

Appendix C – Claim Chart for US Patent No. 9,270,744 Against Accused Cisco Products

Based on information presently available, Damaka Inc. (“Damaka”) contends that Cisco Systems, Inc. (“Cisco” or “Defendant”) infringes Claims 20-27 (the “Asserted Claims”) of U.S. Patent No. 9,270,744 (the “’744 Patent”) through the Accused Products which are manufactured, sold, offered for sale, used, imported into the United States, and/or exported from the United States by Cisco.

The Accused Products comprise all versions and variants of the Webex Application and Webex SDKs¹, including, but not limited to, Webex for iOS, Webex for Android, Webex for Tablet, Webex for Web Browser, Webex for the Windows operating system, Webex for the Mac operating system, and Webex for Linux that are made, used, sold, offered for sale, imported into the United States, and/or exported from the United States or otherwise provided, after May 30, 2019. Damaka reserves the right to amend this list of Accused Products as discovery progresses.

Cisco directly infringes each of the Asserted Claims by using, importing, testing, selling, offering for sale, importing into the United States, and/or exporting from the United States the Accused Products in violation of 35 U.S.C. § 271(a).

Cisco indirectly infringes the Asserted Claims in violation of 35 U.S.C. § 271(b) by inducing third parties, including Cisco’s customers and end-users, to directly infringe through their operation and use of the Accused Products. Cisco has knowingly and intentionally induced this direct infringement by, *inter alia*, manufacturing, selling, distributing, and/or otherwise making available the Accused Products, and providing instructions, documentation, and other information to customers and end-users suggesting that they use the Accused Products in an infringing manner, including technical support, marketing, product manuals, advertisements, and online documentation. Cisco’s marketing and promotional materials for the Accused Products are found, for example, on the Webex website (<https://www.webex.com/>). For example, the Webex website instructs customers to, among other things, use the services in the Accused Products. See <https://www.Webex.com/frequently-asked-questions>. The Webex website also offers support to customers, including instruction on, among other things, how to operate their products. On information and belief, Cisco knows that its actions will result in infringement of the Asserted Claims, or subjectively believes that there is a high probability that its actions will result in infringement of the Asserted Claims but has taken deliberate actions to avoid learning these facts.

Cisco also contributorily infringes each of the Asserted Claims in violation of 35 U.S.C. § 271(c) by selling, importing, offering for sale, and otherwise providing the Accused Products, which when used by third parties, such as customers and end-users, directly infringe the Asserted Claims. The Accused Products constitute a material part of the Asserted Claims, are not staple articles or commodities of commerce, have no substantial non-infringing uses, and are known by Cisco to be especially made or adapted for use in the infringement of the ’744 Patent. On information and belief, Cisco knows that its actions will result in infringement of the Asserted Claims, or subjectively believes that there is a high probability that its actions will result in infringement of the Asserted Claims but has taken deliberate actions to avoid learning these facts.

On information and belief, the charted version of the Webex application and Webex SDK is representative of all the Accused Products as functional equivalents of all claims exist in iOS, Android, and “Web.”²

Damaka does not concede that any claims of the ’744 Patent that are not listed below are not infringed by the identified Accused Products. Moreover, the citations to certain documents and other information below are intended to be exemplary only and in no way foreclose Damaka from citing or relying on additional documents, information, source code, and/or testimony at a later time. These contentions are preliminary in nature, and an analysis of Cisco’s products, internal documentation, source code, and/or testimony from relevant witnesses may more fully and accurately describe the infringing features of its Accused Products. Accordingly, Damaka reserves the right to supplement, correct, modify, and/or amend these contentions once such additional information is made available to Damaka. Furthermore, Damaka reserves the right to supplement, correct, modify, and/or amend

¹ The Webex SDK encompasses both the SDK as and associated Webex API. See *e.g.* <https://developer.webex.com/meeting/docs/meetings>

² See *e.g.* <https://developer.webex.com/meeting/docs/meetings>; See also <https://help.webex.com/en-us/article/fz1e4b/System-requirements-for-Webex-services>

Appendix C – Claim Chart for US Patent No. 9,270,744 Against Accused Cisco Products

these contentions as discovery in this case progresses; in view of the Court’s claim construction order(s); in view of any positions taken by Cisco, including, but not limited to, positions on claim construction, invalidity, and/or non-infringement; and in connection with the preparation and exchange of expert reports.

To the extent Cisco contends that any element of the Accused Products is attributable to a third party, Damaka contends that the activities are attributable to Cisco such that they constitute direct infringement by Cisco under 271(a). The acts may be attributable to Cisco because Cisco directs or controls the others’ performance, and because Cisco and the other entity form a joint enterprise. *Akamai Techs., Inc. v. Limelight Networks, Inc.*, 797 F.3d 1020, 1022 (Fed. Cir. 2015). Additionally, Cisco is vicariously liable for the activities of these other entities. *Centillion Data Sys., LLC v. Qwest Commc’ns Int’l, Inc.*, 631 F.3d 1279, 1286 (Fed. Cir. 2011). Further, the activities of these entities (including third party application developers, and users of the Accused Products consumers), are attributable to Cisco because Cisco (1) conditions participation in an activity or receipt of a benefit upon others’ performance of one or more steps of a patented method, and (2) establishes the manner or timing of that performance. *Eli Lilly & Co. v. Teva Parenteral Medicines, Inc.*, 845 F.3d 1357, 1365 (Fed. Cir. 2017). These acts are also attributed to Cisco because it initiated the activities of their end-users. *SiRF Tech., Inc. v. Int’l Trade Comm’n*, 601 F.3d 1319, 1330 (Fed. Cir. 2010).

Claim	Cisco Accused Products
20.p A method comprising:	Each Accused Product practices a method comprising the steps of claim 20. Plaintiff may supplement these contentions pursuant to Section 3(a)(i) of the model Discovery Order once access to relevant source code is provided.
20a. receiving, by a function block stored in a memory of a mobile device, a request for a communication service from a superblock application stored in the memory of the mobile device, wherein the superblock application is configured to access the function block for functionality that is lacking in the superblock application;	Each Accused Product literally performs the step of “receiving, by a function block stored in a memory of a mobile device, a request for a communication service from a superblock application stored in the memory of the mobile device, wherein the superblock application is configured to access the function block for functionality that is lacking in the superblock application.” For example, Cisco’s Webex SDK, the function block within Cisco’s Webex (the superblock) application, receives a request for communication services. As shown below “Webex Meetings [SDK] offers integrated audio, video...”

Appendix C – Claim Chart for US Patent No. 9,270,744 Against Accused Cisco Products

Claim	Cisco Accused Products
	Webex Meetings Webex Meetings offers integrated audio, video, and content sharing with highly secure web meetings from the cloud. The Webex Meetings REST API allows developers to add basic Webex scheduling functionality to their custom applications or websites. You can: ▪ Create and manage meetings ▪ Add and manage meeting invitees ▪ Get and update meeting preferences, options, and other details ▪ Get, list, and delete recordings https://developer.webex.com/meeting/docs/meetings For example, the function block (Webex SDK) can receive a “Join_a_meeting” call. This allows the superblock application “to start or join a meeting”

Join a Meeting

Operation Id: *Join_a_Meeting*

Description: *Retrieves links for a meeting with a specified `meetingId`, `meetingNumber`, or `webLink` that allow users to start or join the meeting directly without logging in and entering a password.*

- Please note that `meetingId`, `meetingNumber` and `webLink` are mutually exclusive and they cannot be specified simultaneously.
- If `joinDirectly` is true or not specified, the response will have HTTP response code 302 and the request will be redirected to `joinLink`; otherwise, the response will have HTTP response code 200 and `joinLink` will be returned in response body.
- Only the meeting host or cohost can generate the `startLink`.
- An admin user or a [Service App](#) can generate the `startLink` and `joinLink` on behalf of another meeting host using the `hostEmail` parameter. When a [Service App](#) generates the `startLink` and `joinLink`, the `hostEmail` parameter is required. The `hostEmail` parameter only applies to meetings, not webinars.
- For Service Apps, `hostEmail` must be provided in the request.
- Generating a join link or a start link before the time specified by `joinBeforeHostMinutes` for a webinar is not supported.
- The `joinLink` and `startLink` generated by the API only work in a web browser, not in mobile apps. When the `joinLink` or `startLink` is used in a mobile app, the user can't join or start the meeting directly by the dial-in or dial-out phone numbers displayed in the app.
- When `createJoinLinkAsWebLink` or `createStartLinkAsWebLink` is set to true, a user cannot join or start the meeting using the `joinLink` or `startLink` returned in the response, and must complete the login flow. Those options are typically useful if mandatory user login is configured in Control Hub.
- Appending the parameter `&LaunchApp=true` to the weblink will launch directly into the browser experience without showing the user the download selection for the native client.
- [Embedded Apps](#) are not supported in a meeting which is started from the `startLink` generated by this API.

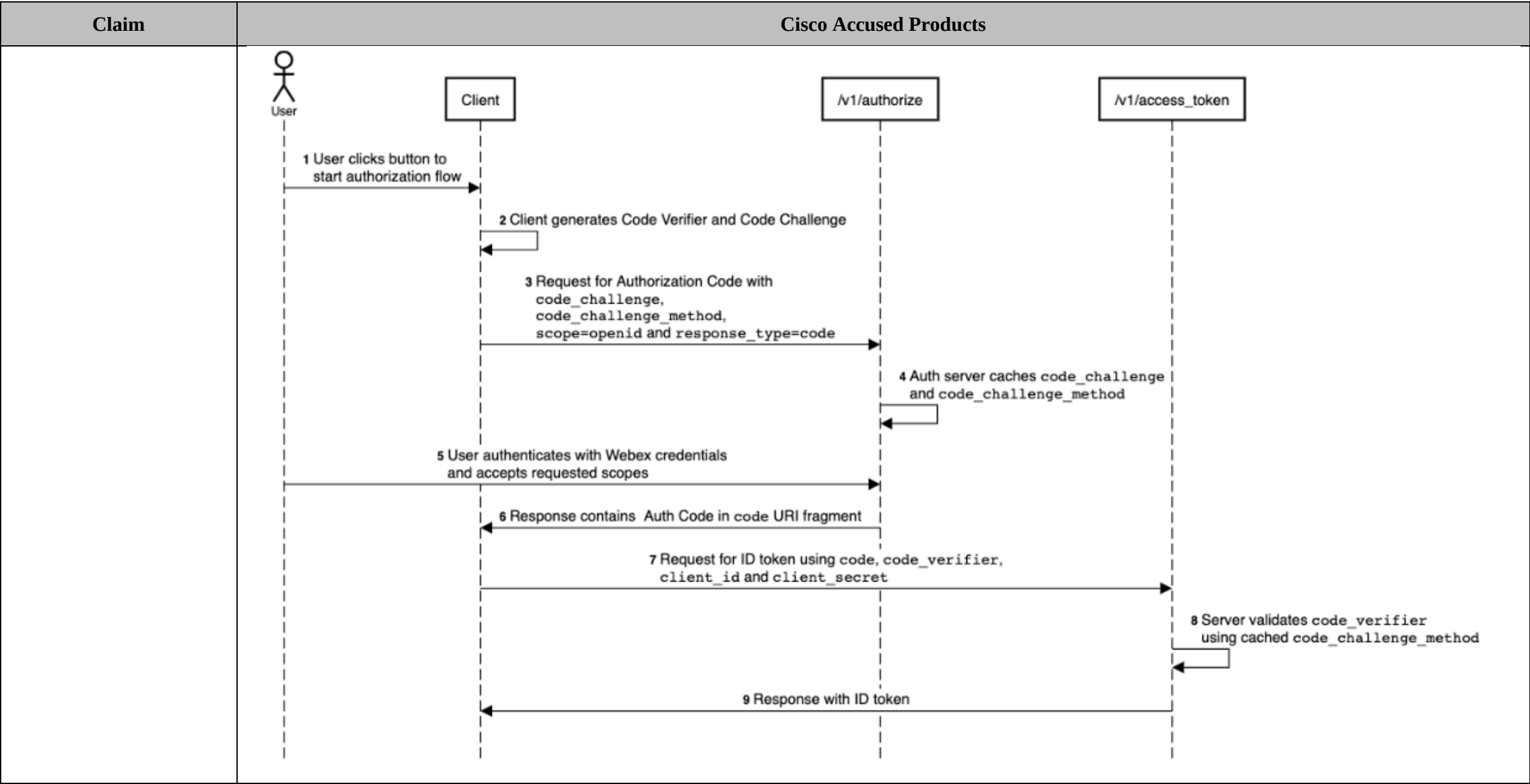
Appendix C – Claim Chart for US Patent No. 9,270,744 Against Accused Cisco Products

Claim	Cisco Accused Products
	https://developer.webex.com/meeting/docs/api/v1/meetings/join-a-meeting There is also equivalent functionality in the formal SDK, as demonstrated by the below. Fetch a Meeting Instance A <code>meeting</code> instance is typically created once a meeting is initiated through the SDK. If a meeting has already been created on the SDK, you can retrieve it using the SDK's <code>meetings</code> object by passing a <code>meetingId</code>.  This method does not fetch the list of all meetings from the Webex cloud. It only lists the meetings that are created either with <code>webex.meetings.create</code> or <code>webex.meetings.syncMeetings</code>. <pre>const meeting = webex.meetings.getAllMeetings()[meetingId];</pre> If participants have already joined the meeting, it can be retrieved from Webex Cloud. Refer to Sync Meetings. Join a Meeting Once you have a <code>Meeting</code> object, you can join the meeting using the <code>join()</code> method: <pre>const join = await meeting.join(joinOptions);</pre> https://developer.webex.com/meeting/docs/sdks/webex-meetings-sdk-web-join-a-meeting

Appendix C – Claim Chart for US Patent No. 9,270,744 Against Accused Cisco Products

Claim	Cisco Accused Products
	Plaintiff may supplement these contentions pursuant to Section 3(a)(i) of the model Discovery Order once access to relevant source code is provided.
20b. determining, by the function block, that providing the communication service to the superblock application requires access to an external device;	Each Accused Product literally performs the step of “determining, by the function block, that providing the communication service to the superblock application requires access to an external device.” For example, when authorizing the superblock application, the function block (Webex SDK) must contact Cisco servers that are external to the mobile device the Webex application is on. As shown below, the function block (Webex SDK) makes a request to a Cisco server to “validate[.]”

Appendix C – Claim Chart for US Patent No. 9,270,744 Against Accused Cisco Products



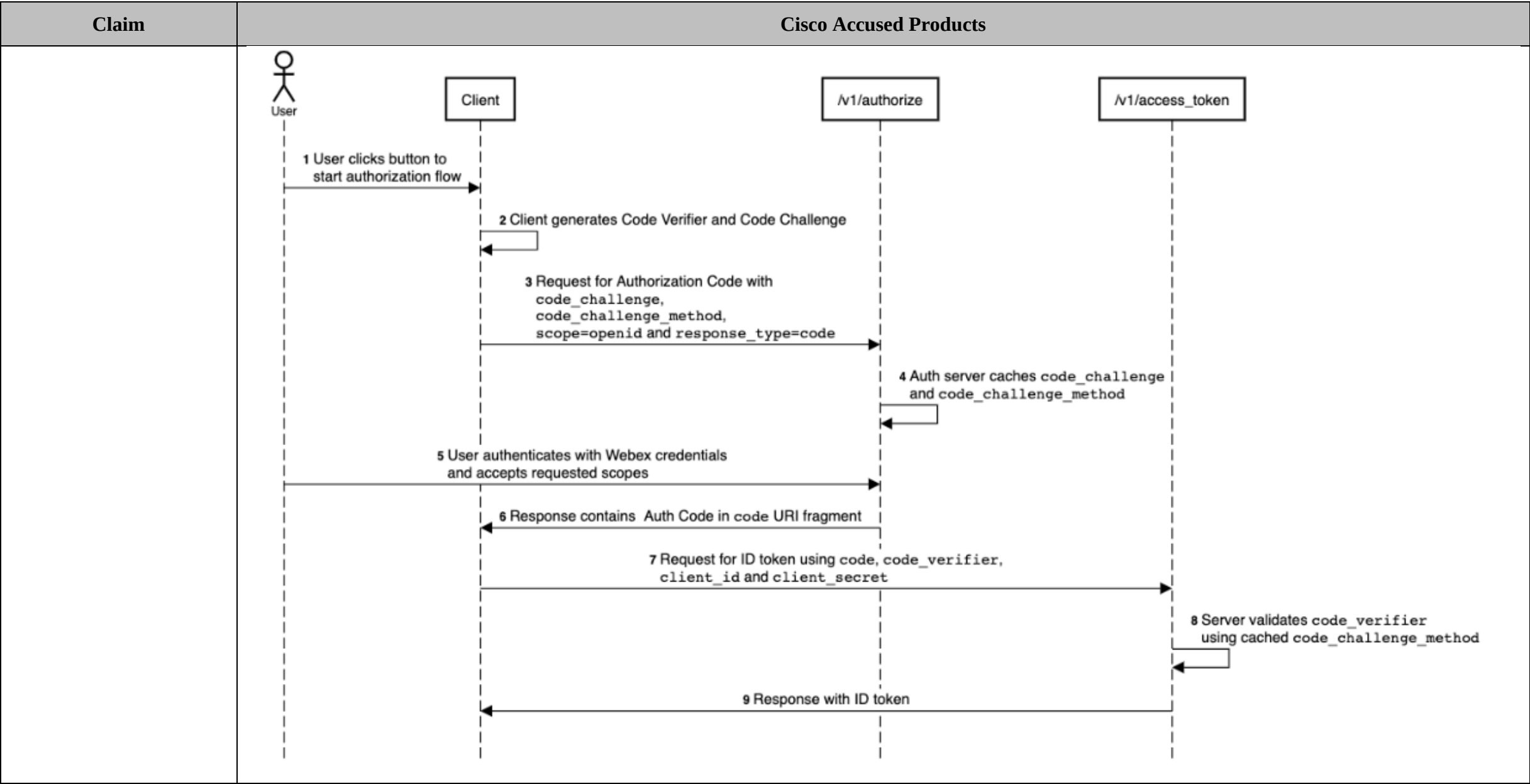
Appendix C – Claim Chart for US Patent No. 9,270,744 Against Accused Cisco Products

Claim	Cisco Accused Products
	https://developer.webex.com/create/docs/login-with-webex If authorization is not successful, the function block will throw a 401 error, as shown below. Status: 401 <i>Unauthorized: Authentication credentials were missing or incorrect.</i> https://developer.webex.com/meeting/docs/api/v1/meetings/join-a-meeting There is analogous functionality within the formal Webex SDK. Initialize the SDK To use the SDK, you'll need a free Webex account; if you don't have a Cisco Webex account, visit Cisco Webex for Developers to create your account and retrieve your access token from Getting and Using your Personal Access Token. To create a new Webex instance: <pre>const webex = Webex.init({ credentials: { access_token: "<AUTHORIZATION/BOT TOKEN>" } });</pre> At this point, the Webex <code>Meetings</code> object is ready for use. The <code>Meetings</code> object contains a list of meetings as well as methods to perform specific operations or listen for certain events. Detailed information is included in this article.

Appendix C – Claim Chart for US Patent No. 9,270,744 Against Accused Cisco Products

Claim	Cisco Accused Products
	https://developer.webex.com/meeting/docs/sdks/webex-meetings-sdk-authorization-device-registration For example, authentication credentials must be acquired to use the Webex SDK. Without proper authorization, Webex SDK will not function, as evidenced by the below. What Are Integrations? Integrations are how you request permission to invoke the Webex REST API on behalf of another Webex user. To do this in a secure way the API supports the OAuth 2 standard which allows third-party integrations to get a temporary access token for authenticating API calls instead of asking users for their password. If you're sure that your integrations require authenticating on behalf of another Webex user, read on, we'll get you there in a few easy steps: 1. Register your integration with Webex 2. Request permission using an OAuth Grant Flow 3. Exchange the resulting authorization code for an access token 4. Use the access token to make your API calls https://developer.webex.com/create/docs/integrations For example, the A/V service requires access the Cisco's server. For example, access to Cisco servers is required for both authentication, as well as hosting the meeting room instance as evident by the "Server validates..." and "MEETING_LINK" shown below.

Appendix C – Claim Chart for US Patent No. 9,270,744 Against Accused Cisco Products



Appendix C – Claim Chart for US Patent No. 9,270,744 Against Accused Cisco Products

Claim	Cisco Accused Products												
	https://developer.webex.com/create/docs/login-with-webex The below shows the “MEETING_LINK” denoting that Cisco servers host the A/V calls.  The screenshot shows a table with the following content: <table><tr><th>2</th><th>type</th><th>String</th><th>No</th></tr><tr><td colspan="4">The meeting type must be one of the following:</td></tr><tr><td colspan="4"> • MEETING_LINK • SIP_URI • CONVERSATION_URL • PERSONAL_ROOM • MEETING_LINK </td></tr></table> https://developer.webex.com/meeting/docs/sdks/webex-meetings-sdk-web-introduction For example, authentication credentials must be acquired to use the Webex SDK. Without proper authorization, Webex SDK will not function, as evidenced by the below.	2	type	String	No	The meeting type must be one of the following:				• MEETING_LINK • SIP_URI • CONVERSATION_URL • PERSONAL_ROOM • MEETING_LINK			
2	type	String	No										
The meeting type must be one of the following:													
• MEETING_LINK • SIP_URI • CONVERSATION_URL • PERSONAL_ROOM • MEETING_LINK													

Appendix C – Claim Chart for US Patent No. 9,270,744 Against Accused Cisco Products

Claim	Cisco Accused Products
	What Are Integrations? Integrations are how you request permission to invoke the Webex REST API on behalf of another Webex user. To do this in a secure way the API supports the OAuth 2 standard which allows third-party integrations to get a temporary access token for authenticating API calls instead of asking users for their password. If you're sure that your integrations require authenticating on behalf of another Webex user, read on, we'll get you there in a few easy steps: 1. Register your integration with Webex 2. Request permission using an OAuth Grant Flow 3. Exchange the resulting authorization code for an access token 4. Use the access token to make your API calls https://developer.webex.com/create/docs/integrations Plaintiff may supplement these contentions pursuant to Section 3(a)(i) of the model Discovery Order once access to relevant source code is provided.
20c. establishing, by the function block, a connection with the external device to support the communication service via a network interface of the mobile device, wherein the establishing is not supported by the superblock application; and	Each Accused Product literally performs the step of “establishing, by the function block, a connection with the external device to support the communication service via a network interface of the mobile device, wherein the establishing is not supported by the superblock application.” For example, the function block needs to send the A/V request over the internet – as such the function block must use the network interface of the mobile device. See support for claim 20b. Plaintiff may supplement these contentions pursuant to Section 3(a)(i) of the model Discovery Order once access to relevant source code is provided.

Appendix C – Claim Chart for US Patent No. 9,270,744 Against Accused Cisco Products

Claim	Cisco Accused Products
20d. providing, by the function block, the communication service to the superblock application, wherein the steps of receiving, determining, establishing, and providing are performed without switching context from the superblock application to another application on the mobile device.	Each Accused Product literally performs the step of “providing, by the function block, the communication service to the superblock application, wherein the steps of receiving, determining, establishing, and providing are performed without switching context from the superblock application to another application on the mobile device.” For example, because the Webex SDK is compiled within the Cisco’s Webex application, there is no context switching from the superblock application. Plaintiff may supplement these contentions pursuant to Section 3(a)(i) of the model Discovery Order once access to relevant source code is provided.
21. The method of claim 20 wherein providing the communication service includes operating, by the function block, as a bridge between the superblock application and the external device.	Each Accused Product literally meets the requirement “wherein providing the communication service includes operating, by the function block, as a bridge between the superblock application and the external device.” For example, the function block is responsible for bridging the superblock application of Cisco’s Webex and the external device, such as a Cisco server (or other server). See support for claim 20.

Appendix C – Claim Chart for US Patent No. 9,270,744 Against Accused Cisco Products

Claim	Cisco Accused Products
	Plaintiff may supplement these contentions pursuant to Section 3(a)(i) of the model Discovery Order once access to relevant source code is provided.
22. The method of claim 20 wherein the request is an application programming interface (API) call from the superblock application to the function block.	Each Accused Product literally meets the requirement “wherein the request is an application programming interface (API) call from the superblock application to the function block.” For example, the Webex SDK offers both a formal SDK and an API as demonstrated in claim 20. Plaintiff may supplement these contentions pursuant to Section 3(a)(i) of the model Discovery Order once access to relevant source code is provided.
23. The method of claim 20 further comprising determining, by the function block, whether the function block has a capability to provide the communication service.	Each Accused Product literally performs the step of “determining, by the function block, whether the function block has a capability to provide the communication service.” For example, the Webex SDK has authorization capabilities as shown in claim 20b. Additionally, the Webex SDK offers numerous checks on resolution capabilities.

Appendix C – Claim Chart for US Patent No. 9,270,744 Against Accused Cisco Products

Claim	Cisco Accused Products																														
	Prerequisites Keep the following prerequisites in mind: - The device camera must support full HD (1080p); if not, the feature is downgraded to the camera's maximum resolution support. To check for FHD support, see Webcam Test Resolution. - Make sure that you're using version webex-js-sdk@2.32.0 or above.  For SDK versions 2.32.0 and above, the value mappings used for resolutions are LOW: 480p, MEDIUM: 720p, and HIGH 1080p. For earlier SDK versions: LOW: 360p, MEDIUM: 480p, and HIGH: 720p. Local Resolution The sharpness, smoothness, and overall quality of the video are directly linked to the frame rate, bitrate, and other video settings. The following table specifies the different resolution settings available. Supported Sending Video Resolutions <table><tr><th>Video Profile</th><th>Resolution (Width*Height)</th><th>Frame Rate (fps)</th><th>Chrome</th><th>Firefox</th><th>Safari</th></tr><tr><td>360p</td><td>640 * 360</td><td>30</td><td>✓</td><td>✓</td><td>✓</td></tr><tr><td>480p</td><td>640 * 480</td><td>30</td><td>✓</td><td>✓</td><td>✓</td></tr><tr><td>720p</td><td>1280 * 720</td><td>30</td><td>✓</td><td>✓</td><td>✓</td></tr><tr><td>1080p</td><td>1920 * 1080</td><td>30</td><td>✓</td><td>✓</td><td>✓</td></tr></table>	Video Profile	Resolution (Width*Height)	Frame Rate (fps)	Chrome	Firefox	Safari	360p	640 * 360	30	✓	✓	✓	480p	640 * 480	30	✓	✓	✓	720p	1280 * 720	30	✓	✓	✓	1080p	1920 * 1080	30	✓	✓	✓
Video Profile	Resolution (Width*Height)	Frame Rate (fps)	Chrome	Firefox	Safari																										
360p	640 * 360	30	✓	✓	✓																										
480p	640 * 480	30	✓	✓	✓																										
720p	1280 * 720	30	✓	✓	✓																										
1080p	1920 * 1080	30	✓	✓	✓																										

Appendix C – Claim Chart for US Patent No. 9,270,744 Against Accused Cisco Products

Claim	Cisco Accused Products															
	https://developer.webex.com/meeting/docs/sdks/webex-meetings-sdk-web-video-resolution For example, the Webex SDK will throw the following errors if either the superblock or function block cannot render the A/V call: Remote Video Resolution Errors The following table lists possible errors when working with remote video resolutions: <table><tr><th>Method</th><th>Error</th><th>Fix</th></tr><tr><td><code>setLocalVideoTrack()</code></td><td><code>Meeting:index#setLocalVideoTrack --> Local video quality of "\${localQualityLevel}" not supported, downscaling to the highest possible resolution of "\${height}p"</code></td><td>Remote resolution is set to 720p.</td></tr><tr><td><code>setRemoteVideoConstraints()</code></td><td><code>setRemoteVideoConstraints: unable to set max framesize, value for level "\${level}" is not defined</code></td><td>Use the defined resolution values.</td></tr><tr><td><code>getMediaStreams()</code></td><td><code>ParameterError: \${audioVideo} not supported. Either pass level from pre-defined resolutions or pass complete audioVideo object</code></td><td>Pass a level from the pre-defined resolutions or use a complete <code>audioVideo</code> object with appropriate video properties</td></tr><tr><td><code>getUserMedia()</code></td><td><code>Media:index#getMedia --> navigator.mediaDevices.getUserMedia failed - [object OverconstrainedError] (facingMode)</code></td><td>If you request the rear camera on a laptop device, you'll get this error. Use either <code>ideal</code> instead of <code>exact</code> or request only cameras appropriate to the device.</td></tr></table> https://developer.webex.com/meeting/docs/sdks/webex-meetings-sdk-web-video-resolution	Method	Error	Fix	<code>setLocalVideoTrack()</code>	<code>Meeting:index#setLocalVideoTrack --> Local video quality of "\${localQualityLevel}" not supported, downscaling to the highest possible resolution of "\${height}p"</code>	Remote resolution is set to 720p.	<code>setRemoteVideoConstraints()</code>	<code>setRemoteVideoConstraints: unable to set max framesize, value for level "\${level}" is not defined</code>	Use the defined resolution values.	<code>getMediaStreams()</code>	<code>ParameterError: \${audioVideo} not supported. Either pass level from pre-defined resolutions or pass complete audioVideo object</code>	Pass a level from the pre-defined resolutions or use a complete <code>audioVideo</code> object with appropriate video properties	<code>getUserMedia()</code>	<code>Media:index#getMedia --> navigator.mediaDevices.getUserMedia failed - [object OverconstrainedError] (facingMode)</code>	If you request the rear camera on a laptop device, you'll get this error. Use either <code>ideal</code> instead of <code>exact</code> or request only cameras appropriate to the device.
Method	Error	Fix														
<code>setLocalVideoTrack()</code>	<code>Meeting:index#setLocalVideoTrack --> Local video quality of "\${localQualityLevel}" not supported, downscaling to the highest possible resolution of "\${height}p"</code>	Remote resolution is set to 720p.														
<code>setRemoteVideoConstraints()</code>	<code>setRemoteVideoConstraints: unable to set max framesize, value for level "\${level}" is not defined</code>	Use the defined resolution values.														
<code>getMediaStreams()</code>	<code>ParameterError: \${audioVideo} not supported. Either pass level from pre-defined resolutions or pass complete audioVideo object</code>	Pass a level from the pre-defined resolutions or use a complete <code>audioVideo</code> object with appropriate video properties														
<code>getUserMedia()</code>	<code>Media:index#getMedia --> navigator.mediaDevices.getUserMedia failed - [object OverconstrainedError] (facingMode)</code>	If you request the rear camera on a laptop device, you'll get this error. Use either <code>ideal</code> instead of <code>exact</code> or request only cameras appropriate to the device.														

Appendix C – Claim Chart for US Patent No. 9,270,744 Against Accused Cisco Products

Claim	Cisco Accused Products
	Plaintiff may supplement these contentions pursuant to Section 3(a)(i) of the model Discovery Order once access to relevant source code is provided.
24. The method of claim 20 further comprising determining, by the function block, whether the function block has a capability to perform any rendering needed by the communication service.	Each Accused Product literally performs the step of “determining, by the function block, whether the function block has a capability to perform any rendering needed by the communication service.” For example, the Webex SDK has rendering capabilities as shown in claim 23. Plaintiff may supplement these contentions pursuant to Section 3(a)(i) of the model Discovery Order once access to relevant source code is provided.
25. The method of claim 20 further comprising determining, by the function block, whether the superblock has a capability to perform any rendering needed by the communication service.	Each Accused Product literally performs the step of “the method of claim 20 further comprising determining, by the function block, whether the superblock has a capability to perform any rendering needed by the communication service.” For example, the Webex SDK performs resolution and rendering checks, as shown in claim 23. Plaintiff may supplement these contentions pursuant to Section 3(a)(i) of the model Discovery Order once access to relevant source code is provided.
26. The method of claim 20 further comprising initializing the function block, by the superblock application, when the superblock application is	Each Accused Product literally performs the step of “initializing the function block, by the superblock application, when the superblock application is loaded into the memory and executed.” For example, on information and belief the Webex application (the super block application), initializes the Webex SDK (function block).

Appendix C – Claim Chart for US Patent No. 9,270,744 Against Accused Cisco Products

Claim	Cisco Accused Products
loaded into the memory and executed.	Plaintiff may supplement these contentions pursuant to Section 3(a)(i) of the model Discovery Order once access to relevant source code is provided.
27. The method of claim 20 further comprising determining, by the function block, that the superblock application is authorized to access functionality provided by the function block for the communication service.	Each Accused Product literally performs the step of “determining, by the function block, that the superblock application is authorized to access functionality provided by the function block for the communication service.” For example, the Webex SDK authorizes the Webex APP to provide communication services, as shown in claim 20b. Plaintiff may supplement these contentions pursuant to Section 3(a)(i) of the model Discovery Order once access to relevant source code is provided.