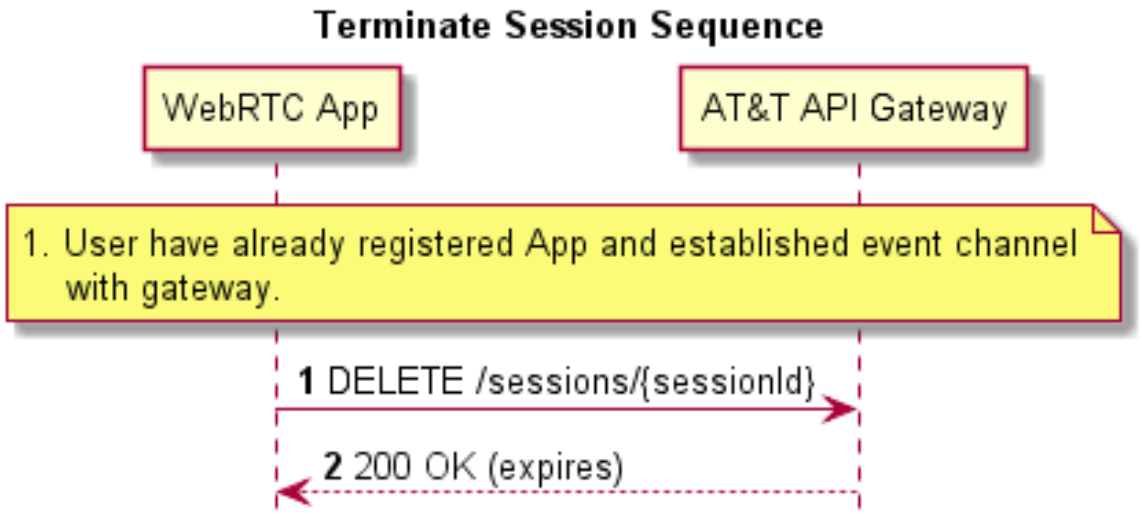
MessageId	Text	Variables	HTTP Status Code
POL0002	Privacy verification failed for address <address>, request is refused	address : <content_here>	403
POL0003	Too many addresses specified in Message part	N/A	403
POL1009	User has not been provisioned for %1	1% : System that has not been provisioned.	403

4.5 Operation: End Session

4.5.1 Functional Behavior

The End Session method is used to end and frees up all resources allocated to it, including the Event Channel and any pending, incoming Events within the session.

4.5.2 Call flow



4.5.3 Version Impact Summary

Service Version	Major or Minor Impact	Changes Introduced by this Version
1	Major	Initial Release

4.5.4 Authentication and Authorization

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description
authorization_code	Yes	RTC	1. Obtain the authorization code by passing App Key and App Secret along with scope. This redirects the user to AT&T authorization page to capture user consent. 2. Obtain OAuth access token using the OAuth authorization code along with the App Key and App Secret by making a Get Access Token request with grant_type=authorization_code. • Supports ICMN case
client_credentials	No	RTC	1. Obtain OAuth access token with the App Key and App Secret along with the scope by making a Get Access Token request with grant_type=client_credentials • Supports VTN case • Supports no-TN case

4.5.5 Representation Formats

4.5.6 Representation Formats

Direction	Supported Respresentation Formats
Request	XML, JSON,URLENCODED
Response	XML, JSON,URLENCODED

4.5.7 Input Parameters

Parameter	Data Type	Req?	Brief description	Location
accept	String	No	Specifies the format of the body of the response. The acceptable values for this parameter are: • application/json • application/xml • application/x-www-form-urlencoded The default value is application/json. Per rfc2616: ”If no Accept header field is present, then it is assumed that the client accepts all media types.” By default our services return application/json. The normal Accept header processing rules shall be followed according to rfc2616. *Note*: If there is no entity body in a normal successful response, this parameter is still needed to specify the format in the case of an error response message.	Header
authorization	String	Yes	Specifies the authorization type and token. The acceptable format for this parameter is the phrase ”Bearer OAuth Token” followed by a space ( ) and an OAuth access token. If this parameter value is missing from the header, then the API Gateway returns a message with an HTTP status code of 400 Invalid Request. If the OAuth access token is not valid, then the API Gateway returns an HTTP status code of 401 Unauthorized with a WWW-Authenticate HTTP header.	Header

Parameter	Data Type	Req?	Brief description	Location
content-length	Integer	No	Specifies the length of the content in octets. This parameter is only required for a non-streaming request.	Header
content-type	String	Yes	Specifies the type of content of the body. The acceptable values for this parameter are: • application/xml • application/json	Header
transfer-encoding	String	Conditional	Specifies the encodingof the message. The only acceptable values for this parameter is: • chunked This parameter is only required for a streaming request.	Header
sessionId	String	Yes	Specifies the session identifier for ending a session.	URI Path

4.5.7.1 Request – Example (End Session - JSON)

This example demonstrates how to end a session resource. The response format is JSON.

```
DELETE /RTC/v1/sessions/f81d4fae-7dec-11d0 HTTP/1.1
Host: api.att.com
Authorization: Bearer abcdef12345678
Content-Type: application/json
Accept: application/json
```

4.5.7.2 Request – Example (End Session - XML) This example demonstrates how to end a session resource. The response format is XML.

```
DELETE /RTC/v1/sessions/f81d4fae-7dec-11d0 HTTP/1.1
Host: api.att.com
Authorization: Bearer abcdef12345678
Content-Type: application/xml
Accept: application/xml
```

4.5.8 Output Parameters

Parameter	Data Type	Req?	Brief description	Location
content-type	String	Yes	Specifies the type of content of the body. The acceptable values for this parameter are: • application/xml • application/json	Header

4.5.8.1 Response – Example (End Session response)

This example shows the response to the End WebRTC Session request.

```
HTTP/1.1 204 No Content
Content-Type: application/json
```

4.5.9 Service Exceptions

MessageId	Text	Variables	HTTP Status Code
SVC0001	A service error has occurred. Error code is <error_explanation>	error_explanation : <content here>	400
SVC0002	Invalid input value for Message part <part_name>	part_name : name of the input parameter that resulted in the error.	400
SVC0003	Invalid input value for Message part <part_name>, valid values are <part_values>	part_name : name of the input parameter that resulted in the error. part_value : value of input parameter that was found to be in error.	400
SVC0004	No valid addresses provided in the Message part <part_name>	part_name : name of the input parameter that resulted in the error.	400

4.5.10 Policy Exceptions

MessageId	Text	Variables	HTTP Status Code
POL0001	A policy error occurred. For example, rate limit error, authentication and authorization error.	N/A	401, 403
POL0002	Privacy verification failed for address <address>, request is refused	address : <content_here>	403
POL0003	Too many addresses specified in Message part	N/A	403
POL1009	User has not been provisioned for %1	1% : System that has not been provisioned.	403

4.6 Operation: Get Events

4.6.1 Functional Behavior

The Get Session Events method retrieves the Events during a specific session using comet long polling mechanism (Please refer rfc6202 for more details). The events are related to call, conference, group chat, 1-to-1 chat and file transfer.

Upon successful creation of a session, the client must request a Get Session Events method at a regular interval to maintain the event channel.

The following is list of events under different scenarios.

4.6.1.1 Events during setup or tear-down of a call This event is received for the Get Session Events method request and when it is available from the receiving app. The type for the session is call and the state indicates the current state of the call in the network.

Event shall be generated under following circumstances.

- When called party receives call alert: Caller receives the event.

- When called party rejects or ends a call: Caller receives the event.
- When called party accepts or answers the call: Caller receives the event.
- When caller cancels the call: Called party receives the event.
- When caller ends the call: Called party receives the event.

4.6.1.2 Events during ongoing call modifications This event is generated when the client initiated a media modifications and should be expected under the following scenarios.

- Media upgrade or downgrade request, for example audio to audiovideo: The receiving app gets the event.
- Call Hold Request: The receiving app receives the event.
- Call Resume Request: The receiving app receives the event.

4.6.1.3 Events during call modification completion This event is an indication that the call modification is complete and the call can continue with initial sdp media values.

4.6.1.4 Events during call move complete This event is an indication that the call move is complete.

4.6.1.5 Events during call transfer initiation This event is an indication that the client initiated a call transfer to third-party and receives this event asynchronously on the event channel indicating that a request is sent to the server to bridge calls.

4.6.1.6 Events during call transfer completion This event is an indication that the client initiated a call transfer and receives this event asynchronously on the event channel indicating fulfillment of request in the network.

4.6.1.7 Events during conference invitation This event is received on the event channel as a result of long polling in the scenario when a Conferences invite is sent from another client.

4.6.1.8 Events during conference Modification This event is received when conference participant requests modification in the conference properties. This event is similar to call modification except that the resource URL and type would indicate them to belong to a conference.

4.6.1.9 Events during Conference Modification completed This event is received when the media conference modification is terminated.This event is similar to call modification except that the resource URL and type would indicate them to belong to a conference.

4.6.1.10 Events during 1-to-1 chat message This event is received over the outstanding channel requested by the recipient.

4.6.1.11 Events during 1-to-1 chat message failure This event is only sent when a message failed to be delivered.

4.6.1.12 Events during an is-composing 1-to-1 chat message This event is an indication that a user started typing a message or stopped typing a message. Is-composing is only sent to the recipient after previously a message has been sent or received to or from the same user on short intervals.

4.6.1.13 Events during 1-to-1 chat media message The event contains an URI from which the media can be retrieved.

4.6.1.14 Events during invitation to join a chat group This event is received through the event channel when a invitation is requested by another client.

4.6.1.15 Events during group Chat group status update This event is received every time a participant joins or leaves the conference to update connected users on which other parties are in the conference. Every client that accepted the conference and is in a connected state in the conference automatically receives the conference status updates.

4.6.1.16 Events during group Chat message This event contains the chat message sent by a chat member to the Group.

4.6.1.17 Events during an is-composing message from group Chat . This event is passed in the event channel when an is-composing message is received from a conference.

4.6.1.18 Events during media chats during group Chat This event is passed in the event channel when a media message is received from a conference.

4.6.1.19 Events during File Transfer Invitation or Status This event is sent through the event channel every time a file transfer session is initiated, changes of state or is ended. The recipient receives the File transfer invitation through this event. Both the sender and recipient receive updates when the File Transfer state changes. Both receive an event when the File Transfer is completed. At the end of the File transfer, the clients do not need to end the File transfer. The network does this for them.

4.6.2 Call flow

4.6.3 Version Impact Summary

Service Version	Major or Minor Impact	Changes Introduced by this Version
1	Major	Initial Release

4.6.4 Authentication and Authorization

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description
authorization_code	Yes	RTC	1. Obtain the authorization code by passing App Key and App Secret along with scope. This redirects the user to AT&T authorization page to capture user consent. 2. Obtain OAuth access token using the OAuth authorization code along with the App Key and App Secret by making a Get Access Token request with grant_type=authorization_code. • Supports ICMN case
client_credentials	No	RTC	1. Obtain OAuth access token with the App Key and App Secret along with the scope by making a Get Access Token request with grant_type=client_credentials • Supports VTN case • Supports no-TN case

4.6.5 Representation Formats

4.6.6 Representation Formats

Direction	Supported Respresentation Formats
Request	XML, JSON,URLENCODED
Response	XML, JSON,URLENCODED

4.6.7 Input Parameters

Parameter	Data Type	Req?	Brief description	Location
accept	String	No	Specifies the format of the body of the response. The acceptable values for this parameter are: • */* The default value is application/json. Note: For this method, this parameter specifies how the entity should be represented in case of an error. This parameter is for setting the format of an error message. If there is no error, then the representation will be the form of the actual content. Per rfc2616: ”If no Accept header field is present, then it is assumed that the client accepts all media types.” By default our services return application/json. The normal Accept header processing rules shall be followed according to rfc2616. *Note*: If there is no entity body in a normal successful response, this parameter is still needed to specify the format in the case of an error response message.	Header
authorization	String	Yes	Specifies the authorization type and token. The acceptable format for this parameter is the phrase ”Bearer OAuth Token” followed by a space ( ) and an OAuth access token. If this parameter value is missing from the header, then the API Gateway returns a message with an HTTP status code of 400 Invalid Request. If the OAuth access token is not valid, then the API Gateway returns an HTTP status code of 401 Unauthorized with a WWW-Authenticate HTTP header.	Header

4.6.7.1 Request – Example (Get Session Event(s) - JSON)

RetrieveEvents during the session. The response format is JSON.

```
GET /RTC/v1/sessions/0045-ab42-89a2/events HTTP/1.1
authorization: Bearer abcdef12345678
Content-Type: application/json
accept: application/json
```

4.6.7.2 Request – Example (Get Session Event(s) - XML)

RetrieveEvents during the session. The response format is XML.

```
GET /RTC/v1/sessions/0045-ab42-89a2/events HTTP/1.1
authorization: Bearer abcdef12345678
Content-Type: application/xml
accept: application/xml
```

4.6.8 Output Parameters

Parameter	Data Type	Required?	Brief description	Location
events	events Object	Yes	Contains the event list.	Body

Structure of events Object

Parameter	Data Type	Required?	Brief description
eventList	List of event Objects	Yes	Contains the list of event objects.

Structure of event Object

Parameter	Data Type	Required?	Brief description
type	String	No	Specifies the Resource Name for which an event is generated. The acceptable values for this parameter are: • calls • conferences • chats • groups • files Note : type,subtype,state combination should indicate the nature of event for a corresponding session.

subType	String	No	Specifies the representation for the type to categorize the event. The acceptable values for this parameter are: • failure : Applicable for 1-to-1 chats and is generated when the message delivery failed. • is-composing : Applicable for 1-to-1 and group chats conveying the user action of message composition. • media : The message is a media content and is applicable for 1-to-1 and group chats. • invitation : An invitation received to join group chat. • info : The group chat information. • message : The event containing the message for group chat. • session-message : The message is transmitted in SIP INVITE. • transfer : The file transfer request.
from	String	No	Specifies the SIP URI or TEL URI of the sender.
to	String	No	Specifies the SIP URI or TEL URI of the receiver.
resourceURL	String	No	Specifies the unique URI identifying the session.

state	String	No	Specifies the state depending on the service being used. For **calls and conference**,the acceptable values for this parameter are: • Invitation-received : A call or conference session invitation has been received and the client can accept, reject, or ignore the invitation. • Session-open : The call or conference session has been accepted by the clients,or SDP, or both is provided. • Session-terminated : The call or conference session has ended. The reason parameter indicates the reason the session has ended, when available. • Session-modified: The SDP for the session is updated. The updated SDP is included in the event. The client is required to apply the updated SDP. It cannot accept or reject the updated SDP. • Add-failed : Adding a user to the conference failed. • Remove-failed : Removing a user form the conference failed. • mod-received : Receiving media modifications for calls or conferences. • mod-terminated : Receiving media modifications completion events for calls or conferences. • transfer-initiated :The call transfer event is initiated. • transfer-terminated : The call transfer has completed.

			For **1-to-1 and group chats**, the acceptable values for this parameter are: • active: The user is currently composing a message. • idle : The user is currently idle or stopped typing. • full : The complete list of the chat Group. • partial : The list of users contains only what has changed since the last event.
			For **file transfer**, the acceptable values for this parameter are: • Invitation-received : A file transfer session invitation has been received and the client can accept, reject or ignore the invitation. • Session-open : The file transfer session has been accepted by the clients . • Session-terminated : The file transfer session has ended. The reason parameter indicates the reason the session has ended, when available. • Invitation-sent : The file transfer invitation has been sent to the receiving client and the local client should wait for the remote user to accept or reject. This is only provided to the invitee.
modId	String	No	Specifies the modification associated with media modification request.

reason	String	No	Specifies the additional information on reason of change based upon success or failure. When the value of state is set to session-terminated, then the acceptable values for this parameter are: • user busy : Invited user is Busy. • call rejected : Invited user rejected the call. • user not found : Invited user does not exist. • not available : Invited user is not online. • SDP parameter error : The SDP in the invitation is incorrect or not acceptable. • transfer-success: The call was disconnected and successfully transferred following a call transfer request. • transfer-in-progress : The call was disconnected following a Call Transfer Request that is currently in progress.

			For session Media modifications, when the value of state is set to session-terminated, then the acceptable values for this parameter are: • success : the modification was accepted by both local and remote client and media can be used. • not_acceptable : the modification was rejected by the remote client. • expired : the modification request did not receive a response in a timely manner. • cancelled : the modification request was cancelled by the local client. • rejected : the modification request was rejected by the remote client. • failed : the modification request failed due to other problem

			When the value of state is set to transfer-terminated, then the acceptable values for this parameter are: • Success : The call is successfully established between transferee and transfer-target. • Forbidden : The client is not allowed to use this operation. • Transfer Target Call Does Not Exist" : The requested transfer-target call could not be found in the network. • Rejected by Transfer Target : The transfer-target has refused the connection to the transferee. • Rejected by Transferee : The transferee has refused the connection to the transfer-target. • Call Terminated Early : The transfer-target or the transferee disconnected the call before the transfer is completed.
			When subtype is failure for type chats and the value of state is set to transfer-terminated, then the acceptable values for this parameter are: • Bad request • Forbidden • Timeout • Undesired message • Content-Type not supported • MSRP session no longer supported" • User not found • User not available • User busy • Rejected

			For file transfer; when an invitation fails, the acceptable values for this parameter are: • 480 : The receiver is offline. • 500 : The receiver is unknown. • 603 : The receiver rejected the file transfer. • Data transfer failed
transferToCallId	String	No	Specifies call identifier of the transfer target.
contentType	String	No	Specifies the MIME content type of the message. The possible values are: • text/plain - for text chat • image/jpeg
body	String	No	Specifies the content of the message.
msgId	String	No	Specifies the identifier for the sent message that is received in the response when the message was sent. This is applicable in case of 1-to-1 chat and group chat.
refresh	String	No	Specifies the timer, in seconds, that should be associated to the is-composing icon. If this parameter is set to 120, then it means the receiver displays an is-composing icon for 120 seconds. The is-composing icon should be removed when the timer expires, a chat message is received, or an is-composing message with the state set to idle is received. **Note**: This parameter is required only in case of sending an is-composing messge
referredBy	String	No	Specifies the SIP URI or TEL URI of the sender of the group chat invitation.
confId	String	No	Specifies the identifier of the group chat .
subject	String	No	Specifies the subject of the chat Group.
participants	List of Strings	No	Specifies the list of participants as SIP URI or TEL URI.
userCount	String	No	Specifies the number of users currently connected in the chat group conference.

users	users Object	No	Contains the participants in the chat group conference.
size	String	No	Specifies the size in bytes of the media message
fileSize	String	No	Specifies the size in bytes of the file.
fileName	String	No	Specifies the name of the file.
ftId	String	No	Specifies the unique identifier for the file transfer.
isIncoming	Boolean	No	Indicates that the file transfer is incoming. If this parameter is set to false, then the file transfer is outgoing.
contentDisposition	String	No	Specifies the invitation and status updates to recipient This parameter indicates how the message body is to be interpreted.
sdp	sdp Object	No	Contains the media associated with the session. The description is formatted according to RFC 4566 SDP Session Description Protocol.

Structure of users Object

Parameter	Data Type	Required?	Brief description
entity	String	No	Specifies the network identifier as SIP URI or TEL URI.
status	String	No	Specifies the state of the user.The value could be “connected” or “disconnected”.

Structure of sdp Object

Parameter	Data Type	Required?	Brief description

m	String	Yes	Specifies the media description in the following format. m=<media> <port> <proto> <fmt> The <media> field is the media type. The only acceptable value for this field is: • audio The <port> field is the transport port to which the media stream is sent. <proto> is the transport protocol. The only acceptable value for this field is: • TCP/RTMP The <fmt> field is a media format description. The fourth and any subsequent sub-fields describe the format of the media. The interpretation of the media format depends on the value of the <proto> sub-field.
c	String	No	Specifies the call description in the following format. c=<nettype> <addrtype> <connection-address> The <nettype> field is the network type, which is a text string giving the type of network. The only acceptable value for this field is: • IN : Internet. The <addrtype> field is the address type. This allows SDP to be used for sessions that are not IP based. The acceptable values for this field is: • IP4 : IPv4. • IP6 : IPv6. The <connection-address> field is the connection address. The subsequent sub-filds may be added after the connection address depending on the value of the <addrtype> field.

attributes	List of Strings	Yes	Contains the attributes for extending SDP. This parameter may be defined to be used as session-level attributes, media-level attributes, or both. The attribute fields may be in one of the following formats. A property attribute is simply of the form: • a=<flag> • These are binary attributes, and the presence of the attribute conveys that the attribute is a property of the session. Example: a=recvonly A value attribute is of the form: • a=<attribute>:<value>

4.6.8.1 Response – Example (Events during setup or tear-down of a call - JSON)

This example shows the response to the Get Session request. The response format is JSON.

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "events": {
    "eventList": {
      "eventObject": {
        "type": "calls",
        "from": "tel:+11234567890",
        "resourceURL": "/v1/session/0045-ab42-89a2/calls/15031864",
        "state": "invitation-received",
        "sdp": [
          {
            "m": "audio 9 RTP/AVP 8",
            "c": "IN IP4 128.128.128.128",
            "attributes": [
              { "a": "sendrecv" },
```

```
            { "a": "rtpmap:8 pcmu/8000/1" },
            { "a": "context-id:84657112" }
          ]
        },
        {
          "m": "video 10 RTP/AVP 98",
          "c": "IN IP4 128.128.128.128",
          "attributes": [
            { "a": "sendrecv" },
            { "a": "rtpmap:98 H264/90000" },
            { "a": "fmtp:98 profile-level-id=1a4db5;packetization-mode=1" },
            { "a": "context-id:84657112" }
          ]
        }
      ]
    }
  }
}
```

4.6.8.2 Response – Example (Events during setup or tear-down of a call - XML)

This example shows the response to the Get Session request. The response format is XML.

```
HTTP/1.1 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" ?>
    <events>
        <eventList>
            <eventObject>
                <type>calls</type>
                <from>tel:+11234567890</from>
                <resourceURL>/v1/session/0045-ab42-89a2/calls/15031864</
                    resourceURL>
                <state>invitation-received</state>
                <sdp>
                    <m>audio 9 RTP/AVP 8</m>
                    <c>IN IP4 128.128.128.128</c>
                    <attributes>
                        <a>sendrecv</a>
                    </attributes>
                    <attributes>
                        <a>rtpmap:8 pcmu/8000/1</a>
                    </attributes>
                    <attributes>
```

```
                    <a>context-id:84657112</a>
                </attributes>
            </sdp>
            <sdp>
                <m>video 10 RTP/AVP 98</m>
                <c>IN IP4 128.128.128.128</c>
                <attributes>
                    <a>sendrecv</a>
                </attributes>
                <attributes>
                    <a>rtpmap:98 H264/90000</a>
                </attributes>
                <attributes>
                    <a>fmtp:98 profile-level-id=1a4db5;packetization-mode
                        =1</a>
                </attributes>
                <attributes>
                    <a>context-id:84657112</a>
                </attributes>
            </sdp>
        </eventObject>
    </eventList>
</events>
```

4.6.8.3 Response – Example (Events during ongoing call modifications - JSON)

This example shows the response to the Get Session Event request. The response format is JSON.

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "events": {
    "eventList": {
      "eventObject": {
        "type": "calls",
        "from": "tel:+11234567890",
        "resourceURL": "/v1/sessions/0045-ab42-89a2/calls/15031864",
        "modId": "15034",
        "state": "mod-received",
        "sdp": [
          {
            "m": "audio9000RTP/AVP8",
            "attributes": [
```

```
            { "a": "sendonly" },
            { "a": "rtpmap: 8pcmu/8000/1" }
          ]
        },
        {
          "m": "video9001RTP/AVP98",
          "attributes": [
            { "a": "sendonly" },
            { "a": "rtpmap: 98H264/90000" },
            { "a": "fmtp: 98profile-levelid=A1438;packetization-mode=1" }
          ]
        }
      ]
    }
  }
}
```

4.6.8.4 Response – Example (Events during ongoing call modifications - XML)

This example shows the response to the Get Session request. The response format is XML.

```
HTTP/1.1 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" ?>
    <events>
        <eventList>
            <eventObject>
                <type>calls</type>
                <from>tel:+11234567890</from>
                <resourceURL>/v1/sessions/0045-ab42-89a2/calls/15031864</
                    resourceURL>
                <modId>15034</modId>
                <state>mod-received</state>
                <sdp>
                    <m>audio9000RTP/AVP8</m>
                    <attributes>
                        <a>sendonly</a>
                    </attributes>
                    <attributes>
                        <a>rtpmap: 8pcmu/8000/1</a>
                    </attributes>
                </sdp>
                <sdp>
                    <m>video9001RTP/AVP98</m>
```

```
                    <attributes>
                        <a>sendonly</a>
                    </attributes>
                    <attributes>
                        <a>rtpmap: 98H264/90000</a>
                    </attributes>
                    <attributes>
                        <a>fmtp: 98 profile-levelid=A1438;packetization-mode
                            =1</a>
                    </attributes>
                </sdp>
            </eventObject>
        </eventList>
    </events>
```

4.6.8.5 Response – Example (Events during call modification completion - JSON)

This example shows the response to the Get Session Event request. The response format is JSON.

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "events": {
    "eventList": {
      "eventObject": {
        "type": "calls",
        "from": "tel:+11234567890",
        "resourceURL": "/V1/sessions/0045-ab42-89a2/audiovideo/15031864",
        "modId": "15034",
        "state": "mod-terminated",
        "reason": "success",
        "sdp": [
          {
            "m": "audio 9000 RTP/AVP 8",
            "attributes": [
              { "a": "sendonly" },
              { "a": "rtpmap:8 pcmu/8000/1" }
            ]
          },
          {
            "m": "video 9001 RTP/AVP 98",
            "attributes": [
              { "a": "sendonly" },
```

```
                { "a": "rtpmap:98 H264/90000" },
                { "a": "fmtp:98 profile-levelid=A1438;packetization-mode=1" }
            ]
          }
        ]
      }
    }
  }
}
```

4.6.8.6 Response – Example (Events during call modification completion- XML)

This example shows the response to the Get Session Event request. The response format is XML.

```
HTTP/1.1 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" ?>
    <events>
        <eventList>
            <eventObject>
                <type>calls</type>
                <from>tel:+11234567890</from>
                <resourceURL>/V1/sessions/0045-ab42-89a2/audiovideo/15031864</
                    resourceURL>
                <modId>15034</modId>
                <state>mod-terminated</state>
                <reason>success</reason>
                <sdp>
                    <m>audio 9000 RTP/AVP 8</m>
                    <attributes>
                        <a>sendonly</a>
                    </attributes>
                    <attributes>
                        <a>rtpmap:8 pcmu/8000/1</a>
                    </attributes>
                </sdp>
                <sdp>
                    <m>video 9001 RTP/AVP 98</m>
                    <attributes>
                        <a>sendonly</a>
                    </attributes>
                    <attributes>
                        <a>rtpmap:98 H264/90000</a>
                    </attributes>
                    <attributes>
```

```
                        <a>fmtp:98 profile-levelid=A1438;packetization-mode
                            =1</a>
                    </attributes>
                </sdp>
            </eventObject>
        </eventList>
    </events>
```

4.6.8.7 Response – Example (Events during call move complete- JSON) This example shows the response to the Get Session Event request. The response format is JSON.

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "events": {
    "eventList": {
      "eventObject": {
        "type": "calls",
        "from": "tel:+11234567890",
        "resourceURL": "/v1/sessions/0045-ab42-89a2/calls/15031864",
        "transferToCallId": "15031865",
        "state": "move-terminated",
        "reason": "success"
      }
    }
  }
}
```

4.6.8.8 Response – Example (Events during call move complete - XML) This example shows the response to the Get Session Event request. The response format is XML.

```
HTTP/1.1 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" ?>
    <events>
        <eventList>
            <eventObject>
                <type>calls</type>
```

```
                <from>tel:+11234567890</from>
                <resourceURL>/v1/sessions/0045-ab42-89a2/calls/15031864</
                    resourceURL>
                <transferToCallId>15031865</transferToCallId>
                <state>move-terminated</state>
                <reason>success</reason>
            </eventObject>
        </eventList>
    </events>
```

4.6.8.9 Response – Example (Events during call transfer initiation - JSON) This example shows the response to the Get Session Event request. The response format is JSON.

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "events": {
    "eventList": {
      "eventObject": {
        "type": "calls",
        "from": "tel:+11234567890",
        "resourceURL": "/v1/sessions/0045-ab42-89a2/calls/15031864",
        "transferToCallId": "15031865",
        "state": "transfer-initiated"
      }
    }
  }
}
```

4.6.8.10 Response – Example (Events during call transfer initiation - XML) This example shows the response to the Get Session Event request. The response format is XML.

```
HTTP/1.1 200 OK
Content-Type: application/xml
    <?xml version="1.0" encoding="UTF-8" ?>
    <events>
        <eventList>
            <eventObject>
```

```
                <type>calls</type>
                <from>tel:+11234567890</from>
                <resourceURL>/v1/sessions/0045-ab42-89a2/calls/15031864</
                    resourceURL>
                <transferToCallId>15031865</transferToCallId>
                <state>transfer-initiated</state>
            </eventObject>
        </eventList>
    </events>
```

4.6.8.11 Response – Example (Events during call transfer completion - JSON) This example shows the response to the Get Session Event request. The response format is JSON.

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "events": {
    "eventList": {
      "eventObject": {
        "type": "calls",
        "from": "tel:+11234567890",
        "resourceURL": "/v1/sessions/0045-ab42-89a2/calls/15031864",
        "transferToCallId": "15031865",
        "state": "transfer-terminated",
        "reason": "success"
      }
    }
  }
}
```

4.6.8.12 Response – Example (Events during call transfer completion - XML) This example shows the response to the Get Session Event request. The response format is XML.

```
HTTP/1.1 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" ?>
    <events>
```

```
        <eventList>
            <eventObject>
                <type>calls</type>
                <from>tel:+11234567890</from>
                <resourceURL>/v1/sessions/0045-ab42-89a2/calls/15031864</
                    resourceURL>
                <transferToCallId>15031865</transferToCallId>
                <state>transfer-terminated</state>
                <reason>success</reason>
            </eventObject>
        </eventList>
    </events>
```

4.6.8.13 Response – Example (Events during conference invitation - JSON) This example shows the response to the Get Session Event request. The response format is JSON.

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "events": {
    "eventList": {
      "eventObject": {
        "type": "conferences",
        "from": "tel:+11234567890",
        "resourceURL": "/v1/sessions/0045-ab42-89a2/conferences/15031864",
        "state": "invitation-received",
        "sdp": {
          "m": "audio 9 TCP/RTMP 8",
          "c": "IN IP4 128.128.128.128",
          "attributes": [
            { "a": "sendrecv" },
            { "a": "rtpmap:8 pcmu/8000/1" },
            { "a": "setup:passive" },
            { "a": "context-id:3974619" }
          ]
        }
      }
    }
  }
}
```

4.6.8.14 Response – Example (Events during conference invitation - XML) This example shows the response to the Get Session Event request. The response format is XML.

```
HTTP/1.1 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" ?>
    <events>
        <eventList>
            <eventObject>
                <type>conferences</type>
                <from>tel:+11234567890</from>
                <resourceURL>/rest/v2/session/0045-ab42-89a2/conferences
                    /15031864</resourceURL>
                <state>invitation-received</state>
                <sdp>
                    <m>audio 9 TCP/RTMP 8</m>
                    <c>IN IP4 128.128.128.128</c>
                    <attributes>
                        <a>sendrecv</a>
                    </attributes>
                    <attributes>
                        <a>rtpmap:8 pcmu/8000/1</a>
                    </attributes>
                    <attributes>
                        <a>setup:passive</a>
                    </attributes>
                    <attributes>
                        <a>context-id:3974619</a>
                    </attributes>
                </sdp>
            </eventObject>
        </eventList>
    </events>
```

4.6.8.15 Response – Example (Events during 1-to-1 chat message - JSON) This example shows the response to the Get Session Event request. The response format is JSON.

```
HTTP/1.1 200 OK
Content-Type: application/json
{
```

```
  "events": {
    "eventList": {
      "eventObject": {
        "type": "chats",
        "subType":"session-message",
        "from": "tel:+11234567890",
        "contentType": "text/plain",
        "body": "This is my message"
      }
    }
  }
}
```

4.6.8.16 Response – Example (Events during 1-to-1 chat message - XML) This example shows the response to the Get Session Event request. The response format is XML.

```
HTTP/1.1 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" ?>
    <events>
        <eventList>
            <eventObject>
                <type>chats</type>
                <subType>session-message</subType>
                <from>tel:+11234567890</from>
                <contentType>text/plain</contentType>
                <body>This is my message</body>
            </eventObject>
        </eventList>
    </events>
```

4.6.8.17 Response – Example (Events during 1-to-1 chat message failure - JSON) This example shows the response to the Get Session Event request. The response format is JSON.

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "events": {
```

```
    "eventList": {
      "eventObject": {
        "type": "chats",
        "to": "tel:+11234567890",
        "msgId": "dd47fb6b-dfc8-4404-8211-2eb20222cf1e",
        "subType": "failure",
        "reason": "Forbidden"
      }
    }
  }
}
```

4.6.8.18 Response – Example (Events during 1-to-1 chat message failure - XML)

This example shows the response to the Get Session Event request. The response format is XML.

```
HTTP/1.1 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" ?>
    <events>
        <eventList>
            <eventObject>
                <type>chats</type>
                <to>tel:+11234567890</to>
                <msgId>dd47fb6b-dfc8-4404-8211-2eb20222cf1e</msgId>
                <subType>failure</subType>
                <reason>Forbidden</reason>
            </eventObject>
        </eventList>
    </events>
```

4.6.8.19 Response – Example (Events during 1-to-1 chat message 1-to-1 chat message - JSON)

This example shows the response to the Get Session Event request. The response format is JSON.

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "events": {
    "eventList": {
```

```
      "eventObject": {
        "type": "chats",
        "subType": "is-composing",
        "contentType": "text/plain",
        "refresh": "120",
        "state": "active",
        "from": "sip:jane@rcs.att.com"
      }
    }
  }
}
```

4.6.8.20 Response – Example (Events during 1-to-1 chat message 1-to-1 chat message - XML) This example shows the response to the Get Session Event request. The response format is XML.

```
HTTP/1.1 200 OK
Content-Type: application/xml
    <?xml version="1.0" encoding="UTF-8" ?>
    <events>
        <eventList>
            <eventObject>
                <type>chats</type>
                <subType>is-composing</subType>
                <contentType>text/plain</contentType>
                <refresh>120</refresh>
                <state>active</state>
                <from>sip:jane@rcs.att.com</from>
            </eventObject>
        </eventList>
    </events>
```

4.6.8.21 Response – Example (Events during 1-to-1 chat media message - JSON) This example shows the response to the Get Session Event request. The response format is JSON.

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "events": {
```

```
    "eventList": {
      "eventObject": {
        "type": "chats",
        "subType":"media",
        "from": "tel:+1234567890",
        "contentType": "image/jpeg",
        "resourceURL": "/v1/sessions/0045-ab42-89a2/chats/?fileRef=43235321",
        "size": "51199"
      }
    }
  }
}
```

4.6.8.22 Response – Example (Events during 1-to-1 chat media message - XML)

This example shows the response to the Get Session Event request. The response format is XML.

```
HTTP/1.1 200 OK
Content-Type: application/xml
    <?xml version="1.0" encoding="UTF-8" ?>
    <events>
        <eventList>
            <eventObject>
                <type>chats</type>
                <subType>media</subType>
                <from>tel:+11234567890</from>
                <contentType>image/jpeg</contentType>
                <resourceURL>/v1/sessions/0045-ab42-89a2/chats/?fileRef
                    =43235321</resourceURL>
                <size>51199</size>
            </eventObject>
        </eventList>
    </events>
```

4.6.8.23 Response – Example (Events during invitation to join a chat group - JSON)

This example shows the response to the Get Session Event request. The response format is JSON.

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "events": {
    "eventList": {
      "eventObject": {
        "type": "groups",
        "subType": "invitation",
        "referredBy": "tel:+11223456789",
        "confId": "357",
        "subject": "Testing Group Chat With multiple participants",
        "participants": "tel:+11234567890"
      }
    }
  }
}
```

4.6.8.24 Response – Example (Events during invitation to join a chat group - XML)

This example shows the response to the Get Session Event request. The response format is XMLt.

```
HTTP/1.1 200 OK
Content-Type: application/xml
    <?xml version="1.0" encoding="UTF-8" ?>
    <events>
        <eventList>
            <eventObject>
                <type>groups</type>
                <subType>invitation</subType>
                <referredBy>tel:+11234567890</referredBy>
                <confId>357</confId>
                <subject>Testing Group Chat With multiple participants</
                    subject>
                <participants>tel:+1234567890</participants>
            </eventObject>
        </eventList>
    </events>
```

4.6.8.25 Response – Example (Events during group Chat group status update - JSON)

This example shows the response to the Get Session Event request. The response format is JSON.

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "events": {
    "eventList": {
      "eventObject": {
        "type": "groups",
        "subType": "info",
        "userCount": "2",
        "confId": "357",
        "users": [
          {
            "entity": "tel:+11234567890",
            "status": "connected"
          },
          {
            "entity": "tel:+11223456789",
            "status": "connected"
          }
        ],
        "state": "full",
        "reason": "<a reason>"
      }
    }
  }
}
```

4.6.8.26 Response – Example (Events during group Chat group status update - XML) This example shows the response to the Get Session Event request. The response format is XML.

```
HTTP/1.1 200 OK
Content-Type: application/xml
    <?xml version="1.0" encoding="UTF-8" ?>
    <events>
        <eventList>
            <eventObject>
                <type>groups</type>
                <subType>info</subType>
                <userCount>2</userCount>
                <confId>357</confId>
                <users>
                    <entity>tel:+11234567890</entity>
                    <status>connected</status>
```

```
                </users>
                <users>
                    <entity>tel:+11223456789</entity>
                    <status>connected</status>
                </users>
                <state>full</state>
                <reason>&lt;a reason&gt;</reason>
            </eventObject>
        </eventList>
    </events>
```

4.6.8.27 Response – Example (Events during group Chat message - JSON) This example shows the response to the Get Session Event request. The response format is JSON.

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "events": {
    "eventList": {
      "eventObject": {
        "type": "groups",
        "subType": "message",
        "from": "tel:+11234567890",
        "confId": "357",
        "body": "Hello"
      }
    }
  }
}
```

4.6.8.28 Response – Example (Events during group Chat message - XML) This example shows the response to the Get Session Event request. The response format is XML.

```
HTTP/1.1 200 OK
Content-Type: application/xml
    <?xml version="1.0" encoding="UTF-8" ?>
    <events>
        <eventList>
```

```
            <eventObject>
                <type>groups</type>
                <subType>message</subType>
                <from>tel:+11234567890</from>
                <confId>357</confId>
                <body>Hello</body>
            </eventObject>
        </eventList>
    </events>
```

4.6.8.29 Response – Example (Events during an is-composing message from group Chat - JSON) This example shows the response to the Get Session Event request. The response format is JSON.

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "events": {
    "eventList": {
      "eventObject": {
        "type": "groups",
        "subType": "is-composing",
        "confId": "357",
        "contentType": "text/plain",
        "refresh": "120",
        "state": "active",
        "from": "tel:+11234567890"
      }
    }
  }
}
```

4.6.8.30 Response – Example (Events during an is-composing message from group Chat - XML) This example shows the response to the Get Session Event request. The response format is XML.

```
HTTP/1.1 200 OK
Content-Type: application/xml
    <?xml version="1.0" encoding="UTF-8" ?>
    <events>
```

```
        <eventList>
            <eventObject>
                <type>groups</type>
                <subType>is-composing</subType>
                <confId>357</confId>
                <contentType>text/plain</contentType>
                <refresh>120</refresh>
                <state>active</state>
                <from>tel:+11234567890</from>
            </eventObject>
        </eventList>
    </events>
```

4.6.8.31 Response – Example (Events during media chats during group Chat - JSON) This example shows the response to the Get Session Event request. The response format is JSON.

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "events": {
    "eventList": {
      "eventObject": {
        "type": "groups",
        "subType": "media",
        "from": "tel:+11234567890",
        "contentType": "image/jpeg",
        "resourceURL": "/v1/sessions/0045-ab42-89a2/groups/?fileRef=43235321",
        "size": "2345",
        "confId": "357"
      }
    }
  }
}
```

4.6.8.32 Response – Example (Events during media chats during group Chat - XML) This example shows the response to the Get Session Event request. The response format is XML.

```
HTTP/1.1 200 OK
Content-Type: application/xml

    <?xml version="1.0" encoding="UTF-8" ?>
    <events>
        <eventList>
            <eventObject>
                <type>groups</type>
                <subType>media</subType>
                <from>tel:+11234567890</from>
                <contentType>image/jpeg</contentType>
                <resourceURL>/v1/sessions/0045-ab42-89a2/groups/?fileRef
                    =43235321</resourceURL>
                <size>2345</size>
                <confId>357</confId>
            </eventObject>
        </eventList>
    </events>
```

4.6.8.33 Response – Example (Events during File Transfer Invitation or Status - JSON) This example shows the response to the Get Session Event request. The response format is JSON.

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "events": {
    "list": {
      "eventObject": {
        "type": "files",
        "subType": "transfer",
        "from": "tel:+11234567890",
        "fileName": "image0127.jpg",
        "fileSize": "103608",
        "contentType": "image/jpeg",
        "contentDisposition": "render",
        "ftId": "15031864",
        "state": "session-received",
        "isIncoming": "true"
      }
    }
  }
```

```
}
```

4.6.8.34 Response – Example (Events during File Transfer Invitation or Status - XML) This example shows the response to the Get Session Event request. The response format is XML.

```
HTTP/1.1 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" ?>
    <events>
        <list>
            <eventObject>
                <type>files</type>
                <subType>transfer</subType>
                <from>tel:+11234567890</from>
                <fileName>image0127.jpg</fileName>
                <fileSize>103608</fileSize>
                <contentType>image/jpeg</contentType>
                <contentDisposition>render</contentDisposition>
                <ftId>15031864</ftId>
                <state>session-received</state>
                <isIncoming>true</isIncoming>
            </eventObject>
        </list>
    </events>
```

4.6.9 HTTP Status Codes

Please refer to the appendix for HTTP errors.

4.6.10 Service Exceptions

MessageId	Text	Variables	HTTP Status Code
SVC0001	A service error has occurred. Error code is <error_explanation>	error_explanation : <content_here>	400

MessageId	Text	Variables	HTTP Status Code
SVC0002	Invalid input value for Message part <part_name>	part_name : name of the input parameter that resulted in the error.	400
SVC0003	Invalid input value for Message part <part_name>, valid values are <part_values>	part_name : name of the input parameter that resulted in the error. part_value : value of input parameter that was found to be in error.	400
SVC0004	No valid addresses provided in the Message part <part_name>	part_name : name of the input parameter that resulted in the error.	400

4.6.11 Policy Exceptions

MessageId	Text	Variables	HTTP Status Code
POL0001	A policy error occurred. For example, rate limit error, authentication and authorization error.	N/A	401, 403
POL0002	Privacy verification failed for address <address>, request is refused	address : <content_here>	403
POL0003	Too many addresses specified in Message part	N/A	403
POL1009	User has not been provisioned for %1	1% : System that has not been provisioned	403

4.7 Operation: Start Call

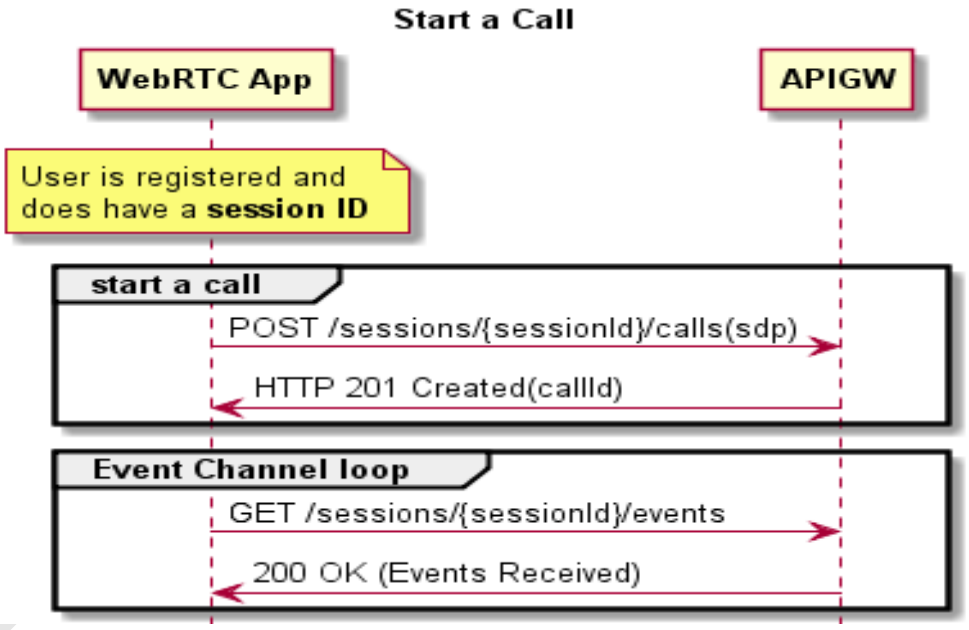
4.7.1 Functional Behavior

The Start Call method sets up or starts an audio-video session. A call identifier is allocated to the client using the SIP-IMS component as a result of successful request. The event channel should be present to communicate the changes of state at the receiving end.

Note: The client must be registered and have a session identifier allocated for using this method.

Note: The client must establish an event channel to get the invitation responses.

4.7.2 Call flow



4.7.3 Version Impact Summary

Service Version	Major or Minor Impact	Changes Introduced by this Version
1	Major	Initial Release

4.7.4 Authentication and Authorization

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description
authorization_code	Yes	RTC	1. Obtain the authorization code by passing App Key and App Secret along with scope. This redirects the user to AT&T authorization page to capture user consent. 2. Obtain OAuth access token using the OAuth authorization code along with the App Key and App Secret by making a Get Access Token request with grant_type=authorization_code. • Supports ICMN case
client_credentials	No	RTC	1. Obtain OAuth access token with the App Key and App Secret along with the scope by making a Get Access Token request with grant_type=client_credentials • Supports VTN case • Supports no-TN case

4.7.5 Representation Formats

4.7.6 Representation Formats

Direction	Supported Respresentation Formats
Request	XML, JSON,URLENCODED
Response	XML, JSON,URLENCODED

4.7.7 Input Parameters

Parameter	Data Type	Required?	Brief description	Location
accept	String	No	Specifies the format of the body of the response. The acceptable values for this parameter are: • application/json • application/xml • application/x-www-form-urlencoded The default value is application/json. Per rfc2616: ”If no Accept header field is present, then it is assumed that the client accepts all media types.” By default our services return application/json. The normal Accept header processing rules shall be followed according to rfc2616. *Note*: If there is no entity body in a normal successful response, this parameter is still needed to specify the format in the case of an error response message.	Header
authorization	String	Yes	Specifies the authorization type and token. The acceptable format for this parameter is the phrase ”Bearer OAuth Token” followed by a space ( ) and an OAuth access token. If this parameter value is missing from the header, then the API Gateway returns a message with an HTTP status code of 400 Invalid Request. If the OAuth access token is not valid, then the API Gateway returns an HTTP status code of 401 Unauthorized with a WWW-Authenticate HTTP header.	Header
calledParty	String	Yes	Specifies URI of the destination (TEL or SIP URI).	Body
call	call Object	Yes	Contains the call object.	Body

Parameter	Data Type	Required?	Brief description	Location
sdp	Sdp Object	Yes	Contains an SDP offer for the media conference session. The description is formatted according to RFC 4566 “SDP Session Description Protocol”.	Body

Structure of call Object

Parameter	Data Type	Required?	Brief description
sdp	Sdp Object	Yes	Contains an SDP offer for the media conference session.

Structure of Sdp Object

Parameter	Data Type	Required?	Brief description
m	String	Yes	Specifies the media description in the following format. m=<media> <port> <proto> <fmt> The <media> field is the media type. The <port> field is the transport port to which the media stream is sent. <proto> is the transport protocol. The <fmt> field is a media format description. The fourth and any subsequent sub-fields describe the format of the media. The interpretation of the media format depends on the value of the <proto> sub-field.

attributes	List of Strings	Yes	Contains the attributes for extending SDP. This parameter may be defined to be used as session-level attributes, media-level attributes, or both. The attribute fields may be in one of the following formats. A property attribute is simply of the form: • a=<flag> • These are binary attributes, and the presence of the attribute conveys that the attribute is a property of the session. Example: a=recvonly A value attribute is of the form: • a=<attribute>:<value>

4.7.7.1 Request – Example (Create a call) - JSON

JSON request to start a call.

```
POST /RTC/v1/sessions/0045-ab42-89a2/calls HTTP/1.1
Authorization: Bearer abcdef12345678
Content-Type: application/json;charset=UTF-8
Accept: application/json, text/html
Content-Length: 134
{"call":
{"calledNumber":"tel:+1234567890",
    "sdp": [
        {
            "m": "audio 9 TCP/RTMP 8",
            "attributes": [
                {
                    "a": "sendrecv"
                },
                {
                    "a": "rtpmap:8 pcmu/8000/1"
                },
                {
                    "a": "setup:active"
                }
            ]
        }
    ]
}
}
```

4.7.7.2 Request – Example (Create a call) - XML XML request to start a call.

```
POST /RTC/v1/sessions/0045-ab42-89a2/calls HTTP/1.1
Authorization: Bearer abcdef12345678
Content-Type: application/json;charset=UTF-8
Accept: application/json, text/html
Content-Length: 134
<?xml version="1.0" encoding="UTF-8" ?>
<call>
<calledNumber>tel:+1234567890</calledNumber>
    <sdp>
        <m>audio 9 TCP/RTMP 8</m>
        <attributes>
            <a>sendrecv</a>
        </attributes>
        <attributes>
            <a>rtpmap:8 pcmu/8000/1</a>
        </attributes>
        <attributes>
            <a>setup:active</a>
        </attributes>
    </sdp>
</call>
```

4.7.8 Output Parameters

Parameter	Data Type	Required?	Brief description	Location
x-state	String	Yes	Specifies the state of the call.	Header
location	String	Yes	Specifies the location of newly created call resource.	Header

4.7.8.1 Response – Example (Create a call) - XML

This shows the response to a call creation in XML format.

```
HTTP/1.1 201 Created
Location: http://128.128.128.128:80/RTC/v1/sessions/0045-ab42-89a2/calls/1234
```

```
x-state: invitation-sent
```

4.7.8.2 Response – Example (Create a call) - JSON

This shows the response to a call creation in JSON format

```
HTTP/1.1 201 Created
Location: http://128.128.128.128:80/RTC/v1/sessions/0045-ab42-89a2/calls/1234
x-state: invitation-sent
```

4.7.9 HTTP Status Codes

Code	Reason Phrase	Description
200	OK	Successful response.
201	Created	Successful response and resource was successfully created.
202	Accepted	Successful response but action has not yet been enacted.
204	No Content	Successful response and the response does not include an entity.
206	Partial Content	The server has fulfilled the partial GET request for the resource.
304	Not Modified	Returned when the client has performed a conditional GET request and access is allowed, but the resource has not been modified.
400	Bad Request	Many possible reasons not specified by the other codes.
401	Authentication Error	Authentication failed or was not provided.
403	Forbidden	Access permission error.
404	Not Found	The server has not found anything matching the Request-URI. No indication is given of whether the condition is temporary or permanent.
405	Method Not Allowed	A request was made of a resource using a request method not supported by that resource , for example using PUT on a REST resource that only supports POST.
406	Not Acceptable	The resource identified by the request is only capable of generating response entities which have content characteristics not acceptable according to the accept headers sent in the request.
408	Request Timeout	The client did not produce a request within the time that the server was prepared to wait. The client may repeat the request without modifications at any later time.

Code	Reason Phrase	Description
200	OK	Successful response.
409	Conflict	The request could not be completed due to a conflict with the current state of the resource. This code is only allowed in situations where it is expected that the user might be able to resolve the conflict and resubmit the request.
411	Length Required	The Content-Length header was not specified.
412	Precondition failed	A conditional header triggered the failure of the request.
413	Request Entity Too Large	The size of the request body exceed the maximum size permitted.
414	Request-URI Too Long	The server is refusing to service the request because the Request-URI is longer than the server is willing to interpret.
415	Unsupported Media Type	The request is in a format not supported by the requested resource for the requested method.
500	Internal Server Error	The server encountered an internal error or timed out; please retry.
502	Bad Gateway	The server, while acting as a gateway or proxy, received an invalid response from the upstream server it accessed in attempting to fulfill the request.
503	Service Unavailable	The server is currently unable to receive requests; please retry.
504	Gateway Timeout	The server, while acting as a gateway or proxy, did not receive a timely response from the upstream server specified by the URI (e.g. HTTP, FTP, LDAP) or some other auxiliary server (e.g. DNS) it needed to access in attempting to complete the request. **Note:** Note to implementors: some deployed proxies are known to return 400 or 500 when DNS lookups time out.

4.7.10 Service Exceptions

MessageId	Text	Variables	Parent HTTP Code
SVC0001	A service error has occurred. Error code is <Error Explanation>	Error Explanation : <content here>	400
SVC0002	Invalid input value for Message part <part name>	part name : name of the input parameter that resulted in the error	400

MessageId	Text	Variables	Parent HTTP Code
SVC0003	Invalid input value for Message part <part name>, valid values are <part values>	part value : value of input parameter that was found to be in error.	400
SVC0004	No valid addresses provided in the Message part <part name>	part name : name of the input parameter that resulted in the error	400

4.7.11 Policy Exceptions

MessageId	Text	Variables	Parent HTTP Code
POL0001	A policy error occurred. For example, rate limit error, authentication and authorization error.	N/A	401,403
POL0002	Privacy verification failed for address <address>, request is refused	address : <content here>	403
POL0003	Too many addresses specified in Message part	N/A	403
POL1009	User has not been provisioned for %1	System that hasn't been provisioned	403

4.8 Operation: Modify Call

4.8.1 Functional Behavior

The Modify Call method shall be used during an ongoing call dialogue and provides following the functionality for the client:

- Modifies the state of the call through session description protocol parameters.
- Enables the client to provide features, such as call move and call transfer.

Following scenarios shall be achieved using this method.

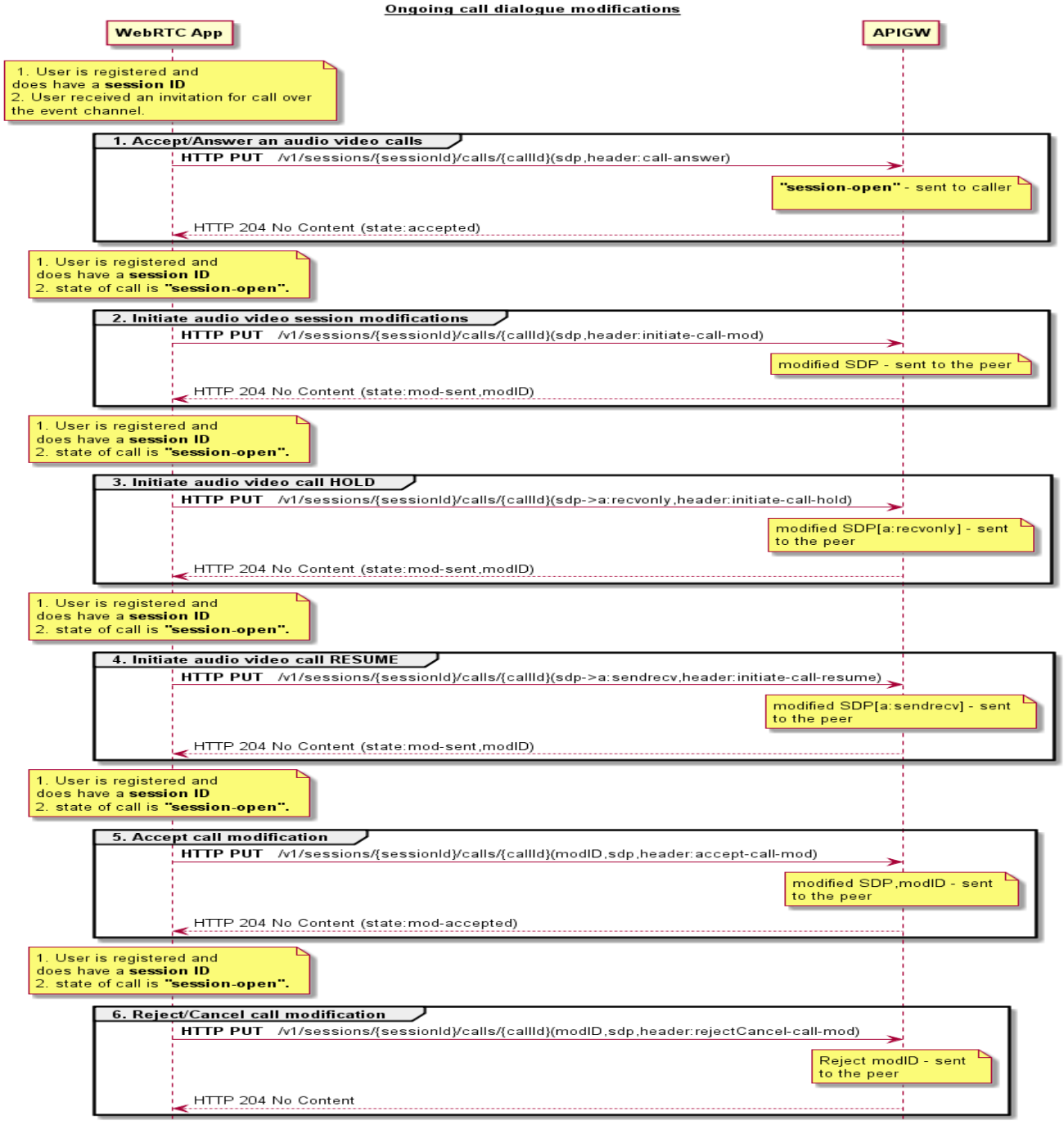
- Accept or Answer calls.
- Initiate : Calls Modifications.
- Initiate : Calls Hold.
- Initiate : Calls Resume.
- Accept Calls Modifications.
- Reject Calls modifications .
- Move call dialogue to different device or WebRTC or IMS registered endpoint .
- Call Transfer : The transferrer is a webRTC client and does have two call dialogue in open state which can be called as transferee and transfer-target. This method triggers a communication between transferee and transfer.

The client must set the following in order to use this method.

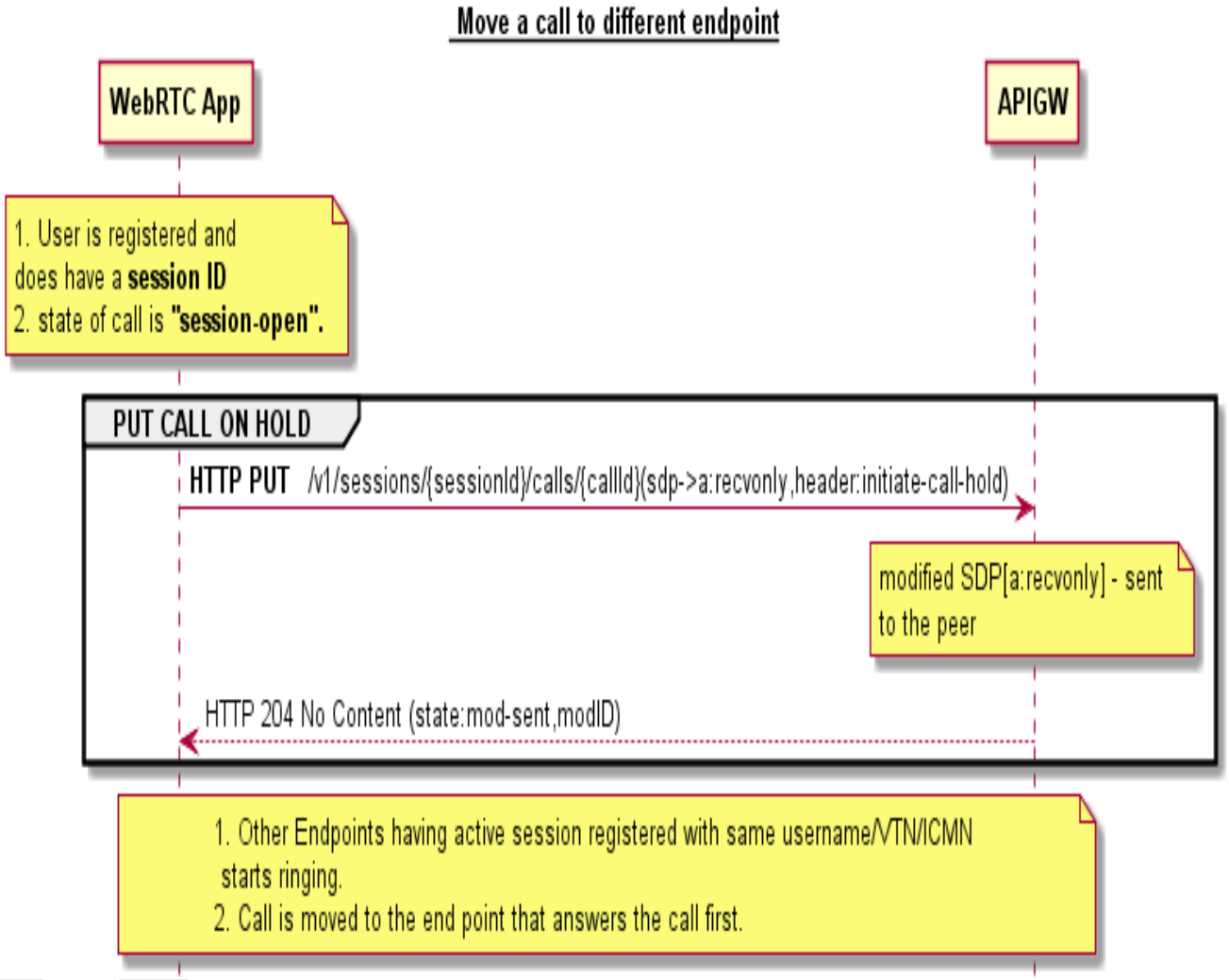
- The client must be registered and have a session ID allocated.
- The client must maintain an event channel to be able to receive the result of the Calls session modification invitation.
- The client has Calls session with the state set to session-open.

4.8.2 Call flow - Ongoing Call dialogue modifications

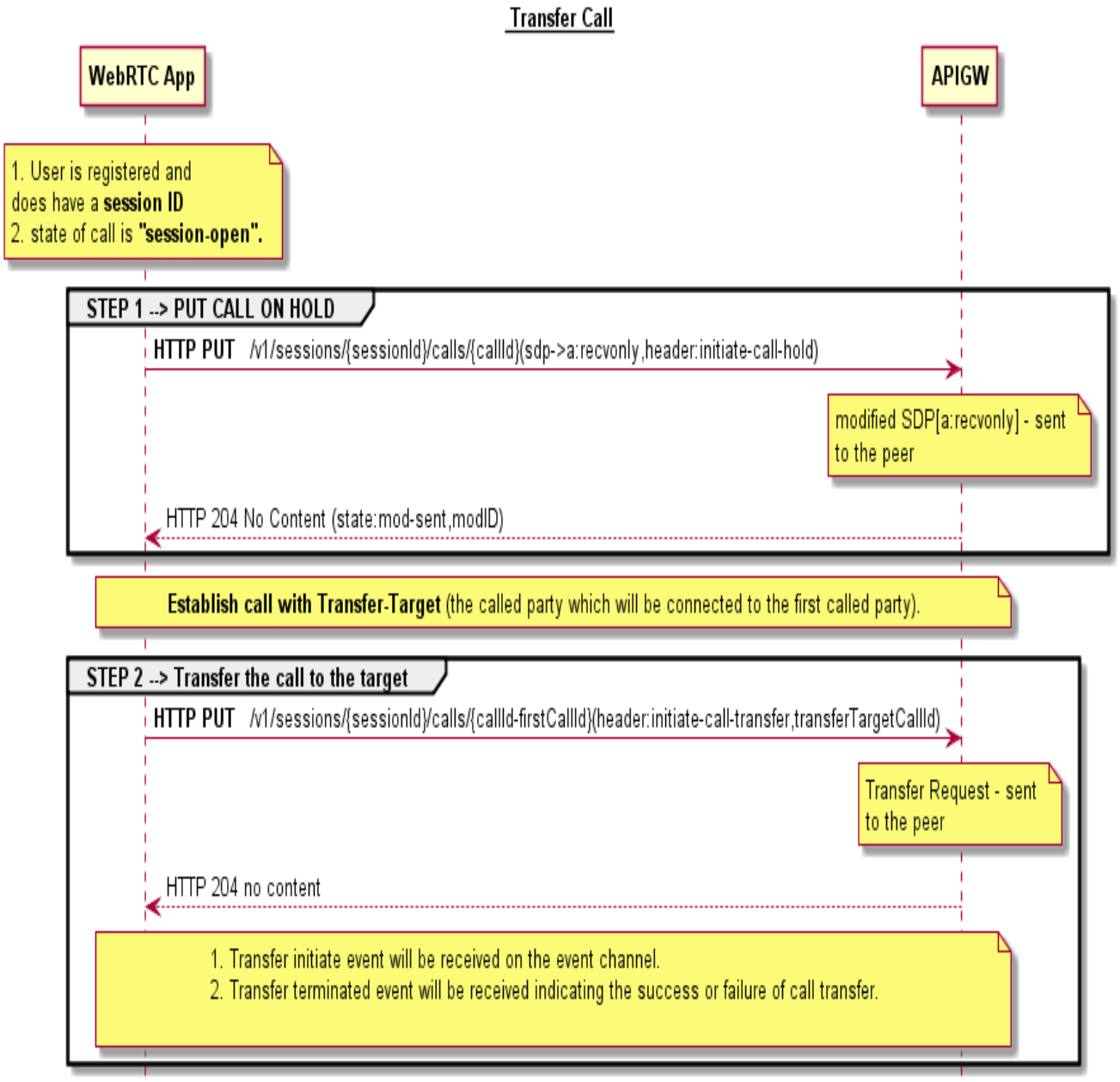
4.8.2.1 Ongoing call modification - Flow 1 Call Modifications



4.8.2.2 **Move a Call - Flow 2** Move Call



4.8.2.3 **Transfer a Call - Flow 3** Transfer Call



4.8.3 Version Impact Summary

Service Version	Major or Minor Impact	Changes Introduced by this Version
1	Major	Initial Release

4.8.4 Authentication and Authorization

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description
authorization_code	Yes	RTC	1. Obtain the authorization code by passing App Key and App Secret along with scope. This redirects the user to AT&T authorization page to capture user consent. 2. Obtain OAuth access token using the OAuth authorization code along with the App Key and App Secret by making a Get Access Token request with grant_type=authorization_code. • Supports ICMN case
client_credentials	No	RTC	1. Obtain OAuth access token with the App Key and App Secret along with the scope by making a Get Access Token request with grant_type=client_credentials • Supports VTN case • Supports no-TN case

4.8.5 Representation Formats

4.8.6 Representation Formats

Direction	Supported Respresentation Formats
Request	XML, JSON,URLENCODED
Response	XML, JSON,URLENCODED

4.8.7 Input Parameters

Parameter	Data Type	Required?	Brief description	Location
accept	String	No	Specifies the format of the body of the response. The acceptable values for this parameter are: • application/json • application/xml • application/x-www-form-urlencoded The default value is application/json. Per rfc2616: ”If no Accept header field is present, then it is assumed that the client accepts all media types.” By default our services return application/json. The normal Accept header processing rules shall be followed according to rfc2616. *Note*: If there is no entity body in a normal successful response, this parameter is still needed to specify the format in the case of an error response message.	Header
authorization	String	Yes	Specifies the authorization type and token. The acceptable format for this parameter is the phrase ”Bearer OAuth Token” followed by a space ( ) and an OAuth access token. If this parameter value is missing from the header, then the API Gateway returns a message with an HTTP status code of 400 Invalid Request. If the OAuth access token is not valid, then the API Gateway returns an HTTP status code of 401 Unauthorized with a WWW-Authenticate HTTP header.	Header

Parameter	Data Type	Required?	Brief description	Location
x-calls-action	String	Yes	Specifies the action for a conference session . The acceptable values for this parameter are: • call-answer : The client accepts the call invitation. • initiate-call-mod : The client request for change in media attributes, for example remove a video stream from an ongoing call session. • initiate-call-hold : The Client request to put the media path in a hold state. • initiate-call-transfer : The client put the call in a hold state, calls another number, and bridges the two parties. • initiate-call-resume : The Client request to resume the media path which was in a hold state. • initiate-call-move : The Client requests to move the call to different registered active endpoint. • accept-call-mod : The client accepts the changes in media attributes, for example remove a video stream from an ongoing conference session. • rejectCancel-call-mod : The proposal from the remote client to reject a media modification in an AudioVideo mod-received event or the Initiate WebRTC Session request from the local client to cancel a media modification.	Header
x-transferTargetCallId	String	Yes	Specifies the call identifier of the transfer target.	Header

Parameter	Data Type	Required?	Brief description	Location
x-modId	String	Conditional	Specifies the modification identifier for media changes. This parameter is required if the x-calls-action parameter is set to accept-call-mod or rejectCancel-call-mod.	Header
x-reject-reason	String	No	Specifies the reason for rejecting the session media changes. This parameter is only applicable if the x-calls-action parameter is set to rejectCancel-call-mod.	Header
callsMediaModifications	callsMediaModifications Object	Yes	Contains the call media information.	Body

Structure of callsMediaModifications Object

Parameter	Data Type	Required?	Brief description
sdp	Sdp Object	Yes	Contains an SDP offer for the audio video session.

Structure of sdp Object

Parameter	Data Type	Required?	Brief description
m	String	Yes	Specifies the media description in the following format. m=<media> <port> <proto> <fmt> The <media> field is the media type. The only acceptable values for this field is: • audio The <port> field is the transport port to which the media stream is sent. <proto> is the transport protocol. The only acceptable values for this field is: • TCP/RTMP The <fmt> field is a media format description. The fourth and any subsequent sub-fields describe the format of the media. The interpretation of the media format depends on the value of the <proto> sub-field.

attributes	List of Strings	Yes	Contains the attributes for extending SDP. This parameter may be defined to be used as session-level attributes, media-level attributes, or both. The attribute fields may be in one of the following formats. A property attribute is simply of the form: • a=<flag> • These are binary attributes, and the presence of the attribute conveys that the attribute is a property of the session. Example: a=recvonly A value attribute is of the form: • a=<attribute>:<value>

4.8.7.1 Request – Example (Accept/Answer Calls) - JSON

Request with Content-Type as JSON.

```
PUT /RTC/v1/sessions/0045-ab42-89a2/calls/1234 HTTP/1.1
Host: api.att.com
Authorization: Bearer abcdef12345678
Content-Type: application/json;charset=UTF-8
Accept: application/json
x-calls-action: call-answer
Content-Length: xxx

{
    "callsMediaModifications": {
        "sdp": [
            {
                "m": "audio 9 TCP/RTMP 8",
                "attributes": [
                    {
                        "a": "sendrecv"
                    },
                    {
                        "a": "rtpmap:8 pcmu/8000/1"
                    },
                    {
                        "a": "setup:active"
                    }
                ]
            }
        ]
    }
}
```

4.8.7.2 Request – Example (Accept/Answer Calls) - XML

Request with Content-Type as XML.

```
PUT /RTC/v1/sessions/0045-ab42-89a2/calls/1234 HTTP/1.1
Host: api.att.com
Authorization: Bearer abcdef12345678
Content-Type: application/json;charset=UTF-8
Accept: application/xml
x-calls-action:call-answer
Content-Length: yyy

<?xml version="1.0" encoding="UTF-8" ?>
    <callsMediaModifications>
        <sdp>
            <m>audio 9 TCP/RTMP 8</m>
            <attributes>
                <a>sendrecv</a>
            </attributes>
            <attributes>
                <a>rtpmap:8 pcmu/8000/1</a>
            </attributes>
            <attributes>
                <a>setup:active</a>
            </attributes>
        </sdp>
    </callsMediaModifications>
```

4.8.7.3 Request – Example (initiate call call session modifications) - JSON

Request with Content-Type as JSON.

```
PUT /RTC/v1/sessions/0045-ab42-89a2/calls/1234 HTTP/1.1
Host: api.att.com
Authorization: Bearer abcdef12345678
Content-Type: application/json;charset=UTF-8
Accept: application/json
x-calls-action: initiate-call-mod
Content-Length: xxx

{
  "callsMediaModifications": {
    "sdp": [
      {
        "m": "audio 9000 RTP/AVP 8",
        "attributes": [
          { "a": "rcvonly" },
          { "a": "rtpmap:8 pcmu/8000/1" }
        ]
      },
      {
        "m": "video 9001 RTP/AVP 98",
        "attributes": [
          { "a": "rcvonly" },
          { "a": "rtpmap:98 H264/90000" },
          { "a": "fmtp:98 profile-levelid=A1438;packetization-mode=1" }
        ]
      }
    ]
  }
}
```

4.8.7.4 Request – Example (initiate call call session modifications) - XML Request with Content-Type as XML.

```
PUT /RTC/v1/sessions/0045-ab42-89a2/calls/1234 HTTP/1.1
Host: api.att.com
Authorization: Bearer abcdef12345678
Content-Type: application/xml;charset=UTF-8
Accept: application/xml
x-calls-action: initiate-call-mod
Content-Length: xxx

<?xml version="1.0" encoding="UTF-8" ?>
    <callsMediaModifications>
        <sdp>
            <m>audio 9000 RTP/AVP 8</m>
            <attributes>
                <a>rcvonly</a>
            </attributes>
            <attributes>
                <a>rtpmap:8 pcmu/8000/1</a>
            </attributes>
        </sdp>
        <sdp>
            <m>video 9001 RTP/AVP 98</m>
            <attributes>
                <a>rcvonly</a>
            </attributes>
            <attributes>
                <a>rtpmap:98 H264/90000</a>
            </attributes>
            <attributes>
                <a>fmtp:98 profile-levelid= A1438;packetization-mode=1</a>
            </attributes>
        </sdp>
    </callsMediaModifications>
```

4.8.7.5 Request – Example (initiate call call hold) - JSON

Request with Content-Type as JSON.

```
PUT /RTC/v1/sessions/0045-ab42-89a2/calls/1234 HTTP/1.1
Host: api.att.com
Authorization: Bearer abcdef12345678
Content-Type: application/json;charset=UTF-8
Accept: application/json
x-calls-action: initiate-call-hold
Content-Length: xxx

{
    "callsMediaModifications": {
        "sdp": [
            {
                "m": "audio 9 TCP/RTMP 8",
                "attributes": [
                    {
                        "a": "recvonly"
                    },
                    {
                        "a": "rtpmap:8 pcmu/8000/1"
                    }
                ]
            }
        ]
    }
}
```

4.8.7.6 Request – Example (initiate call call hold) - XML

Request with Content-Type as XML.

```
PUT /RTC/v1/sessions/0045-ab42-89a2/calls/1234 HTTP/1.1
Content-Type: application/xml;charset=UTF-8
Host: api.att.com
Authorization: Bearer abcdef12345678
Accept: application/xml
x-calls-action: initiate-call-hold
Content-Length: xxx

<?xml version="1.0" encoding="UTF-8" ?>
    <callsMediaModifications>
        <sdp>
            <m>audio 9 TCP/RTMP 8</m>
            <attributes>
                <a>recvonly</a>
            </attributes>
            <attributes>
                <a>rtpmap:8 pcmu/8000/1</a>
            </attributes>
        </sdp>
    </callsMediaModifications>
```

4.8.7.7 Request – Example (initiate call call resume) - JSON

Request with Content-Type as JSON.

```
PUT /RTC/v1/sessions/0045-ab42-89a2/calls/1234 HTTP/1.1
Content-Type: application/json;charset=UTF-8
Host: api.att.com
Authorization: Bearer abcdef12345678
Accept: application/json
x-calls-action: initiate-call-resume
Content-Length: xxx

{
    "callsMediaModifications": {
        "sdp": [
            {
                "m": "audio 9 TCP/RTMP 8",
                "attributes": [
                    {
                        "a": "sendrecv"
                    },
                    {
                        "a": "rtpmap:8 pcmu/8000/1"
                    }
                ]
            }
        ]
    }
}
```

4.8.7.8 Request – Example (initiate call call resume) - XML

Request with Content-Type as XML.

```
PUT /RTC/v1/sessions/0045-ab42-89a2/calls/1234 HTTP/1.1
Content-Type: application/xml;charset=UTF-8
Host: api.att.com
Authorization: Bearer abcdef12345678
Accept: application/xml
x-calls-action: initiate-call-resume
Content-Length: xxx

<?xml version="1.0" encoding="UTF-8" ?>
    <callsMediaModifications>
        <sdp>
            <m>audio 9 TCP/RTMP 8</m>
            <attributes>
                <a>sendrecv</a>
            </attributes>
            <attributes>
                <a>rtpmap:8 pcmu/8000/1</a>
            </attributes>
        </sdp>
    </callsMediaModifications>
```

4.8.7.9 Request – Example (Accept call Media modifications) - JSON

Request with Content-Type as JSON.

```
PUT /RTC/v1/sessions/0045-ab42-89a2/calls/1234 HTTP/1.1
Content-Type: application/json;charset=UTF-8
Host: api.att.com
Authorization: Bearer abcdef12345678
Accept: application/json
x-calls-action: accept-call-mod
x-modId:15011
Content-Length: xxx

{
  "callsMediaModifications": {
    "sdp": [
      {
        "m": "audio 9000 RTP/AVP 8",
        "attributes": [
          { "a": "rcvonly" },
          { "a": "rtpmap:8 pcmu/8000/1" }
        ]
      },
      {
        "m": "video 9001 RTP/AVP 98",
        "attributes": [
          { "a": "rcvonly" },
          { "a": "rtpmap:98 H264/90000" },
          { "a": "fmtp:98 profile-levelid=A1438;packetization-mode=1" }
        ]
      }
    ]
  }
}
```

4.8.7.10 Request – Example (Accept call Media modifications) - XML

Request with Content-Type as XML.

```
PUT /RTC/v1/sessions/0045-ab42-89a2/calls/1234 HTTP/1.1
Host: api.att.com
Authorization: Bearer abcdef12345678
Content-Type: application/xml;charset=UTF-8
Accept: application/xml
x-calls-action: accept-call-mod
x-modId:15011
Content-Length: xxx

<?xml version="1.0" encoding="UTF-8" ?>
    <callsMediaModifications>
        <sdp>
            <m>audio 9000 RTP/AVP 8</m>
            <attributes>
                <a>rcvonly</a>
            </attributes>
            <attributes>
                <a>rtpmap:8 pcmu/8000/1</a>
            </attributes>
        </sdp>
        <sdp>
            <m>video 9001 RTP/AVP 98</m>
            <attributes>
                <a>rcvonly</a>
            </attributes>
            <attributes>
                <a>rtpmap:98 H264/90000</a>
            </attributes>
            <attributes>
                <a>fmtp:98 profile-levelid= A1438;packetization-mode=1</a>
            </attributes>
        </sdp>
    </callsMediaModifications>
```

4.8.7.11 Request – Example (Reject/Cancel call Media modifications) - JSON

Request with Content-Type as JSON.

```
PUT /RTC/v1/sessions/0045-ab42-89a2/calls/1234 HTTP/1.1
Host: api.att.com
Authorization: Bearer abcdef12345678
Content-Type: application/json;charset=UTF-8
Accept: application/json
x-calls-action: rejectCancel-call-mod
x-modId:15011
Content-Length: xxx

{
  "callsMediaModifications": {
    "sdp": [
      {
        "m": "audio 9000 RTP/AVP 8",
        "attributes": [
          { "a": "rcvonly" },
          { "a": "rtpmap:8 pcmu/8000/1" }
        ]
      },
      {
        "m": "video 9001 RTP/AVP 98",
        "attributes": [
          { "a": "rcvonly" },
          { "a": "rtpmap:98 H264/90000" },
          { "a": "fmtp:98 profile-levelid= A1438;packetization-mode=1" }
        ]
      }
    ]
  }
}
```

4.8.7.12 Request – Example (Reject/Cancel call Media modifications) - XML

Request with Content-Type as XML.

```
PUT /RTC/v1/sessions/0045-ab42-89a2/calls/1234 HTTP/1.1
Host: api.att.com
Authorization: Bearer abcdef12345678
Content-Type: application/xml;charset=UTF-8
Accept: application/xml
x-calls-action: rejectCancel-call-mod
x-modId:15011
Content-Length: xxx

<?xml version="1.0" encoding="UTF-8" ?>
    <callsMediaModifications>
        <sdp>
            <m>audio 9000 RTP/AVP 8</m>
            <attributes>
                <a>rcvonly</a>
            </attributes>
            <attributes>
                <a>rtpmap:8 pcmu/8000/1</a>
            </attributes>
        </sdp>
        <sdp>
            <m>video 9001 RTP/AVP 98</m>
            <attributes>
                <a>rcvonly</a>
            </attributes>
            <attributes>
                <a>rtpmap:98 H264/90000</a>
            </attributes>
            <attributes>
                <a>fmtp:98 profile-levelid=A1438;packetization-mode=1</a>
            </attributes>
        </sdp>
    </callsMediaModifications>
```

4.8.7.13 Request – Example (Transfer a call) - JSON

Request with Content-Type as JSON.

```
PUT /RTC/v1/sessions/0045-ab42-89a2/calls/1234 HTTP/1.1
Host: api.att.com
Authorization: Bearer abcdef12345678
Content-Type: application/json;charset=UTF-8
Accept: application/json
x-calls-action: initiate-call-transfer
x-transferTargetCallId:1235
Content-Length: xxx

{
  "callsMediaModifications": {
    "sdp": [
      {
        "m": "audio 9000 RTP/AVP 8",
        "attributes": [
          { "a": "rcvonly" },
          { "a": "rtpmap:8 pcmu/8000/1" }
        ]
      },
      {
        "m": "video 9001 RTP/AVP 98",
        "attributes": [
          { "a": "rcvonly" },
          { "a": "rtpmap:98 H264/90000" },
          { "a": "fmtp:98 profile-levelid= A1438;packetization-mode=1" }
        ]
      }
    ]
  }
}
```

4.8.7.14 Request – Example (Transfer a call) - XML

Request with Content-Type as XML.

```
PUT /RTC/v1/sessions/0045-ab42-89a2/calls/1234 HTTP/1.1
Host: api.att.com
Authorization: Bearer abcdef12345678
Content-Type: application/xml;charset=UTF-8
Accept: application/xml
x-calls-action: initiate-call-transfer
x-transferTargetCallId:1235
Content-Length: xxx

<?xml version="1.0" encoding="UTF-8" ?>
    <callsMediaModifications>
        <sdp>
            <m>audio 9000 RTP/AVP 8</m>
            <attributes>
                <a>rcvonly</a>
            </attributes>
            <attributes>
                <a>rtpmap:8 pcmu/8000/1</a>
            </attributes>
        </sdp>
        <sdp>
            <m>video 9001 RTP/AVP 98</m>
            <attributes>
                <a>rcvonly</a>
            </attributes>
            <attributes>
                <a>rtpmap:98 H264/90000</a>
            </attributes>
            <attributes>
                <a>fmtp:98 profile-levelid= A1438;packetization-mode=1</a>
            </attributes>
        </sdp>
    </callsMediaModifications>
```

4.8.7.15 Request – Example (Move a call) - JSON

Request with Content-Type as JSON.

```
PUT /RTC/v1/sessions/0045-ab42-89a2/calls/1234 HTTP/1.1
Host: api.att.com
Authorization: Bearer abcdef12345678
Content-Type: application/json;charset=UTF-8
Accept: application/json
x-calls-action: initiate-call-move
x-transferTargetCallId:1235
Content-Length: xxx

{
  "callsMediaModifications": {
    "sdp": [
      {
        "m": "audio 9000 RTP/AVP 8",
        "attributes": [
          { "a": "rcvonly" },
          { "a": "rtpmap:8 pcmu/8000/1" }
        ]
      },
      {
        "m": "video 9001 RTP/AVP 98",
        "attributes": [
          { "a": "rcvonly" },
          { "a": "rtpmap:98 H264/90000" },
          { "a": "fmtp:98 profile-levelid= A1438;packetization-mode=1" }
        ]
      }
    ]
  }
}
```

4.8.7.16 Request – Example (Move a cal) - XML Request with Content-Type as XML.

```
PUT /RTC/v1/sessions/0045-ab42-89a2/calls/1234 HTTP/1.1
Host: api.att.com
Authorization: Bearer abcdef12345678
Content-Type: application/xml;charset=UTF-8
Accept: application/xml
x-calls-action: initiate-call-move
x-transferTargetCallId:1235
Content-Length: xxx

<?xml version="1.0" encoding="UTF-8" ?>
    <callsMediaModifications>
        <sdp>
            <m>audio 9000 RTP/AVP 8</m>
            <attributes>
                <a>rcvonly</a>
            </attributes>
            <attributes>
                <a>rtpmap:8 pcmu/8000/1</a>
            </attributes>
        </sdp>
        <sdp>
            <m>video 9001 RTP/AVP 98</m>
            <attributes>
                <a>rcvonly</a>
            </attributes>
            <attributes>
                <a>rtpmap:98 H264/90000</a>
            </attributes>
            <attributes>
                <a>fmtp:98 profile-levelid= A1438;packetization-mode=1</a>
            </attributes>
        </sdp>
    </callsMediaModifications>
```

4.8.8 Output Parameters

Parameter	Data Type	Required?	Brief description	Location

Parameter	Data Type	Required?	Brief description	Location
x-modId	String	Conditional	Specifies the Identifier associated with the modification of media attributes. This parameter must be returned as a response to the initiation of media modifications. Note: This parameter is not applicable for a call session response .	Header
x-state	String	Conditional	Specifies the state of the modifications request.	Header

4.8.8.1 Response – Example (Accept/Answer Calls) - JSON

```
HTTP/1.1 204 No Content
x-state: accepted
```

4.8.8.2 Response – Example (Accept/Answer Calls) - (XML)

```
HTTP/1.1 204 No Content
x-state: accepted
```

4.8.8.3 Response – Example (initiate call call Media modifications) - JSON

```
HTTP/1.1 204 No Content
x-modID: abc-1234-def-567
x-state: mod-sent
```

4.8.8.4 Response – Example (initiate call call session modifications) - (XML)

```
HTTP/1.1 204 No Content
x-modID: abc-1234-def-567
x-state: mod-sent
```

4.8.8.5 Response – Example (initiate call call hold) - JSON

```
HTTP/1.1 204 No Content
x-modID: abc-1234-def-567
x-state: mod-sent
```

4.8.8.6 Response – Example (initiate call call hold) - (XML)

```
HTTP/1.1 204 No Content
x-modID: abc-1234-def-567
x-state: mod-sent
```

4.8.8.7 Response – Example (initiate call call resume) - JSON

```
HTTP/1.1 204 No Content
x-modID: abc-1234-def-567
x-state: mod-sent
```

4.8.8.8 Response – Example (initiate call call resume) - (XML)

```
HTTP/1.1 204 No Content
x-modID: abc-1234-def-567
x-state: mod-sent
```

4.8.8.9 Response – Example (Accept call Media modifications) - JSON

```
HTTP/1.1 204 No Content
x-modID: abc-1234-def-567
x-state: mod-accepted
```

4.8.8.10 Response – Example (Accept call Media modifications) - (XML)

```
HTTP/1.1 204 No Content
x-modID: abc-1234-def-567
x-state: mod-accepted
```

4.8.8.11 Response – Example (Reject/Cancel call Media modifications) - JSON

```
HTTP/1.1 204 No Content
x-state: mod-rejectcancel
```

4.8.8.12 Response – Example (Reject/Cancel call Media modifications) - (XML)

```
HTTP/1.1 204 No Content
x-state: mod-sent
```

4.8.8.13 Response – Example (Transfer a call) - JSON

```
HTTP/1.1 204 No Content
```

4.8.8.14 Response – Example (Transfer a call) - (XML)

```
HTTP/1.1 204 No Content
```

4.8.8.15 Response – Example (Move a call) - JSON

```
HTTP/1.1 204 No Content
```

4.8.8.16 Response – Example (Move a call) - (XML)

```
HTTP/1.1 204 No Content
```

4.8.9 HTTP Status Codes

Code	Reason Phrase	Description
200	OK	Successful response.
201	Created	Successful response and resource was successfully created.

Code	Reason Phrase	Description
200	OK	Successful response.
202	Accepted	Successful response but action has not yet been enacted.
204	No Content	Successful response and the response does not include an entity.
206	Partial Content	The server has fulfilled the partial GET request for the resource.
304	Not Modified	Returned when the client has performed a conditional GET request and access is allowed, but the resource has not been modified.
400	Bad Request	Many possible reasons not specified by the other codes.
401	Authentication Error	Authentication failed or was not provided.
403	Forbidden	Access permission error.
404	Not Found	The server has not found anything matching the Request-URI. No indication is given of whether the condition is temporary or permanent.
405	Method Not Allowed	A request was made of a resource using a request method not supported by that resource , for example using PUT on a REST resource that only supports POST.
406	Not Acceptable	The resource identified by the request is only capable of generating response entities which have content characteristics not acceptable according to the accept headers sent in the request.
408	Request Timeout	The client did not produce a request within the time that the server was prepared to wait. The client may repeat the request without modifications at any later time.
409	Conflict	The request could not be completed due to a conflict with the current state of the resource. This code is only allowed in situations where it is expected that the user might be able to resolve the conflict and resubmit the request.
411	Length Required	The Content-Length header was not specified.
412	Precondition failed	A conditional header triggered the failure of the request.
413	Request Entity Too Large	The size of the request body exceed the maximum size permitted.
414	Request-URI Too Long	The server is refusing to service the request because the Request-URI is longer than the server is willing to interpret.
415	Unsupported Media Type	The request is in a format not supported by the requested resource for the requested method.
500	Internal Server Error	The server encountered an internal error or timed out; please retry.
502	Bad Gateway	The server, while acting as a gateway or proxy, received an invalid response from the upstream server it accessed in attempting to fulfill the request.

Code	Reason Phrase	Description
200	OK	Successful response.
503	Service Unavailable	The server is currently unable to receive requests; please retry.
504	Gateway Timeout	The server, while acting as a gateway or proxy, did not receive a timely response from the upstream server specified by the URI (e.g. HTTP, FTP, LDAP) or some other auxiliary server (e.g. DNS) it needed to access in attempting to complete the request. **Note:** Note to implementors: some deployed proxies are known to return 400 or 500 when DNS lookups time out.

4.8.10 Service Exceptions

MessageId	Text	Variables	Parent HTTP Code
SVC0001	A service error has occurred. Error code is <Error Explanation>	Error Explanation : <content here>	400
SVC0002	Invalid input value for Message part <part name>	part name : name of the input parameter that resulted in the error	400
SVC0003	Invalid input value for Message part <part name>, valid values are <part values>	part value : value of input parameter that was found to be in error.	400
SVC0004	No valid addresses provided in the Message part <part name>	part name : name of the input parameter that resulted in the error	400
SVC8501	Call dialogue opened in Early Media	NA	409
SVC8502	Call transfer ongoing	NA	409
SVC8503	Call <callId > move in progress; proceed with move before doing a <action>	callId : Call of the end user. action: this can be media modification, call transfer or call move.	409
SVC8506	Modification ongoing	NA	409

4.8.11 Policy Exceptions

MessageId	Text	Variables	Parent HTTP Code
POL0001	A policy error occurred. For example, rate limit error, authentication and authorization error.	N/A	401,403
POL0002	Privacy verification failed for address <address>, request is refused	address : <content here>	403
POL0003	Too many addresses specified in Message part	N/A	403
POL1009	User has not been provisioned for %1	System that hasn't been provisioned	403

4.9 Operation: End Call

4.9.1 Functional Behavior

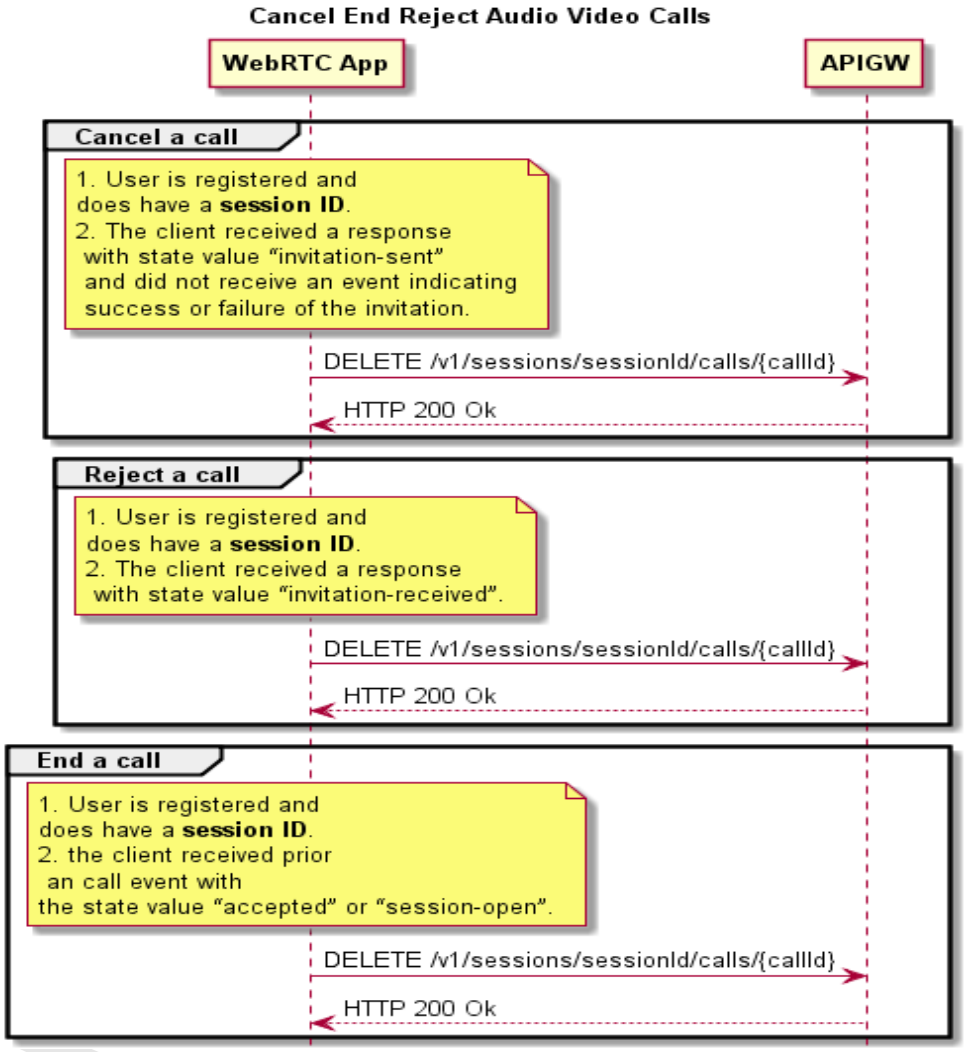
This operation is used to reject a call invitation or terminate an ongoing call . This operation shall be called under following circumstances:

1. Cancel a call - This request is sent from the originating side.
2. Reject a call - This operation is applicable for the called party.
3. Terminate a call - This scenario is applicable when the call does have an established media.

The Following conditions must be set to call this operation**:**

- Client needs to be registered and have a session ID allocated for using this operation.
- To cancel an invitation: The client received a response with state value "invitation-sent" and did not receive an event indicating success or failure of the invitation.
- To reject an invitation: the client received an AudioVideo event with state value "invitation-received".
- To terminate participation in a session: the client received prior an AudioVideo event with the state value "accepted" or "session-open".

4.9.2 Call flow



4.9.3 Version Impact Summary

Service Version	Major or Minor Impact	Changes Introduced by this Version
1	Major	Initial Release

4.9.4 Authentication and Authorization

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description
authorization_code	Yes	RTC	1. Obtain the authorization code by passing App Key and App Secret along with scope. This redirects the user to AT&T authorization page to capture user consent. 2. Obtain OAuth access token using the OAuth authorization code along with the App Key and App Secret by making a Get Access Token request with grant_type=authorization_code. • Supports ICMN case
client_credentials	No	RTC	1. Obtain OAuth access token with the App Key and App Secret along with the scope by making a Get Access Token request with grant_type=client_credentials • Supports VTN case • Supports no-TN case

4.9.5 Representation Formats

4.9.6 Representation Formats

Direction	Supported Respresentation Formats
Request	XML, JSON,URLENCODED
Response	XML, JSON,URLENCODED

4.9.7 Input Parameters

Parameter	Data Type	Required?	Brief description	Location
accept	String	No	Specifies the format of the body of the response. The acceptable values for this parameter are: • application/json • application/xml • application/x-www-form-urlencoded The default value is application/json. Per rfc2616: ”If no Accept header field is present, then it is assumed that the client accepts all media types.” By default our services return application/json. The normal Accept header processing rules shall be followed according to rfc2616. *Note*: If there is no entity body in a normal successful response, this parameter is still needed to specify the format in the case of an error response message.	Header
authorization	String	Yes	Specifies the authorization type and token. The acceptable format for this parameter is the phrase ”Bearer OAuth Token” followed by a space ( ) and an OAuth access token. If this parameter value is missing from the header, then the API Gateway returns a message with an HTTP status code of 400 Invalid Request. If the OAuth access token is not valid, then the API Gateway returns an HTTP status code of 401 Unauthorized with a WWW-Authenticate HTTP header.	Header
x-delete-reason	String	Yes	Specifies the reason for ending the call. The acceptable values for this parameter are: • cancel • reject • terminate	Header
callId	String	Yes	Specifies the call identifier.	URI

4.9.7.1 Request – Example (Cancel End or Reject a call) - JSON

JSON request for cancel, End or Reject a call.

```
DELETE /RTC/v1/sessions/0045-ab42-89a2/calls/15031864 HTTP/1.1
x-delete-reason=<reasonValue>
Content-Type: application/json;charset=UTF-8
Accept: application/json, text/html
Content-Length: 134
```

4.9.7.2 Request – Example (Cancel End or Reject a call) - XML

XML request for cancel, End or Reject a call.

```
DELETE /RTC/v1/sessions/0045-ab42-89a2/calls/15031864 HTTP/1.1
x-delete-reason=<reasonValue>
Content-Type: application/json;charset=UTF-8
Accept: application/json, text/html
Content-Length: 134
```

4.9.8 Output Parameters

Parameter	Data Type	Required?	Brief description	Location

4.9.8.1 Response – (Cancel/Terminate or Reject a call) Example (JSON)

```
HTTP/1.1 204 No Content
```

4.9.8.2 Response – (Cancel/Terminate or Reject a call) Example (XML)

```
HTTP/1.1 204 No Content
```

4.9.9 HTTP Response Codes

4.9.10 Service Exceptions

MessageId	Text	Variables	Parent HTTP Code
SVC0001	A service error has occurred. Error code is <Error Explanation>	Error Explanation : <content here>	400
SVC0002	Invalid input value for Message part <part name>	part name : name of the input parameter that resulted in the error	400
SVC0003	Invalid input value for Message part <part name>, valid values are <part values>	part value : value of input parameter that was found to be in error.	400
SVC0004	No valid addresses provided in the Message part <part name>	part name : name of the input parameter that resulted in the error	400

4.9.11 Policy Exceptions

MessageId	Text	Variables	Parent HTTP Code
POL0001	A policy error occurred. For example, rate limit error, authentication and authorization error.	N/A	401,403
POL0002	Privacy verification failed for address <address>, request is refused	address : <content here>	403
POL0003	Too many addresses specified in Message part	N/A	403
POL1009	User has not been provisioned for %1	System that hasn't been provisioned	403

4.10 Operation:Create Conference

4.10.1 Functional Behavior

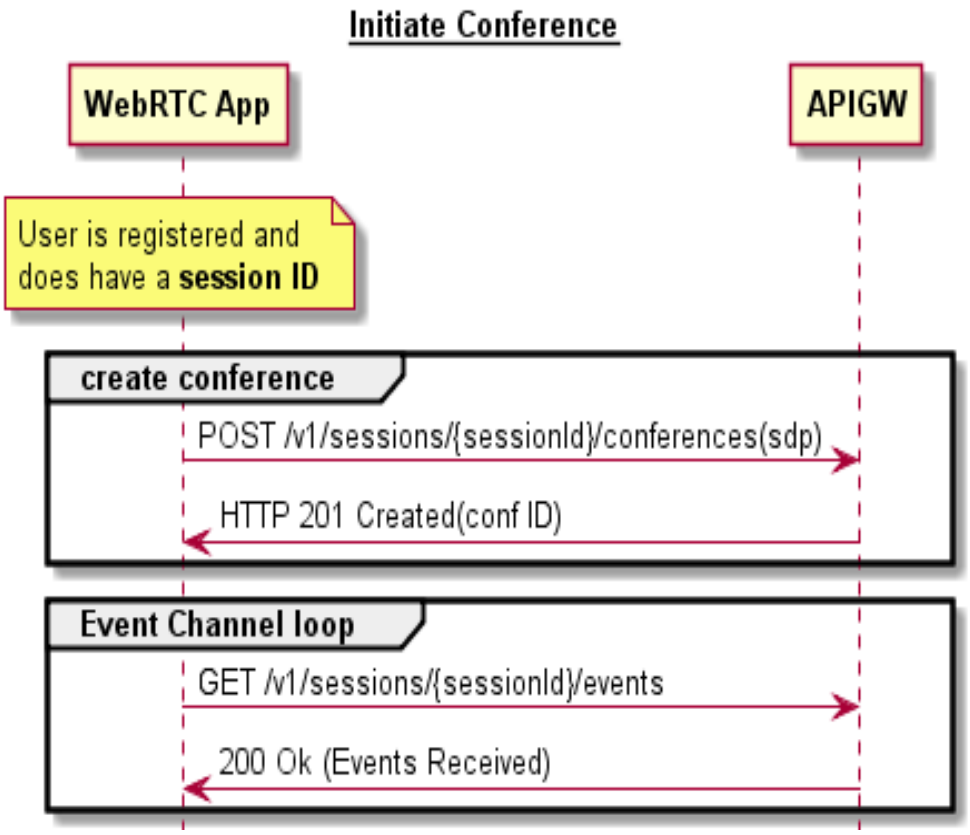
This operation is used to setup or start a conference session. A conference ID is allocated to the client by the conference server as a result of successful execution of this operation.

The participants can be added to the conference resource allocated to the client using this identifier .

Following condition should be met to start a conference.

- Client needs to be registered and have a session ID allocated for using this operation.
- Client needs to establish an event channel to get the invitation responses.

4.10.2 Call flow



4.10.3 Version Impact Summary

Service Version	Major or Minor Impact	Changes Introduced by this Version
1	Major	Initial Release

4.10.4 Authentication and Authorization

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description
authorization_code	Yes	RTC	1. Obtain the authorization code by passing App Key and App Secret along with scope. This redirects the user to AT&T authorization page to capture user consent. 2. Obtain OAuth access token using the OAuth authorization code along with the App Key and App Secret by making a Get Access Token request with grant_type=authorization_code. • Supports ICMN case
client_credentials	No	RTC	1. Obtain OAuth access token with the App Key and App Secret along with the scope by making a Get Access Token request with grant_type=client_credentials • Supports VTN case • Supports no-TN case

4.10.5 Representation Formats

4.10.6 Representation Formats

Direction	Supported Respresentation Formats
Request	XML, JSON,URLENCODED
Response	XML, JSON,URLENCODED

4.10.7 Input Parameters

Parameter	Data Type	Required?	Brief description	Location
accept	String	No	Specifies the format of the body of the response. The acceptable values for this parameter are: • application/json • application/xml • application/x-www-form-urlencoded The default value is application/json. Per rfc2616: ”If no Accept header field is present, then it is assumed that the client accepts all media types.” By default our services return application/json. The normal Accept header processing rules shall be followed according to rfc2616. *Note*: If there is no entity body in a normal successful response, this parameter is still needed to specify the format in the case of an error response message.	Header
authorization	String	Yes	Specifies the authorization type and token. The acceptable format for this parameter is the phrase ”Bearer OAuth Token” followed by a space ( ) and an OAuth access token. If this parameter value is missing from the header, then the API Gateway returns a message with an HTTP status code of 400 Invalid Request. If the OAuth access token is not valid, then the API Gateway returns an HTTP status code of 401 Unauthorized with a WWW-Authenticate HTTP header.	Header
mediaConference	mediaConference Object	Yes	Contains the media conference session.	Body

Structure of mediaConference Object

Parameter	Data Type	Required?	Brief description
sdp	Sdp Object	Yes	Contains an SDP offer for the media conference session.

Structure of Sdp Object

Parameter	Data Type	Required?	Brief description
m	String	Yes	Specifies the media description in the following format. m=<media> <port> <proto> <fmt> The <media> field is the media type. The only acceptable value for this field is: • audio The <port> field is the transport port to which the media stream is sent. <proto> is the transport protocol. The only acceptable value for this field is: • TCP/RTMP The <fmt> field is a media format description. The fourth and any subsequent sub-fields describe the format of the media. The interpretation of the media format depends on the value of the <proto> sub-field.
attributes	List of Strings	Yes	Contains the attributes for extending SDP. This parameter may be defined to be used as session-level attributes, media-level attributes, or both. The attribute fields may be in one of the following formats. A property attribute is simply of the form: • a=<flag> • These are binary attributes, and the presence of the attribute conveys that the attribute is a property of the session. Example: a=recvonly A value attribute is of the form: • a=<attribute>:<value>

4.10.7.1 Request – Example (Create a Conference) - JSON

JSON Request to start a conference.

```
POST /RTC/v1/sessions/0045-ab42-89a2/conferences HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json, text/html
Content-Length: 134

{
 "mediaConference": {
    "sdp": [
        {
            "m": "audio 9 TCP/RTMP 8",
            "attributes": [
                {
                    "a": "sendrecv"
                },
                {
                    "a": "rtpmap:8 pcmu/8000/1"
                },
                {
                    "a": "setup:active"
                }
            ]
        }
    ]
 }
}
```

4.10.7.2 Request – Example (Create a Conference) - XML

XML Request to start a conference.

```
POST /RTC/v1/sessions/0045-ab42-89a2/conferences HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json, text/html
Content-Length: 134
<?xml version="1.0" encoding="UTF-8" ?>
<mediaConference>
    <sdp>
        <m>audio 9 TCP/RTMP 8</m>
        <attributes>
            <a>sendrecv</a>
        </attributes>
        <attributes>
            <a>rtpmap:8 pcmu/8000/1</a>
```

```
        </attributes>
        <attributes>
            <a>setup:active</a>
        </attributes>
    </sdp>
</mediaConference>
```

4.10.8 Output Parameters

Parameter	Data Type	Required?	Brief description	Location
x-state	String	Yes	Specifies the state of the conference resource.	Header
location	String	Yes	Specifies the location of the newly created conference object.	Header

4.10.8.1 Response – Example (XML)

This shows the response to a conference creation in XML format.

```
HTTP/1.1 201 Created
Location: http://128.128.128.128:80/RTC/v1/sessions/0045-ab42-89a2/conferences
    /15031864
x-state: invitation-sent
```

4.10.8.2 Response – Example (JSON)

This shows the response to a conference creation in JSON format

```
HTTP/1.1 201 Created
Location: http://128.128.128.128:80/RTC/v1/sessions/0045-ab42-89a2/conferences
    /15031864
x-state: invitation-sent
```

4.10.9 HTTP Response Codes

Please Refer to Appendix for HTTP Error Codes.

4.10.10 Service Exceptions

MessageId	Text	Variables	Parent HTTP Code
SVC0001	A service error has occurred. Error code is <Error Explanation>	Error Explanation : <content here>	400
SVC0002	Invalid input value for Message part <part name>	part name : name of the input parameter that resulted in the error	400
SVC0003	Invalid input value for Message part <part name>, valid values are <part values>	part value : value of input parameter that was found to be in error.	400
SVC0004	No valid addresses provided in the Message part <part name>	part name : name of the input parameter that resulted in the error	400

4.10.11 Policy Exceptions

MessageId	Text	Variables	Parent HTTP Code
POL0001	A policy error occurred. For example, rate limit error, authentication and authorization error.	N/A	401,403
POL0002	Privacy verification failed for address <address>, request is refused	address : <content here>	403
POL0003	Too many addresses specified in Message part	N/A	403
POL1009	User has not been provisioned for %1	System that hasn't been provisioned	403

4.11 **Operation:** Add Participant To Conference

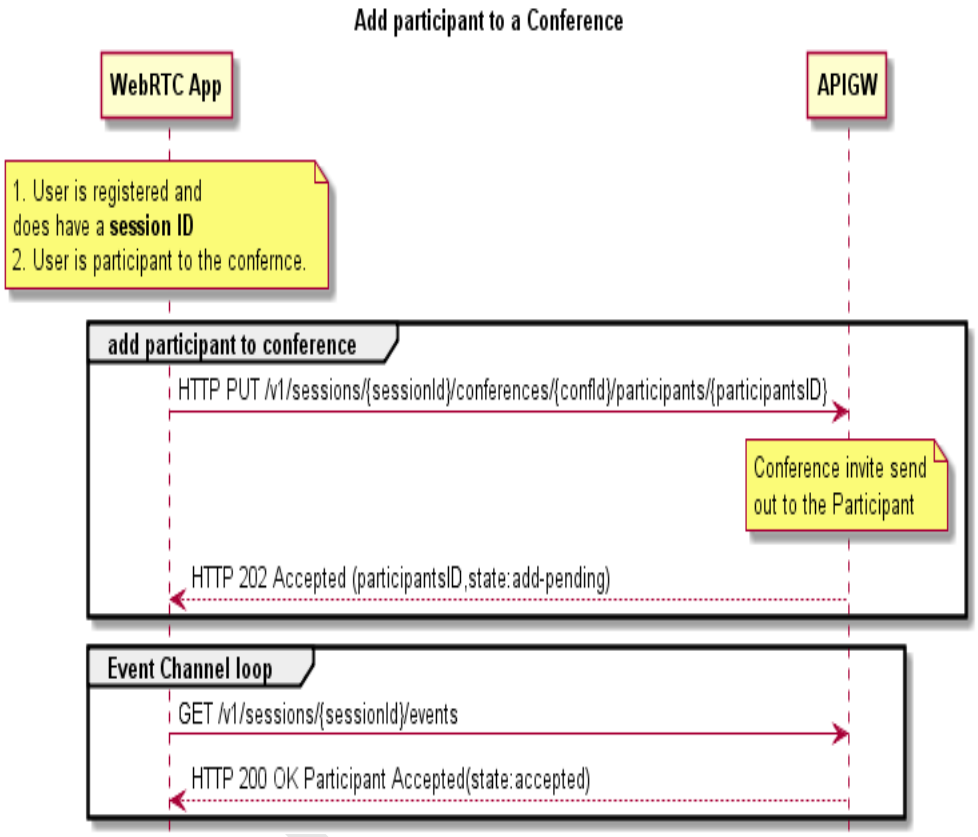
4.11.1 Functional Behavior

The Add Participants To Conference method adds or removes a conference participant during an on-going conference . The member identifier is sent to the server and ian invite is sent to the specified participant.

The following condition must be met to add participants.

- Client must to be registered and have a session identifier allocated for using this method.
- Client must to be participant.
- Client must to establish an event channel to receive the result of add participants.

4.11.2 Call flow



4.11.3 Version Impact Summary

Service Version	Major or Minor Impact	Changes Introduced by this Version
1	Major	Initial Release

4.11.4 Authentication and Authorization

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description
authorization_code	Yes	RTC	1. Obtain the authorization code by passing App Key and App Secret along with scope. This redirects the user to AT&T authorization page to capture user consent. 2. Obtain OAuth access token using the OAuth authorization code along with the App Key and App Secret by making a Get Access Token request with grant_type=authorization_code. • Supports ICMN case
client_credentials	No	RTC	1. Obtain OAuth access token with the App Key and App Secret along with the scope by making a Get Access Token request with grant_type=client_credentials • Supports VTN case • Supports no-TN case

4.11.5 Representation Formats

4.11.6 Representation Formats

Direction	Supported Respresentation Formats
Request	XML, JSON,URLENCODED
Response	XML, JSON,URLENCODED

4.11.7 Input Parameters

Parameter	Data Type	Required?	Brief description	Location
accept	String	No	Specifies the format of the body of the response. The acceptable values for this parameter are: • application/json • application/xml • application/x-www-form-urlencoded The default value is application/json. Per rfc2616: ”If no Accept header field is present, then it is assumed that the client accepts all media types.” By default our services return application/json. The normal Accept header processing rules shall be followed according to rfc2616. *Note*: If there is no entity body in a normal successful response, this parameter is still needed to specify the format in the case of an error response message.	Header
authorization	String	Yes	Specifies the authorization type and token. The acceptable format for this parameter is the phrase ”Bearer OAuth Token” followed by a space ( ) and an OAuth access token. If this parameter value is missing from the header, then the API Gateway returns a message with an HTTP status code of 400 Invalid Request. If the OAuth access token is not valid, then the API Gateway returns an HTTP status code of 401 Unauthorized with a WWW-Authenticate HTTP header.	Header
conferenceId	String	Yes	Specified the conference identifier for which the this method request is sent.	URI Path
participantsId	String	Yes	Specifies the SIP URI or TEL URI of the participant being added.	URI Path

4.11.7.1 Request – Example (Add Participant to a Conference) - JSON

JSON request to add a participant.

```
PUT /RTC/v1/sessions/0045-ab42-89a2/conferences/15031864/participants/tel
    :+1234567890 HTTP/1.1
Content-Type: application/json
Accept: application/json
```

4.11.7.2 Request – Example (Add Participant to a Conference) - XML

XML request to add a participant.

```
PUT /RTC/v1/sessions/0045-ab42-89a2/conferences/15031864/participants/tel
    :+1234567890 HTTP/1.1
Content-Type: application/xml
Accept: application/json
```

4.11.8 Output Parameters

Parameter	Data Type	Required?	Brief description	Location
x-state	string	yes	Specifies the current status of the add participant. The acceptable values for this parameter are: • add-pending : The conference session invitation has been sent and the local client should wait for the result of the invitation.	Header
x-modId	string	yes	Specifies the participant acceptance request. This parameter is generated to be sent over the event channel for participants acceptance request.	Header

4.11.8.1 Response – Example (XML)

This shows the response to a conference creation in XML format.

```
HTTP/1.1 204 No Content
x-modId:28463715
x-state: "add-pending"
```

4.11.8.2 Response – Example (JSON)

This shows the response to a conference creation in JSON format

```
HTTP/1.1 204 No Content
x-modId:28463715
x-state: "add-pending"
```

4.11.9 HTTP Response Codes

Please refer to appendix for HTTP Errors.

4.11.10 Service Exceptions

MessageId	Text	Variables	Parent HTTP Code
SVC0001	A service error has occurred. Error code is <Error Explanation>	Error Explanation : <content here>	400
SVC0002	Invalid input value for Message part <part name>	part name : name of the input parameter that resulted in the error	400
SVC0003	Invalid input value for Message part <part name>, valid values are <part values>	part value : value of input parameter that was found to be in error.	400
SVC0004	No valid addresses provided in the Message part <part name>	part name : name of the input parameter that resulted in the error	400

4.11.11 Policy Exceptions

MessageId	Text	Variables	Parent HTTP Code
POL0001	A policy error occurred. For example, rate limit error, authentication and authorization error.	N/A	401,403
POL0002	Privacy verification failed for address <address>, request is refused	address : <content here>	403
POL0003	Too many addresses specified in Message part	N/A	403
POL1009	User has not been provisioned for %1	System that hasn't been provisioned	403

4.12 **Operation:** Remove Participant from Conference

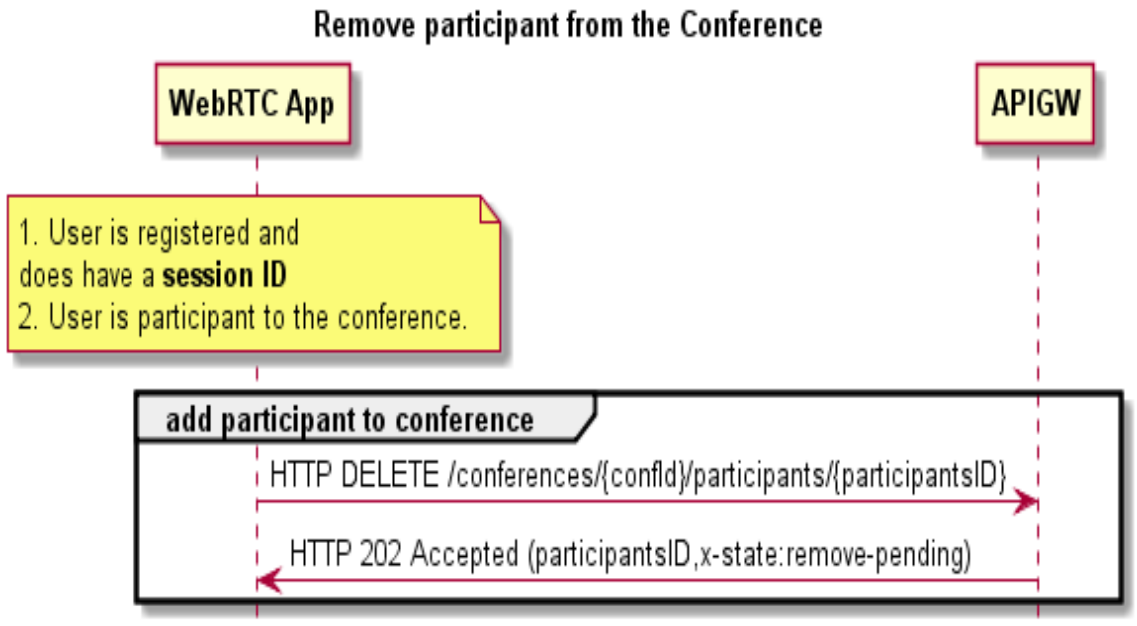
4.12.1 Functional Behavior

This operation is used to remove a conference participant during an ongoing conference.

Pre-Requisites :

- Client needs to be registered and have a session ID allocated for using this operation.
- Client needs to be participant.
- Client needs to establish an event channel to receive the result of add participant.

4.12.2 Call flow



4.12.3 Version Impact Summary

Service Version	Major or Minor Impact	Changes Introduced by this Version
1	Major	Initial Release

4.12.4 Authentication and Authorization

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description
authorization_code	Yes	RTC	1. Obtain the authorization code by passing App Key and App Secret along with scope. This redirects the user to AT&T authorization page to capture user consent. 2. Obtain OAuth access token using the OAuth authorization code along with the App Key and App Secret by making a Get Access Token request with grant_type=authorization_code. • Supports ICMN case
client_credentials	No	RTC	1. Obtain OAuth access token with the App Key and App Secret along with the scope by making a Get Access Token request with grant_type=client_credentials • Supports VTN case • Supports no-TN case

4.12.5 Representation Formats

4.12.6 Representation Formats

Direction	Supported Respresentation Formats
Request	XML, JSON,URLENCODED
Response	XML, JSON,URLENCODED

4.12.7 Input Parameters

Parameter	Data Type	Required?	Brief description	Location
accept	String	No	Specifies the format of the body of the response. The acceptable values for this parameter are: • application/json • application/xml • application/x-www-form-urlencoded The default value is application/json. Per rfc2616: ”If no Accept header field is present, then it is assumed that the client accepts all media types.” By default our services return application/json. The normal Accept header processing rules shall be followed according to rfc2616. *Note*: If there is no entity body in a normal successful response, this parameter is still needed to specify the format in the case of an error response message.	Header
authorization	String	Yes	Specifies the authorization type and token. The acceptable format for this parameter is the phrase ”Bearer OAuth Token” followed by a space ( ) and an OAuth access token. If this parameter value is missing from the header, then the API Gateway returns a message with an HTTP status code of 400 Invalid Request. If the OAuth access token is not valid, then the API Gateway returns an HTTP status code of 401 Unauthorized with a WWW-Authenticate HTTP header.	Header
conferenceId	String	Yes	Specifies conference identifier that is generated during conference initiatation.	URI Path
participantsId	String	Yes	A list with only one SIP or TEL URI of the participant to be removed.	URI Path

4.12.7.1 Request – Example (Remove participants from a Conference) - JSON

```
DELETE /RTC/v1/sessions/0045-ab42-89a2/conferences/15031864/participants/tel
    :+646264827 HTTP/1.1
Content-Type: application/json
Accept: application/json
```

4.12.7.2 Request – Example (Remove participants from a Conference) - XML

```
DELETE /RTC/v1/sessions/0045-ab42-89a2/conferences/15031864/participants/tel
    :+646264827 HTTP/1.1
Content-Type: application/xml
Accept: application/xml
```

4.12.8 Output Parameters

Parameter	Data Type	Required?	Brief description	Location
x-state	string	yes	The current status of the removed participant. The acceptable values for this parameter are: • Remove-pending : The conference session ending has been sent and the local client should wait for the result of the removal.	Header
x-modID	String	No	Specifies the modification identifier.	???

4.12.8.1 Response – Example (Remove participants from a Conference) (JSON)

```
HTTP/1.1 202 Accepted
x-modId:28463715
x-state: remove-pending
```

4.12.8.2 Response – Example (Remove participants from a Conference) (XML)

This shows the response to a conference creation in JSON format

```
HTTP/1.1 202 Accepted
x-modId:28463715
x-state: remove-pending
```

4.12.9 Service Exceptions

MessageId	Text	Variables	HTTP Status Code
SVC0001	A service error has occurred. Error code is <error_explanation>	error_explanation : <content_here>	400
SVC0002	Invalid input value for Message part <part_name>	part_name : name of the input parameter that resulted in the error.	400
SVC0003	Invalid input value for Message part <part_name>, valid values are <part_values>	part_name : name of the input parameter that resulted in the error. part_value : value of input parameter that was found to be in error.	400
SVC0004	No valid addresses provided in the Message part <part_name>	part_name : name of the input parameter that resulted in the error.	400

4.12.10 Policy Exceptions

MessageId	Text	Variables	HTTP Status Code
POL0001	A policy error occurred. For example, rate limit error, authentication and authorization error.	N/A	401, 403
POL0002	Privacy verification failed for address <address>, request is refused	address : <content_here>	403
POL0003	Too many addresses specified in Message part	N/A	403
POL1009	User has not been provisioned for %1	1% : System that has not been provisioned.	403

4.13 **Operation:** Modify Conference

4.13.1 Functional Behavior

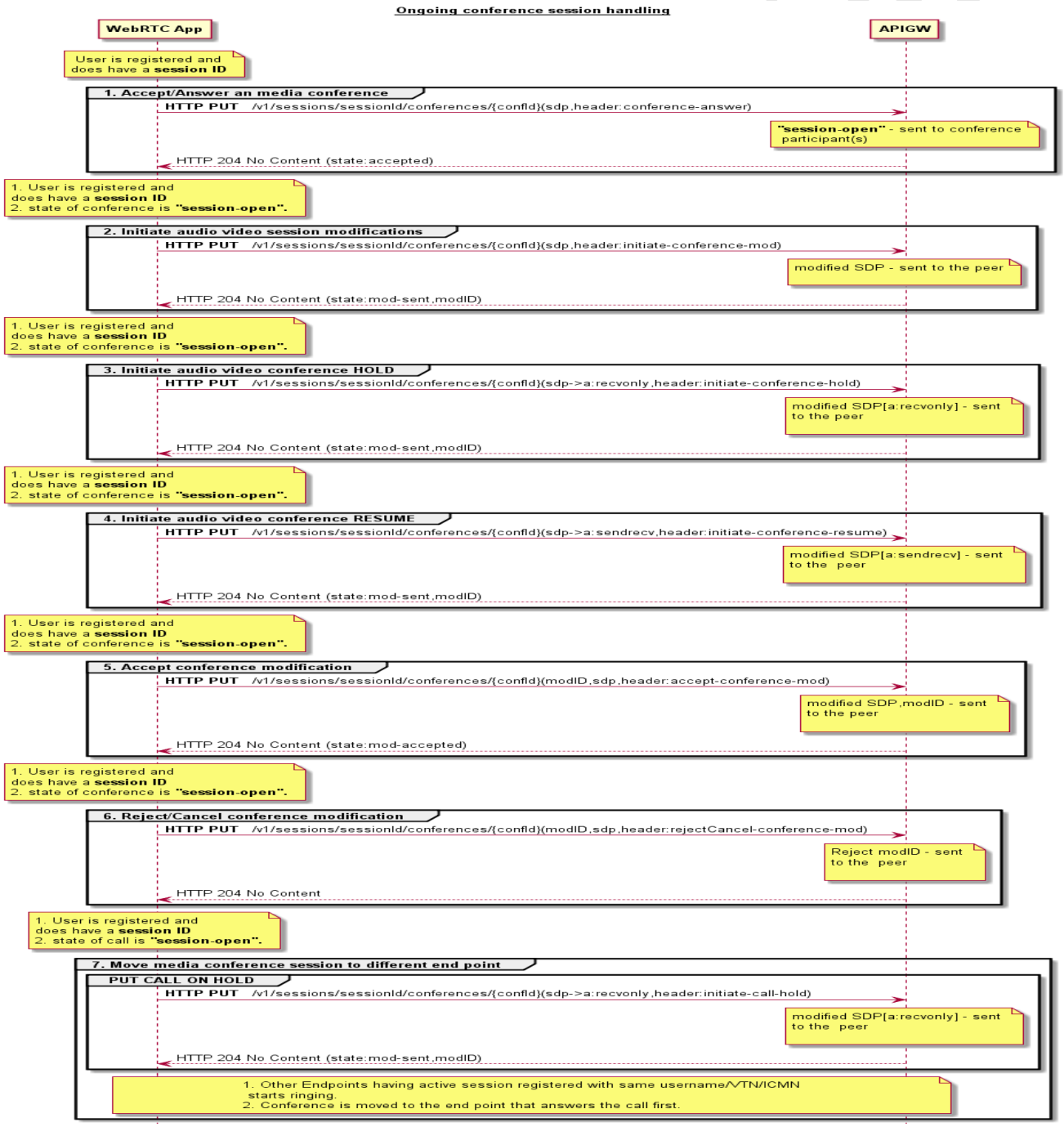
This operation is used to change the attributes of the media through session description protocol parameters. Below mentioned use cases can be achieved using this operation

1. Accept or Answer media Conference call.
2. Initiate Conference **media modifications.**
3. Initiate - Media Conference Session **Hold**.
4. Initiate - Media Conference Session **Resume**.
5. Accept Conference **media modifications** .
6. Reject/Cancel Conference **media modifications** .
7. **Move** media conference session to different device/Endpoint(webRTC browser).

Pre-Requisites :

- Client needs to be registered and have a session ID allocated for using this operation.
- The client needs to maintain an event channel to be able to receive the result of the conference session modification invitation.
- The client has conference a session with the state value in state **“session-open”**.

4.13.2 Call flow



4.13.3 Version Impact Summary

Service Version	Major or Minor Impact	Changes Introduced by this Version
1	Major	Initial Release

4.13.4 Authentication and Authorization

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description
authorization_code	Yes	RTC	1. Obtain the authorization code by passing App Key and App Secret along with scope. This redirects the user to AT&T authorization page to capture user consent. 2. Obtain OAuth access token using the OAuth authorization code along with the App Key and App Secret by making a Get Access Token request with grant_type=authorization_code. • Supports ICMN case
client_credentials	No	RTC	1. Obtain OAuth access token with the App Key and App Secret along with the scope by making a Get Access Token request with grant_type=client_credentials • Supports VTN case • Supports no-TN case

4.13.5 Representation Formats

4.13.6 Representation Formats

Direction	Supported Respresentation Formats
Request	XML, JSON,URLENCODED
Response	XML, JSON,URLENCODED

4.13.7 Input Parameters

Parameter	Data Type	Required?	Brief description	Location
accept	String	No	Specifies the format of the body of the response. The acceptable values for this parameter are: • application/json • application/xml • application/x-www-form-urlencoded The default value is application/json. Per rfc2616: ”If no Accept header field is present, then it is assumed that the client accepts all media types.” By default our services return application/json. The normal Accept header processing rules shall be followed according to rfc2616. *Note*: If there is no entity body in a normal successful response, this parameter is still needed to specify the format in the case of an error response message.	Header

Parameter	Data Type	Required?	Brief description	Location
authorization	String	Yes	Specifies the authorization type and token. The acceptable format for this parameter is the phrase ”Bearer OAuth Token” followed by a space ( ) and an OAuth access token. If this parameter value is missing from the header, then the API Gateway returns a message with an HTTP status code of 400 Invalid Request. If the OAuth access token is not valid, then the API Gateway returns an HTTP status code of 401 Unauthorized with a WWW-Authenticate HTTP header.	Header

Parameter	Data Type	Required?	Brief description	Location
x-conference-action	String	Yes	Specifies the action for a conference session. The acceptable values for this parameter are: • call-answer : When client accepts the request to join a conference. • initiate-media-mod : The client request for change in media attributes,for example removing a video stream from an ongoing conference session. • initiate-hold : The Client request to put the media path in a hold state. • initiate-resume : The Client request to resume the media path which was in a hold state. • initiate-conference-move: The Client requests to move the conference to a different endpoint. • accept-media-mod : The client accepts the changes in media attributes, for example removing a video stream from an ongoing conference session. • accept-hold : The Client accepts to put the media path in a hold state. • accept-resume : The Client accepts to resume the media path which was in a hold state. • reject-media-mod : The client accepts the changes in media attributes, for example removing a video stream from an ongoing conference session.	Header
x-mod-id	String	Conditional	Specifies the modification ID for media changes. This parameter is required if the **x-conference-action** parameter is set to accept-media-mod, accept-hold, accept-resume, or reject-media-mod.	Header

Parameter	Data Type	Required?	Brief description	Location
x-reject-reason	String	No	Specifies the reason for rejecting the session media changes.	Header
conferenceModifications	conferenceModifications Object	Yes	Contains the media modifications.	Body

Structure of conferenceModifications Object

Parameter	Data Type	Required?	Brief description
sdp	Sdp Object	Yes	Contains an SDP offer for the media conference session.

Structure of Sdp Object

Parameter	Data Type	Required?	Brief description
m	String	Yes	Specifies the media description in the following format. m=<media> <port> <proto> <fmt> The <media> field is the media type. The only acceptable value for this field is: • audio The <port> field is the transport port to which the media stream is sent. <proto> is the transport protocol. The only acceptable value for this field is: • TCP/RTMP The <fmt> field is a media format description. The fourth and any subsequent sub-fields describe the format of the media. The interpretation of the media format depends on the value of the <proto> sub-field.

attributes	List of Strings	Yes	Contains the attributes for extending SDP. This parameter may be defined to be used as session-level attributes, media-level attributes, or both. The attribute fields may be in one of the following formats. A property attribute is simply of the form: • a=<flag> • These are binary attributes, and the presence of the attribute conveys that the attribute is a property of the session. Example: a=recvonly A value attribute is of the form: • a=<attribute>:<value>

4.13.7.1 Request – Example (Accept/Answer media conference session) - JSON

```
PUT /RTC/v1/sessions/0045-ab42-89a2/conferences/1234 HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
x-conference-action: call-answer
Content-Length: xxx

{
    "conferenceModifications": {
        "sdp": [
            {
                "m": "audio 9 TCP/RTMP 8",
                "attributes": [
                    {
                        "a": "sendrecv"
                    },
                    {
                        "a": "rtpmap:8 pcmu/8000/1"
                    },
                    {
                        "a": "setup:active"
                    }
                ]
            }
        ]
    }
```

```
}
```

4.13.7.2 Request – Example (Accept/Answer media conference session) - XML

```
PUT /RTC/v1/sessions/0045-ab42-89a2/conferences/1234 HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/xml
x-conference-action:call-answer
Content-Length: yyy

<?xml version="1.0" encoding="UTF-8" ?>
    <conferenceModifications>
        <sdp>
            <m>audio 9 TCP/RTMP 8</m>
            <attributes>
                <a>sendrecv</a>
            </attributes>
            <attributes>
                <a>rtpmap:8 pcmu/8000/1</a>
            </attributes>
            <attributes>
                <a>setup:active</a>
            </attributes>
        </sdp>
    </conferenceModifications>
```

4.13.7.3 Request – Example (initiate Conference Media modifications) - JSON

```
PUT /RTC/v1/sessions/0045-ab42-89a2/conferences/1234 HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json, text/html
x-conference-action: initiate-media-mod
Content-Length: xxx

{
    "sdp": [
        {
            "m": "audio 9 TCP/RTMP 8",
            "attributes": [
                {
                    "a": "recvonly"
                },
                {
                    "a": "rtpmap:8 pcmu/8000/1"
                }
            ]
```

```
        }
    ]
}
```

4.13.7.4 Request – Example (initiate Conference Media modifications) - XML

```
PUT /RTC/v1/sessions/0045-ab42-89a2/conferences/1234 HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/xml, text/html
x-conference-action: initiate-media-mod
Content-Length: 134
<?xml version="1.0" encoding="UTF-8" ?>
    <sdp>
        <m>audio 9 TCP/RTMP 8</m>
        <attributes>
            <a>recvonly</a>
        </attributes>
        <attributes>
            <a>rtpmap:8 pcmu/8000/1</a>
        </attributes>
    </sdp>
```

4.13.7.5 Request – Example (initiate conference call hold) - JSON

```
PUT /RTC/v1/sessions/0045-ab42-89a2/conferences/1234 HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
x-conference-action: initiate-conference-hold
Content-Length: xxx

{
    "conferenceModifications": {
        "sdp": [
            {
                "m": "audio 9 TCP/RTMP 8",
                "attributes": [
                    {
                        "a": "recvonly"
                    },
                    {
                        "a": "rtpmap:8 pcmu/8000/1"
                    }
                ]
            }
        ]
    }
```

```
}
```

4.13.7.6 Request – Example (initiate conference call hold) - XML

```
PUT /RTC/v1/sessions/0045-ab42-89a2/conferences/1234 HTTP/1.1
Content-Type: application/xml;charset=UTF-8
Accept: application/xml
x-conference-action: initiate-conference-hold
Content-Length: xxx

<?xml version="1.0" encoding="UTF-8" ?>
    <conferenceModifications>
        <sdp>
            <m>audio 9 TCP/RTMP 8</m>
            <attributes>
                <a>recvonly</a>
            </attributes>
            <attributes>
                <a>rtpmap:8 pcmu/8000/1</a>
            </attributes>
        </sdp>
    </conferenceModifications>
```

4.13.7.7 Request – Example (initiate conference call resume) - JSON

```
PUT /RTC/v1/sessions/0045-ab42-89a2/conferences/1234 HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
x-conference-action: initiate-conference-resume
Content-Length: xxx

{
    "conferenceModifications": {
        "sdp": [
            {
                "m": "audio 9 TCP/RTMP 8",
                "attributes": [
                    {
                        "a": "sendrecv"
                    },
                    {
                        "a": "rtpmap:8 pcmu/8000/1"
                    }
                ]
            }
        ]
```

```
    }
}
```

4.13.7.8 Request – Example (initiate conference call resume) - XML

```
PUT /RTC/v1/sessions/0045-ab42-89a2/conferences/1234 HTTP/1.1
Content-Type: application/xml;charset=UTF-8
Accept: application/xml
x-conference-action: initiate-conference-resume
Content-Length: xxx

<?xml version="1.0" encoding="UTF-8" ?>
    <conferenceModifications>
        <sdp>
            <m>audio 9 TCP/RTMP 8</m>
            <attributes>
                <a>sendrecv</a>
            </attributes>
            <attributes>
                <a>rtpmap:8 pcmu/8000/1</a>
            </attributes>
        </sdp>
    </conferenceModifications>
```

4.13.7.9 Request – Example (initiate conference move) - JSON

```
PUT /RTC/v1/sessions/0045-ab42-89a2/conferences/1234 HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
x-conference-action: initiate-conference-move
Content-Length: xxx

{
    "conferenceModifications": {
        "sdp": [
            {
                "m": "audio 9 TCP/RTMP 8",
                "attributes": [
                    {
                        "a": "sendrecv"
                    },
                    {
                        "a": "rtpmap:8 pcmu/8000/1"
                    }
                ]
```

```
            }
        ]
    }
}
```

4.13.7.10 Request – Example (initiate conference move) - XML

```
PUT /RTC/v1/sessions/0045-ab42-89a2/conferences/1234 HTTP/1.1
Content-Type: application/xml;charset=UTF-8
Accept: application/xml
x-conference-action: initiate-conference-move
Content-Length: xxx

<?xml version="1.0" encoding="UTF-8" ?>
    <conferenceModifications>
        <sdp>
            <m>audio 9 TCP/RTMP 8</m>
            <attributes>
                <a>sendrecv</a>
            </attributes>
            <attributes>
                <a>rtpmap:8 pcmu/8000/1</a>
            </attributes>
        </sdp>
    </conferenceModifications>
```

4.13.7.11 Request – Example (Accept conference Media modifications) - JSON

```
PUT /RTC/v1/sessions/0045-ab42-89a2/conferences/1234 HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
x-conference-action: accept-conference-mod
x-modId:15011
Content-Length: xxx

{
  "conferenceModifications": {
    "sdp": [
      {
        "m": "audio 9000 RTP/AVP 8",
        "attributes": [
          { "a": "rcvonly" },
          { "a": "rtpmap:8 pcmu/8000/1" }
        ]
```

```
      },
      {
        "m": "video 9001 RTP/AVP 98",
        "attributes": [
          { "a": "rcvonly" },
          { "a": "rtpmap:98 H264/90000" },
          { "a": "fmtp:98 profile-levelid=A1438;packetization-mode=1" }
        ]
      }
    ]
  }
}
```

4.13.7.12 Request – Example (Accept conference Media modifications) - XML

```
PUT /RTC/v1/sessions/0045-ab42-89a2/conferences/1234 HTTP/1.1
Content-Type: application/xml;charset=UTF-8
Accept: application/xml
x-conference-action: accept-conference-mod
x-modId:15011
Content-Length: xxx

<?xml version="1.0" encoding="UTF-8" ?>
    <conferenceModifications>
        <sdp>
            <m>audio 9000 RTP/AVP 8</m>
            <attributes>
                <a>rcvonly</a>
            </attributes>
            <attributes>
                <a>rtpmap:8 pcmu/8000/1</a>
            </attributes>
        </sdp>
        <sdp>
            <m>video 9001 RTP/AVP 98</m>
            <attributes>
                <a>rcvonly</a>
            </attributes>
            <attributes>
                <a>rtpmap:98 H264/90000</a>
            </attributes>
            <attributes>
                <a>fmtp:98 profile-levelid=A1438;packetization-mode=1</a>
            </attributes>
        </sdp>
    </conferenceModifications>
```

4.13.7.13 Request – Example (Reject/Cancel conference Media modifications) - JSON

```
PUT /RTC/v1/sessions/0045-ab42-89a2/conferences/1234 HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
x-conference-action: rejectCancel-conference-mod
x-modId:15011
Content-Length: xxx

{
  "conferenceModifications": {
    "sdp": [
      {
        "m": "audio 9000 RTP/AVP 8",
        "attributes": [
          { "a": "rcvonly" },
          { "a": "rtpmap:8 pcmu/8000/1" }
        ]
      },
      {
        "m": "video 9001 RTP/AVP 98",
        "attributes": [
          { "a": "rcvonly" },
          { "a": "rtpmap:98 H264/90000" },
          { "a": "fmtp:98 profile-levelid= A1438;packetization-mode=1" }
        ]
      }
    ]
  }
}
```

4.13.7.14 Request – Example (Reject/Cancel conference Media modifications) - XML

```
PUT /RTC/v1/sessions/0045-ab42-89a2/conferences/1234 HTTP/1.1
Content-Type: application/xml;charset=UTF-8
Accept: application/xml
x-conference-action: rejectCancel-conference-mod
x-modId:15011
Content-Length: xxx

<?xml version="1.0" encoding="UTF-8" ?>
    <conferenceModifications>
        <sdp>
            <m>audio 9000 RTP/AVP 8</m>
            <attributes>
```

```
                <a>rcvonly</a>
            </attributes>
            <attributes>
                <a>rtpmap:8 pcmu/8000/1</a>
            </attributes>
        </sdp>
        <sdp>
            <m>video 9001 RTP/AVP 98</m>
            <attributes>
                <a>rcvonly</a>
            </attributes>
            <attributes>
                <a>rtpmap:98 H264/90000</a>
            </attributes>
            <attributes>
                <a>fmtp:98 profile-levelid=A1438;packetization-mode=1</a>
            </attributes>
        </sdp>
    </conferenceModifications>
```

4.13.8 Output Parameters

Parameter	Data Type	Required?	Brief description	Location
x-modId	String	Conditional	Specifies the identifier associated with the modification of media attributes. Note: This parameter must be returned as a response for the initiation of media modifications.	Header
x-state	String	Conditional	Specifies the current state of the media modifications. Note: This parameter must be returned as a response for the initiation or acceptance of media modifications.	Header

4.13.8.1 Response – Example (Accept/Answer media conference session) - JSON

```
HTTP/1.1 204 No Content
x-state: accepted
```

4.13.8.2 Response – Example (Accept/Answer media conference session) - (XML)

```
HTTP/1.1 204 No Content
x-state: accepted
```

4.13.8.3 Response – Example (initiate conference call Media modifications) - JSON

```
HTTP/1.1 204 No Content
x-modID: abc-1234-def-567
x-state: mod-sent
```

4.13.8.4 Response – Example (initiate conference call session modifications) - (XML)

```
HTTP/1.1 204 No Content
x-modID: abc-1234-def-567
x-state: mod-sent
```

4.13.8.5 Response – Example (initiate conference call hold) - JSON

```
HTTP/1.1 204 No Content
x-modID: abc-1234-def-567
x-state: mod-sent
```

4.13.8.6 Response – Example (initiate conference call hold) - (XML)

```
HTTP/1.1 204 No Content
x-modID: abc-1234-def-567
x-state: mod-sent
```

4.13.8.7 Response – Example (initiate conference call resume) - JSON

```
HTTP/1.1 204 No Content
x-modID: abc-1234-def-567
x-state: mod-sent
```

4.13.8.8 Response – Example (initiate conference call resume) - (XML)

```
HTTP/1.1 204 No Content
x-modID: abc-1234-def-567
x-state: mod-sent
```

4.13.8.9 Response – Example (initiate conference call move) - JSON

```
HTTP/1.1 204 No Content
x-modID: abc-1234-def-567
x-state: mod-sent
```

4.13.8.10 Response – Example (initiate conference call move) - (XML)

```
HTTP/1.1 204 No Content
x-modID: abc-1234-def-567
x-state: mod-sent
```

4.13.8.11 Response – Example (Accept conference Media modifications) - JSON

```
HTTP/1.1 204 No Content
x-modID: abc-1234-def-567
x-state: mod-accepted
```

4.13.8.12 Response – Example (Accept conference Media modifications) - (XML)

```
HTTP/1.1 204 No Content
x-modID: abc-1234-def-567
x-state: mod-accepted
```

4.13.8.13 Response – Example (Reject/Cancel conference Media modifications) - JSON

```
HTTP/1.1 204 No Content
x-state: mod-rejectcancel
```

4.13.8.14 Response – Example (Reject/Cancel conference Media modifications) - (XML)

```
HTTP/1.1 204 No Content
x-state: mod-rejectcancel
```

4.13.9 HTTP Response Codes

Code	Reason Phrase	Description
200	OK	Successful response
201	Created	Successful response and resource was successfully created.
202	Accepted	Successful response but action has not yet been enacted
204	No Content	Successful response and the response does not include an entity.
206	Partial Content	The server has fulfilled the partial GET request for the resource
304	Not Modified	Returned when the client has performed a conditional GET request and access is allowed, but the resource has not been modified.
400	Bad Request	Many possible reasons not specified by the other codes.
401	Authentication Error	Authentication failed or was not provided.
403	Forbidden	Access permission error.
404	Not Found	The server has not found anything matching the Request-URI. No indication is given of whether the condition is temporary or permanent.
405	Method Not Allowed	A request was made of a resource using a request method not supported by that resource (e.g., using PUT on a REST resource that only supports POST).
406	Not Acceptable	The resource identified by the request is only capable of generating response entities which have content characteristics not acceptable according to the accept headers sent in the request.
408	Request Timeout	The client did not produce a request within the time that the server was prepared to wait. The client MAY repeat the request without modifications at any later time.

Code	Reason Phrase	Description
200	OK	Successful response
409	Conflict	The request could not be completed due to a conflict with the current state of the resource. This code is only allowed in situations where it is expected that the user might be able to resolve the conflict and resubmit the request.
411	Length Required	The Content-Length header was not specified.
413	Request Entity Too Large	The size of the request body exceed the maximum size permitted.
414	Request-URI Too Long	The server is refusing to service the request because the Request-URI is longer than the server is willing to interpret.
415	Unsupported Media Type	The request is in a format not supported by the requested resource for the requested method.
500	Internal Server Error	The server encountered an internal error or timed out; please retry.
503	Service Unavailable	The server is currently unable to receive requests; please retry.

4.13.10 Service Exceptions

MessageId	Text	Variables	Parent HTTP Code
SVC0001	A service error has occurred. Error code is <Error Explanation>	Error Explanation : <content here>	400
SVC0002	Invalid input value for Message part <part name>	part name : name of the input parameter that resulted in the error	400
SVC0003	Invalid input value for Message part <part name>, valid values are <part values>	part value : value of input parameter that was found to be in error.	400
SVC0004	No valid addresses provided in the Message part <part name>	part name : name of the input parameter that resulted in the error	400

4.13.11 Policy Exceptions

MessageId	Text	Variables	Parent HTTP Code
POL0001	A policy error occurred. For example, rate limit error, authentication and authorization error.	N/A	401,403
POL0002	Privacy verification failed for address <address>, request is refused	address : <content here>	403
POL0003	Too many addresses specified in Message part	N/A	403
POL1009	User has not been provisioned for %1	System that hasn't been provisioned	403

4.14 Operation: Cancel End Reject Conference

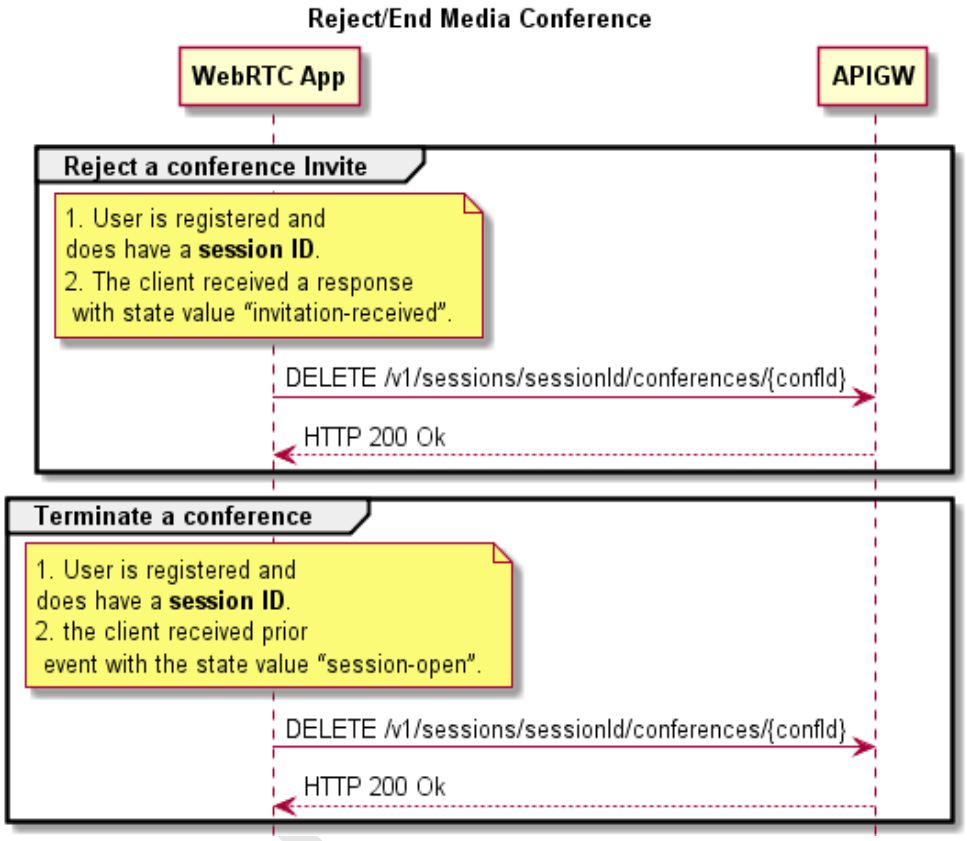
4.14.1 Functional Behavior

This operation is used to reject a conference invitation or terminate an ongoing conference .

The Following criteria should be met to terminate or end conference:

- Client needs to be registered and have a session ID allocated for using this operation.
- To reject an invitation: the client received a media conference event with state value **"invitation-received"**.
- To terminate participation in a conference: the client received a media conference event with the state value **"session-open"**.

4.14.2 Call flow



4.14.3 Version Impact Summary

Service Version	Major or Minor Impact	Changes Introduced by this Version
1	Major	Initial Release

4.14.4 Authentication and Authorization

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description
authorization_code	Yes	RTC	1. Obtain the authorization code by passing App Key and App Secret along with scope. This redirects the user to AT&T authorization page to capture user consent. 2. Obtain OAuth access token using the OAuth authorization code along with the App Key and App Secret by making a Get Access Token request with grant_type=authorization_code. • Supports ICMN case
client_credentials	No	RTC	1. Obtain OAuth access token with the App Key and App Secret along with the scope by making a Get Access Token request with grant_type=client_credentials • Supports VTN case • Supports no-TN case

4.14.5 Representation Formats

4.14.6 Representation Formats

Direction	Supported Respresentation Formats
Request	XML, JSON,URLENCODED
Response	XML, JSON,URLENCODED

4.14.7 Input Parameters

Parameter	Data Type	Required?	Brief description	Location
accept	String	No	Specifies the format of the body of the response. The acceptable values for this parameter are: • application/json • application/xml • application/x-www-form-urlencoded The default value is application/json. Per rfc2616: ”If no Accept header field is present, then it is assumed that the client accepts all media types.” By default our services return application/json. The normal Accept header processing rules shall be followed according to rfc2616. *Note*: If there is no entity body in a normal successful response, this parameter is still needed to specify the format in the case of an error response message.	Header
authorization	String	Yes	Specifies the authorization type and token. The acceptable format for this parameter is the phrase ”Bearer OAuth Token” followed by a space ( ) and an OAuth access token. If this parameter value is missing from the header, then the API Gateway returns a message with an HTTP status code of 400 Invalid Request. If the OAuth access token is not valid, then the API Gateway returns an HTTP status code of 401 Unauthorized with a WWW-Authenticate HTTP header.	Header
x-delete-reason	String	Yes	Specifies the reason for ending the conference. The acceptable values for this parameter are: • reject : The participant receives an invitation and is unable to join. • terminate : The session is ongoing and participant is exiting the conference.	Header
conferenceId	String	Yes	Specifies the conference identifier allocated during initiation of conference.	URI

4.14.7.1 Request – Example (Terminate or Reject a Conference) - JSON

JSON request to End or Reject the conference

```
DELETE /RTC/v1/sessions/0045-ab42-89a2/conferences/15031864 HTTP/1.1
x-delete-reason=<reasonValue>
Content-Type: application/json
Accept: application/json
```

4.14.7.2 Request – Example (Terminate or Reject a Conference) - XML

XML request to End or Reject the conference

```
DELETE /RTC/v1/sessions/0045-ab42-89a2/conferences/15031864 HTTP/1.1
x-delete-reason=<reasonValue>
Content-Type: application/xml
Accept: application/xml
```

4.14.8 Output Parameters

Parameter	Data Type	Required?	Brief description	Location

4.14.8.1 Response – (Terminate or Reject a Conference) Example (JSON)

```
HTTP/1.1 200 OK
```

4.14.8.2 Response – (Terminate or Reject a Conference) Example (XML)

```
HTTP/1.1 200 OK
```

4.14.9 HTTP Response Codes

Please refer to Appendix for HTTP Errors.

4.14.10 Service Exceptions

MessageId	Text	Variables	Parent HTTP Code
SVC0001	A service error has occurred. Error code is <Error Explanation>	Error Explanation : <content here>	400
SVC0002	Invalid input value for Message part <part name>	part name : name of the input parameter that resulted in the error	400
SVC0003	Invalid input value for Message part <part name>, valid values are <part values>	part value : value of input parameter that was found to be in error.	400
SVC0004	No valid addresses provided in the Message part <part name>	part name : name of the input parameter that resulted in the error	400

4.14.11 Policy Exceptions

MessageId	Text	Variables	Parent HTTP Code
POL0001	A policy error occurred. For example, rate limit error, authentication and authorization error.	N/A	401,403
POL0002	Privacy verification failed for address <address>, request is refused	address : <content here>	403
POL0003	Too many addresses specified in Message part	N/A	403
POL1009	User has not been provisioned for %1	System that hasn't been provisioned	403

4.15 **Operation:** Create Group Chat

4.15.1 Functional Behavior

The Create Group Chat method is used to create chat group where mutlitile participants are able to exchange chats. A group chat is identified with a group identifier. This **group id** should be included in all of the following requests related to that chat group

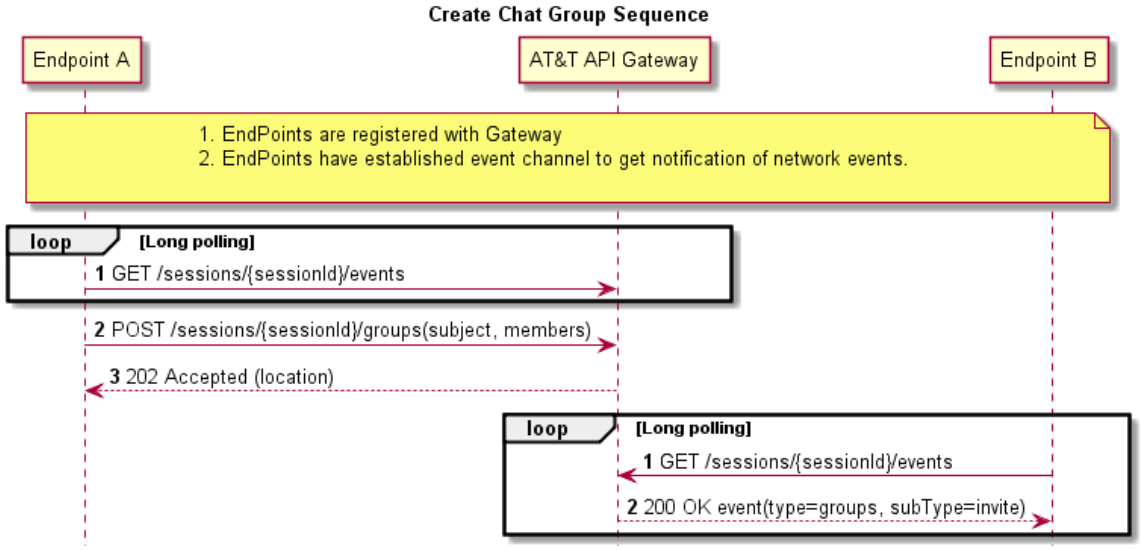
Note: The endpoint must be registered with AT&T network before using this method.

Note: The endpoint also must establish an event channel to receive notification of possible failures and other status updates from the AT&T network.

Note: Invited users receive an event named invite. The endpoint receives events named info when users are joining or leaving the group chat. After the first user accepted, the inviter is able to send chats to the group chat.

Note: A network configurable parameter limits the maximum number of participants that are able to be added to the group chat.

4.15.2 Call flow



4.15.3 Version Impact Summary

Service Version	Major or Minor Impact	Changes Introduced by this Version
1	Major	Initial Release

4.15.4 Authentication and Authorization

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description
authorization_code	Yes	RTC	1. Obtain the authorization code by passing App Key and App Secret along with scope. This redirects the user to AT&T authorization page to capture user consent. 2. Obtain OAuth access token using the OAuth authorization code along with the App Key and App Secret by making a Get Access Token request with grant_type=authorization_code. • Supports ICMN case
client_credentials	No	RTC	1. Obtain OAuth access token with the App Key and App Secret along with the scope by making a Get Access Token request with grant_type=client_credentials • Supports VTN case • Supports no-TN case

4.15.5 Representation Formats

4.15.6 Representation Formats

Direction	Supported Respresentation Formats
Request	XML, JSON,URLENCODED
Response	XML, JSON,URLENCODED

4.15.7 Input Parameters

Parameter	Data Type	Req?	Brief description	Location
accept	String	No	Specifies the format of the body of the response. The acceptable values for this parameter are: • application/json • application/xml • application/x-www-form-urlencoded The default value is application/json. Per rfc2616: ”If no Accept header field is present, then it is assumed that the client accepts all media types.” By default our services return application/json. The normal Accept header processing rules shall be followed according to rfc2616. *Note*: If there is no entity body in a normal successful response, this parameter is still needed to specify the format in the case of an error response message.	Header
authorization	String	Yes	Specifies the authorization type and token. The acceptable format for this parameter is the phrase ”Bearer OAuth Token” followed by a space ( ) and an OAuth access token. If this parameter value is missing from the header, then the API Gateway returns a message with an HTTP status code of 400 Invalid Request. If the OAuth access token is not valid, then the API Gateway returns an HTTP status code of 401 Unauthorized with a WWW-Authenticate HTTP header.	Header

Parameter	Data Type	Req?	Brief description	Location
content-length	Integer	No	Specifies the length of the content in octets. This parameter is only required for a non-streaming request.	Header
content-type	String	Yes	Specifies the type of content of the body. The acceptable values for this parameter are: • application/xml • application/json • image/jpeg	Header
transfer-encoding	String	Conditional	Specifies the encodingof the message. The only acceptable values for this parameter is: • chunked This parameter is only required for a streaming request.	Header
group	Object	Conditional	Contains the destination details of the new group. • This object is not required for accepting, rejecting, or joining using HTTP PUT.	Body

Structure of group Object

Parameter	Data Type	Req?	Brief description
subject	String	Yes	Specifies the name or description for chat group which is being created.
members	List of Strings	Yes	Specifies the list of members for the chat group which is being created.

4.15.7.1 Request – Example (Create a chat group)

This demostrates how to create chat group using POST, and specifying the response to be in XML.

```
POST /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/groups HTTP/1.1
Host: api.att.com
```

```
authorization: Bearer abcdef12345678
accept: application/xml
content-type: application/xml
<?xml version="1.0" encoding="UTF-8" ?>
<group>
  <subject>Global warming</subject>
  <members>
    <member>tel:+2345678901</member>
    <member>tel:+2345678902</member>
    <member>tel:+2345678903</member>
  </members>
</group>
```

4.15.7.2 Request – Example (Create chat group json)

This demostrates how create a chat group. The response format will be in JSON.

```
POST /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/groups HTTP/1.1
Host: api.att.com
authorization: Bearer 38C2399A23999
accept: application/json
content-length: 300
content-type: application/json
{
    "group":
        {
        "subject": "Global warming",
        "members":["tel:+2345678902","tel:+2345678903","tel:+2345678901"]
        }
}
```

4.15.8 Output Parameters

Parameter	Data Type	Req?	Brief description	Location
content-type	String	Yes	Specifies the type of content of the body. The acceptable values for this parameter are: • application/xml • application/json	Header

Parameter	Data Type	Req?	Brief description	Location
location	String	Yes	Specifies the location of the newly created group chat. Location uri contains newly created chat group id.	Uri

4.15.8.1 Response – Example (Create chat group)

This shows the response to a chat group creation in XML format.

```
HTTP/1.1 202 Accepted.
Content-Type: application/xml
Content-Length: 100
Date: Tue, 04 Dec 2013 02:51:59 GMT
location: /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/groups/j7q5bL
```

4.15.9 Service Exceptions

MessageId	Text	Variables	Parent HTTP Code
SVC0001	A service error has occurred. Error code is <error_explanation>	error_explanation : <content_here>	400
SVC0002	Invalid input value for Message part <part_name>	part_name : name of the input parameter that resulted in the error.	400
SVC0003	Invalid input value for Message part <part_name>, valid values are <part_values>	part_name : name of the input parameter that resulted in the error. part_value : value of input parameter that was found to be in error.	400
SVC0004	No valid addresses provided in the Message part <part_name>	part_name : name of the input parameter that resulted in the error.	400

4.15.10 Policy Exceptions

MessageId	Text	Variables	Parent HTTP Code
POL0001	A policy error occurred. For example, rate limit error, authentication and authorization error.	N/A	401, 403
POL0002	Privacy verification failed for address <address>, request is refused	address : <content_here>	403
POL0003	Too many addresses specified in Message part	N/A	403
POL1009	User has not been provisioned for %1	1% : System that has not been provisioned	403

4.16 Operation: Modify Group Chat

4.16.1 Functional Behavior

The Modify Group Chat method performs the following actions.

- Accept : The invitation to join a group chat is accepted.
- Reject : The invitation to join a group chat is rejected.
- Join : An on-going group chat conversation is joined.

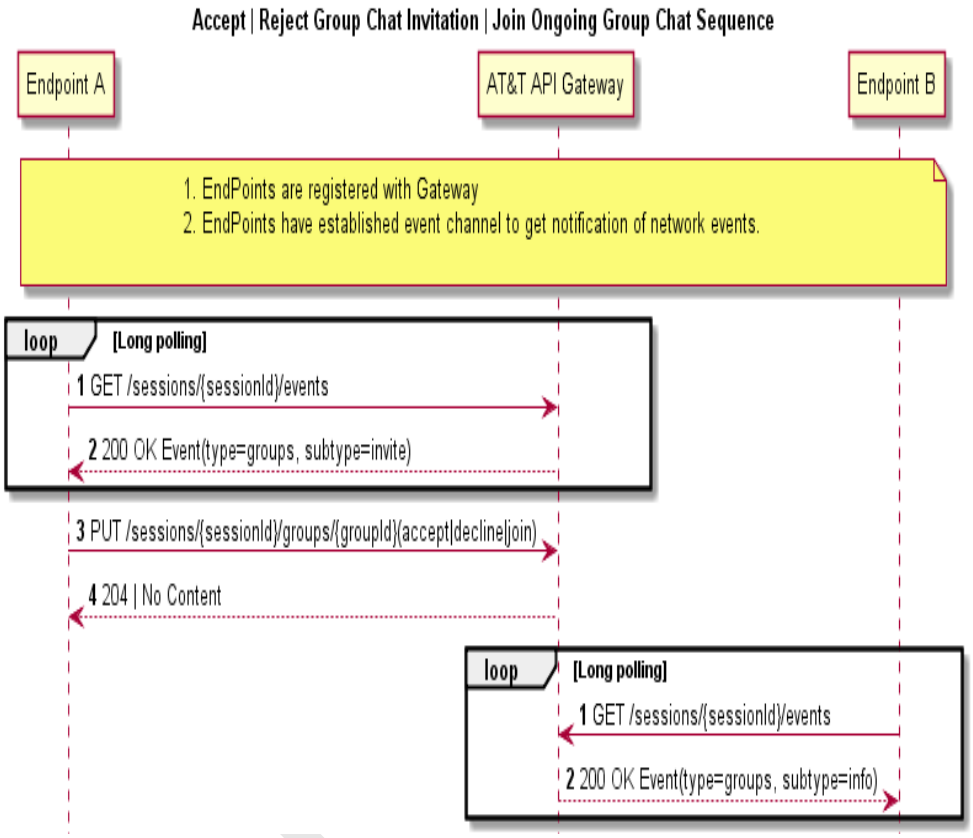
Note: The endpoint needs to be registered with AT&T network before accepting or rejecting an invitation to join a group.

Note: The endpoint also needs to establish an event channel to receive notification of possible failures and other status updates from the AT&T network.

Note: The endpoint must have received an invitation with a group identifier for this request to succeed.

Note: A network configurable parameter limits the maximum number of participants that is able to be added to the group. Refer to parameters appendix for more details on global netwrok settings

4.16.2 Call flow



4.16.3 Version Impact Summary

Service Version	Major or Minor Impact	Changes Introduced by this Version
1	Major	Initial Release

4.16.4 Authentication and Authorization

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description
authorization_code	Yes	RTC	1. Obtain the authorization code by passing App Key and App Secret along with scope. This redirects the user to AT&T authorization page to capture user consent. 2. Obtain OAuth access token using the OAuth authorization code along with the App Key and App Secret by making a Get Access Token request with grant_type=authorization_code. • Supports ICMN case
client_credentials	No	RTC	1. Obtain OAuth access token with the App Key and App Secret along with the scope by making a Get Access Token request with grant_type=client_credentials • Supports VTN case • Supports no-TN case

4.16.5 Representation Formats

4.16.6 Representation Formats

Direction	Supported Respresentation Formats
Request	XML, JSON,URLENCODED
Response	XML, JSON,URLENCODED

4.16.7 Input Parameters

Parameter	Data Type	Req?	Brief description	Location
accept	String	No	Specifies the format of the body of the response. The acceptable values for this parameter are: • application/json • application/xml • application/x-www-form-urlencoded The default value is application/json. Per rfc2616: ”If no Accept header field is present, then it is assumed that the client accepts all media types.” By default our services return application/json. The normal Accept header processing rules shall be followed according to rfc2616. *Note*: If there is no entity body in a normal successful response, this parameter is still needed to specify the format in the case of an error response message.	Header

Parameter	Data Type	Req?	Brief description	Location
authorization	String	Yes	Specifies the authorization type and token. The acceptable format for this parameter is the phrase ”Bearer OAuth Token” followed by a space ( ) and an OAuth access token. If this parameter value is missing from the header, then the API Gateway returns a message with an HTTP status code of 400 Invalid Request. If the OAuth access token is not valid, then the API Gateway returns an HTTP status code of 401 Unauthorized with a WWW-Authenticate HTTP header.	Header
content-length	Integer	No	Specifies the length of the content in octets. This parameter is only required for a non-streaming request.	Header
content-type	String	Yes	Specifies the type of content of the body. The acceptable values for this parameter are: • application/xml • application/json	Header
transfer-encoding	String	Conditional	Specifies the encodingof the message. The only acceptable values for this parameter is: • chunked This parameter is only required for a streaming request.	Header
x-invitation	String	Yes	Specifies the answer to the invitation. The acceptable values for this parameter are: • accept : The participant accepts the chat invitation. • decline : The participant rejects the chat invitation. • join : Th participant joins ongoing group chat conversation.	Header

4.16.7.1 Request – Example (Accept invitation to join a chat group - XML)

This demostrates how to accept invitation to join a group using xml.

```
PUT /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/groups/j7q5bL
    HTTP/1.1
Host: api.att.com
authorization: Bearer abcdef12345678
accept: application/xml
content-type: application/xml
x-invitation: accept
```

4.16.7.2 Request – Example (Accept invitation to join a chat group - JSON)

This demostrates how to accept invitation to join a group using JSON.

```
PUT /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/groups/j7q5bL
    HTTP/1.1
Host: api.att.com
authorization: Bearer 38C2399A23999
accept: application/json
content-length: 300
content-type: application/json
x-invitation: accept
```

4.16.7.3 Request – Example (Reject invitation to join a chat group - XML)

This demostrates how to accept invitation to join a group using xml.

```
PUT /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/groups/j7q5bL
    HTTP/1.1
Host: api.att.com
authorization: Bearer abcdef12345678
accept: application/xml
content-type: application/xml
x-invitation: decline
```

4.16.7.4 Request – Example (Reject invitation to join a chat group- JSON)

This demostrates how to reject invitation to join a group using JSON.

```
PUT /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/groups/j7q5bL
    HTTP/1.1
Host: api.att.com
```

```
authorization: Bearer 38C2399A23999
accept: application/json
content-length: 300
x-invitation: decline
```

4.16.7.5 Request – Example (Join ongoing chat conversation - JSON)

This demostrates how to reject invitation to join a group using JSON.

```
PUT /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/chatGroups/j7q5bL
    HTTP/1.1
Host: api.att.com
authorization: Bearer 38C2399A23999
accept: application/json
content-length: 300
content-type: application/json
x-invitation: join
```

4.16.8 Output Parameters

Parameter	Data Type	Req?	Brief description	Location
content-type	String	Yes	Specifies the type of content of the body. The acceptable values for this parameter are: • application/xml • application/json	Header

4.16.8.1 Response – Example (Accept invitation to join a chat group - XML)

This shows the response to a accept/reject invitation to join a chat group XML format.

```
HTTP/1.1 204 No Content
Content-Type: application/xml
Content-Length: 100
Date: Tue, 04 Dec 2013 02:51:59 GMT
```

4.16.8.2 Response – Example (Joining ongoing group conversation - XML)

This shows the response to joining ongoing group conversation XML format.

```
HTTP/1.1 202 Accepted
Content-Type: application/xml
Content-Length: 100
Date: Tue, 04 Dec 2013 02:51:59 GMT
```

4.16.9 Service Exceptions

MessageId	Text	Variables	Parent HTTP Code
SVC0001	A service error has occurred. Error code is <error_explanation>	error_explanation : <content_here>	400
SVC0002	Invalid input value for Message part <part_name>	part_name : name of the input parameter that resulted in the error.	400
SVC0003	Invalid input value for Message part <part_name>, valid values are <part_values>	part_name : name of the input parameter that resulted in the error. part_value : value of input parameter that was found to be in error.	400
SVC0004	No valid addresses provided in the Message part <part_name>	part_name : name of the input parameter that resulted in the error.	400

4.16.10 Policy Exceptions

MessageId	Text	Variables	Parent HTTP Code

MessageId	Text	Variables	Parent HTTP Code
POL0001	A policy error occurred. For example, rate limit error, authentication and authorization error.	N/A	401, 403
POL0002	Privacy verification failed for address <address>, request is refused	address : <content_here>	403
POL0003	Too many addresses specified in Message part	N/A	403
POL1009	User has not been provisioned for %1	1% : System that has not been provisioned	403

4.17 Operation: Send Message To Group Chat

4.17.1 Functional Behavior

This operation used send chats to the chat group.

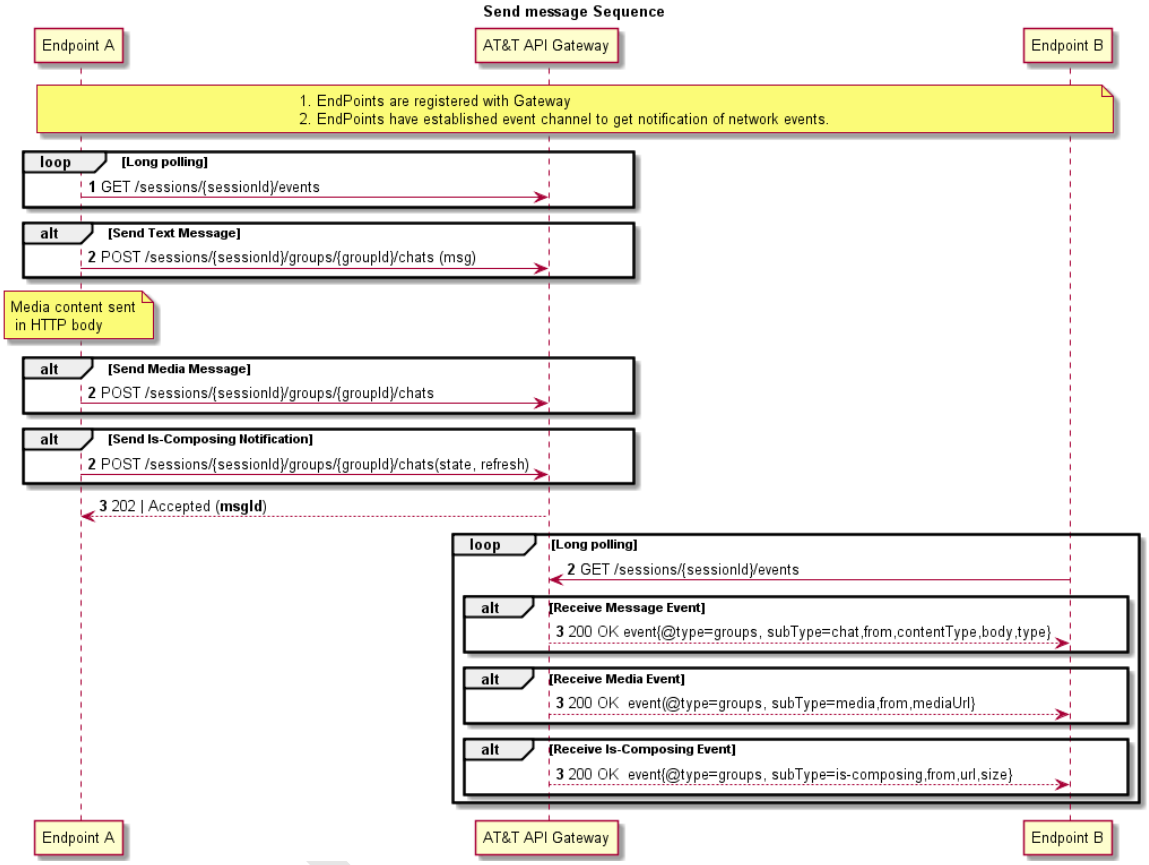
Following use cases will be cover under this operation.

1. Sending **text** messsage.
2. Sending **media** message.
3. Send **is-composing** message.

Note:

1. A media message is only sent by AT&T network to the recipient after previously a message has been sent or received to/from the same user on short intervals.
2. End point needs to be registered with AT&T network before sending the message to the peer(s).
3. End point also needs to establish an event channel to receive notification of possible failures from the AT&T network.

4.17.2 Call flow



4.17.3 Version Impact Summary

Service Version	Major or Minor Impact	Changes Introduced by this Version
1	Major	Initial Release

4.17.4 Authentication and Authorization

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description
authorization_code	Yes	RTC	1. Obtain the authorization code by passing App Key and App Secret along with scope. This redirects the user to AT&T authorization page to capture user consent. 2. Obtain OAuth access token using the OAuth authorization code along with the App Key and App Secret by making a Get Access Token request with grant_type=authorization_code. • Supports ICMN case
client_credentials	No	RTC	1. Obtain OAuth access token with the App Key and App Secret along with the scope by making a Get Access Token request with grant_type=client_credentials • Supports VTN case • Supports no-TN case

4.17.5 Representation Formats

4.17.6 Representation Formats

Direction	Supported Respresentation Formats
Request	XML, JSON,URLENCODED
Response	XML, JSON,URLENCODED

4.17.7 Input Parameters

Parameter	Data Type	Req?	Brief description	Location
accept	String	No	Specifies the format of the body of the response. The acceptable values for this parameter are: • application/json • application/xml • application/x-www-form-urlencoded The default value is application/json. Per rfc2616: ”If no Accept header field is present, then it is assumed that the client accepts all media types.” By default our services return application/json. The normal Accept header processing rules shall be followed according to rfc2616. *Note*: If there is no entity body in a normal successful response, this parameter is still needed to specify the format in the case of an error response message.	Header
authorization	String	Yes	Specifies the authorization type and token. The acceptable format for this parameter is the phrase ”Bearer OAuth Token” followed by a space ( ) and an OAuth access token. If this parameter value is missing from the header, then the API Gateway returns a message with an HTTP status code of 400 Invalid Request. If the OAuth access token is not valid, then the API Gateway returns an HTTP status code of 401 Unauthorized with a WWW-Authenticate HTTP header.	Header
content-length	Integer	No	Specifies the length of the content in octets. This parameter is only required for a non-streaming request.	Header

Parameter	Data Type	Req?	Brief description	Location
content-type	String	Yes	Specifies the type of content of the body. The acceptable values for this parameter are: • application/xml • application/json • image/jpeg	Header
transfer-encoding	String	Conditional	Specifies the encodingof the message. The only acceptable values for this parameter is: • chunked This parameter is only required for a streaming request.	Header
chat	Object	Conditional	Contains the details of chat message. This object is not requied to send media message. Media is sent as http payload.	Body

Structure of chat Object

Parameter	Data Type	Req?	Brief description
body	String	Conditional	Specifies the text message being sent. Note: This parameter is required only in case of sending a text message.
state	String	Conditional	Specifies the state of user. The acceptable values for this field are: • active : The user is currently composing a message • idle : The user is currently idle or stopped typing. Note: This parameter is required only in case of sending an is-composing message.

Parameter	Data Type	Req?	Brief description
refresh	Integer	Conditional	Specifies the timer, in seconds, that should be associated to the is-composing icon. If this parameter is set to 120, then it means the receiver displays an is-composing icon for 120 seconds. The is-composing icon should be removed when the timer expires, a chat message is received, or an is-composing message with the state set to idle is received. Note: This parameter is required only in case of sending an is-composing messge

4.17.7.1 Request – Example (Send a text chat message)

This demostrates how to send chat to a group using POST, and specifying the response to be in XML.

```
POST /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/groups/j7q5bL/chats
     HTTP/1.1
Host: api.att.com
authorization: Bearer abcdef12345678
accept: application/xml
content-type: application/xml
<?xml version="1.0" encoding="UTF-8" ?>
<chat>
    <body>This is my message</body>
</chat>
```

4.17.7.2 Request – Example (Send a media chat message)

This demostrates how to send an media chat message . The response format will be in JSON.

```
POST /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/groups/j7q5bL/
    chats HTTP/1.1
Host: api.att.com
authorization: Bearer 38C2399A23999
accept: application/json
content-length: 5655
Content-Type: image/jpeg
```

```
00105e0 e6b0 343b 9c74 0804 e7bc 0804 e7d5 0804
00105f0 e7e4 0804 e6b0 0804 e7f0 0804 e7ff 0804
0010600 e80b 0804 e81a 0804 e6b0 0804 e6b0 0804
```

4.17.7.3 Request – Example (Send text chat message in json)

This demostrates how to send text message using JSON.

```
POST /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/groups/j7q5bL/
    chats HTTP/1.1
Host: api.att.com
authorization: Bearer abcdef12345678
accept: application/json
content-type: application/json
{
    "chat":
    {
        "body": "This is my message"
    }
}
```

4.17.7.4 Request – Example (Sending is-composing message in json)

This demostrates how to send is-composing notification using JSON.

```
POST /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/groups/j7q5bL/
    chats HTTP/1.1
Host: api.att.com
authorization: Bearer abcdef12345678
accept: application/json
content-type: application/json
{
    "chat":
    {
    "state": "active",
    "refresh": "120"
    }
}
```

4.17.8 Output Parameters

Parameter	Data Type	Req?	Brief description	Location
content-type	String	Yes	Specifies the type of content of the body. The acceptable values for this parameter are: • application/json • application/xml	Header
location	String	Yes	Specifies the location which contains identifier of the sent message.	Header

4.17.8.1 Response – Example (Send text message in XML)

This shows the response to sent chat text message in XML format.

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Date: Tue, 04 Dec 2013 02:51:59 GMT
location:/RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/groups/j7q5bL/
    chats/4ff7a06-762e-4d1a-abc5
```

4.17.8.2 Response – Example (Send media message in json)

This shows the response to sent media message in JSON format

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 100
Date: Tue, 04 Dec 2013 02:51:59 GMT
location:/RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/groups/j7q5bL/
    chats/4ff7a06-762e-4d1a-abc5
```

4.17.9 Service Exceptions

MessageId	Text	Variables	HTTP Status Code
SVC0001	A service error has occurred. Error code is <error_explanation>	error_explanation : <content_here>	400
SVC0002	Invalid input value for Message part <part_name>	part_name : name of the input parameter that resulted in the error.	400
SVC0003	Invalid input value for Message part <part_name>, valid values are <part_values>	part_name : name of the input parameter that resulted in the error. part_value : value of input parameter that was found to be in error.	400
SVC0004	No valid addresses provided in the Message part <part_name>	part_name : name of the input parameter that resulted in the error.	400
SVC8505	Media is larger than max <media size>. Check <body> parameter or media size or File too large	media size : Allowed content size. body : MIME payload sent as request in body.	413

4.17.10 Policy Exceptions

MessageId	Text	Variables	HTTP Status Code
POL0001	A policy error occurred. For example, rate limit error, authentication and authorization error.	N/A	401, 403
POL0002	Privacy verification failed for address <address>, request is refused	address : <content_here>	403
POL0003	Too many addresses specified in Message part	N/A	403
POL1009	User has not been provisioned for %1	1% : System that has not been provisioned.	403

4.18 **Operation:** Get Media From Group Chat

4.18.1 Functional Behavior

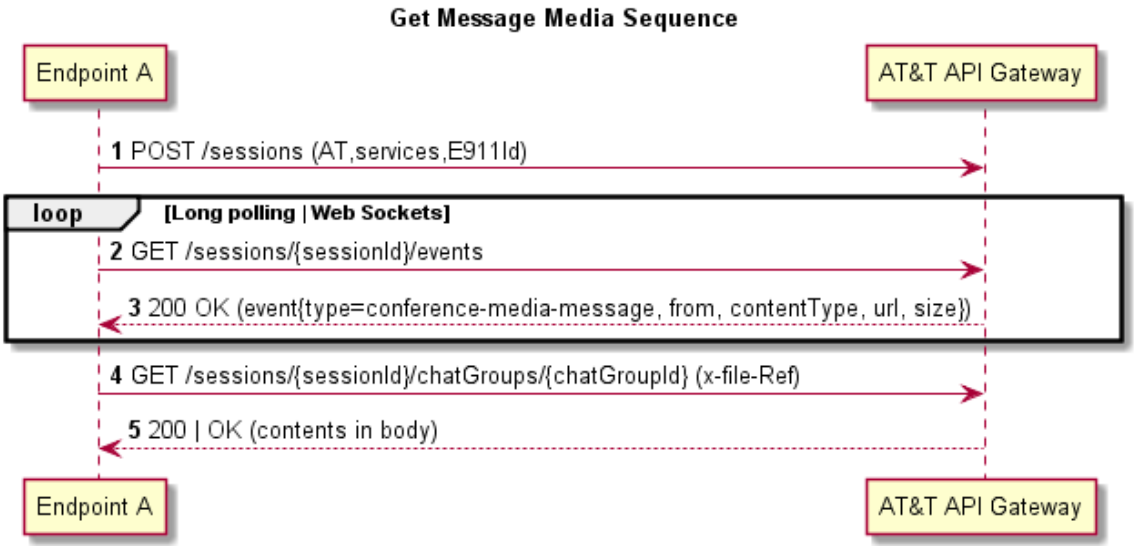
The Get Media From WebRTC Group Chat method fetches the binary content of the media message. The details of URI from where to get media is received in the event named **conference-mediamessage** received by the endpoint earlier on event channel.

Note: The end

point must be registered with AT&T Wireless network before fetching the media associated with the media message.

Note: The endpoint also must establish an event channel to receive notification of possible failures from the AT&T Wireless network.

4.18.2 Call flow



4.18.3 Version Impact Summary

Service Version	Major or Minor Impact	Changes Introduced by this Version
1	Major	Initial Release

4.18.4 Authentication and Authorization

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description
OAuth authorization_code	Yes	WEBRTC	1. Redirect the user to AT&T authorization page to capture user consent. 2. Obtain OAuth access token using the OAuth authorization code issued by the API Gateway along with the App Key and App Secret by making a Get Access Token request with GrantType=authorization_code. (a) Supports ICMN case 3. 4. (a)
OAuth client_credentials	No	WEBRTC	1. Obtain OAuth access token with the App Key and App Secret by making a Get Access Token request. (a) Supports VTN case (b) Supports no-TN case 2. (a) (b)

4.18.5 Representation Formats

Direction	Supported Respresentation Formats
Request	XML, JSON, URLENCODED
Response	XML, JSON, URLENCODED

4.18.6 Input Parameters

Parameter	Data Type	Req?	Brief description	Location
accept	String	No	Specifies the format of the body of the response. The acceptable values for this parameter are: • */* The default value is application/json. Note: For this method, this parameter specifies how the entity should be represented in case of an error. This parameter is for setting the format of an error message. If there is no error, then the representation will be the form of the actual content. Per rfc2616: ”If no Accept header field is present, then it is assumed that the client accepts all media types.” By default our services return application/json. The normal Accept header processing rules shall be followed according to rfc2616. *Note*: If there is no entity body in a normal successful response, this parameter is still needed to specify the format in the case of an error response message.	Header

Parameter	Data Type	Req?	Brief description	Location
authorization	String	Yes	Specifies the authorization type and token. The acceptable format for this parameter is the phrase ”Bearer OAuth Token” followed by a space ( ) and an OAuth access token. If this parameter value is missing from the header, then the API Gateway returns a message with an HTTP status code of 400 Invalid Request. If the OAuth access token is not valid, then the API Gateway returns an HTTP status code of 401 Unauthorized with a WWW-Authenticate HTTP header.	Header
content-length	Integer	No	Specifies the length of the content in octets. This parameter is only required for a non-streaming request.	Header
content-type	String	Yes	Specifies the type of content of the body. The acceptable values for this parameter are: • */*	Header
transfer-encoding	String	Conditional	Specifies the encodingof the message. The only acceptable values for this parameter is: • chunked This parameter is only required for a streaming request.	Header
fileRef	String	Yes	Specifies the reference number for the content file.	Query Parameter

4.18.6.1 Request – Example (Get media message content)

This demostrates how to get binary data of the media message using GET

```
GET /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/groups/43235321/
    chats?fileRef=12345 HTTP/1.1
Host: api.att.com
authorization: Bearer abcdef12345678
accept: */*
content-type: */*
```

4.18.7 Output Parameters

Parameter	Data Type	Req?	Brief description	Location
content-type	String	Yes	Specifies the type of content of the body. The acceptable values for this parameter are: • */*	Header

4.18.7.1 Response – Example (Respone get media message content)

This shows the response to get media content

```
HTTP/1.1 200 OK
Content-Type: */*
Content-Length: 624
content-type: */*
Date: Thu, 04 Jun 2010 02:51:59 GMT

binary data.....
00105e0 e6b0 343b 9c74 0804 e7bc 0804 e7d5 0804
00105f0 e7e4 0804 e6b0 0804 e7f0 0804 e7ff 0804
0010600 e80b 0804 e81a 0804 e6b0 0804 e6b0 0804
```

4.18.8 Service Exceptions

MessageId	Text	Variables	HTTP Status Code
SVC0001	A service error has occurred. Error code is <error_explanation>	error_explanation : <content_here>	400
SVC0002	Invalid input value for Message part <part_name>	part_name : name of the input parameter that resulted in the error.	400
SVC0003	Invalid input value for Message part <part_name>, valid values are <part_values>	part_name : name of the input parameter that resulted in the error. part_value : value of input parameter that was found to be in error.	400

MessageId	Text	Variables	HTTP Status Code
SVC0004	No valid addresses provided in the Message part <part_name>	part_name : name of the input parameter that resulted in the error.	400

4.18.9 Policy Exceptions

MessageId	Text	Variables	HTTP Status Code
POL0001	A policy error occurred. For example, rate limit error, authentication and authorization error.	N/A	401, 403
POL0002	Privacy verification failed for address <address>, request is refused	address : <content_here>	403
POL0003	Too many addresses specified in Message part	N/A	403
POL1009	User has not been provisioned for %1	1% : System that hasn't been provisioned.	403

4.19 Operation: Add Participant(s) To Group Chat

4.19.1 Functional Behavior

The Add Participant To Chat Group method is used to add one or more participants to a chat group.

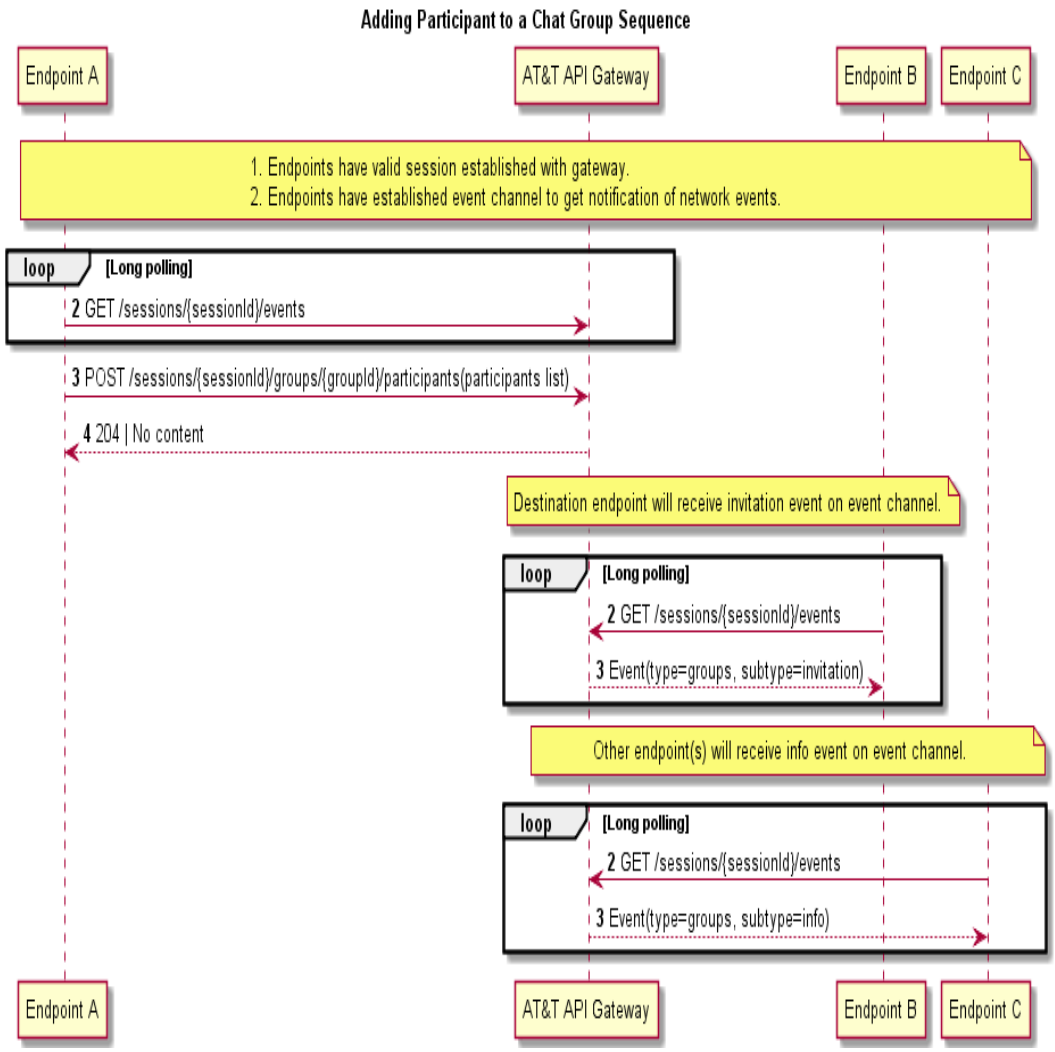
Note: The endpoint must to be registered with AT&T network before adding the participant(s) to the chat group.

Note: The endpoint also must to establish an event channel to receive notification of possible failures and other status updates from the AT&T network.

Note: The endpoint must to be a connected and must be a participant in a conference for this request to succeed.

Note: A network configurable parameter limits the maximum number of participants that are able to be added to the group.

4.19.2 Call flow



4.19.3 Version Impact Summary

Service Version	Major or Minor Impact	Changes Introduced by this Version
1	Major	Initial Release

4.19.4 Authentication and Authorization

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description
authorization_code	Yes	RTC	1. Obtain the authorization code by passing App Key and App Secret along with scope. This redirects the user to AT&T authorization page to capture user consent. 2. Obtain OAuth access token using the OAuth authorization code along with the App Key and App Secret by making a Get Access Token request with grant_type=authorization_code. • Supports ICMN case
client_credentials	No	RTC	1. Obtain OAuth access token with the App Key and App Secret along with the scope by making a Get Access Token request with grant_type=client_credentials • Supports VTN case • Supports no-TN case

4.19.5 Representation Formats

4.19.6 Representation Formats

Direction	Supported Respresentation Formats
Request	XML, JSON,URLENCODED
Response	XML, JSON,URLENCODED

4.19.7 Input Parameters

Parameter	Data Type	Req?	Brief description	Location
accept	String	No	Specifies the format of the body of the response. The acceptable values for this parameter are: • application/json • application/xml • application/x-www-form-urlencoded The default value is application/json. Per rfc2616: ”If no Accept header field is present, then it is assumed that the client accepts all media types.” By default our services return application/json. The normal Accept header processing rules shall be followed according to rfc2616. *Note*: If there is no entity body in a normal successful response, this parameter is still needed to specify the format in the case of an error response message.	Header
authorization	String	Yes	Specifies the authorization type and token. The acceptable format for this parameter is the phrase ”Bearer OAuth Token” followed by a space ( ) and an OAuth access token. If this parameter value is missing from the header, then the API Gateway returns a message with an HTTP status code of 400 Invalid Request. If the OAuth access token is not valid, then the API Gateway returns an HTTP status code of 401 Unauthorized with a WWW-Authenticate HTTP header.	Header

Parameter	Data Type	Req?	Brief description	Location
content-length	Integer	No	Specifies the length of the content in octets. This parameter is only required for a non-streaming request.	Header
content-type	String	Yes	Specifies the type of content of the body. The acceptable values for this parameter are: • application/xml • application/json	Header
transfer-encoding	String	Conditional	Specifies the encodingof the message. The only acceptable values for this parameter is: • chunked This parameter is only required for a streaming request.	Header
participants	List of Strings	Yes	Specifies the details for one of more participants who need to added to the group.	Body

4.19.7.1 Request – Example (Add participant(s) to the chat group - XML)

This demostrates how to add participant(s) to a chat group using xml.

```
POST /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/groups/j7q5bL/
    participants HTTP/1.1
Host: api.att.com
authorization: Bearer abcdef12345678
accept: application/xml
content-type: application/xml
<?xml version="1.0" encoding="UTF-8" ?>
<participants>
        <participant>tel:+1234567890</participant>
        <participant>tel:+1234567891</participant>
</participants>
```

4.19.7.2 Request – Example (Add participant(s) to the chat group - JSON)

This demostrates how to add participant(s) to a chat group using JSON.

```
POST /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/groups/j7q5bL/
    participants HTTP/1.1
```

```
Host: api.att.com
authorization: Bearer 38C2399A23999
accept: application/json
content-length: 300
content-type: application/json
{
    "participants": ["tel:+1234567890","tel:+1234567891"]
}
```

4.19.8 Output Parameters

Parameter	Data Type	Req?	Brief description	Location
content-type	String	Yes	Specifies the type of content of the body. The acceptable values for this parameter are: • application/xml • application/json	Header

4.19.8.1 Response – Example (Join a onging group conversation in XML)

This shows the response joining group in xml.

```
HTTP/1.1 204 No Content
Content-Type: application/xml
Content-Length: 100
Date: Tue, 04 Dec 2013 02:51:59 GMT
```

4.19.9 Service Exceptions

MessageId	Text	Variables	Parent HTTP Code
SVC0001	A service error has occurred. Error code is <error_explanation>	error_explanation : <content_here>	400

MessageId	Text	Variables	Parent HTTP Code
SVC0002	Invalid input value for Message part <part_name>	part_name : name of the input parameter that resulted in the error.	400
SVC0003	Invalid input value for Message part <part_name>, valid values are <part_values>	part_name : name of the input parameter that resulted in the error. part_value : value of input parameter that was found to be in error.	400
SVC0004	No valid addresses provided in the Message part <part_name>	part_name : name of the input parameter that resulted in the error.	400
SVC8504	Allowed number of participants %1 exceeded.	%1 : Maximum allowed participants in a group chat.	400

4.19.10 Policy Exceptions

MessageId	Text	Variables	Parent HTTP Code
POL0001	A policy error occurred. For example, rate limit error, authentication and authorization error.	N/A	401, 403
POL0002	Privacy verification failed for address <address>, request is refused	address : <content_here>	403
POL0003	Too many addresses specified in Message part	N/A	403
POL1009	User has not been provisioned for %1	1% : System that has not been provisioned.	403

4.20 **Operation:** Exit Group Chat

4.20.1 Functional Behavior

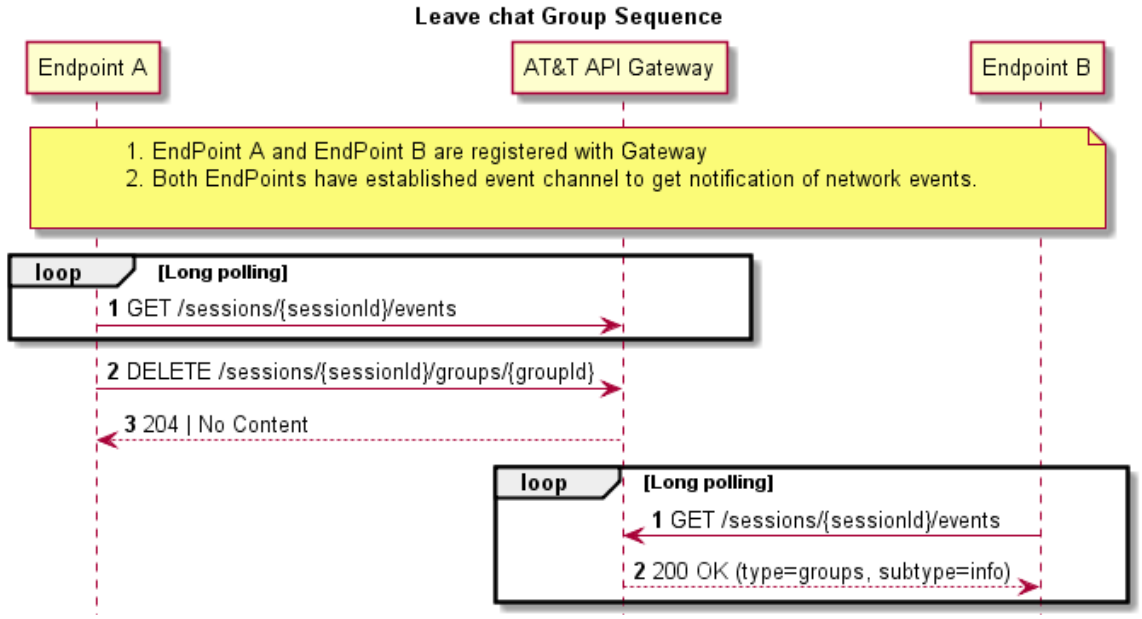
The **leave group** method is used to leave the chat group.

1. End point needs to be registered with AT&T network and should be participant in the chat group before using leave chat group method.
2. End point also needs to establish an event channel to receive notification of possible failures and other status updates from the AT&T network.

Note:

1. The end points receives **info** events when users are joining or leaving the group.
2. A network configurable parameter limits the maximum number of participants that can be added to the group. Refer to parameters in appendix for more detais.

4.20.2 Call flow



4.20.3 Version Impact Summary

Service Version	Major or Minor Impact	Changes Introduced by this Version
1	Major	Initial Release

4.20.4 Authentication and Authorization

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description
authorization_code	Yes	RTC	1. Obtain the authorization code by passing App Key and App Secret along with scope. This redirects the user to AT&T authorization page to capture user consent. 2. Obtain OAuth access token using the OAuth authorization code along with the App Key and App Secret by making a Get Access Token request with grant_type=authorization_code. • Supports ICMN case
client_credentials	No	RTC	1. Obtain OAuth access token with the App Key and App Secret along with the scope by making a Get Access Token request with grant_type=client_credentials • Supports VTN case • Supports no-TN case

4.20.5 Representation Formats

4.20.6 Representation Formats

Direction	Supported Respresentation Formats
Request	XML, JSON,URLENCODED
Response	XML, JSON,URLENCODED

4.20.7 Input Parameters

Parameter	Data Type	Req?	Brief description	Location
accept	String	No	Specifies the format of the body of the response. The acceptable values for this parameter are: • application/json • application/xml • application/x-www-form-urlencoded The default value is application/json. Per rfc2616: "If no Accept header field is present, then it is assumed that the client accepts all media types." By default our services return application/json. The normal Accept header processing rules shall be followed according to rfc2616. *Note*: If there is no entity body in a normal successful response, this parameter is still needed to specify the format in the case of an error response message.	Header

Parameter	Data Type	Req?	Brief description	Location
authorization	String	Yes	Specifies the authorization type and token. The acceptable format for this parameter is the phrase ”Bearer OAuth Token” followed by a space ( ) and an OAuth access token. If this parameter value is missing from the header, then the API Gateway returns a message with an HTTP status code of 400 Invalid Request. If the OAuth access token is not valid, then the API Gateway returns an HTTP status code of 401 Unauthorized with a WWW-Authenticate HTTP header.	Header
content-length	Integer	No	Specifies the length of the content in octets. This parameter is only required for a non-streaming request.	Header
content-type	String	Yes	Specifies the type of content of the body. The acceptable values for this parameter are: • application/xml • application/json	Header
transfer-encoding	String	Conditional	Specifies the encoding of the message. The only acceptable values for this parameter is: • chunked This parameter is only required for a streaming request.	Header
groupId	String	Yes	Specifies the group identifier to which the user was joined or created earlier.	URI Path

4.20.7.1 Request – Example (Leave a chat group)

This demostrates how to leave the chat group using DELETE method.

```
DELETE /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/groups/j7q5bL
    HTTP/1.1
Host: api.att.com
authorization: Bearer abcdef12345678
accept: application/xml
content-type: application/xml
```

4.20.8 Output Parameters

Parameter	Data Type	Req?	Brief description	Location
content-type	String	Yes	Specifies the type of content of the body. The acceptable values for this parameter are: • application/json • application/xml	Header

4.20.8.1 Response – Example (Leave a chat group in xml)

This shows the response to a leave chat group in XML format.

```
HTTP/1.1 204 OK
Content-Type: application/xml
Content-Length: 100
Date: Tue, 04 Dec 2013 02:51:59 GMT
```

4.20.9 Service Exceptions

MessageId	Text	Variables	Parent HTTP Code
SVC0001	A service error has occurred. Error code is <Error Explanation>	Error Explanation : <content here>	400
SVC0002	Invalid input value for Message part <part name>	part name : name of the input parameter that resulted in the error	400
SVC0003	Invalid input value for Message part <part name>, valid values are <part values>	part value : value of input parameter that was found to be in error.	400
SVC0004	No valid addresses provided in the Message part <part name>	part name : name of the input parameter that resulted in the error	400

4.20.10 Policy Exceptions

MessageId	Text	Variables	Parent HTTP Code
POL0001	A policy error occurred. For example, rate limit error, authentication and authorization error.	N/A	401,403
POL0002	Privacy verification failed for address <address>, request is refused	address : <content here>	403
POL0003	Too many addresses specified in Message part	N/A	403
POL1009	User has not been provisioned for %1	System that hasn't been provisioned	403

4.21 Operation: Send Message To One-to-One Chat

4.21.1 Functional Behavior

Send one-to-one chat message operation is used for one-to-one chat with peer.

Following use cases can be achieved using this operation.

1. Sending **text** messsage.
2. Sending **media** message.
3. Sending **is-composing** notification.

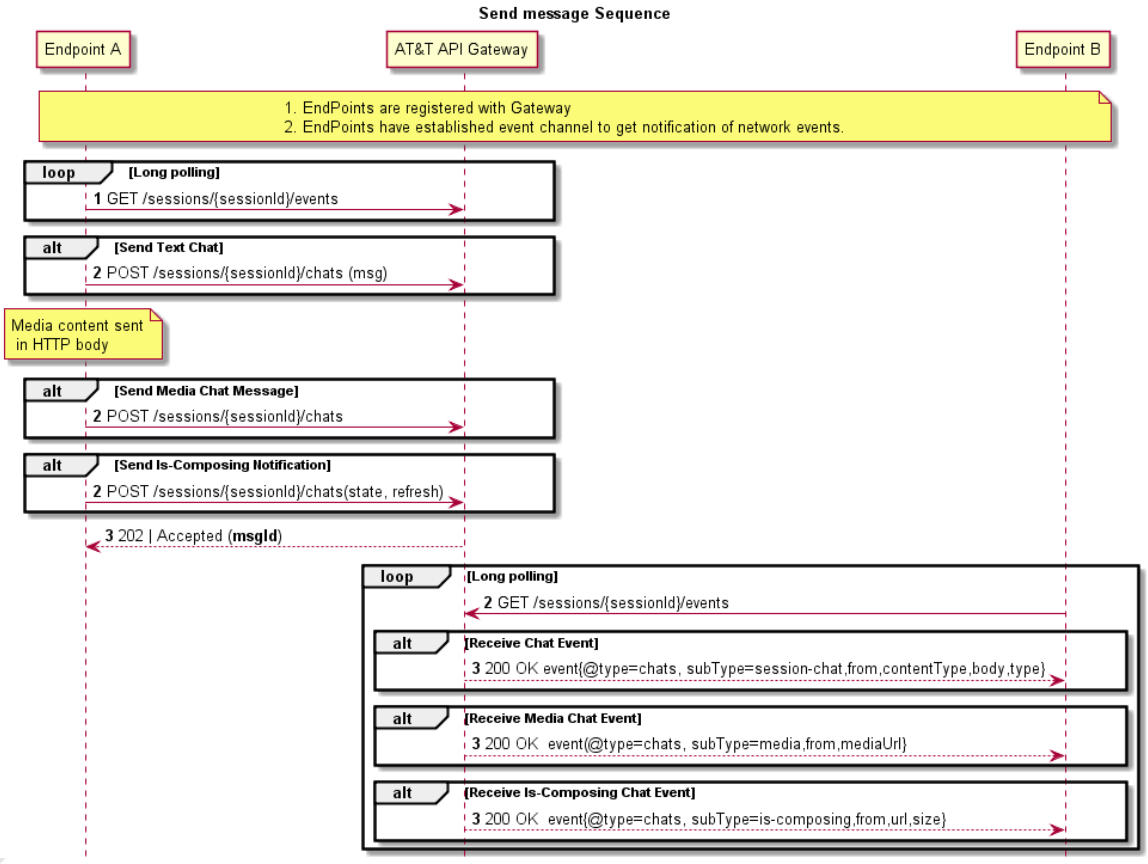
Note:

1. End point needs to be registered with AT&T network before sending the chat message to the peer.

2. End point also needs to establish an event channel to receive notification of possible failures from the AT&T network.

3. When the client switches destination URI between TEL URI and SIP URI, mapping of the message to previous exchanged chats might fail.

4.21.2 Call flow



4.21.3 Version Impact Summary

Service Version	Major or Minor Impact	Changes Introduced by this Version
1	Major	Initial Release

4.21.4 Authentication and Authorization

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description
authorization_code	Yes	RTC	1. Obtain the authorization code by passing App Key and App Secret along with scope. This redirects the user to AT&T authorization page to capture user consent. 2. Obtain OAuth access token using the OAuth authorization code along with the App Key and App Secret by making a Get Access Token request with grant_type=authorization_code. • Supports ICMN case
client_credentials	No	RTC	1. Obtain OAuth access token with the App Key and App Secret along with the scope by making a Get Access Token request with grant_type=client_credentials • Supports VTN case • Supports no-TN case

4.21.5 Representation Formats

4.21.6 Representation Formats

Direction	Supported Respresentation Formats
Request	XML, JSON,URLENCODED
Response	XML, JSON,URLENCODED

4.21.7 Input Parameters

Parameter	Data Type	Req?	Brief description	Location
accept	String	No	Specifies the format of the body of the response. The acceptable values for this parameter are: • application/json • application/xml • application/x-www-form-urlencoded The default value is application/json. Per rfc2616: ”If no Accept header field is present, then it is assumed that the client accepts all media types.” By default our services return application/json. The normal Accept header processing rules shall be followed according to rfc2616. *Note*: If there is no entity body in a normal successful response, this parameter is still needed to specify the format in the case of an error response message.	Header
authorization	String	Yes	Specifies the authorization type and token. The acceptable format for this parameter is the phrase ”Bearer OAuth Token” followed by a space ( ) and an OAuth access token. If this parameter value is missing from the header, then the API Gateway returns a message with an HTTP status code of 400 Invalid Request. If the OAuth access token is not valid, then the API Gateway returns an HTTP status code of 401 Unauthorized with a WWW-Authenticate HTTP header.	Header

Parameter	Data Type	Req?	Brief description	Location
content-length	Integer	No	Specifies the length of the content in octets. This parameter is only required for a non-streaming request.	Header
content-type	String	Yes	Specifies the type of content of the body. The acceptable values for this parameter are: • application/xml • application/json • multipart/mixed • image/jpeg	Header
transfer-encoding	String	Conditional	Specifies the encodingof the message. The only acceptable values for this parameter is: • chunked This parameter is only required for a streaming request.	Header
chat	Object	Yes	Contains the destination details of chat as text in case it is text chat or as binary data in the body in case it is media chat.	Body

Structure of chat Object

Parameter	Data Type	Req?	Brief description
to	String	Yes	Specifies the SIP URI or TEL URI of the destination.
body	String	Conditional	Specifies the text message being sent. Note: This parameter is required only in case of sending a text message.
contentType	String	No	Specifies the file content type. The MIME content type of the message. For a text message, this parameter is set to text/plain. The default value is text/plain. Note: This parameter is required only in case of sending a text message.

Parameter	Data Type	Req?	Brief description
type	String	Conditional	Specifies the type of message. The acceptable values for this field are: • chat: Send the message in pager mode. • session-chat : Send the message in a session. The default value session-chat. Note: This parameter is required only in case of sending a text message
state	String	Conditional	Specifies the state of user. The acceptable values for this field are: • active : The user is currently composing a message • idle : The user is currently idle or stopped typing. Note: This parameter is required only in case of sending an is-composing message.
refresh	Integer	Conditional	Specifies the timer, in seconds, that should be associated to the is-composing icon. If this parameter is set to 120, then it means the receiver displays an is-composing icon for 120 seconds. The is-composing icon should be removed when the timer expires, a chat message is received, or an is-composing message with the state set to idle is received. Note: This parameter is required only in case of sending an is-composing messge

4.21.7.1 Request – Example (Send a one-to-one chat text message)

This demostrates how to send a one to one text chat using POST, and specifying the response to be in XML.

```
POST /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/chats HTTP/1.1
Host: api.att.com
authorization: Bearer abcdef12345678
accept: application/xml
```

```
content-type: application/xml
<?xml version="1.0" encoding="UTF-8" ?>
<chat>
    <to>tel:+1234567890</to>
    <body>This is my chat </body>
    <contentType>text/plain</contentType>
    <type>session-message</type>
</chat>
```

4.21.7.2 Request – Example (Send a one-to-onemedia message)

This demostrates how to send a media chat message . The response format will be in JSON.

```
POST /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/chats HTTP/1.1
Host: api.att.com
authorization: Bearer 38C2399A23999
accept: application/json
content-length: 5655
Content-Type: multipart/mixed;boundary=webrtc
--webrtc
Content-Type:application/json
{
    "chat":
    {
        "to": "tel:+1234567890"
    }
}
--webrtc
Content-Type: image/jpeg
00105e0 e6b0 343b 9c74 0804 e7bc 0804 e7d5 0804
00105f0 e7e4 0804 e6b0 0804 e7f0 0804 e7ff 0804
0010600 e80b 0804 e81a 0804 e6b0 0804 e6b0 0804
--webrtc--
```

4.21.7.3 Request – Example (Send a one-to-one text chat in JSON)

This demostrates how to send text message using JSON.

```
POST /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/chats HTTP/1.1
Host: api.att.com
authorization: Bearer abcdef12345678
accept: application/json
content-type: application/json
{
```

```
    "chat":
    {

        "to": "tel:+1234567890",
        "body": "This is my chat",
        "contentType": "text/plain",
        "type": "session-chat"
    }
}
```

4.21.8 Output Parameters

Parameter	Data Type	Req?	Brief description	Location
content-type	String	Yes	Specifies the type of content of the body. The acceptable values for this parameter are: • application/json • application/xml • application/x-www-form-urlencoded	Header
location	String	Yes	Specifies the identifier of the sent chat message.	Header

4.21.8.1 Response – Example (Send a one-to-one text chat in XML)

This shows the response to sending text message in XML format.

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Date: Tue, 04 Dec 2013 02:51:59 GMT
location: /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/chats/74ff7a06
    -762e-4d1a-abc5-aafc4c5accb1
```

4.21.8.2 Response – Example (Send a one-to-one media chat in json)

This shows the response to sending media chat in JSON format

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 100
Date: Tue, 04 Dec 2013 02:51:59 GMT
location: /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/chats/74ff7a06
    -762e-4d1a-abc5-aafc4c5accb1
```

4.21.9 Service Exceptions

MessageId	Text	Variables	HTTP Status Code
SVC0001	A service error has occurred. Error code is <error_explanation>	error_explanation : <content _here>	400
SVC0002	Invalid input value for Message part <part_name>	part_name : name of the input parameter that resulted in the error.	400
SVC0003	Invalid input value for Message part <part_name>, valid values are <part_values>	part_name : name of the input parameter that resulted in the error. part_value : value of input parameter that was found to be in error.	400
SVC0004	No valid addresses provided in the Message part <part_name>	part_name : name of the input parameter that resulted in the error.	400
SVC8505	Media is larger than max <media size>. Check <body> parameter or media size or File too large	media size : Allowed content size. body : MIME payload sent as request in body.	413

4.21.10 Policy Exceptions

MessageId	Text	Variables	HTTP Status Code

MessageId	Text	Variables	HTTP Status Code
POL0001	A policy error occurred. For example, rate limit error, authentication and authorization error.	N/A	401, 403
POL0002	Privacy verification failed for address <address>, request is refused	address : <content_here>	403
POL0003	Too many addresses specified in Message part	N/A	403
POL1009	User has not been provisioned for %1	1% : System that has not been provisioned.	403

4.22 Operation: Get Media From One-to-One Chat

4.22.1 Functional Behavior

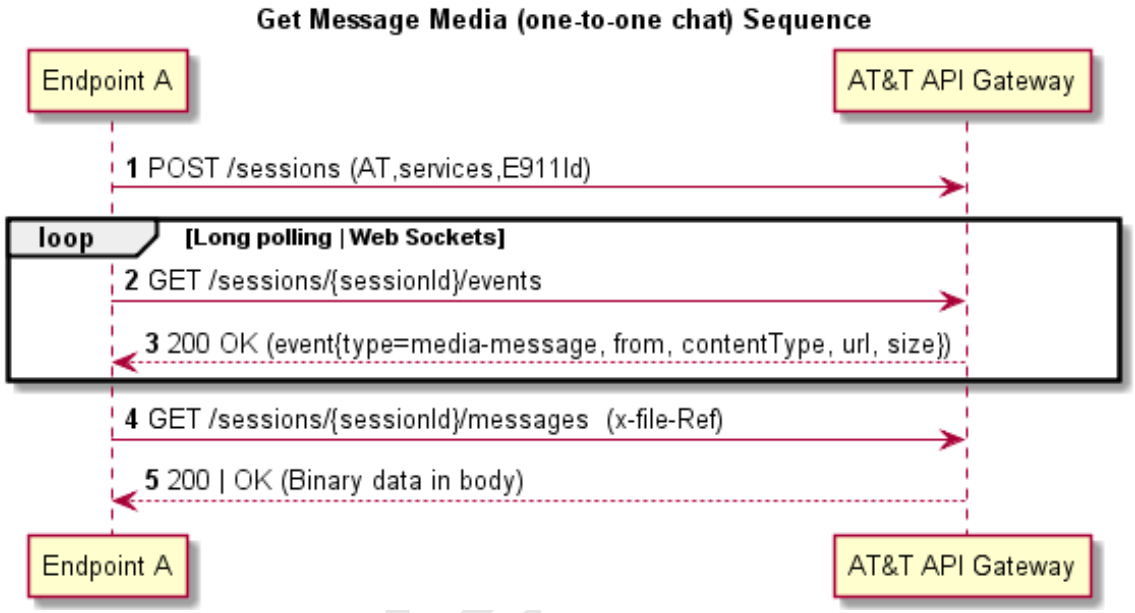
The Get Media From WebRTC Single Chat method fetches the binary content of the media message. The details of the URI from where to get media is received in the event named media-message received by the endpoint earlier on event channel.

Note: The end

point must be registered with AT&T Wireless network before fetching the media associated the the media message.

Note: The endpoint also must establish an event channel to receive notification of possible failures from the AT&T Wireless network.

4.22.2 Call flow



4.22.3 Version Impact Summary

Service Version	Major or Minor Impact	Changes Introduced by this Version
1	Major	Initial Release

4.22.4 Authentication and Authorization

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description
OAuth authorization_code	Yes	WEBRTC	1. Redirect the user to AT&T authorization page to capture user consent. 2. Obtain OAuth access token using the OAuth authorization code issued by the API Gateway along with the App Key and App Secret by making a Get Access Token request with GrantType=authorization_code. (a) Supports ICMN case 3. 4. (a)
OAuth client_credentials	No	WEBRTC	1. Obtain OAuth access token with the App Key and App Secret by making a Get Access Token request. (a) Supports VTN case (b) Supports no-TN case 2. (a) (b)

4.22.5 Representation Formats

Direction	Supported Respresentation Formats
Request	XML, JSON, URLENCODED
Response	XML, JSON

4.22.6 Input Parameters

Parameter	Data Type	Req?	Brief description	Location
accept	String	No	Specifies the format of the body of the response. The acceptable values for this parameter are: • */* The default value is application/json. Note: For this method, this parameter specifies how the entity should be represented in case of an error. This parameter is for setting the format of an error message. If there is no error, then the representation will be the form of the actual content. Per rfc2616: ”If no Accept header field is present, then it is assumed that the client accepts all media types.” By default our services return application/json. The normal Accept header processing rules shall be followed according to rfc2616. *Note*: If there is no entity body in a normal successful response, this parameter is still needed to specify the format in the case of an error response message.	Header
authorization	String	Yes	Specifies the authorization type and token. The acceptable format for this parameter is the phrase ”Bearer OAuth Token” followed by a space ( ) and an OAuth access token. If this parameter value is missing from the header, then the API Gateway returns a message with an HTTP status code of 400 Invalid Request. If the OAuth access token is not valid, then the API Gateway returns an HTTP status code of 401 Unauthorized with a WWW-Authenticate HTTP header.	Header
content-length	Integer	No	Specifies the length of the content in octets. This parameter is only required for a non-streaming request.	Header

Parameter	Data Type	Req?	Brief description	Location
content-type	String	Yes	Specifies the type of content of the body. The acceptable values for this parameter are: • */*	Header
transfer-encoding	String	Conditional	Specifies the encodingof the message. The only acceptable values for this parameter is: • chunked This parameter is only required for a streaming request.	Header
fileRef	String	Yes	Specifies the identifier for fetching the message content.	URI Path

4.22.6.1 Request – Example (Get media message content)

This demostrates how to get binary data of the media message using GET

```
GET /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/chats?fileRef=1234
    HTTP/1.1
Host: api.att.com
authorization: Bearer abcdef12345678
accept: */*
content-type: */*
```

4.22.7 Output Parameters

Parameter	Data Type	Req?	Brief description	Location
content-type	String	Yes	Specifies the type of content of the body. The acceptable values for this parameter are: • */*	Header

4.22.7.1 Response – Example (Respone get media message content)

This shows the response to get media content

```
HTTP/1.1 200 OK
Content-Type: */*
Content-Length: 624
content-type: */*
Date: Thu, 04 Jun 2010 02:51:59 GMT

binary data ...
00105e0 e6b0 343b 9c74 0804 e7bc 0804 e7d5 0804
00105f0 e7e4 0804 e6b0 0804 e7f0 0804 e7ff 0804
0010600 e80b 0804 e81a 0804 e6b0 0804 e6b0 0804
```

4.22.8 Service Exceptions

MessageId	Text	Variables	HTTP Status Code
SVC0001	A service error has occurred. Error code is <error_explanation>	error_explanation : <content_here>	400
SVC0002	Invalid input value for Message part <part_name>	part_name : name of the input parameter that resulted in the error.	400
SVC0003	Invalid input value for Message part <part_name>, valid values are <part_values>	part_name : name of the input parameter that resulted in the error. part_value : value of input parameter that was found to be in error.	400
SVC0004	No valid addresses provided in the Message part <part_name>	part_name : name of the input parameter that resulted in the error.	400

4.22.9 Policy Exceptions

MessageId	Text	Variables	HTTP Status Code

MessageId	Text	Variables	HTTP Status Code
POL0001	A policy error occurred. For example, rate limit error, authentication and authorization error.	N/A	401, 403
POL0002	Privacy verification failed for address <address>, request is refused	address : <content_here>	403
POL0003	Too many addresses specified in Message part	N/A	403
POL1009	User has not been provisioned for %1	1% : System that has not been provisioned	403

4.23 Operation: Initiate File Transfer

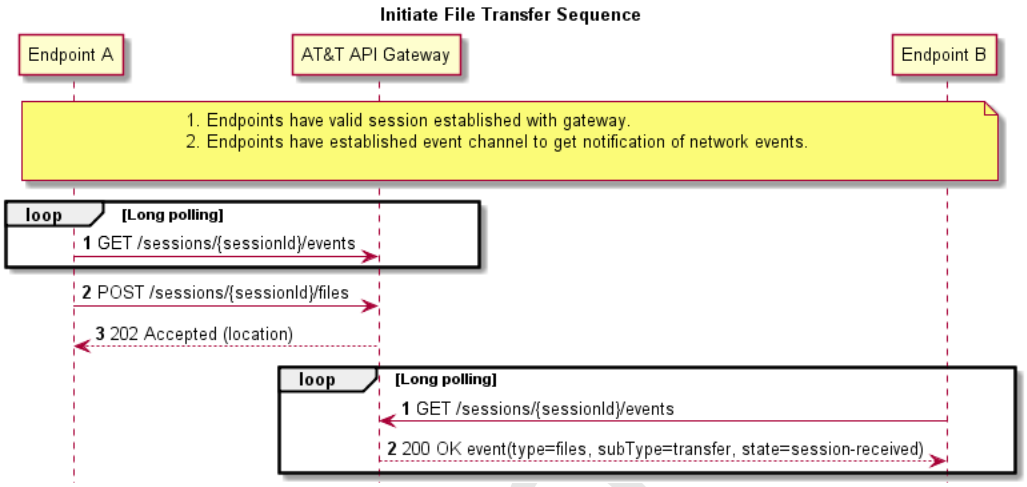
4.23.1 Functional Behavior

Initiate file transfer method initiates a file transfer session with the peer. The recipient receives a files event with status set to invitation-received. The invoking client receives a files event with status set to session-open indicating if the recipient accepted the file transfer or files event with status set to session-terminated indicating that the recipient ignored or rejected it.

Note: The endpoint must be registered with AT&T network before initiating the file transfer session with the peer.

Note: The endpoint also must establish an event channel to receive notification of possible failures from the AT&T network.

4.23.2 Call flow



4.23.3 Version Impact Summary

Service Version	Major or Minor Impact	Changes Introduced by this Version
1	Major	Initial Release

4.23.4 Authentication and Authorization

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description
authorization_code	Yes	RTC	1. Obtain the authorization code by passing App Key and App Secret along with scope. This redirects the user to AT&T authorization page to capture user consent. 2. Obtain OAuth access token using the OAuth authorization code along with the App Key and App Secret by making a Get Access Token request with grant_type=authorization_code. • Supports ICMN case
client_credentials	No	RTC	1. Obtain OAuth access token with the App Key and App Secret along with the scope by making a Get Access Token request with grant_type=client_credentials • Supports VTN case • Supports no-TN case

4.23.5 Representation Formats

4.23.6 Representation Formats

Direction	Supported Respresentation Formats
Request	XML, JSON,URLENCODED
Response	XML, JSON,URLENCODED

4.23.7 Input Parameters

Parameter	Data Type	Req?	Brief description	Location
accept	String	No	Specifies the format of the body of the response. The acceptable values for this parameter are: • application/json • application/xml • application/x-www-form-urlencoded The default value is application/json. Per rfc2616: ”If no Accept header field is present, then it is assumed that the client accepts all media types.” By default our services return application/json. The normal Accept header processing rules shall be followed according to rfc2616. *Note*: If there is no entity body in a normal successful response, this parameter is still needed to specify the format in the case of an error response message.	Header
authorization	String	Yes	Specifies the authorization type and token. The acceptable format for this parameter is the phrase ”Bearer OAuth Token” followed by a space ( ) and an OAuth access token. If this parameter value is missing from the header, then the API Gateway returns a message with an HTTP status code of 400 Invalid Request. If the OAuth access token is not valid, then the API Gateway returns an HTTP status code of 401 Unauthorized with a WWW-Authenticate HTTP header.	Header
content-length	Integer	No	Specifies the length of the content in octets. This parameter is only required for a non-streaming request.	Header

Parameter	Data Type	Req?	Brief description	Location
content-type	String	Yes	Specifies the type of content of the body. The acceptable values for this parameter are: • application/xml • application/json • image/jpeg	Header
transfer-encoding	String	Conditional	Specifies the encodingof the message. The only acceptable values for this parameter is: • chunked This parameter is only required for a streaming request.	Header
file	Object	Conditional	Contains the destination details in order to initiate file transfer. This object is specified only if file transfer is initiated. This object is not requiredfor accepting invitation and uploading data using HTTP PUT .	Body

Structure of file Object

Parameter	Data Type	Req?	Brief description
to	String	Yes	Specifies the SIP URI or TEL URI of the destination.
fileName	String	Yes	Specifies the name of the file that is sent.
filesize	String	Yes	Specifies the size of the file that is sent.
contentType	String	Yes	Specifies the MIME type of the file that is sent.
contentDisposition	String	Yes	Specifies how the message body is to be interpreted.

4.23.7.1 Request – Example (Initiate file transfer dialog with peer)

This demostrates how to initiate file transfer using xml.

```
POST /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/files HTTP/1.1
Host: api.att.com
authorization: Bearer abcdef12345678
```

```
accept: application/xml
content-type: application/xml
<?xml version="1.0" encoding="UTF-8" ?>
<file>
    <to>tel:+1234567890</to>
    <contentDisposition>attachment</contentDisposition>
    <contentType>image/jpeg</contentType>
    <fileName>dummyFilename_accept.txt</fileName>
    <fileSize>2345</fileSize>
</file>
```

4.23.7.2 Request – Example (Initiate file transfer dialog with peer using json)

This demostrates how to initiate file transfer using json payload.

```
POST /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/files HTTP/1.1
Host: api.att.com
authorization: Bearer abcdef12345678
accept: application/json
content-type: application/json
{
    "file":
        {
        "to": "tel:+1234567890",
        "contentDisposition": "attachment",
        "contentType": "image/jpeg",
        "fileName": "dummyFilename_accept.txt",
        "fileSize": "2345"
        }
}
```

4.23.8 Output Parameters

Parameter	Data Type	Req?	Brief description	Location
content-type	String	Yes	Specifies the type of content of the body. The acceptable values for this parameter are: • application/json • application/xml	Header

Parameter	Data Type	Req?	Brief description	Location
location	String	Yes	Specifies the location of the file transfer resource. Location contains **fileId** which uniquely identify file transfer resource in file transfer dialog.	Header
x-state	String	Yes	Specifies the state of file resource	Header

4.23.8.1 Response – Example (Respone initiate file transfer) This shows the response to initiate file transfer with peer.

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Date: Thu, 04 Jun 2010 02:51:59 GMT
location: /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/files
    /182332923
x-state:invitation-sent
```

4.23.9 Service Exceptions

MessageId	Text	Variables	HTTP Status Code
SVC0001	A service error has occurred. Error code is <error_explanation>	error_explanation : <content_here>	400
SVC0002	Invalid input value for Message part <part_name>	part_name : name of the input parameter that resulted in the error.	400
SVC0003	Invalid input value for Message part <part_name>, valid values are <part_values>	part_name : name of the input parameter that resulted in the error. part_value : value of input parameter that was found to be in error.	400
SVC0004	No valid addresses provided in the Message part <part_name>	part name : name of the input parameter that resulted in the error.	400

4.23.10 Policy Exceptions

MessageId	Text	Variables	HTTP Status Code
POL0001	A policy error occurred. For example, rate limit error, authentication and authorization error.	N/A	401, 403
POL0002	Privacy verification failed for address <address>, request is refused	address : <content_here>	403
POL0003	Too many addresses specified in Message part	N/A	403
POL1009	User has not been provisioned for %1	1% : System that has not been provisioned.	403

4.24 Operation: Get File Data

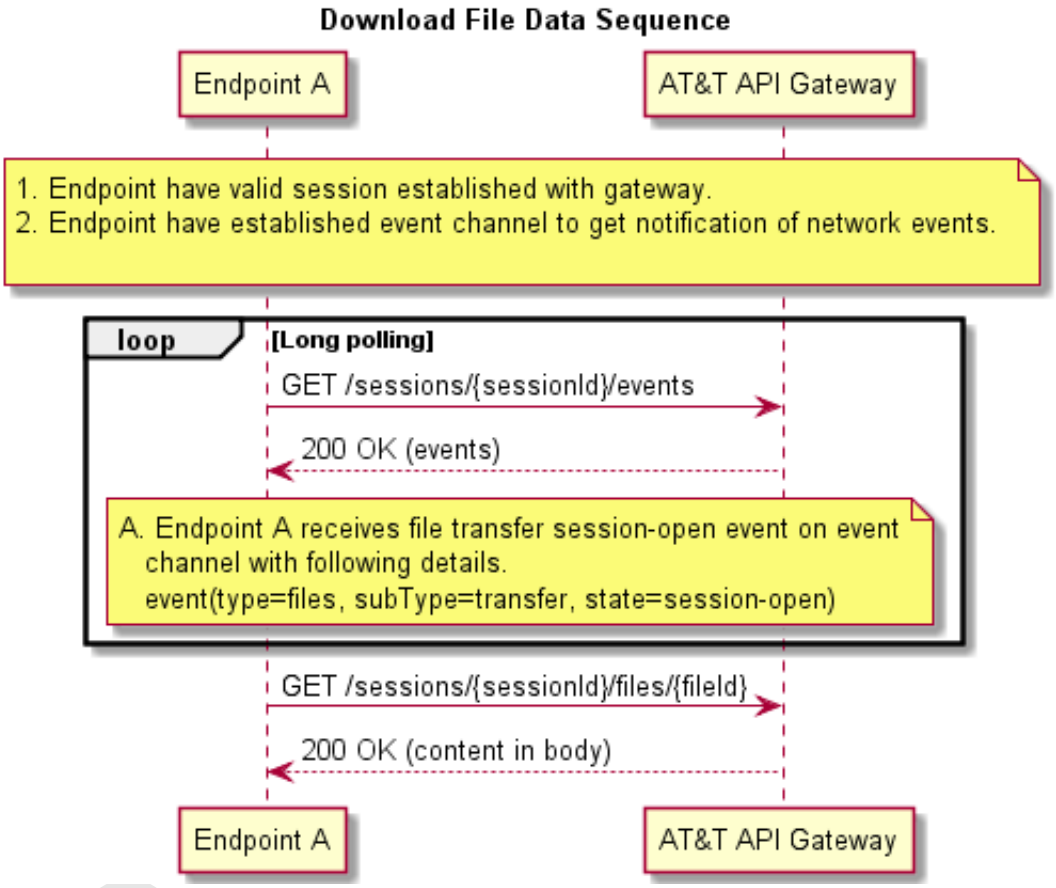
4.24.1 Functional Behavior

Download file data method is used to download file in the file dialog which was initiated using initiate file transfer method.

Note: The endpoint must be registered with the AT&T network before downloading data in a specified file transfer dialog.

Note: The endpoint also must establish an event channel in order to receive notification of possible failures from the AT&T network.

4.24.2 Call flow



4.24.3 Version Impact Summary

Service Version	Major or Minor Impact	Changes Introduced by this Version
1	Major	Initial Release

4.24.4 Authentication and Authorization

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description
authorization_code	Yes	RTC	1. Obtain the authorization code by passing App Key and App Secret along with scope. This redirects the user to AT&T authorization page to capture user consent. 2. Obtain OAuth access token using the OAuth authorization code along with the App Key and App Secret by making a Get Access Token request with grant_type=authorization_code. • Supports ICMN case
client_credentials	No	RTC	1. Obtain OAuth access token with the App Key and App Secret along with the scope by making a Get Access Token request with grant_type=client_credentials • Supports VTN case • Supports no-TN case

4.24.5 Representation Formats

4.24.6 Representation Formats

Direction	Supported Respresentation Formats
Request	XML, JSON,URLENCODED
Response	XML, JSON,URLENCODED

4.24.7 Input Parameters

Parameter	Data Type	Req?	Brief description	Location
accept	String	No	Specifies the format of the body of the response. The acceptable values for this parameter are: • application/json • application/xml • application/x-www-form-urlencoded The default value is application/json. Note: For this method, this parameter specifies how the entity should be represented in case of an error. This parameter is for setting the format of an error message. If there is no error, then the representation will be the form of the actual content. Per rfc2616: ”If no Accept header field is present, then it is assumed that the client accepts all media types.” By default our services return application/json. The normal Accept header processing rules shall be followed according to rfc2616. *Note*: If there is no entity body in a normal successful response, this parameter is still needed to specify the format in the case of an error response message.	Header
authorization	String	Yes	Specifies the authorization type and token. The acceptable format for this parameter is the phrase ”Bearer OAuth Token” followed by a space ( ) and an OAuth access token. If this parameter value is missing from the header, then the API Gateway returns a message with an HTTP status code of 400 Invalid Request. If the OAuth access token is not valid, then the API Gateway returns an HTTP status code of 401 Unauthorized with a WWW-Authenticate HTTP header.	Header

Parameter	Data Type	Req?	Brief description	Location
content-length	Integer	No	Specifies the length of the content in octets. This parameter is only required for a non-streaming request.	Header
content-type	String	Yes	Specifies the type of content of the body. The acceptable values for this parameter are: • application/xml • application/json • image/jpeg	Header
transfer-encoding	String	Conditional	Specifies the encodingof the message. The only acceptable values for this parameter is: • chunked This parameter is only required for a streaming request.	Header

4.24.7.1 Request – Example (Download file data)

This demostrates how to download file data.

```
GET /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/files/182332923
    HTTP/1.1
Host: api.att.com
authorization: Bearer abcdef12345678
accept: application/xml
content-type: image/jpeg
```

4.24.8 Output Parameters

Parameter	Data Type	Req?	Brief description	Location
content-type	String	Yes	Specifies the type of content of the body. The acceptable values for this parameter are: • application/json • application/xml	Header

4.24.8.1 Response – Example (Respone download file data)

This shows the response to download file data.

```
HTTP/1.1 200 OK
Content-Type: application/json
Date: Thu, 04 Jun 2010 02:51:59 GMT

00105e0 e6b0 343b 9c74 0804 e7bc 0804 e7d5 0804
00105f0 e7e4 0804 e6b0 0804 e7f0 0804 e7ff 0804
0010600 e80b 0804 e81a 0804 e6b0 0804 e6b0 0804
```

4.24.9 Service Exceptions

MessageId	Text	Variables	HTTP Status Code
SVC0001	A service error has occurred. Error code is <error_explanation>	error_explanation : <content_here>	400
SVC0002	Invalid input value for Message part <part_name>	part_name : name of the input parameter that resulted in the error.	400
SVC0003	Invalid input value for Message part <part_name>, valid values are <part_values>	part_name : name of the input parameter that resulted in the error. part_value : value of input parameter that was found to be in error.	400
SVC0004	No valid addresses provided in the Message part <part_name>	part_name : name of the input parameter that resulted in the error.	400

4.24.10 Policy Exceptions

MessageId	Text	Variables	HTTP Status Code

MessageId	Text	Variables	HTTP Status Code
POL0001	A policy error occurred. For example, rate limit error, authentication and authorization error.	N/A	401, 403
POL0002	Privacy verification failed for address <address>, request is refused	address : <content_here>	403
POL0003	Too many addresses specified in Message part	N/A	403
POL1009	User has not been provisioned for %1	1% : System that has not been provisioned.	403

4.25 Operation: Modify File Transfer

4.25.1 Functional Behavior

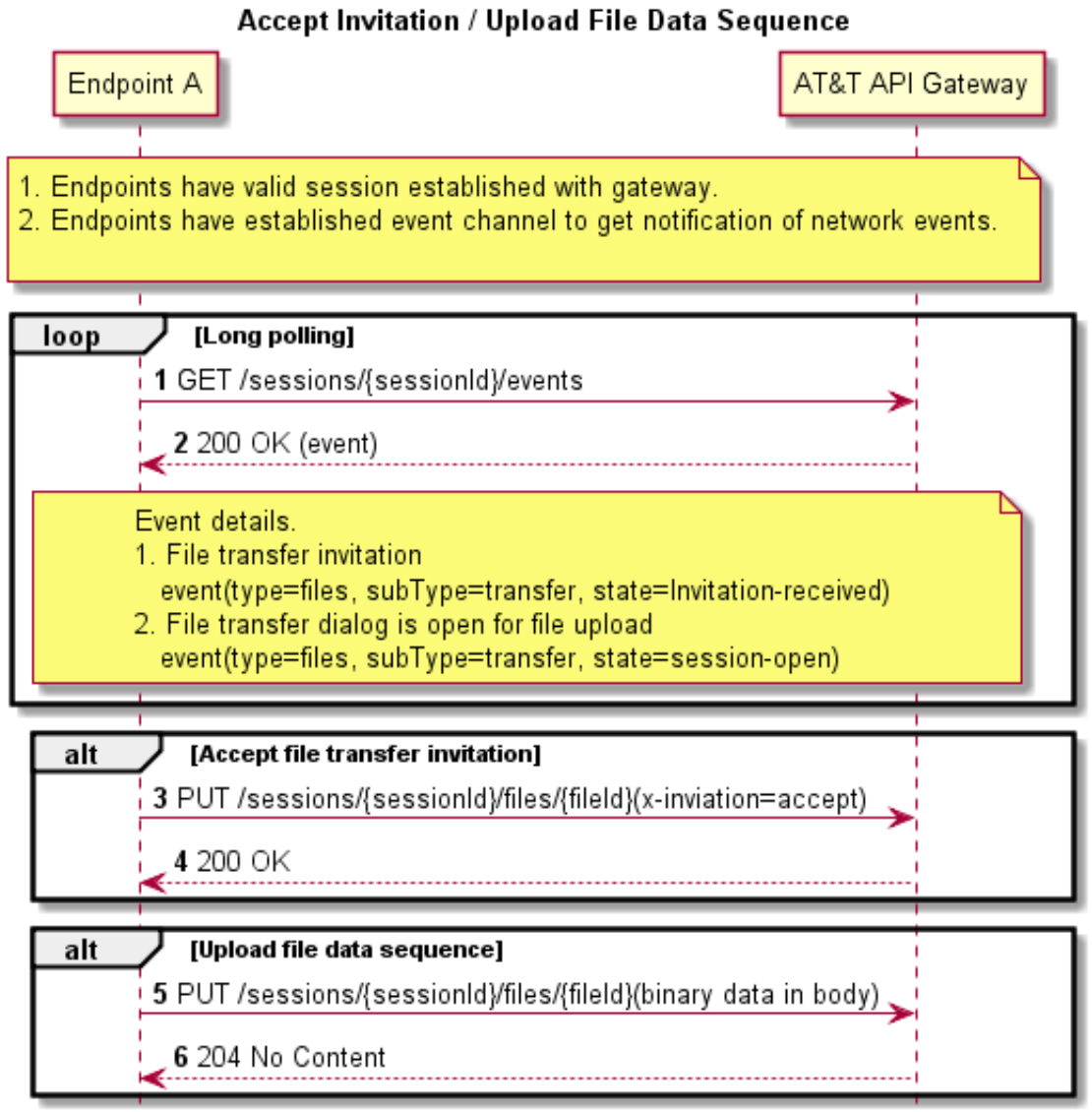
This operation is used to achive following use cases

1. Upload file data in the file transfer dialog which was initiated using initiate file transfer method.
2. Accept the file transfer invitation.

Note:

1. End point needs to be registered with AT&T network before uploading data in a specified file transfer dialog.
2. End point also needs to establish an event channel to receive notification of possible failures from the AT&T network.
3. The maximum size of a file which can be transfered can be limited in the network. Refer to global parameter section for details.
4. The End point must have a valid File Transfer ID and the file transfer needs to be in the state “**invitation-received**” for accept file transfer invitation request to succeed.

4.25.2 Call flow



4.25.3 Version Impact Summary

Service Version	Major or Minor Impact	Changes Introduced by this Version
1	Major	Initial Release

4.25.4 Authentication and Authorization

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description
authorization_code	Yes	RTC	1. Obtain the authorization code by passing App Key and App Secret along with scope. This redirects the user to AT&T authorization page to capture user consent. 2. Obtain OAuth access token using the OAuth authorization code along with the App Key and App Secret by making a Get Access Token request with grant_type=authorization_code. • Supports ICMN case
client_credentials	No	RTC	1. Obtain OAuth access token with the App Key and App Secret along with the scope by making a Get Access Token request with grant_type=client_credentials • Supports VTN case • Supports no-TN case

4.25.5 Representation Formats

4.25.6 Representation Formats

Direction	Supported Respresentation Formats
Request	XML, JSON,URLENCODED
Response	XML, JSON,URLENCODED

4.25.7 Input Parameters

Parameter	Data Type	Req?	Brief description	Location
accept	String	No	Specifies the format of the body of the response. The acceptable values for this parameter are: • application/json • application/xml • application/x-www-form-urlencoded The default value is application/json. Per rfc2616: "If no Accept header field is present, then it is assumed that the client accepts all media types." By default our services return application/json. The normal Accept header processing rules shall be followed according to rfc2616. *Note*: If there is no entity body in a normal successful response, this parameter is still needed to specify the format in the case of an error response message.	Header
authorization	String	Yes	Specifies the authorization type and token. The acceptable format for this parameter is the phrase "Bearer OAuth Token" followed by a space ( ) and an OAuth access token. If this parameter value is missing from the header, then the API Gateway returns a message with an HTTP status code of 400 Invalid Request. If the OAuth access token is not valid, then the API Gateway returns an HTTP status code of 401 Unauthorized with a WWW-Authenticate HTTP header.	Header
content-length	Integer	No	Specifies the length of the content in octets. This parameter is only required for a non-streaming request.	Header

Parameter	Data Type	Req?	Brief description	Location
content-type	String	Yes	Specifies the type of content of the body. The acceptable values for this parameter are: • application/xml • application/json • image/jpeg	Header
transfer-encoding	String	Conditional	Specifies the encodingof the message. The only acceptable values for this parameter is: • chunked This parameter is only required for a streaming request.	Header
x-invitation	String	Yes	Specifies the answer of the invitation. The only acceptable values for this parameter is: • accept : The file transfer is accepted.	Header

4.25.7.1 Request – Example (Upload file data - XML)

This demostrates how to upload file data.

```
PUT /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/files/182332923
    HTTP/1.1
Host: api.att.com
authorization: Bearer abcdef12345678
accept: application/xml
content-type: image/jpeg

Binary data ....
00105e0 e6b0 343b 9c74 0804 e7bc 0804 e7d5 0804
00105f0 e7e4 0804 e6b0 0804 e7f0 0804 e7ff 0804
0010600 e80b 0804 e81a 0804 e6b0 0804 e6b0 0804
```

4.25.7.2 Request – Example (Upload file data - JSON)

This demostrates how to upload file data.

```
PUT /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/files/182332923
    HTTP/1.1
Host: api.att.com
authorization: Bearer abcdef12345678
accept: application/json
content-type: image/jpeg

Binary data ....
00105e0 e6b0 343b 9c74 0804 e7bc 0804 e7d5 0804
00105f0 e7e4 0804 e6b0 0804 e7f0 0804 e7ff 0804
0010600 e80b 0804 e81a 0804 e6b0 0804 e6b0 0804
```

4.25.7.3 Request – Example (Accepting file transfer request - JSON)

This demostrates how to accept file transfer.

```
PUT /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/files/182332923
    HTTP/1.1
Host: api.att.com
authorization: Bearer abcdef12345678
accept: application/json
x-invitation: accept
```

4.25.7.4 Request – Example (Accepting file transfer request - XML)

This demostrates how to accept file transfer.

```
PUT /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/files/182332923
    HTTP/1.1
Host: api.att.com
authorization: Bearer abcdef12345678
accept: application/xml
x-invitation: accept
```

4.25.8 Output Parameters

Parameter	Data Type	Req?	Brief description	Location

Parameter	Data Type	Req?	Brief description	Location
content-type	String	Yes	Specifies the type of content of the body. The acceptable values for this parameter are: • application/json • application/xml	Header
x-state	String	Yes	Specifies the state after invitation. The only acceptable value for this parameter is: • accepted : The file transfer is accepted.	Header

4.25.8.1 Response – Example (Respone upload file data)

This shows the response to upload file data.

```
HTTP/1.1 204 No Content
Content-Type: application/xml
Date: Thu, 04 Jun 2010 02:51:59 GMT
```

4.25.8.2 Response – Example (Respone accept file transfer invitation)

This shows the response to accept file transfer request.

```
HTTP/1.1 200 OK
Content-Type: application/json
Date: Thu, 04 Jun 2010 02:51:59 GMT
x-state: accepted
```

4.25.9 Service Exceptions

MessageId	Text	Variables	HTTP Status Code
SVC0001	A service error has occurred. Error code is <error_explanation>	error_explanation : <content_here>	400

MessageId	Text	Variables	HTTP Status Code
SVC0002	Invalid input value for Message part <part_name>	part_name : name of the input parameter that resulted in the error	400
SVC0003	Invalid input value for Message part <part_name>, valid values are <part_values>	part_name : name of the input parameter that resulted in the error. part_value : value of input parameter that was found to be in error.	400
SVC0004	No valid addresses provided in the Message part <part_name>	part_name : name of the input parameter that resulted in the error	400

4.25.10 Policy Exceptions

MessageId	Text	Variables	HTTP Status Code
POL0001	A policy error occurred. For example, rate limit error, authentication and authorization error.	N/A	401, 403
POL0002	Privacy verification failed for address <address>, request is refused	address : <content_here>	403
POL0003	Too many addresses specified in Message part	N/A	403
POL1009	User has not been provisioned for %1	1% : System that has not been provisioned.	403

4.26 Operation: Cancel End File Transfer

4.26.1 Functional Behavior

Cancel or End file transfer method is used to cancel or end file transfer dialog respectively.

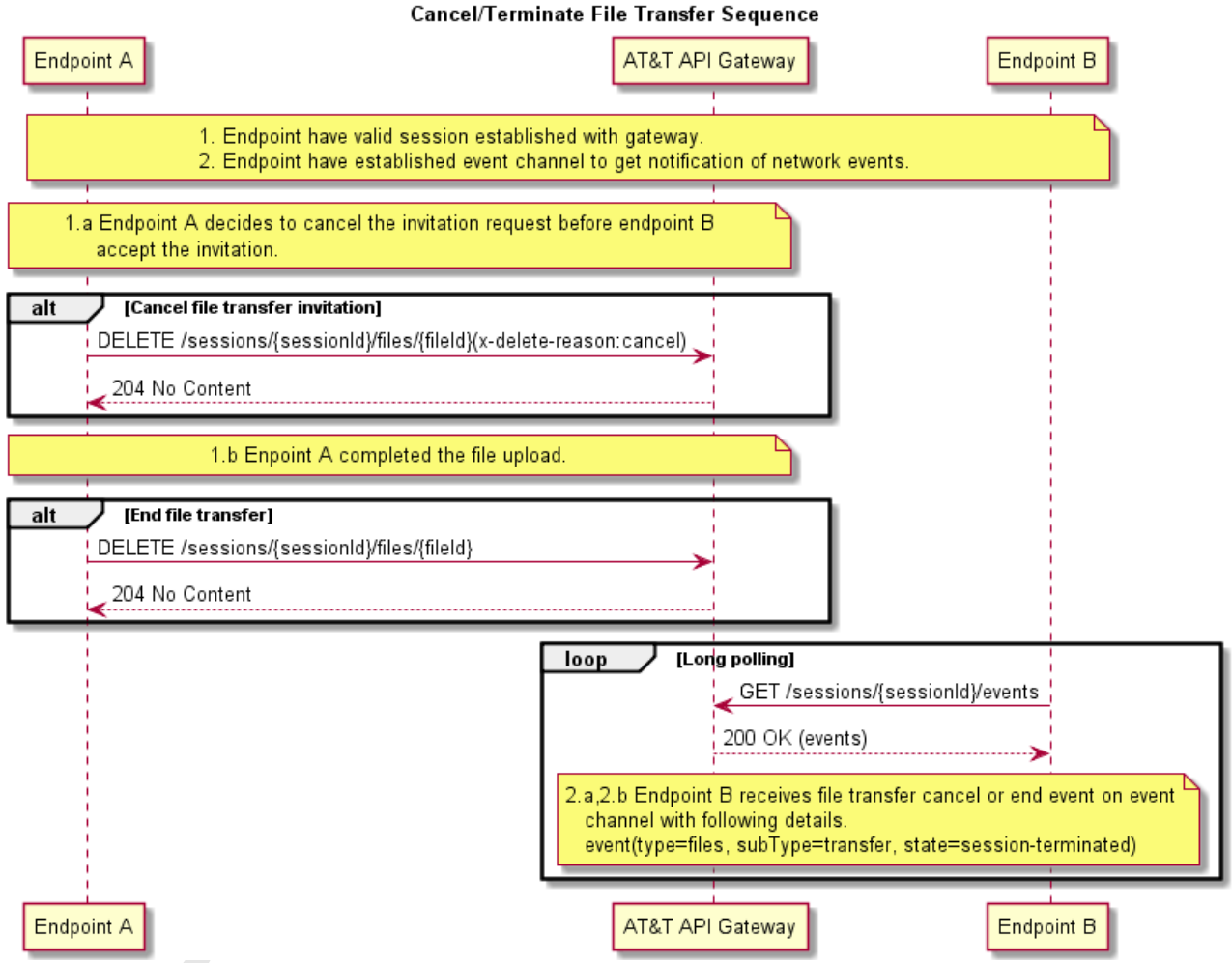
Note: The endpoint must be registered with AT&T network before accepting a cancel or end file transfer dialog request.

Note: The endpoint also must establish an event channel to receive notification of possible failures, success, or changes from the AT&T network.

Note: End point must have a valid File Transfer ID and the file transfer needs to be in the state “invitation-received” for cancel request to succeed. If user calls the HTTP DELETE method before user have accepted the invitation, the operation is called cancel, otherwise it is called terminate.

Note: The endpoint must be registered, have a valid file transfer ID, and the file transfer must set the state to invitation-received or session-open for a end file transfer dialog request to succeed.

4.26.2 Call flow



4.26.3 Version Impact Summary

Service Version	Major or Minor Impact	Changes Introduced by this Version
1	Major	Initial Release

4.26.4 Authentication and Authorization

Authorization Model	Subscriber Autho-rization required?	OAuth Scope Value	Brief Description
authorization_code	Yes	RTC	1. Obtain the authorization code by passing App Key and App Secret along with scope. This redirects the user to AT&T authorization page to capture user consent. 2. Obtain OAuth access token using the OAuth authorization code along with the App Key and App Secret by making a Get Access Token request with grant_type=authorization_code. • Supports ICMN case
client_credentials	No	RTC	1. Obtain OAuth access token with the App Key and App Secret along with the scope by making a Get Access Token request with grant_type=client_credentials • Supports VTN case • Supports no-TN case

4.26.5 Representation Formats

4.26.6 Representation Formats

Direction	Supported Respresentation Formats
Request	XML, JSON,URLENCODED
Response	XML, JSON,URLENCODED

4.26.7 Input Parameters

Parameter	Data Type	Req?	Brief description	Location
accept	String	No	Specifies the format of the body of the response. The acceptable values for this parameter are: • application/json • application/xml • application/x-www-form-urlencoded The default value is application/json. Note: For this method, this parameter specifies how the entity should be represented in case of an error. This parameter is for setting the format of an error message. If there is no error, then the representation will be the form of the actual content. Per rfc2616: ”If no Accept header field is present, then it is assumed that the client accepts all media types.” By default our services return application/json. The normal Accept header processing rules shall be followed according to rfc2616. *Note*: If there is no entity body in a normal successful response, this parameter is still needed to specify the format in the case of an error response message.	Header
authorization	String	Yes	Specifies the authorization type and token. The acceptable format for this parameter is the phrase ”Bearer OAuth Token” followed by a space ( ) and an OAuth access token. If this parameter value is missing from the header, then the API Gateway returns a message with an HTTP status code of 400 Invalid Request. If the OAuth access token is not valid, then the API Gateway returns an HTTP status code of 401 Unauthorized with a WWW-Authenticate HTTP header.	Header

Parameter	Data Type	Req?	Brief description	Location
content-length	Integer	No	Specifies the length of the content in octets. This parameter is only required for a non-streaming request.	Header
content-type	String	Yes	Specifies the type of content of the body. The acceptable values for this parameter are: • application/xml • application/json • image/jpeg	Header
transfer-encoding	String	Conditional	Specifies the encodingof the message. The only acceptable values for this parameter is: • chunked This parameter is only required for a streaming request.	Header
x-delete-reason	String	Yes	Specifies the the type of ending file transfer. The acceptable values for this parameter are: • cancel • terminate The default value is terminate.	Body

4.26.7.1 Request – Example (Terminating the file transfer dialog)

This demostrates how to end the file transfer dialog.

```
DELETE /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/files/182332923
    HTTP/1.1
Host: api.att.com
authorization: Bearer abcdef12345678
accept: application/xml
content-type: application/xml
```

4.26.7.2 Request – Example (Cancelling file transfer dialog)

This demostrates how to cancel file transfer dialog.

```
DELETE /RTC/v1/sessions/4ba569b5-290d-4f1f-b3af-255731383204/files/182332923
    HTTP/1.1
Host: api.att.com
authorization: Bearer abcdef12345678
accept: application/xml
content-type: application/xml
x-delete-reason:cancel
```

4.26.8 Output Parameters

Parameter	Data Type	Req?	Brief description	Location
content-type	String	Yes	Specifies the type of content of the body. The acceptable values for this parameter are: • application/xml • application/json • •	Header
state	String	Conditional	Specifies that the call is accepted. Note: This parameter is returned only when the call is accepted.	Body

4.26.8.1 Response – Example (Response accept file transfer invitation) This shows the response to cancel file transfer session request.

```
HTTP/1.1 204 No Content
Content-Type: application/xml
Date: Thu, 04 Jun 2010 02:51:59 GMT
```

4.26.9 Service Exceptions

MessageId	Text	Variables	Parent HTTP Code
SVC0001	A service error has occurred. Error code is <Error Explanation>	Error Explanation : <content here>	400
SVC0002	Invalid input value for Message part <part name>	part name : name of the input parameter that resulted in the error	400
SVC0003	Invalid input value for Message part <part name>, valid values are <part values>	part value : value of input parameter that was found to be in error.	400
SVC0004	No valid addresses provided in the Message part <part name>	part name : name of the input parameter that resulted in the error	400

4.26.10 Policy Exceptions

MessageId	Text	Variables	Parent HTTP Code
POL0001	A policy error occurred. For example, rate limit error, authentication and authorization error.	N/A	401,403
POL0002	Privacy verification failed for address <address>, request is refused	address : <content here>	403
POL0003	Too many addresses specified in Message part	N/A	403
POL1009	User has not been provisioned for %1	System that hasn't been provisioned	403

5 Structure of Exceptions

5.1 RESTful Web Services Exceptions

Two types of RESTful exceptions are supported:

- Service Exceptions: These exceptions occur when a service is unable to process a request and retrying of the request will result in a consistent failure. For example, if an application provides invalid input.
- Policy Exceptions: These exceptions occur when a policy criteria has not been met. For example, if (N+1)th request arrives when an application's service level agreement only allows for N transactions per second.

Both service and policy exceptions contain the following elements:

Field Name	Data Type	Required?	Description
MessageId	xs:string	Yes	Unique message identifier of the format 'ABCnnnn' where 'ABC' is either 'SVC' for Service Exceptions or 'POL' for Policy Exception. Exception numbers may be in the range of 0001 to 9999 where : • 0001 to 0199 are reserved for common exception messages • 0200 to 0999 are reserved for Parlay Web Services specification use • 1000-9999 are available for exceptions
Text	xs:string	Yes	Message text, with replacement variables marked with %n, where n is an index into the list of <variables> elements, starting at 1
Variables	xs:string [0..unbounded]	No	List of zero or more strings that represent the contents of the variables used by the message text.

An example is provided below:

```
HTTP/1.1 405 Method Not Allowed
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"RequestError":{
    "PolicyException":{
    "MessageId":"POL123"
    "Text":"The resource was invoked with an unsupported operation: %1.",
```

```
    "Variables":"PUT"}
}}
```

Notes:

1. The response status code is provided on the first line.
2. The appropriate content type is described on the second line.
3. The RequestError structure in expressed in an appropriate format (e.g., JSON or XML).
4. ‘RequestError’ contains either a ‘PolicyException’ or a ‘ServiceException’ structure.

6 Terms

Term	Description
API	In the context of this document, API is used to refer to a specific instance of operations identified specific URI.
Callback	In the context of this document, callback is used to refer to an API that is exposed by a 3rd party application and consumed by AT&T per a specification provided by AT&T.
Compatible	In the context of this document, compatible implies the API is available and responsive but it does not meet all aspects of compliance in one or more of these areas: 1. URI 2. Parameters 3. Syntax 4. Input and Output 5. Success/Error Codes 6. Behavior of API service

Term	Description
Compliance	In context of this document, compliance implies every aspect of the API implementation is fully compliant with the relevant specification.
Consumption Pattern	<<description needed>>
End-User	<< check marketing description or mCommerce team description>>
Method	In the context of this document, method is used to refer to an API that is exposed by AT&T to be consumed by a 3rd party application.
MSISDN	MSISDN is a number uniquely identifying a subscription in a GSM or a UMTS mobile network. Simply put, it is the telephone number to the SIM card in a mobile/cellular phone.
Operation	In the context of this document, operation is used to refer to a logical feature of the related service referenced in manner independent of whether it is characterized as a “method” or a “callback”. An operation is associated with one or more URIs. An operation may have multiple URIs if multiple provides offer the operation. Examples of operations: • Send SMS • Receive SMS
Short code	Short codes (also known as short numbers) are special telephone numbers, significantly shorter than full telephone numbers, which can be used to address SMS and MMS messages from certain service provider’s mobile phones or fixed phones.
Verbs	In context of this document, the term verbs refers to HTTP methods e.g. GET, PUT, POST,. per IETF RFC 2616.

7 Acronyms

Item	Description
ACR	Anonymous Customer Reference
DC	Device Capabilities query
DL	Device Location query
DTD	Data Type Document
FQDN	Fully Qualified Domain Name
IMEI	International Mobile Equipment Identity
JSON	JavaScript Object Notation

Item	Description
JSONP	JavaScript Object Notation with Padding
MSISDN	Mobile Subscriber ISDN Number
NSG	Network Services Gateway
PAP	Push Access Protocol
REST	Representational State Transfer
SI	Service Indication
UE	User Equipment, for example the customer's mobile device
URI	Uniform Resource Indicator
URL	Uniform Resource Locator
WAP	Wireless Application Protocol
VTN	Virtual Telephone Number
ICMN	In App Calling Mobile Number
NoTN	No Telephone Number, using sip URI instead to make a call
E911	Enhanced 9-1-1
PSTN	Public switched telephone network
ESN	Emergency Service Number
ALI	Automatic Location Information database
PSAP	Public Safety Answering Point

8 Approvals

8.1 Assertion Checklist

By approving below; the Project Manager, CSO, Dev Ecosystem, and the Project Team assert that:

- Data elements exposed by Northbound APIs described in this document have been reviewed for privacy or legal and security concerns.
- The authentication and authorization approach meets current standards and best practices.
- The API naming and scope is aligned with the service catalog.
- The API design conforms to all other requirements of the Northbound API Practice requirements except where exceptions have been approved.

8.2 Approvers

ATTUID	Name	Role	Latest Version Approved, Approval Date and Evidence
		Project Manager	
		Solution Architect	
		CSO	
		Privacy	
		Legal	
		API Realization	
		Dev Ecosystem	
		Arch and Planning	

9 Appendices

9.1 Parameters

Parameter	Data Type	Unit	Brief description
chat_enable	Boolean		Indicates that the chat service is active.
chat_max_media_size	Integer	bytes	Specifies the maximum number of bytes a send media message allows in an ongoing chat session. • Default: 51200 • Range: 0 through 2147483647

Parameter	Data Type	Unit	Brief description
chat_timeout_received_media	Integer	seconds	Specifies the number of seconds the received chat media message is available on the server before it is automatically removed. • Default: 120 • Range: 30 through 2147483647
chat_invite_reject_delay	Integer	seconds	Specifies the number of seconds before an incoming chat session is automatically rejected when the client does not accept the session. • Default: 120 • Range: 0 through 2147483647
groupchat_enable	Boolean		Indicates that the group chat service is active.
groupchat_max_media_size	Integer	bytes	Specifies the maximum number of bytes a send media message allows in an ongoing group chat session. • Default: 51200 • Range: 0 through 2147483647

Parameter	Data Type	Unit	Brief description
groupchat_max_participants	Integer	participant	Specifies the maximum number of participants allowed in a group chat session including the initiator. • Default: 10 • Range: 2 through 2147483647
groupchat_timeout_received_media	Integer	seconds	Specifies the number of seconds the received group chat media message is available on the server before it is automatically removed. • Default: 120 • Range: 30 through 2147483647
groupchat_invite_reject_delay	Integer	seconds	Specifies the number of seconds before a group chat invitation is automatically rejected by the server when the client does not accept the session. • Default: 120 • Range: 0 through 2147483647
calling_services_enable	Boolean		Indicates that the AudioVideo calling service is active.

Parameter	Data Type	Unit	Brief description
call_invite_reject_delay	Integer		Specifies the number of seconds before an AudioVideo session request is automatically rejected by the server when the client does not accept the session. • Default: 120 • Range: 0 through 2147483647
conference_services_enable	Boolean		Indicates that the media conferencing service is active.
filetransfer_enable	Boolean		Indicates if file transfer service is active
filetransfer_invite_reject_delay	Integer	seconds	Indicating the number of seconds before a file transfer request is automatically rejected by the server if the client does not accept the session. Default: **120** Range: 0-2147483647
filetransfer_max_size	Integer	bytes	Maximum file size allowed for originating file transfer Default: 5000000 Range: 0-2147483647 bytes
user_session_timeout	Integer	Seconds	Timeout duration (seconds) for a client not renewing the event channel. At timeout the client is de-registered. • Default: 3600

9.2 SIP-Error-Mapping

9.2.1 SIP Error Response (Mapped to standard errors)

SIP Response Code	Description	Standard Cause Code	Description
400	Bad Request	41	Temporary Failure
401	Unauthorized	21	Call Rejected
402	Payment Required	21	Call Rejected
403	Forbidden	21	Call Rejected
404	Not Found	1	Unallocated Number
405	Method Not Allowed	63	Service or Option Unavailable
406	Not Acceptable	79	Service/Option Not Implemented
407	Proxy Authentication Required	21	Call Rejected
408	Request Timeout	102	Recovery On Timer Expiry
409	Conflict		
410	Gone	22	Number Changed (no diagnostic)
411	Length Required		
413	Request Entry Too Long	127	Interworking Unspecified
414	Request URI Too Long	127	Interworking Unspecified
415	Unsupported Media Type	79	Service/Option Not Implemented
416	Unsupported URI Scheme	127	Interworking Unspecified
420	Bad Extension	127	Interworking Unspecified
421	Extension Required	127	Interworking Unspecified
423	Interval Too Brief	127	Interworking Unspecified
480	Temporarily Unavailable	18	User Not Responding
481	Call Transaction Does Not Exist	41	Temporary Failure
482	Loop Detected	25	Exchange Routing Error

483	Too Many Hops	25	Exchange Routing Error
484	Address Incomplete	28	Invalid Number Format
485	Ambiguous	1	Unallocated Number
486	Busy Here	17	Busy Here
487	Request Terminated		
488	Not Acceptable		
500	Server Internal Error	41	Temporary Failure
501	Not Implemented	79	Service/Option Not Implemented
502	Bad Gateway	38	Network Out of Order
503	Service Unavailable	41	Temporary Failure
504	Server Timeout	102	Recovery On Timer Expiry
505	Version Not Supported	127	Interworking Unspecified
513	Message Too Long	127	Interworking Unspecified
600	Busy Everywhere	17	Busy Here
603	Decline	21	Decline
604	Does Not Exist Anywhere	1	Unallocated Number
606	Not Acceptable		
Any Other Status Code		31	Normal Unspecified

9.3 HTTP Response Codes

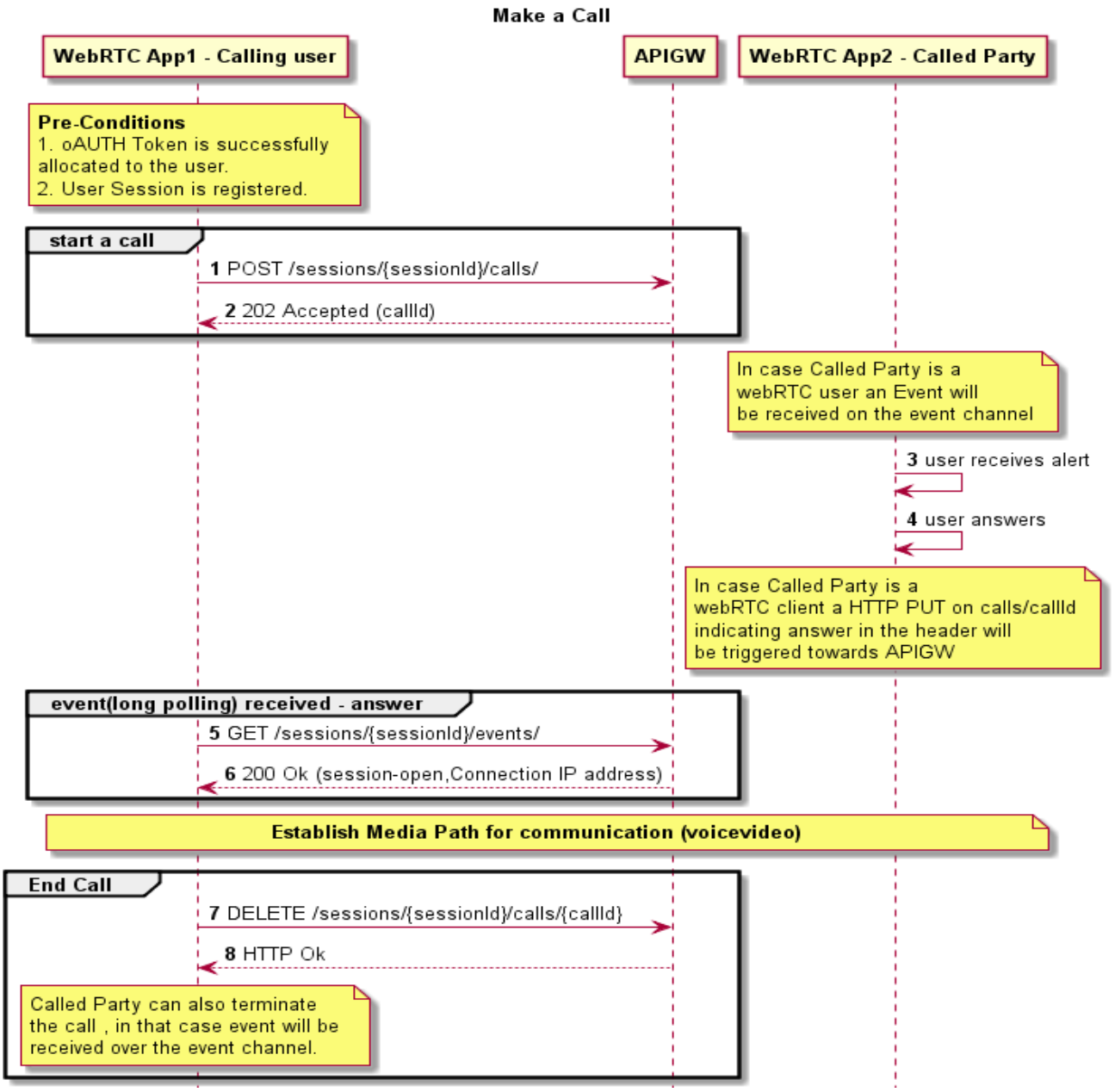
Code	Reason Phrase	Description
200	OK	Successful response.
201	Created	Successful response and resource was successfully created.
202	Accepted	Successful response but action has not yet been enacted.

Code	Reason Phrase	Description
200	OK	Successful response.
204	No Content	Successful response and the response does not include an entity.
206	Partial Content	The server has fulfilled the partial GET request for the resource.
304	Not Modified	Returned when the client has performed a conditional GET request and access is allowed, but the resource has not been modified.
400	Bad Request	Many possible reasons not specified by the other codes.
401	Authentication Error	Authentication failed or was not provided.
403	Forbidden	Access permission error.
404	Not Found	The server has not found anything matching the Request-URI. No indication is given of whether the condition is temporary or permanent.
405	Method Not Allowed	A request was made of a resource using a request method not supported by that resource , for example using PUT on a REST resource that only supports POST.
406	Not Acceptable	The resource identified by the request is only capable of generating response entities which have content characteristics not acceptable according to the accept headers sent in the request.
408	Request Timeout	The client did not produce a request within the time that the server was prepared to wait. The client may repeat the request without modifications at any later time.
409	Conflict	The request could not be completed due to a conflict with the current state of the resource. This code is only allowed in situations where it is expected that the user might be able to resolve the conflict and resubmit the request.
411	Length Required	The Content-Length header was not specified.
412	Precondition failed	A conditional header triggered the failure of the request.
413	Request Entity Too Large	The size of the request body exceed the maximum size permitted.
414	Request-URI Too Long	The server is refusing to service the request because the Request-URI is longer than the server is willing to interpret.

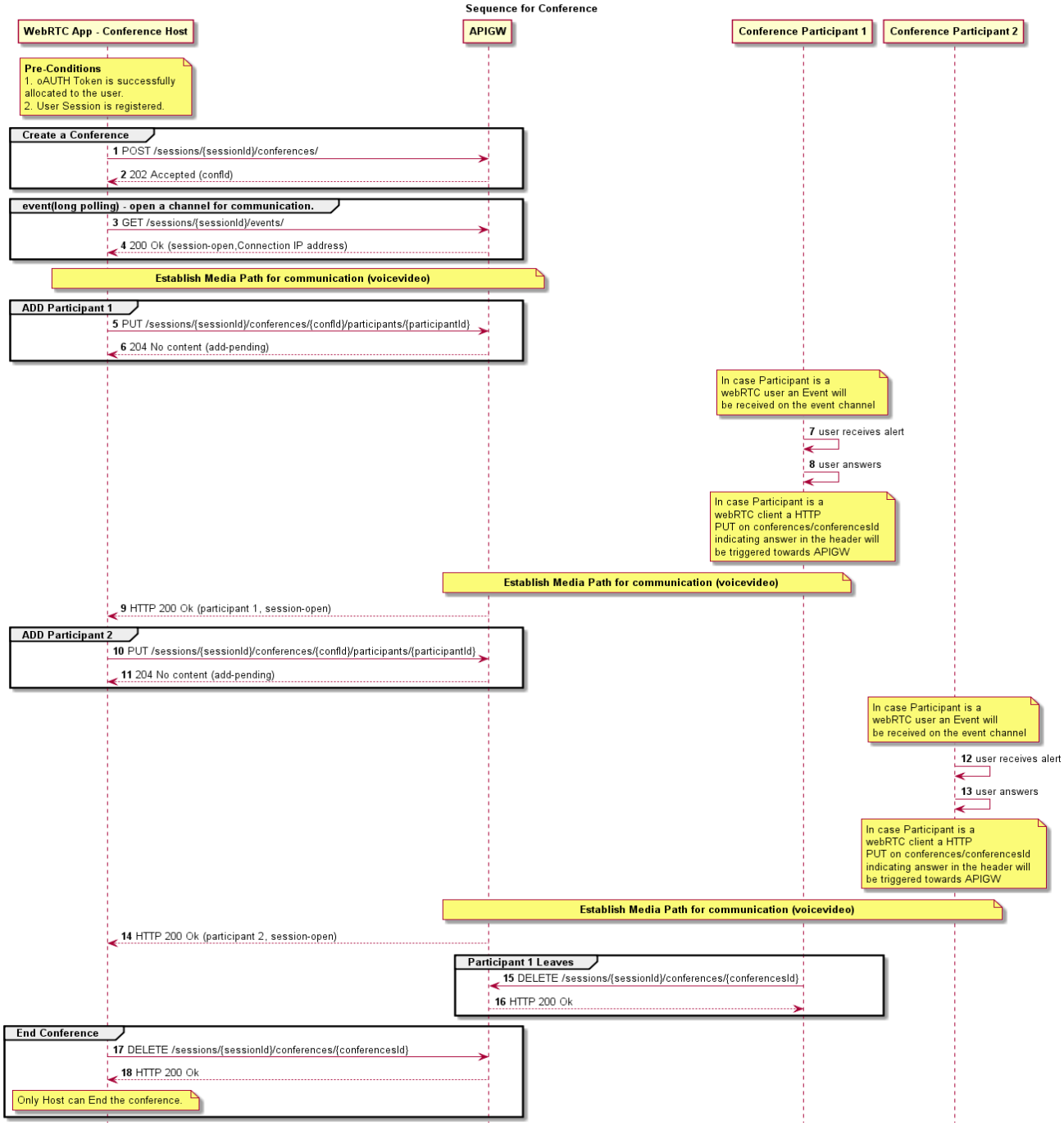
Code	Reason Phrase	Description
200	OK	Successful response.
415	Unsupported Media Type	The request is in a format not supported by the requested resource for the requested method.
500	Internal Server Error	The server encountered an internal error or timed out; please retry.
502	Bad Gateway	The server, while acting as a gateway or proxy, received an invalid response from the upstream server it accessed in attempting to fulfill the request.
503	Service Unavailable	The server is currently unable to receive requests; please retry.
504	Gateway Timeout	The server, while acting as a gateway or proxy, did not receive a timely response from the upstream server specified by the URI (e.g. HTTP, FTP, LDAP) or some other auxiliary server (e.g. DNS) it needed to access in attempting to complete the request. **Note:** Note to implementors: some deployed proxies are known to return 400 or 500 when DNS lookups time out.

9.4 Overall Call flows

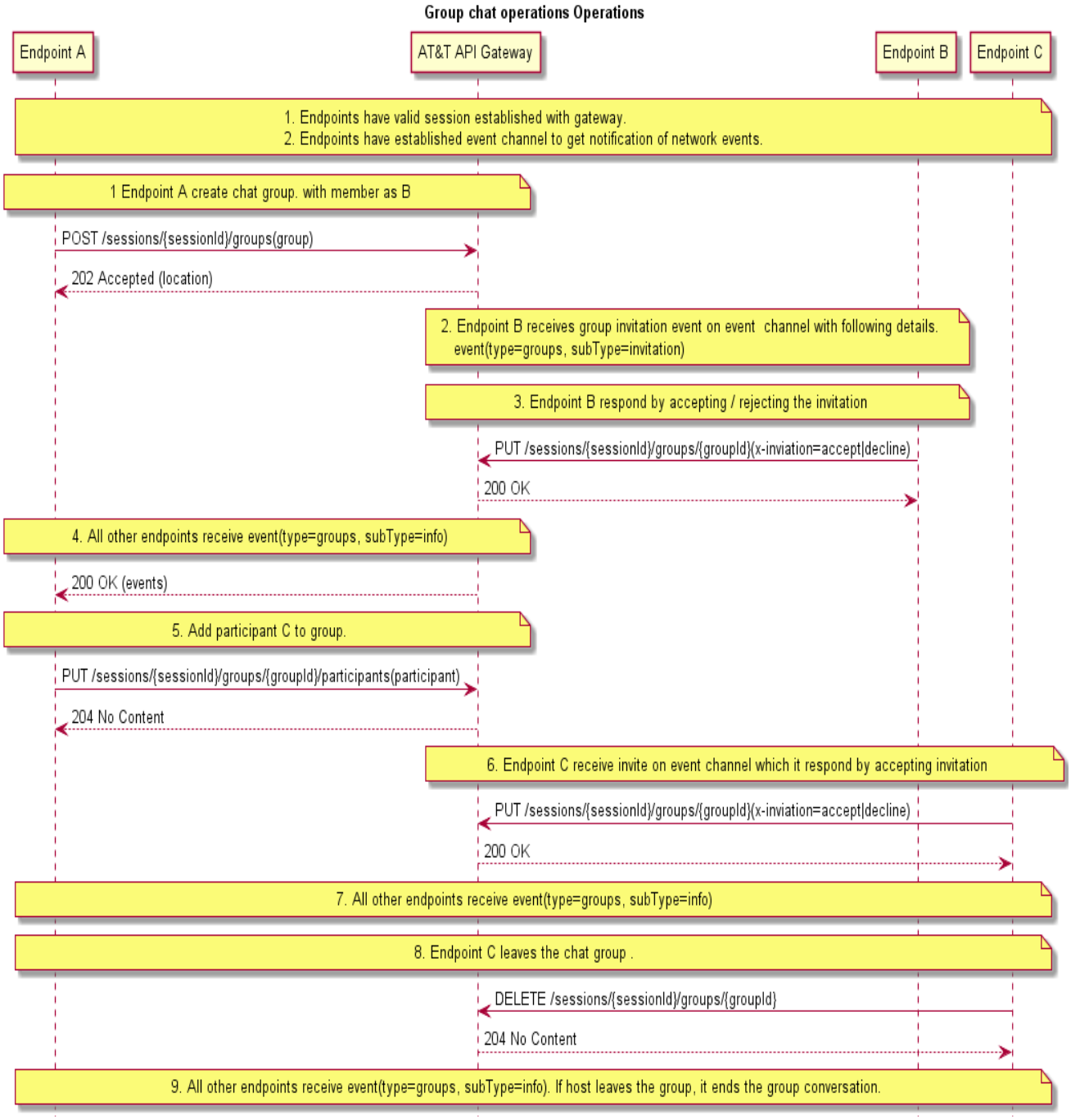
9.4.1 Sequence for Call



9.4.2 Sequence for Conference



9.4.3 Sequence for Group Chat



9.4.4 Sequence for File Transfer