Avaya Analytics™

REST & WebSocket Open Interface API Guide

Release 4.2.0.0

Issue 1.0

July 2022

AVAYA ANALYTICS REAL TIME OPEN INTERFACE SOFTWARE DEVELOPMENT KIT LICENSE AGREEMENT

REVISED: January 14, 2022

READ THIS CAREFULLY BEFORE ELECTRONICALLY ACCESSING OR USING THIS PROPRIETARY PRODUCT!

THIS IS A LEGAL AGREEMENT ("AGREEMENT") BETWEEN YOU, INDIVIDUALLY, AND/OR THE LEGAL ENTITY FOR WHOM YOU ARE OPENING, INSTALLING, DOWNLOADING, COPYING OR OTHERWISE USING THE AVAYA SOFTWARE DEVELOPMENT KIT ("SDK") (COLLECTIVELY, AS REFERENCED HEREIN, "YOU", "YOUR", OR "LICENSEE") AND AVAYA INC. OR ANY AVAYA AFFILIATE (COLLECTIVELY, "AVAYA"). IF YOU ARE ACCEPTING THE TERMS AND CONDITIONS OF THIS AGREEMENT ON BEHALF OF A LEGAL ENTITY, YOU REPRESENT AND WARRANT THAT YOU HAVE FULL LEGAL AUTHORITY TO ACCEPT ON BEHALF OF AND BIND SUCH LEGAL ENTITY TO THIS AGREEMENT. BY OPENING THE MEDIA CONTAINER, BY INSTALLING, DOWNLOADING, COPYING OR OTHERWISE USING THE AVAYA SOFTWARE DEVELOPMENT KIT ("SDK") OR AUTHORIZING OTHERS TO DO SO, YOU SIGNIFY THAT YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT HAVE SUCH AUTHORITY OR DO NOT WISH TO BE BOUND BY THE TERMS OF THIS AGREEMENT, SELECT THE "DECLINE" BUTTON AT THE END OF THE TERMS OF THIS AGREEMENT OR THE EQUIVALENT OPTION AND YOU SHALL HAVE NO RIGHT TO USE THE SDK.

1.0 DEFINITIONS.

1.1 "Affiliates" means any entity that is directly or indirectly controlling, controlled by, or under common control with Avaya Inc. For purposes of this definition, "control" means the power to direct the management and policies of such party, directly or indirectly, whether through ownership of voting securities, by contract or otherwise; and the terms "controlling" and "controlled" have meanings correlative to the foregoing.

1.2 "Avaya Software Development Kit" or "SDK" means Avaya technology, which may include Software, Client Libraries, Specification Documents, Software libraries, application programming interfaces ("API"), Software tools, Sample Application Code and Documentation.

1.3 "Client Libraries" mean any enabler code specifically designated as such and included in a SDK. Client Libraries may also be referred to as "DLLs", and represent elements of the SDK required at runtime to communicate with Avaya products or other SDK elements.

1.4 "Change In Control" shall be deemed to have occurred if any person, entity or group comes to own or control, directly or indirectly, beneficially or of record, voting securities (or any other form of controlling interest) which represent more than fifty percent (50%) of the total voting power of the Licensee.

1.5 "Derivative Work(s)" means any translation (including translation into other computer languages), port, compiling of Source Code into object code, combination with a pre-existing work, modification, correction, addition, extension, upgrade, improvement, compilation, abridgment or other form in which an existing work may be recast, transformed or adapted or which would otherwise constitute a derivative work under the United States Copyright Act. Permitted Modifications will be considered Derivative Works.

1.6 "Documentation" includes programmer guides, CDs, manuals, materials, and information appropriate or necessary for use in connection with the SDK. Documentation may be provided in machine-readable, electronic or hard copy form.

1.7 "Intellectual Property" means any and all: (i) rights associated with works of authorship throughout the world, including copyrights, neighboring rights, moral rights, and mask works, (ii) trademark and trade name rights and similar rights, (iii) trade secret rights, (iv) patents, algorithms, designs and other industrial property rights, (v) all other

intellectual and industrial property rights (of every kind and nature throughout the world and however designated) whether arising by operation of law, contract, license, or otherwise, and (vi) all registrations, initial applications, renewals, extensions, continuations, divisions or reissues thereof now or hereafter in force (including any rights in any of the foregoing).

1.8 "Permitted Modification(s)" means Licensee's modifications of the Sample Application Code as needed to create applications, interfaces, workflows or processes for use with Avaya products.

1.9 "Specification Document" means any notes or similar instructions in hard copy or machine readable form, including any technical, interface and/or interoperability specifications that define the requirements and conditions for connection to and/or interoperability with Avaya products, systems and solutions.

1.10 "Source Code" means human readable or high-level statement version of software written in the source language used by programmers and includes one or more programs. Source Code programs may include one or more files, such as user interface markup language (.mxml), action script (.as), precompiled Flash code (.swc), java script (.js), hypertext markup language (.html), active server pages (.asp), C# or C# .Net source code (.cs), java source code (.java), java server pages (.jsp), java archives (.jar), graphic interchange format (.gif), cascading style sheet (.css), audio files (.wav) and extensible markup language (.xml) files.

1.11 "Sample Application Code" means Software provided for the purposes of demonstrating functionality of an Avaya product through the Avaya Software Development Kit.

1.12 "Software" means data or information constituting one or more computer or apparatus programs, including Source Code or in machine-readable, compiled object code form.

2.0 LICENSE GRANT.

2.1 SDK License.

A. Provided Licensee pays to Avaya the applicable license fee (if any), Avaya hereby grants Licensee a limited, non-exclusive, non-transferable license (without the right to sublicense, except as set forth in 2.1B(iii)) under the Intellectual Property of Avaya and, if applicable, its licensors and suppliers to (i) use the SDK solely for the purpose of Licensee's internal development efforts to develop applications, interfaces, value-added services and/or solutions, workflows or processes to work in conjunction with Avaya products; (ii) to package Client Libraries for redistribution with Licensee's complementary applications that have been developed using this SDK, subject to the terms and conditions set forth herein; (iii) use Specification Documents solely to enable Licensee's products, services and application solutions to exchange messages and signals with Avaya products, systems and solutions to which the Specification Document(s) apply; (iv) modify and create Derivative Works of the Sample Application Code, Specification Documents and Documentation solely for internal development of applications, interfaces, workflows or processes for use with Avaya products, integration of such applications, interfaces, workflows and processes with Avaya products and interoperability testing of the foregoing with Avaya products; and (v) compile or otherwise prepare for distribution the Sample Application Code with Permitted Modifications, into an object code or other machine-readable program format for distribution and distribute the same subject to the conditions set forth in Section 2.1B.

B. The foregoing license to use Sample Application Code is contingent upon the following: (i) Licensee must ensure that the modifications made to the Sample Application Code as permitted in clause (iv) of Section 2.1A are compatible and/or interoperable with Avaya products and/or integrated therewith, (ii) Licensee may distribute Licensee's application that has been created using this SDK, provided that such distribution is subject to an end user pursuant to Licensee's current end user license agreement ("Licensee EULA") that is consistent with the terms of this Agreement and, if applicable, any other agreement with Avaya (e.g., the Avaya DevConnect Program Agreement), and is equally as protective as Licensee's standard software license terms, but in no event shall the standard of care be less than a reasonable degree of care, and (iii) Licensee ensures that each end user who receives Client Libraries or Sample Application Code with Permitted Modifications has all

necessary licenses for all underlying Avaya products associated with such Client Libraries or Sample Application Code.

Your Licensee EULA must include terms concerning restrictions on use, protection of proprietary rights, disclaimer of warranties, and limitations of liability. You must ensure that Your End Users using applications, interfaces, value-added services and/or solutions, workflows or processes that incorporate the API, Client Libraries, Sample Code or Permitted Modifications adhere to these terms, and You agree to notify Avaya promptly if You become aware of any breach of the terms of Licensee EULA that may impact Avaya. You will take all reasonable precautions to prevent unauthorized access to or use of the SDK and notify Avaya promptly of any such unauthorized access or use.

C. Licensee acknowledges and agrees that it is licensed to use the SDK only in connection with Avaya products (and if applicable, in connection with services provided by or on behalf of Avaya).

D. With respect to Software that contains elements provided by third party suppliers, Licensee may install and use the Software in accordance with the terms and conditions of the applicable license agreements, such as "shrinkwrap" or "click-through" licenses, accompanying or applicable to the Software.

2.2 No Standalone Product. Nothing in this Agreement authorizes or grants Licensee any rights to distribute or otherwise make available to a third party the SDK, in whole or in part, or any Derivative Work in source or object code format on a standalone basis other than the modifications permitted in Section 2.1B of this Agreement.

2.3 Proprietary Notices. Licensee shall not remove any copyright, trade mark or other proprietary notices incorporated in the copies of the SDK, Sample Application Code and redistributable files in Licensee's possession or control or any modifications thereto. Redistributions in binary form or other suitable program format for distribution, to the extent expressly permitted, must also reproduce Avaya's copyright, trademarks or other proprietary notices as incorporated in the SDK in any associated Documentation or "splash screens" that display Licensee copyright notices.

2.4 Third-Party Components. You acknowledge certain software programs or portions thereof included in the SDK may contain software distributed under third party agreements ("Third Party Components"), which may contain terms that expand or limit rights to use certain portions of the SDK ("Third Party Terms"). Information identifying the copyright holders of the Third Party Components and the Third Party Terms that apply is available in the attached Schedule 1 (if any), SDK, Documentation, or on Avaya's web site at: http://support.avaya.com/Copyright (or such successor site as designated by Avaya). The open source software license terms provided as Third Party Terms are consistent with the license rights granted in this Agreement, and may contain additional rights benefiting You, such as modification and distribution of the open source software. The Third Party Terms shall take precedence over this Agreement, solely with respect to the applicable Third Party Components, to the extent that this Agreement imposes greater restrictions on You than the applicable Third Party Terms. Licensee is solely responsible for procuring any necessary licenses for Third Party Components, including payment of licensing royalties or other amounts to third parties, for the use thereof.

2.5 Copies of SDK. Licensee may copy the SDK only as necessary to exercise its rights hereunder.

2.6a No Reverse Engineering. Licensee shall have no rights to any Source Code for any of the software in the SDK, except for the explicit rights to use the Source Code as provided to Licensee hereunder. Licensee agrees that it shall not cause or permit the disassembly, decompilation or reverse engineering of the Software. Notwithstanding the foregoing, if the SDK is rightfully located in a member state of the European Union and Licensee needs information about the Software in the SDK in order to achieve interoperability of an independently created software program with the Software in the SDK, Licensee will first request such information from Avaya. Avaya may charge Licensee a reasonable fee for the provision of such information. If Avaya refuses to make such information available, then Licensee may take steps, such as reverse assembly or reverse compilation, to the extent necessary solely in order to achieve interoperability of the Software in the SDK with an independently created software program. To the extent that the Licensee is expressly permitted by applicable mandatory law to undertake any of the activities listed in this

section, Licensee will not exercise those rights until Licensee has given Avaya twenty (20) days written notice of its intent to exercise any such rights.

2.6.b License Restrictions. To the extent permissible under applicable law, Licensee agrees not to: (i) publish, sell, sublicense, lease, rent, loan, assign, convey or otherwise transfer the SDK; (ii) distribute, disclose or allow use the SDK, in any format, through any timesharing service, service bureau, network or by any other means; (iii) distribute or otherwise use the Software in the SDK in any manner that causes any portion of the Software that is not already subject to an OSS License to become subject to the terms of any OSS License; (iv) link the Source Code for any of the software in the SDK with any software licensed under the Affero General Public License (Affero GPL) v.3 or similar licenses; (v) access information that is solely available to root administrators of the Avaya products, systems, and solutions; (vi) develop applications, interfaces, value-added services and/or solutions, workflows or processes that causes adverse effects to Avaya and third-party products, services, solutions, such as, but not limited to, poor performance, software crashes and cessation of their proper functions; and (vii) develop applications, interfaces, value-added services and/or solutions, workflows or processes that blocks or delays emergency calls; (viii) emulate an Avaya SIP endpoint by form or user interface design confusingly similar as an Avaya product ; (ix) reverse engineer Avaya SIP protocol messages; or (x) permit or encourage any third party to do any of (i) through (x), inclusive, above.

2.7 Responsibility for Development Tools. Licensee acknowledges that effective utilization of the SDK may require the use of a development tool, compiler and other software and technology of third parties, which may be incorporated in the SDK pursuant to Section 2.4. Licensee is solely responsible for procuring such third party software and technology and the necessary licenses, including payment of licensing royalties or other amounts to third parties, for the use thereof.

2.8 U.S. Government End Users. The SDK shall be classified as "commercial computer software" and the Documentation is classified as "commercial computer software documentation" or "commercial items," pursuant to FAR 12.212 or DFAR 227.7202, as applicable. Any use, modification, reproduction, release, performance, display or disclosure of the SDK or Documentation by the Government of the United States shall be governed solely by the terms of the Agreement and shall be prohibited except to the extent expressly permitted by the terms of the Agreement.

2.9 Limitation of Rights. No right is granted to Licensee to sublicense its rights hereunder. All rights not expressly granted are reserved by Avaya or its licensors or suppliers and, except as expressly set forth herein, no license is granted by Avaya or its licensors or suppliers under this Agreement directly, by implication, estoppel or otherwise, under any Intellectual Property right of Avaya or its licensors or suppliers. Nothing herein shall be deemed to authorize Licensee to use Avaya's trademarks or trade names in Licensee's advertising, marketing, promotional, sales or related materials.

2.10 Independent Development.

2.10.1 Licensee understands and agrees that Avaya, Affiliates, or Avaya’s licensees or suppliers may acquire, license, develop for itself or have others develop for it, and market and/or distribute applications, interfaces, value-added services and/or solutions, workflows or processes similar to that which Licensee may develop. Nothing in this Agreement shall restrict or limit the rights of Avaya, Affiliates, or Avaya’s licensees or suppliers to commence or continue with the development or distribution of such applications, interfaces, value-added services and/or solutions, workflows or processes.

2.10.2 Nonassertion by Licensee. Licensee agrees not to assert any Intellectual Property related to the SDK or applications, interfaces, value-added services and/or solutions, workflows or processes developed using the SDK against Avaya, Affiliates, Avaya’s licensors or suppliers, distributors, customers, or other licensees of the SDK.

2.11 Feedback and Support. Licensee agrees to provide any information, comments, problem reports, enhancement requests and suggestions regarding the performance of the SDK (collectively, “Feedback”) via any public or private support mechanism, forum or process otherwise indicated by Avaya. Avaya monitors applicable mechanisms, forums, or processes but is under no obligation to implement any of Feedback, or be required to respond to any questions

asked via the applicable mechanism, forum, or process. Licensee hereby assigns to Avaya all right, title, and interest in and to Feedback provided to Avaya.

2.12(a) Fees and Taxes. To the extent that fees are associated with the license of the SDK, Licensee agrees to pay to Avaya or pay directly to the applicable government or taxing authority, if requested by Avaya, all taxes and charges, including without limitation, penalties and interest, which may be imposed by any federal, state or local governmental or taxing authority arising hereunder excluding, however, all taxes computed upon Avaya's net income. If You move any Software, including the SDK, and as a result of such move, a jurisdiction imposes a duty, tax, levy or fee (including withholding taxes, fees, customs or other duties for the import and export of any such Software), then You are solely liable for, and agree to pay, any such duty, taxes, levy or other fees.

2.12(b) Audit. Avaya shall have the right, at its cost and expense, to inspect and/or audit (i) by remote polling or other reasonable electronic means at any time and (ii) in person during normal business hours and with reasonable notice Licensee's books, records, and accounts, to determine Licensee's compliance with this Agreement. In the event such inspection or audit uncovers non-compliance with this Agreement, then without prejudice to Avaya's termination rights hereunder, Licensee shall promptly pay Avaya any applicable license fees. Licensee agrees to keep a current record of the location of the SDK.

2.13 No Endorsement. Neither the name Avaya, Affiliates nor the names of contributors may be used to endorse or promote products derived from the Avaya SDK without specific prior written permission from Avaya.

2.14 High Risk Activities. The Avaya SDK is not fault-tolerant, and is not designed, manufactured or intended for use or resale as on-line control equipment or in hazardous environments requiring failsafe performance, such as in the operation of nuclear facilities, aircraft navigation or aircraft communications systems, mass transit, air traffic control, medical or direct life support machines, dedicated emergency call handling systems or weapons systems, in which the failure of the Avaya SDK could lead directly to death, personal injury, or severe physical or environmental damage ("high risk activities"). If Licensee uses the Avaya SDK for high risk activities, Licensee does so at Licensee's own risk and Licensee assumes all responsibility and liability for such use to the maximum extent such limitation or exclusion is permitted by applicable law. Licensee agrees that Avaya and its suppliers will not be liable for any claims or damages arising from or related to use of the Avaya SDK for high risk activities to the maximum extent such limitation or exclusion is permitted by law.

2.15 No Virus. Licensee warrants that (i) the applications, interfaces, value-added services and/or solutions, workflows or processes Licensee develops using this SDK will not contain any computer program file that includes time code limitations, disabling devices, or any other mechanism which will prevent the Avaya product (including other software, firmware, hardware), services and networks from being functional at all times (collectively "Time Bombs"); and (ii) the applications, interfaces, value-added services and/or solutions, workflows or processes Licensee develops using this SDK will be free of computer viruses, malicious or other harmful code, black boxes, malware, trapdoors, and other mechanisms which could: a) damage, destroy or adversely affect Avaya product, or services and/or end users; b) allow remote/hidden attacks or access through unauthorized computerized command and control; c) spy (network sniffers, keyloggers), and d) damage or erase such applications, interfaces, value-added services and/or solutions, workflows or processes developed using this SDK or data, or any computer files or systems of Avaya, Affiliates, and/or end users (collectively "Virus"). In addition to any other remedies permitted in the Agreement, if Licensee breaches its warranties under this Section, Licensee will, at its expense, take remedial action to eliminate any Time Bombs and/or Viruses and prevent re-occurrence (including implementing appropriate processes to prevent further occurrences) as well as provide prompt, reasonable assistance to Avaya to materially reduce the effects of the Time Bomb and/or Virus.

2.16 Disclaimer. Any software security feature is not a guaranty against malicious code, deleterious routines, and other techniques and tools employed by computer "hackers" and other third parties to create security exposures. Compromised passwords represent a major security risk. Avaya encourages You to create strong passwords using

three different character types, change Your password regularly and refrain from using the same password regularly. You must treat such information as confidential. You agree to notify Avaya immediately upon becoming aware of any unauthorized use or breach of Your user name, password, account, API Key, or other credentials as provided by Avaya for use of the SDK, or subscription. You are responsible for ensuring that Your networks and systems are adequately secured against unauthorized intrusion or attack and regularly back up of Your data and files in accordance with good computing practices.

2.17 Third Party Licensed Software

A. "Commercial Third Party Licensed Software" is software developed by a business with the purpose of making money from the use of that licensed software. "Freeware Licensed Software" is software which is made available for use, free of charge and for an unlimited time, but is not Open Source Licensed Software. "Open Source Software" or "OSS" is as defined by the Open Source Initiative ("OSI") https://opensource.org/osd and is software licensed under an OSI approved license as set forth at https://opensource.org/licenses/alphabetical (or such successor site as designated by OSI). These are collectively referred to herein as "Third Party Licensed Software".

B. Licensee represents and warrants that Licensee, including any employee, contractor, subcontractor, or consultant engaged by Licensee, is to the Licensee's knowledge, in compliance and will continue to comply with all license obligations for Third Party Licensed Software used in the Licensee application created using the SDK including providing to end users all information required by such licenses as may be necessary. LICENSEE REPRESENTS AND WARRANTS THAT, TO THE LICENSEE'S KNOWLEDGE, THE OPEN SOURCE LICENSED SOFTWARE EMBEDDED IN OR PROVIDED WITH LICENSEE APPLICATION OR SERVICES DOES NOT INCLUDE ANY OPEN SOURCE LICENSED SOFTWARE CONTAINING TERMS REQUIRING ANY INTELLECTUAL PROPERTY OWNED OR LICENSED BY AVAYA OR END USERS TO BE (A) DISCLOSED OR DISTRIBUTED IN SOURCE CODE OR OBJECT CODE FORM; (B) LICENSED FOR THE PURPOSE OF MAKING DERIVATIVE WORKS; OR (C) REDISTRIBUTABLE ON TERMS AND CONDITION NOT AGREED UPON BY AVAYA OR END USERS.

C. Subject to any confidentiality obligations, trade secret or other rights or claims of Licensee suppliers, Licensee will respond to requests from Avaya or end users relating to Third Party Licensed Software associated with Licensee's use of Third Party Licensed Software. Licensee will cooperate in good faith by furnishing the relevant information to Avaya or end users and the requester within two (2) weeks from the time Avaya or end user provided the request to Licensee.

3. OWNERSHIP.

3.1 As between Avaya and Licensee, Avaya or its licensors or suppliers shall own and retain all Intellectual Property rights, in and to the SDK and any corrections, bug fixes, enhancements, updates, improvements, or modifications thereto and Licensee hereby irrevocably transfers, conveys and assigns to Avaya, its licensors and its suppliers all of its right, title, and interest therein. Avaya or its licensors or suppliers shall have the exclusive right to apply for or register any patents, mask work rights, copyrights, and such other proprietary protections with respect thereto. Licensee acknowledges that the license granted under this Agreement does not provide Licensee with title or ownership to the SDK, but only a right of limited use under the terms and conditions of this Agreement.

3.2 Grant Back License to Avaya. Licensee hereby grants to Avaya an irrevocable, perpetual, non-exclusive, sublicensable, royalty-free, fully paid up, worldwide license under any and all of Licensee's Intellectual Property rights related to any Permitted Modifications, to (i) use, make, sell, execute, adapt, translate, reproduce, display, perform,

prepare derivative works based upon, distribute (internally and externally) and sublicense the Permitted Modifications and their derivative works, and (ii) sublicense others to do any, some, or all of the foregoing.

4.0 SUPPORT.

4.1 No Avaya Support. Avaya will not provide any support for the SDK provided under this Agreement or for any Derivative Works, including, without limitation, modifications to the Source Code or applications built by Licensee using the SDK. Avaya shall have no obligation to provide support for the use of the SDK, or Licensee's application, services or solutions which may or may not include redistributable Client Libraries or Sample Application Code, to any third party to whom Licensee delivers such applications, services or solutions. Avaya further will not provide fixes, patches or repairs for any defects that might exist in the SDK or the Sample Application Code provided under this Agreement. In the event that Licensee desires support services for the SDK, and, provided that Avaya offers such support services (in its sole discretion), Licensee will be required to enter into an Avaya DevConnect Program Agreement or other support agreement with Avaya.

4.2 Licensee Obligations. Licensee acknowledges and agrees that it is solely responsible for developing and supporting any applications, interfaces, value-added services and/or solutions, workflows or processes developed under this Agreement, including but not limited to (i) developing, testing and deploying such applications, interfaces, value-added services and/or solutions, workflows or processes; (ii) configuring such applications, interfaces, value-added services and/or solutions, workflows or processes to interface and communicate properly with Avaya products; and (iii) updating and maintaining such applications, interfaces, value-added services and/or solutions, workflows or processes as necessary for continued use with the same or different versions of end user and/or third party licensor products, and Avaya products.

5.0 CONFIDENTIALITY.

5.1 Protection of Confidential Information. Licensee acknowledges and agrees that the SDK and any other Avaya technical information obtained by it under this Agreement (collectively, "Confidential Information") is confidential information of Avaya. Licensee shall take all reasonable measures to maintain the confidentiality of the Confidential Information. Licensee further agrees at all times to protect and preserve the SDK in strict confidence in perpetuity, and shall not use such Confidential Information other than as expressly authorized by Avaya under this Agreement, nor shall Licensee disclose any Confidential Information to third parties without Avaya's written consent. Licensee further agrees to immediately 1) cease all use of all Confidential Information (including copies thereof) in Licensee's possession, custody, or control; 2) stop reproducing or distributing the Confidential Information; and 3) destroy the Confidential Information in Licensee's possession or under its control, including Confidential Information on its computers, disks, and other digital storage devices upon termination of this Agreement at any time and for any reason. Upon request, Licensee will certify in writing its compliance with this Section. The obligations of confidentiality shall not apply to information which (a) has entered the public domain except where such entry is the result of Licensee's breach of this Agreement; (b) prior to disclosure hereunder was already rightfully in Licensee's possession; (c) subsequent to disclosure hereunder is obtained by Licensee on a non-confidential basis from a third party who has the right to disclose such information to the Licensee; (d) is required to be disclosed pursuant to a court order, so long as Avaya is given adequate notice and the ability to challenge such required disclosure.

5.2 Press Releases. Any press release or publication regarding this Agreement is subject to prior written approval of Avaya.

6.0 NO WARRANTY.

The SDK and Documentation are provided "AS-IS" without any warranty whatsoever. AVAYA SPECIFICALLY AND EXPRESSLY DISCLAIMS ANY WARRANTIES OR CONDITIONS, STATUTORY OR OTHERWISE, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT AND SATISFACTORY QUALITY. AVAYA DOES NOT WARRANT THAT

THE SDK AND DOCUMENTATION ARE SUITABLE FOR LICENSEE'S USE, THAT THE SDK OR DOCUMENTATION ARE WITHOUT DEFECT OR ERROR, THAT OPERATION WILL BE UNINTERRUPTED, OR THAT DEFECTS WILL BE CORRECTED. FURTHER, AVAYA MAKES NO WARRANTY REGARDING THE RESULTS OF THE USE OF THE SDK AND DOCUMENTATION. NEITHER AVAYA NOR ITS SUPPLIERS MAKE ANY WARRANTY, EXPRESS OR IMPLIED, THAT THE SDK OR DOCUMENTATION IS SECURE, SECURITY THREATS AND VULNERABILITIES WILL BE DETECTED OR SOFTWARE WILL RENDER AN END USER'S OR LICENSEE'S NETWORK OR PARTICULAR NETWORK ELEMENTS SAFE FROM INTRUSIONS AND OTHER SECURITY BREACHES.

7.0 CONSEQUENTIAL DAMAGES WAIVER.

EXCEPT FOR PERSONAL INJURY CLAIMS, AVAYA SHALL NOT BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH, ARISING OUT OF OR RELATING TO THIS AGREEMENT OR USE OF THE SDK, OR FOR THE LOSS OR CORRUPTION OF DATA, INFORMATION OF ANY KIND, BUSINESS, PROFITS, OR OTHER COMMERCIAL LOSS, HOWEVER CAUSED, AND WHETHER OR NOT AVAYA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

8.0 LIMITATION OF LIABILITY.

EXCEPT FOR PERSONAL INJURY CLAIMS, IN NO EVENT SHALL AVAYA'S TOTAL LIABILITY TO LICENSEE IN CONNECTION WITH, ARISING OUT OF OR RELATING TO THIS AGREEMENT EXCEED FIVE HUNDRED DOLLARS ($500). THE PARTIES AGREE THAT THE LIMITATIONS SPECIFIED IN THIS SECTION WILL APPLY EVEN IF ANY LIMITED REMEDY PROVIDED IN THIS AGREEMENT IS FOUND TO HAVE FAILED OF ITS ESSENTIAL PURPOSE.

9.0 INDEMNIFICATION.

Licensee shall defend, indemnify and hold harmless Avaya, Affiliates and their respective officers, directors, agents, suppliers, customers and employees "Indemnified Parties") from and against all claims, demand, suit, actions or proceedings ("Claims") and damages, losses, liabilities, costs, expenses, and fees (including fees of attorneys and other professionals) ("Damages") based upon an allegation pertaining to wrongful use, misappropriation, or infringement of a third party's Intellectual Property right arising from or relating to Licensee's use of the SDK, alone or in combination with other software, such as operating systems and codecs, and the, direct or indirect, use, distribution or sale of any software, Derivative Works or other products (including but not limited to applications, interfaces, and application programming interfaces) developed utilizing the SDK.

Licensee shall defend, indemnify and hold harmless the Indemnified Parties from and against all Claims and Damages arising out of or related to: (i) personal injury (including death); (ii) damage to any person or tangible property caused, or alleged to be caused by Licensee or Licensee's application created by using the SDK; (iii) the failure by Licensee or Licensee's application created by using the SDK to comply with the terms of this Agreement or any applicable laws; (iv) the breach of any representation, or warranty made by Licensee herein; or (v) Licensee's breach of any obligation under the Licensee EULA.

10.0 TERM AND TERMINATION.

10.1 This Agreement will continue through December 31st of the current calendar year. The Agreement will automatically renew for one (1) year terms, unless terminated as specified in Section 10.2 or 10.3 below.

10.2 Either party shall have the right to terminate the Agreement, upon thirty (30) days written notice to the other party.

10.3 Notwithstanding language to the contrary, Avaya may terminate this Agreement immediately, upon written notice to Licensee for breach of Section 2 (License Grant), Section 5 (Confidentiality) or Section 12 (Compliance with Laws). Avaya may also terminate this Agreement immediately by giving written notice if a Change In Control should occur

or if Licensee becomes insolvent, or voluntary or involuntary proceedings by or against Licensee are instituted in bankruptcy or under any insolvency law, or a receiver or custodian is appointed for Licensee, or proceedings are instituted by or against Licensee for corporate reorganization or the dissolution of Licensee, which proceedings, if involuntary, have not been dismissed within thirty (30) days after the date of filing, or Licensee makes an assignment for the benefit of its creditors, or substantially all of the assets of Licensee are seized or attached and not released within sixty (60) days thereafter, or if Licensee has ceased or threatened to cease to do business in the regular course.

10.4 Upon termination or earlier termination of this Agreement, Licensee will immediately cease a) all uses of the Confidential Information; b) Licensee agrees to destroy all adaptations or copies of the Confidential Information stored in any tangible medium including any document or work containing or derived (in whole or in part) from the Confidential Information, and certify its destruction to Avaya upon termination of this License. Licensee will promptly cease use of, distribution and sales of Licensee products that embody any such Confidential Information, and destroy all Confidential Information belonging to Avaya as well as any materials that embody any such Confidential Information. All licenses granted will terminate.

10.5 The rights and obligations of the parties contained in Sections 2.3, 2.6, 2.7, 2.10, 2.11, 2.12, 3, and 5 through 17 shall survive any expiration or termination of this Agreement.

11.0 ASSIGNMENT.

Avaya may assign all or any part of its rights and obligations hereunder. Licensee may not assign this Agreement or any interest or rights granted hereunder to any third party without the prior written consent of Avaya. The term "assign" includes, but is not limited to, any transaction in which there is a Change In Control or reorganization of Licensee pursuant to a merger, sale of assets or stock. This Agreement shall terminate immediately upon occurrence of any prohibited assignment.

12.0 COMPLIANCE WITH LAWS AND IMPORT/EXPORT CONTROL.

Licensee shall comply with all applicable laws and regulations, including without limitation those applicable to data privacy, intellectual property, trade secret, and fraud. Licensee is advised that the Technical Information is of U.S. origin and subject to the U.S. Export Administration Regulations ("EAR") and may be subject to applicable local country import/export laws and regulations. Diversion contrary to U.S. and/or applicable local country law and/or regulation is prohibited. Licensee agrees not to directly or indirectly export, re-export, import, download, or transmit the Technical Information to any country, end user or for any use that is contrary to applicable U.S. and/or local country regulation or statute (including but not limited to those countries embargoed by the U.S. government). Licensee represents that any governmental agency has not issued sanctions against Licensee or otherwise suspended, revoked or denied Licensee's import/export privileges. Licensee agrees not to use or transfer the Technical Information for any use relating to nuclear, chemical or biological weapons, or missile technology, unless authorized by the U.S. and/or any applicable local government by regulation or specific written license. Additionally, Licensee is advised that the Technical Information may contain encryption algorithm or source code that may not be exported to government or military end users without a license issued by the U.S. Bureau of Industry and Security and any other country's governmental agencies, where applicable.

13.0 WAIVER.

The failure to assert any rights under this Agreement, including, but not limited to, the right to terminate in the event of breach or default, will not be deemed to constitute a waiver of the right to enforce each and every provision of this Agreement in accordance with their terms.

14.0 SEVERABILITY.

If any provision of this Agreement is determined to be unenforceable or invalid, this Agreement will not be rendered unenforceable or invalid as a whole, and the provision will be changed and interpreted so as to best accomplish the objectives of the original provision within the limits of applicable law.

15.0GOVERNING LAW AND DISPUTE RESOLUTION.

15.1 Governing Law. This Agreement and any dispute, claim or controversy arising out of or relating to this Agreement ("Dispute"), including without limitation the formation, interpretation, breach or termination of this Agreement, or any issue regarding whether a Dispute is subject to arbitration under this Agreement, will be governed by New York State laws, excluding conflict of law principles, and the United Nations Convention on Contracts for the International Sale of Goods.

15.2 Dispute Resolution. Any Dispute will be resolved in accordance with the provisions of this Section 15. The disputing party shall give the other party written notice of the Dispute in accordance with the notice provision of this Agreement. The parties will attempt in good faith to resolve each controversy or claim within 30 days, or such other longer period as the parties may mutually agree, following the delivery of such notice, by negotiations between designated representatives of the parties who have dispute resolution authority.

15.3 Arbitration of Non-US Disputes. If a Dispute that arose anywhere other than in the United States or is based upon an alleged breach committed anywhere other than in the United States cannot be settled under the procedures and within the timeframe set forth in Section 15.2, it will be conclusively determined upon request of either party by a final and binding arbitration proceeding to be held in accordance with the Rules of Arbitration of the International Chamber of Commerce by a single arbitrator appointed by the parties or (failing agreement) by an arbitrator appointed by the President of the International Chamber of Commerce (from time to time), except that if the aggregate claims, cross claims and counterclaims by any one party against the other party exceed One Million US Dollars at the time all claims, including cross claims and counterclaims are filed, the proceeding will be held in accordance with the Rules of Arbitration of the International Chamber of Commerce by a panel of three arbitrator(s) appointed in accordance with the Rules of Arbitration of the International Chamber of Commerce. The arbitration will be conducted in the English language, at a location agreed by the parties or (failing agreement) ordered by the arbitrator(s). The arbitrator(s) will have authority only to award compensatory damages within the scope of the limitations of Section 8 and will not award punitive or exemplary damages. The arbitrator(s) will not have the authority to limit, expand or otherwise modify the terms of this Agreement. The ruling by the arbitrator(s)) will be final and binding on the parties and may be entered in any court having jurisdiction over the parties or any of their assets. The parties will evenly split the cost of the arbitrator(s)' fees, but Avaya and Customer will each bear its own attorneys' fees and other costs associated with the arbitration. The parties, their representatives, other participants and the arbitrator(s) will hold the existence, content and results of the arbitration in strict confidence to the fullest extent permitted by law. Any disclosure of the existence, content and results of the arbitration will be as limited and narrowed as required to comply with the applicable law. By way of illustration, if the applicable law mandates the disclosure of the monetary amount of an arbitration award only, the underlying opinion or rationale for that award may not be disclosed.

15.4 Choice of Forum for US Disputes. If a Dispute by one party against the other that arose in the United States or is based upon an alleged breach committed in the United States cannot be settled under the procedures and within the timeframe set forth in Section 15.2, then either party may bring an action or proceeding solely in either the Supreme Court of the State of New York, New York County, or the United States District Court for the Southern District of New York. Except as otherwise stated in Section 15.3 each party consents to the exclusive jurisdiction of those courts, including their appellate courts, for the purpose of all actions and proceedings arising out of or relating to this Agreement.

15.5 Injunctive Relief. Nothing in this Agreement will be construed to preclude either party from seeking provisional remedies, including, but not limited to, temporary restraining orders and preliminary injunctions from any court of competent jurisdiction in order to protect its rights, including its rights pending arbitration, at any time. The parties agree that the arbitration provision in Section 15.3 may be enforced by injunction or other equitable order, and no bond or security of any kind will be required with respect to any such injunction or order.

15.6 Time Limit. Actions on Disputes between the parties must be brought in accordance with this Section within 2 years after the cause of action arises.

16.0 AGREEMENT IN ENGLISH.

The parties confirm that it is their wish that the Agreement, as well as all other documents relating hereto, including all notices, have been and shall be drawn up in the English language only. Les parties aux présentes confirment leur volonté que cette convention, de même que tous les documents, y compris tout avis, qui s'y rattachent, soient rédigés en langue anglaise.

17.0 ENTIRE AGREEMENT.

This Agreement, its exhibits, schedules and other agreements or documents referenced herein, constitute the full and complete understanding and agreement between the parties and supersede all contemporaneous and prior understandings, agreements and representations relating to the subject matter hereof. No modifications, alterations or amendments shall be effective unless in writing signed by both parties to this Agreement.

18. REDISTRIBUTABLE CLIENT FILES.

The list of SDK client files that can be redistributed, if any, are in the SDK in a file called Redistributable.txt.

Schedule 1 to Avaya SDK License Agreement
Third Party Notices

1. **CODECS**: WITH RESPECT TO ANY CODECS IN THE SDK, YOU ACKNOWLEDGE AND AGREE YOU ARE RESPONSIBLE FOR ANY AND ALL RELATED FEES AND/OR ROYALTIES, IF ANY. IT IS YOUR RESPONSIBILITY TO CHECK.

THE H.264 (AVC) CODEC IS LICENSED UNDER THE AVC PATENT PORTFOLIO LICENSE FOR THE PERSONAL USE OF A CONSUMER OR OTHER USES IN WHICH IT DOES NOT RECEIVE REMUNERATION TO: (I) ENCODE VIDEO IN COMPLIANCE WITH THE AVC STANDARD ("AVC VIDEO") AND/OR (II) DECODE AVC VIDEO THAT WAS ENCODED BY A CONSUMER ENGAGED IN A PERSONAL ACTIVITY AND/OR WAS OBTAINED FROM A VIDEO PROVIDER LICENSED TO PROVIDE AVC VIDEO. NO LICENSE IS GRANTED OR SHALL BE IMPLIED FOR ANY OTHER USE. ADDITIONAL INFORMATION FOR THE H.264 (AVC) CODEC MAY BE OBTAINED FROM MPEG LA, L.L.C. SEE HTTP://WWW.MPEGLA.COM.

Table of Contents

Contents

Summary

The Avaya Analytics™ Streams Event Processors produce real time measures that are published to authenticated subscribers via the REST & WebSocket interfaces.

The REST interface handles administration and subscription requests whereas the WebSocket is purely used as a transport mechanism for real-time data.

The Event Processor applications produce measure data at a dimensional level and send updated real time measure values to Kafka topics which are filtered by the Data Publisher and then forwarded to WebSocket endpoints.

REST interface

This interface is available over HTTPS and provides the following APIs.

- User login
- List sources
- List producers
- Translations of the measure names for a given producer (multiple locales supported)
- Subscribe/unsubscribe to producers
- List dimensional data (list of agents)

WebSocket interface

This interface is available over HTTPS/WSS and provides the following functionalities:

- Read data from the Kafka topics of all the measure producers.
- Filter data as per subscriptions

Measure Processors

- Read UCM & UCA data from Kafka (inbound) topics
- Calculate metrics
- Write metrics in Kafka (outbound) topics
- Register as a measure producer with the REST interface.

Introduction

Intended audience

This document is for third-party vendors who want to develop applications that rely on real-time measure data feeds from Avaya Analytics™.

The protocols for subscribing for and consuming real-time data are described here.

Purpose

This document describes a new Open Interface for retrieving real-time data from Avaya Analytics™ 4.1.2.0.
This Open Interface is based on a combination of REST and WebSocket technologies and will replace the existing Open Interface based on Kafka in a future release.

Status the Kafka Open Interface.

Avaya will no longer make commercially available the Kafka interface for new sale opportunities once Avaya Oceana 3.8.1/Analytics 4.1.1 becomes generally available.

Existing Avaya Oceana/Analytics customers will continue to be supported.

Third-party vendors who plan to write an application connecting to Avaya Analytics should use the REST & WebSocket interface.

Compatibility with the Kafka Open Interface

The APIs for subscribing and consuming events introduced in the REST & WebSocket Open Interface are fundamentally different to the APIs defined in the Kafka Open Interface for Avaya Analytics™.

Third-party applications that already use the Kafka Open Interface are therefore not compatible with the REST & WebSocket Open Interface and will need to be rewritten to make use of it.

The format of the real-time data exposed by both Open Interfaces is identical.

Pre-Requisites

The reader of this document must be familiar with:

- Maintaining and Troubleshooting Avaya Analytics™ for Avaya Oceana® solution. See Applicable Documentation for Avaya Analytics™.
- Oceana® Call Routing Features and be familiar with Oceana® Attribute Based Routing and Administration of agents with account and attributes. See Applicable Documentation for Avaya Oceana® solution.
- Kubernetes. https://kubernetes.io/
- REST APIs. https://www.redhat.com/en/topics/api/what-is-a-rest-api
- WebSocket protocol. https://tools.ietf.org/html/rfc6455

Licensing

Avaya Analytics Real-Time interface is licensed as an add on option for Avaya Analytics that allows for a maximum of 5 subscriptions on Avaya Analytics Real-Time data.

Licenses must be purchased on a per site basis.

For access keys contact your local Avaya representative.

Support

For support contact http://avaya.com/devconnect

Terminology and Acronyms

Term	Meaning
ACM	Avaya Control Manager
Avaya Analytics™	Reporting and Analytics platform
AWA	Avaya Work Assignment
BHCC	Busy Hour Call Completion
Avaya Breeze® platform	Platform for deployment of snap-ins, used to manage a Contact Center
Crunchy DB	Crunchy Data PostgreSQL database
CCM	Cluster Control Manager
EWT	Expected Wait Time
ITD	Interval To Date
JMS	Java Message Service
JSON	JavaScript Object Notation
REF	Reliable Event Framework
UCA	Unified Collaboration Administration
UCAM	Universal Configuration and Management
UCID	Universal Call Identifier
UCM	Unified Collaboration Model

Avaya Analytics™ Overview

Architecture Overview

Avaya Analytics™ is deployed on top of Kubernetes, an open-source container orchestration solution.

At a high level, Avaya Analytics™ consumes events from the Avaya Oceana solution and is composed of the following services.

1. Event Processor Applications (event processors)
2. Kafka Real-Time Open Interface (Kafka open interface - to be deprecated)
3. Apache Kafka (real-time bus)
4. Publisher (WebSocket interface)
5. REST interface
6. Redis (in-memory database)
7. Authentication service (bridge to breeze authentication service)
8. Crunchy DB (historical data store)
9. Avaya Analytics Historical Reporting (presentation layer)

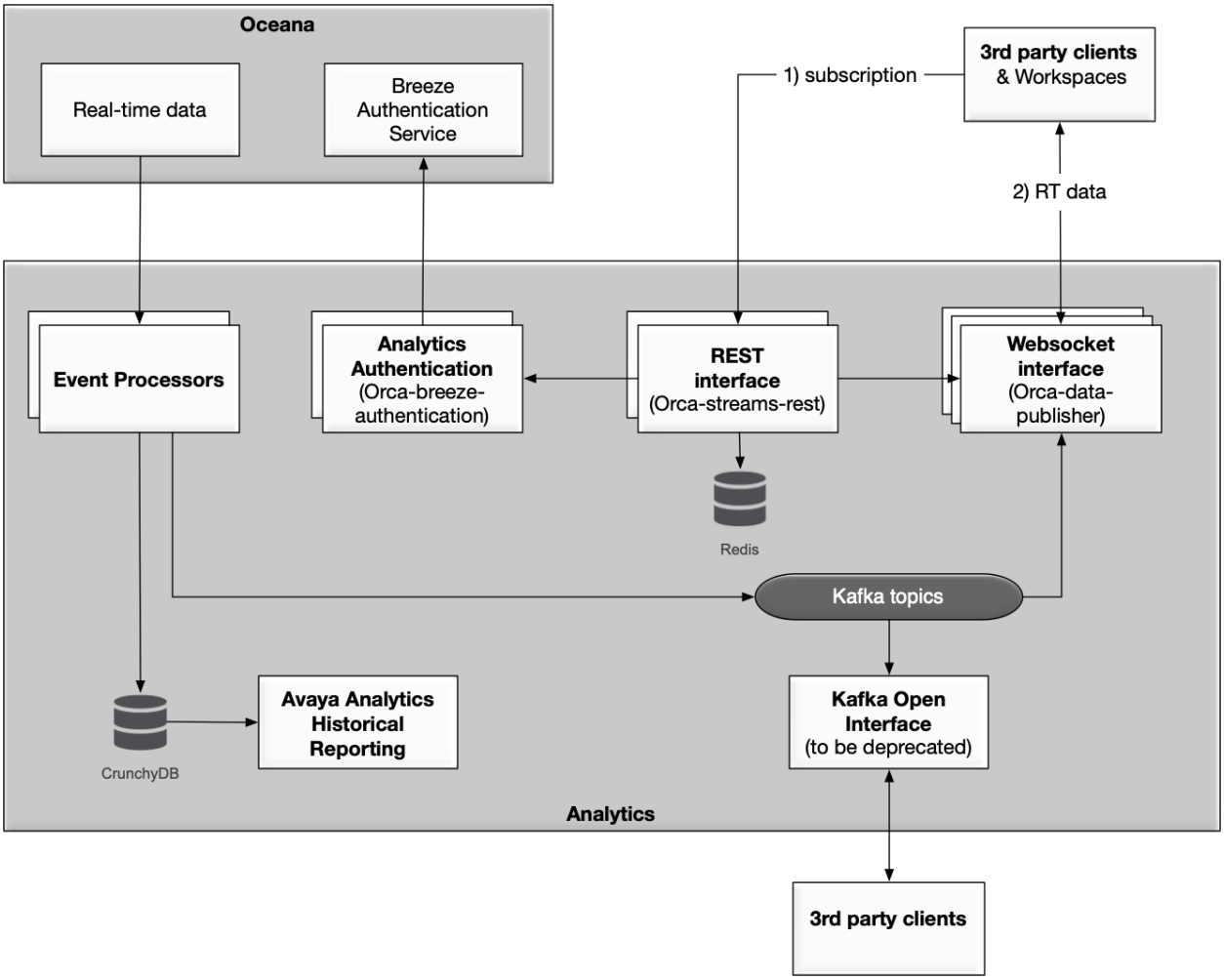


Figure 1 – Avaya Analytics™ Architecture

The Kafka Open Interface shown in the diagram will be deprecated in a future release of Avaya Analytics™.

The Open Interface described in this document is composed of two entities:

2. The REST interface.
3. The WebSocket interface.

Event Processor

The Event Processors consist of a set of Kubernetes pods which process events received from the Avaya Oceana solution and to calculate measures.

Each measure is independent and is defined by specific events relating to that measure.

Real Time and Historical Data Store

The Event Processor (EP) pushes data to a Crunchy database deployed as part of Avaya Analytics™. Historical data is pushed to the Avaya Analytics™ data store while the real time data is pushed to Kafka topics.

The initial contact point for the push of Historical data is a staging area which has a simple schema consisting of tables (representing the dimensional combinations output by the EP), where each table consists of a number of columns (representing the specific measures output by the EP within those dimensions).

Persisted Historical data includes:

- 15-minute fixed interval records
- Contact Detail Records

Subsequent processing aggregates the 15-minute fixed interval to larger grains (e.g. daily, weekly, and monthly). Other processing may also add further processing to generate new measures as the data is moved from the staging area through to an Analytics data mart which acts as the data provider for the presentation layer.

Presentation Layer

Avaya Analytics Historical Reporting is the historical data presentation layer for Avaya Analytics™. This component includes internal modelling components to map from the physical data mart schema and JMS topics through a business model and finally into several Subject Areas - these are the artifacts that the report creator and consumer of Avaya Analytics Historical Reporting visualizations interacts with.

Avaya Analytics Historical Reporting acts as a repository of their respective models, and also their visual widgets and reports (whether output to a screen, printed or emailed).

Avaya Analytics server

The Event Processor applications write real time data to dedicated Kafka topics on the Analytics server.

The Kafka Open Interface application manages the access to the real time data from third-party clients. The mechanisms used by third-party clients to subscribe and consume data is described in the Avaya Analytics™ Real Time Kafka Open Interface API Guide.

The REST/WebSocket Open Interface exposes a REST interface for querying the list of producers configured in the system and subscribing for their data feed. The WebSocket interface transfers the real-time data to the third-party clients.

Workspaces presentation layer

Workspaces uses the REST interface to retrieve the list of producers configured in the system and to subscribe to the data feed. The real-time data generated on the Kafka topics by the measure producers are transferred to the WebSocket endpoint where the presentation layer is registered as a consumer.

Measures

For the full list of Measures available please refer to the *Avaya Analytics™ Data Dictionary 4.1.2.0*

REST & WebSocket Open Interface Specification

Overview

The Avaya Analytics™ Streams Event Processors produce real time measures that are published to authenticated subscribers via the REST/WebSocket interface.

The REST interface handles administration and subscription requests whereas the WebSocket is purely used as a transport mechanism for real-time data.

The Event Processor applications produce measure data at a dimensional level and send updated real time measure values to Kafka topics which are filtered by the Data Publisher and then forwarded to WebSocket endpoints.

High Level Architecture

At a high level, the REST/WebSocket interface relies on 2 main components, the REST interface and the WebSocket (a.k.a data-publisher) interface.

However, more components are a play and below is a short presentation of their responsibilities.

REST interface

This interface is available over HTTPS and provides the following APIs.

1. User login
2. List sources
3. List producers
4. Provide translations of the measure names for a given producer (multiple locales supported)
5. Subscribe/unsubscribe to producers
6. List dimensional data (list of agents)

WebSocket interface

This interface is available over HTTPS and provides the following functionalities:

1. Read data from the Kafka topics of all the measure producers.
2. Filter data as per subscriptions

Redis

1. Store information regarding the measure producers on behalf of the REST interface (Sources, producers, dictionaries, dimension data)
2. Store third-party client Subscription Redirect Requests on behalf of the Data Publisher.

Third-party applications do not interact directly with the Redis database.

Orca-breeze-authentication

1. Connect to the Avaya Oceana Breeze Authentication for authenticating users.
2. Validate tokens

This component is used by the REST interface to login users and validate authentication tokens. Third-party applications do not interact directly with it.

Measure Processors

1. Read UCM & UCA data from Kafka (inbound) topics
2. Calculate metrics
3. Write metrics in Kafka (outbound) topics
4. Register as a measure producer with the REST interface.

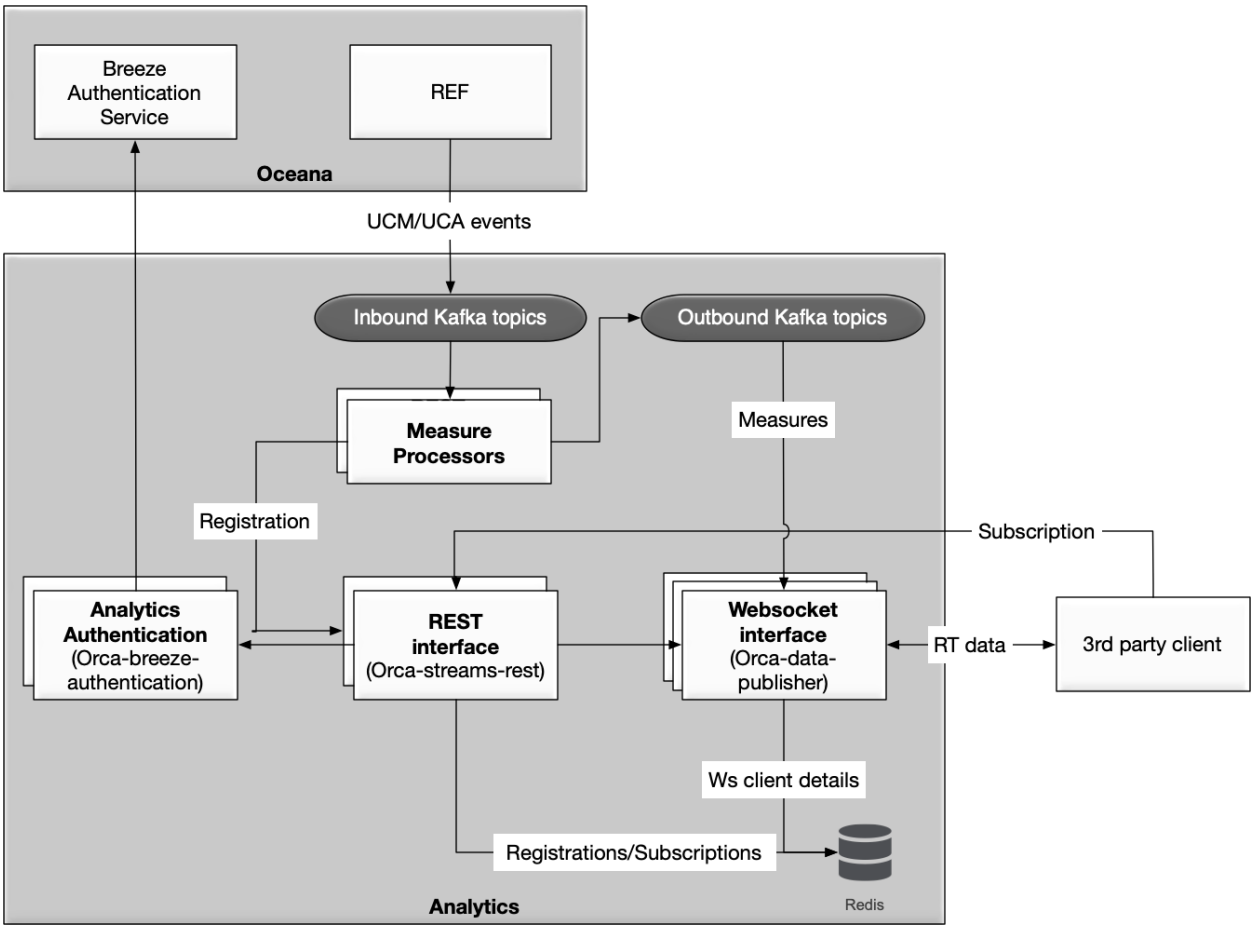


Figure 2 – Avaya Analytics™ Detailed Architecture

Detailed Description

Data exposed via the Open Interface

The following dimensions and intervals are supported on the real time interface

Dimension	Intervals
Agent	Start of Day
Agent By Account	Start of Day
Agent By Routing Service	Start of Day
Routing Service	Start of Day
Routing Point	Start of Day
Site	Start of Day
Agent By Not Ready Reason Code	Start of Day
Agent Group	Start of Day
Contact Details	Start of Day
Routing Service Group	Start of Day

For the full list of Measures available please refer to the *Avaya Analytics™ Data Dictionary 4.1.2.0*

All real-time measures for each of the intervals and dimensions are exposed on the open interface. Start of Day is 12.00 am server time.

Data is published on the real-time interface when the measure is updated.

ACM manages the creation of users that can subscribe for data, manages groups of agents, groups of routing services and assigns groups to supervisors.

Capacity considerations

The Rest & WebSocket Open Interface supports the load defined in the Avaya Oceana® solution Release 4.1.2.0 capacity requirements with the following limitations:

- Supports 10 groups of agents
- Supports 5 concurrent supervisors

Clients may subscribe simultaneously to all the measure streams configured in the system.

Requests sent to the REST interface are expected to be staggered by 100ms.

Each supervisor has its own WebSocket connection, and all the traffic for all the measure streams are sent over that connection. Each WebSocket message can be up to 64Kb in size.

REST APIs

The REST APIs exposed by the Orca Streams server are aimed at two types of applications:

1. **Producers**: These applications produce real-time measures and register themselves with the Analytics Streams Server as “producers”. They are also referred to as measure producers.
2. **Consumers**: These applications subscribe to the Analytics Streams Server to consumer the real-time date published by the Measure Producers. Workspaces is an example of Consumer application.

Producers and Consumers have different needs in terms of REST APIs and therefore use different endpoints. The primary focus of this document is to detail how to use the API from a Consumer point of view and will not cover the APIs used by Producers.

For a complete description of the REST endpoints, including the definitions of the data structures, please refer to the `openinterface-api.yaml` provided with the WebSocket client for Real-Time Open Interface on DevConnect. For more information on how to use the `openinterface-api.yaml` file, please consult https://swagger.io/

Connectivity

The REST and WebSocket interfaces are only accessible over SSL through the ingress gateway of the cluster.

To find the IP address of the ingress gateway, connect to the Cluster Control Manager (CCM) over SSH, switch to the root user and run this command:

```
kubectl get configmap common-services-kube-keepalived-vip -o yaml
```

The IP address of the ingress gateway is displayed on the third line of the output.

To get FQDN for the ingress gateway run this command:

```
nslookup <ingress-gw-ip>
```

All the REST interface APIs are accessible under the `orca-streams-rest` domain. As an example, here is a curl command for sending a login request to the REST endpoint on a lab where the ingress gateway is at IP 10.134.44.134.

```
curl -k --location --request GET 'https://10.134.44.134:443/orca-streams-
rest/users/login?tenant=0&username=cplabagentone@async.galwaylab.avaya
.com&password=abcd123'
```

For simplicity, all the endpoints listed in the rest of this document show no IP address, port or domain.

Subscription Management

The third-party application starts by sending supervisor's username and password to the REST's login endpoint, and once they are validated, receives a token which **must be passed with all subsequent HTTP requests.**

The application then sends a Subscription Request object over REST and receives a Subscription Response in return which contains the WebSocket URL used for sending real-time data.

- A unique subscription is required per dimension. For example, a user needs to send a subscription for Agent By Account, and another subscription for Routing Services.
- The user that was passed in as part of the subscription is validated against the list of supervisors configured in ACM and assigned an Avaya Analytics™ group.
- The agent data published on the WebSocket is based on the Avaya Analytics™ group assigned to the supervisor in ACM.

If an agent is removed from an Avaya Analytics™ Group in ACM, then that agent is removed from the stream of data feeding to all supervisors assigned to that group. A delete event is sent to the client.

If an agent is added to an Avaya Analytics™ Group, then that agent is added to the stream of data feeding to all supervisors assigned to that group. No added event is sent to the client.

If a group is removed from a supervisor, then all agents unique to that group are removed from the stream of data feeding to all supervisors assigned to that group. No delete event is sent to the client.

If a group is added to a supervisor, then all agents unique to that group are added to the stream of data feeding to all supervisors assigned to that group. No added event is sent to the client.

If a supervisor is deleted, all subscriptions and Kafka topics assigned to the user are deleted.

There are heartbeats on each WebSocket endpoint – see table "heartbeat message is sent every 8 seconds on the WebSocket endpoint that was received in the subscription response.

All communications with the REST and WebSocket interfaces are encrypted over SSL.

Data Filtering and Access Control

Users only see agents configured in ACM that they have access to. On subscription to the REST interface, their user credentials are sent, and Avaya Analytics™ checks if that user is administered in ACM and assigned to Avaya Analytics™ groups.

The subscription does not allow for any further filtering of data. It is not possible to subscribe for an individual Avaya Analytics™ group or specify specific agents of interest.

The subscription does not allow for the client to specify a subset of the measures available for a dimension – the client receives all measures for that dimension.

At present, the WebSocket interface provides a mechanism for filtering Routing Services which is enabled by setting this flag in the ConfigMap for orca-streams-data-publisher
`OCEANA.websocketSender.enableRoutingServiceGroupFilter=true`

Functional Model

Authorization token

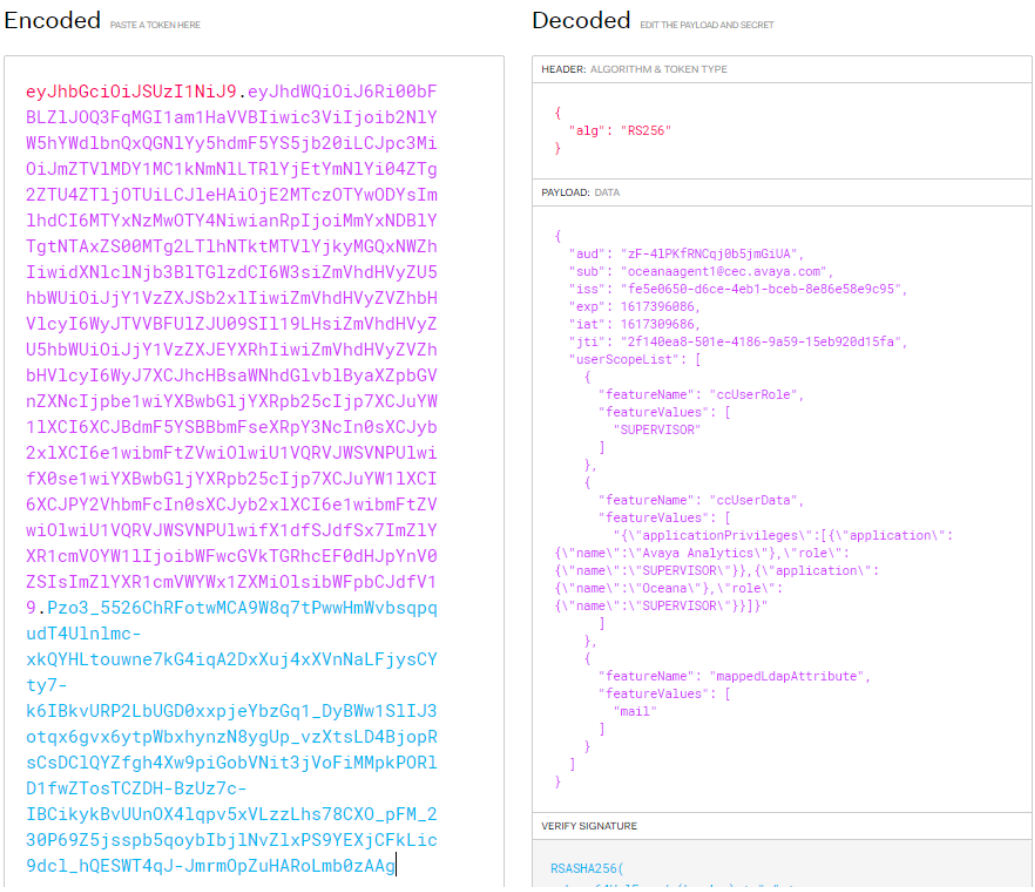
The third-party client starts by sending an HTTP GET request to the `/user/login` endpoint containing valid supervisor credentials for whom real-time data are being sought. The REST server checks the credentials received and, if successful, returns an Authorization token in the "authdata" field of the response.

The third-party application must pass this Authorization token to all subsequent REST API requests as part of the header, and to any WebSocket connection requests.

The token is a JSON Web Token (JWT) defined by open standard RFC 7519. The token is Base64Url encoded, but it is not encrypted. The third-party application is responsible for validating the token's signature to guarantee the token has not been tampered with. Examples of token decoding and signature validation are available for all common programming languages at this address: https://jwt.io/

Example of JWT token

The screenshot below shows the base64 encoded JWT token on the left, and the decoded equivalent on the right.



Subscribing to real-time data

Overview

Subscribing to a real-time data feed is a multi-stage process.

First, the third-party application retrieves the list of Sources and Producers configured in the system.

Second, the application sends a Subscription Request and in response, receives the URL of the WebSocket used for sending real-time data.

Third, the application sends a SubscriptionRedirect request over to the WebSocket to initiate the transfer of real-time data.

Sources

The third-party client sends an HTTP GET request to the `/sources` endpoint to retrieve a list of sources configured in the system. In Avaya Analytics ™ 4.1.2.0, there is currently only one source configured. The Source object, like all the resources returned by the REST interface, is in JSON format.

The most important field in the Source object is the "**sourceId"**, which the third-party application will need to query the list of Producers attached to this source.

Producers

The client then sends an HTTP GET request to the /`sources/{sourceId}/producer` endpoint to retrieve the list of measure producers attached to the Source.

At a high level, a Producer is composed of a header, dimensions, measures and locales.

Header

The header is composed of the following information:

Name	Description
sourceId	The unique id of the source this producer is attached to.
producerId	The unique id of the producer.
producerName	The name of the producer.
streamType	The only streamType currently supported is Start Of Day (SoD).
description	A description field provided by the Measure Producer.
version	The version of the provider.

Dimensions

Name	Description
name	Name of the dimension. For example: routingServiceName
type	The type associated with this dimension. For example: String
value	The value associated with the dimension. s
displayName	The display name for the dimension
description	The description for the dimension

Measures

Name	Description
name	The name of the measure. For example: “abandoned”
type	The type of the measure. For example: String, Number, Duration
format	For example: empty field, String, Number, Seconds
values	For example: Voice, SMS, Email
displayName	The default display name. For example: “Abandoned”
description	A description for the measure.

Locales

This is an array of strings where each entry contains the country code of a supported locale. Avaya Analytics™ support the following locales:

Code	Locale
de	German
en-us	English - United States
es	Spanish
fr	French
it	Italian
ja	Japanese
ko	Korean
pt_BR	Portuguese - Brazil

ru	Russian
zh-cn	Chinese - China
zh-tw	Chinese - Taiwan

Subscription request

The third-party application sends an HTTP POST request to `/source/{sourceId}/producer/{producerId/subscriptions` to subscribe to a real-time data feed. The response is a SubscriptionResponse containing the URL of the WebSocket used to publish the real-time data.

The structure of the SubscriptionRequest is as follows:

Name	Description
sourceId	The id of the source we want to subscribe to.
producerId	The id of the producer we want to subscribe to.
streamType	The stream type. Currently, “SoD” is the only supported value. SoD stands for Start of Day.
transport	Currently, “websocket” is the only supported value.
tenantId	The tenantId field is not currently used. Set this field to “0”.
username	The username of the supervisor.
dimensionFilters	An array of dimension filters to apply before sending the real-time data over the WebSocket interface. (not currently used)
measureFilters	An array of measure filters to apply before sending the real-time data over the WebSocket interface. (not currently used)

Filters

The dimensionFilters and measureFilters share the same data structure, which is defined as follows:

Name	Description
name	Name of the dimension or measure this filter is applicable to.
Filters	An array of Strings containing the values to be filtered.

If the subscription is successful, the REST interface sends back a SubscriptionResponse object which is defined as follows:

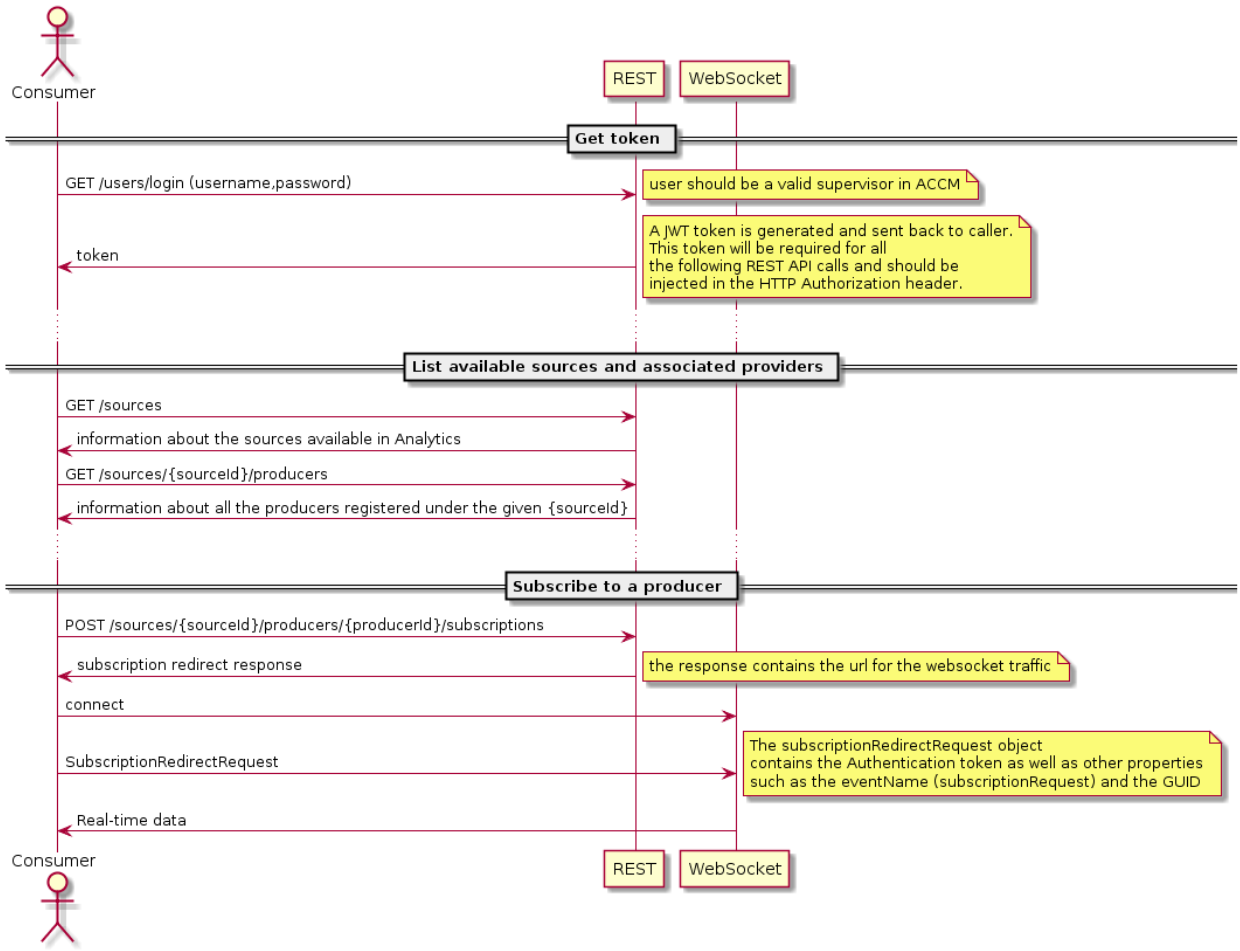
Name	Description
source	The name of the source subscribed to.
sourceId	The id of the source subscribed to.
producer	The name of the producer subscribed to.
producerId	The id of the producer subscribed to.
streamType	The type of streams subscribed to. Currently, only Start of Day (SoD) is supported.
version	The version of the producer subscribed to
transport	Currently, only “WebSocket” is supported.
endpoint	The WebSocket URL used to publish real-time data
guid	A unique ID generated by the WebSocket interface.

WebSocket connection

The third-party application connects to the WebSocket URL specified in the SubscriptionResponse and sends a SubscriptionRedirectRequest message defined as follows:

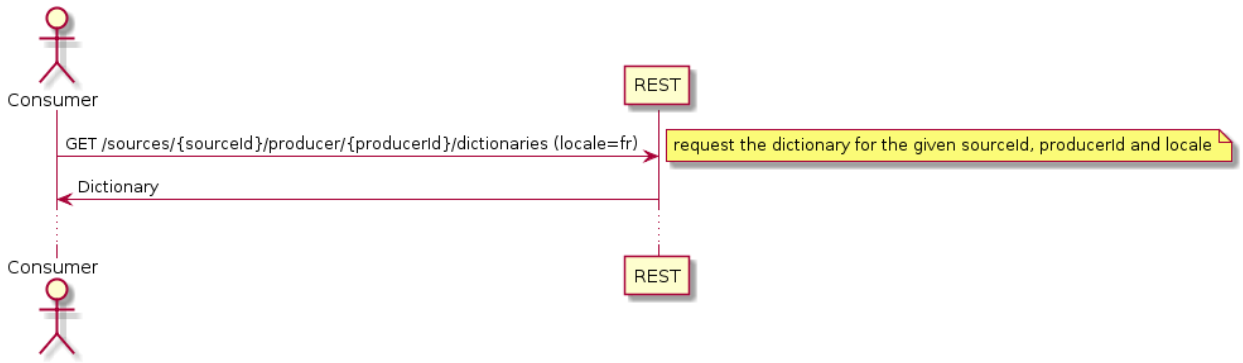
Name	Description
authorization	JWT token
eventName	This parameter is a string and its value can be: - subscriptionRedirect - unsubscriptionRedirect
source	Copy the source value from the SubscriptionResponse
sourceId	Copy the sourceId value from the SubscriptionResponse
producer	Copy the producer value from the SubscriptionResponse
producerId	Copy the producerId value from the SubscriptionResponse
streamType	Copy the streamType value from the SubscriptionResponse
version	Copy the version value from the SubscriptionResponse
transport	Copy the transport value from the SubscriptionResponse
endpoint	Copy the endpoint value from the SubcriptionResponse.
guid	Copy the GUID value from the SubscriptionResponse.
currentGuids	For internal use only. Do not fill in.

Subscription overall sequence diagram



Retrieving the Dictionary

Each Producer is associated with a dictionary that provides the necessary translations to build a fully internationalized user interface. To retrieve a dictionary for a particular locale, the third-party application sends an HTTP GET `/sources/{sourceId}/producer/{producerId}/dictionaries` request and sets the “locale” query parameter to the desired value.



Retrieving Dimension Data

AdminData is a special Measure Producer in charge of exposing Administrative information also called Dimension Data.

List of Dimension Data available:

1. Agents
2. Accounts
3. Groups
4. Group Members
5. Group Privileges
6. Reason Codes
7. Routing Points
8. Routing Services
9. Channel

To receive dimension data, third-party applications send an HTTP GET request to `/sources/{sourceId}/producers/Admin_Admin_3.5/dimdata/{dimDataName}/datatype/{dimDataType}`

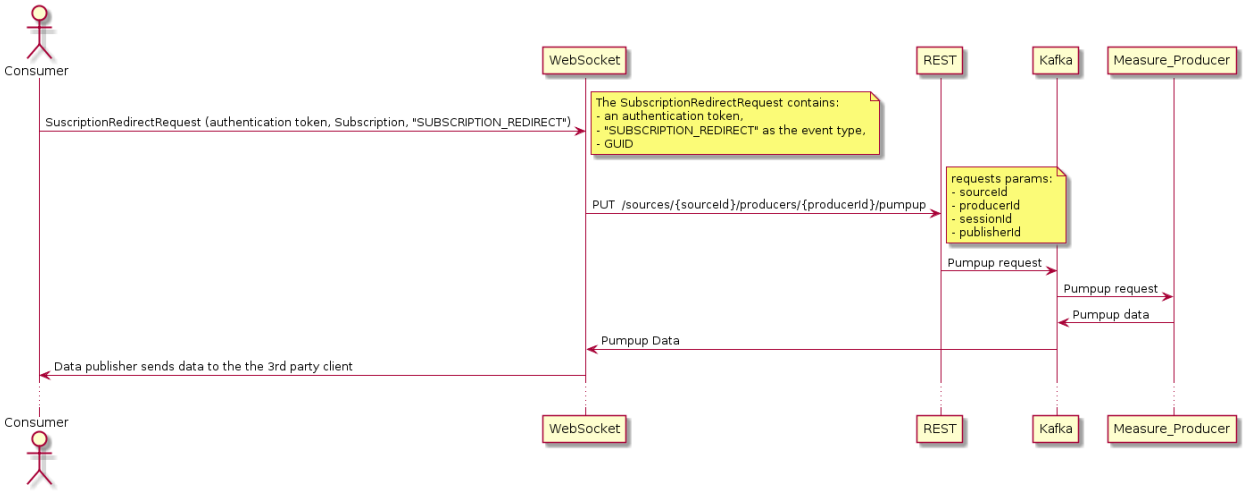
Please note that in this release the values for dimDataName and dimDataType need to be identical. Possible values include:

1. Agent
2. Account
3. Group
4. GroupMember
5. GroupPrivilege
6. ReasonCode
7. RoutingPoint
8. RoutingService
9. Channel

Pumpup request

Pumpup requests are sent by the third-party application to the WebSocket interface which then issues a specialised request on the REST interface.

Third-party clients should not send a Pumpup request directly to the rest interface.



Real Time Data Processing

The real-time data sent by the Measure Producers over the WebSocket interface are in JSON format and will continue to be sent until the third-party client either unsubscribe, or the client WebSocket connection is closed.

Unsubscribing

Unsubscribing from the Open Interface is a 3-stage process.

The first stage consists in sending a SubscriptionRedirectRequest to the WebSocket interface. The Unsubscription object is similar to the Subscription object used to initiate the real-time data feed and is described below.

Name	Description
authorization	JWT token
eventName	This parameter is a string and its value must be: “unsubscriptionRedirect”
source	Copy the source value from the SubscriptionResponse
sourceId	Copy the sourceId value from the SubscriptionResponse
producer	Copy the producer value from the SubscriptionResponse
producerId	Copy the producerId value from the SubscriptionResponse
streamType	Copy the streamType value from the SubscriptionResponse
version	Copy the version value from the SubscriptionResponse
transport	Copy the transport value from the SubscriptionResponse
endpoint	Copy the endpoint value from the SubcriptionResponse.
guid	Copy the GUID value received in the SubscriptionResponse.
currentGuids	For internal use only. Do not fill in.

The second stage consists in sending an HTTP DELETE request to the `/source/{sourceId}/producer/{producerId/subscriptions` REST endpoint. No additional information is necessary to complete this stage.

Lastly, the third-party application can safely disconnect from the WebSocket.

Heartbeat and pumpup messages

When a subscription request has been successful, a heartbeat message is sent every 8 seconds on the WebSocket interface.

The following table defines the details for **heartbeat** data message:

Measures Stream	Message Format (JSON)
All streams	{"dimension":{},"realtimeData":{},"pumpup":false,"pumpupComplete":false,"heartbeat":true}

If a pumpup of data is requested, the current measure data is sent on the WebSocket. When all relevant data has been sent, a “pumpup complete” message is sent. See below for details of the data in a pumpup.

The following table defines the details for **pumpup complete** data message:

Measures Stream	Message Format (JSON)
All streams	{"dimension":{},"realtimeData":{},"pumpup":true,"pumpupComplete":true,"heartbeat":false}

Third-party clients must ensure they sent a “--heartbeat--” message to the WebSocket interface every 8 seconds to keep the connection alive.

The server will terminate the WebSocket connection if it fails to receive a heartbeat from the client.

If required, the timeout for the heartbeat can be set by editing the value of `OCEANA.websocketSender.heartbeatIntervalSeconds` in the ConfigMap for `orca-streams-data-publisher` using this command:
`k edit cm orca-streams-data-publisher`

Beware that updating the timeout value will affect other clients such as Workspaces.

Recovering from the loss of connectivity

The client should try to reconnect to the server in the event of the WebSocket connection going down. The process for reconnecting involves resubscribing to the REST interface and reconnecting to the WebSocket endpoint.

In the case of a HA node or pod failover, the third-party client will detect the connection is down and will repeatedly retry the connection until it is restored. In case of one `orca-streams-data-publisher` pod going down, the connection will be restored to the remaining functioning `orca-streams-data-publisher` pod. Kubernetes will take care of starting a new instance of the `orca-streams-data-publisher` pod to maintain the desired replica count defined in the deployment. The same principle applies to the `orca-streams-rest` pod.

Error codes

The REST interface uses the error codes defined inside the `openinterface-api.yaml` file provided with the sample client.

The WebSocket interface uses the following errors codes:

- 200 OK
- 500 Internal Server Error. The error text contains an explanation for the error. Typically, the corresponding stack trace is available in the log file of the `orca-streams-data-publisher`.

Open Interface Measures Streams and Data

For the full list of Measures available please refer to the Avaya Analytics™ *Data Dictionary 4.1.2.0*

The following table defines the details for measures data messages:

List of producers:

1. Agent
2. AgentByAccount
3. AgentGroup
4. Agent By Not Ready Reason Code
5. ContactDetail
6. RoutingService
7. RoutingServiceByAgent
8. Site
9. VDN
10. RoutingServiceGroup

Producer name	Message Format	Example
Agent	{"dimension":{"agentId":"agentId"},"realtimeData":{"measure":value...},"pumpup":true | false,"pumpupComplete":true | false, heartbeat":true | false} dimension : Map<String, String> realtimeData : Map<String, Object> pumpup : Boolean pumpupComplete : Boolean heartbeat : Boolean	{"dimension":"{\"agentId\":\"traffic181\"}","realtimeData":"{\"offered\":1358,\"notAnswered\":1358}","pumpupIdentifier":null,"pumpup":false,"pumpupComplete":false,"heartbeat":false,"extra":null,"sourceId":"Oceana_Streams_127.0.0.1_3.5","producerId":"Agent_SoD_3.5","eventAction":"UPSERT"} Note: This example was taken from a traffic lab where the agent was not answering the contacts presented to him. This is clearly visible here as the number of “offered” contacts is equal to the number of “not_answered”. The “pumpupIdentifier” field is null because “pumpup” is false. The “extra” field is a placeholder for metric timestamps. This is an Avaya

Producer name	Message Format	Example
		internal feature which is not supported for customer use.
Agent By Account	{"dimension":{"agentId":"agentId","accountId":"accountId", "channelId":"channelId"}, "realtimeData":{"measure":value...},"pumpup":true | false,"pumpupComplete":true | false, heartbeat":true | false } dimension : Map<String, String> realtimeData : Map<String, Object> pumpup : Boolean pumpupComplete : Boolean heartbeat : Boolean	{"dimension":{"accountId":"8881001","agentId":"888101","channelId":"Voice"},"realtimeData":{"activeWorkCount":"0","lastStateReasonTimestamp":"00000001471351598494"},"pumpup":false,"pumpupComplete":false, heartbeat": false }
Agent Group	{"dimension":{"groupid":"groupid"},"realtimeData":{"measure":value…},"pumpup":true | false,"pumpupComplete":true | false, heartbeat":true | false } dimension : Map<String, String> realtimeData : Map<String, Object> pumpup : Boolean pumpupComplete : Boolean heartbeat : Boolean	{"dimension":"{\"groupId\":\"10003\"}","realtimeData":"{\"acwDuration\":7,\"shortAcw\":4}","pumpupIdentifier":null,"pumpup":false,"pumpupComplete":false,"heartbeat":false,"extra":null,"sourceId":"Oceana_Streams_127.0.0.1_3.5","producerId":"AgentGroup_SoD_3.5","eventAction":"UPSERT"}
Agent By Not Ready Reason Code	{"dimension":{"agentId":"agentId","nrReasonCode":"nrReasonCode"},"realtimeData":{"measure":value…},"pumpup":true | false,"pumpupComplete":true | false, heartbeat":true | false } dimension : Map<String, String> realtimeData : Map<String, Object> pumpup : Boolean pumpupComplete : Boolean heartbeat : Boolean	{"dimension":{"agentId":"nbolshak","nrReasonCode":"101"},"realtimeData":{"nrReasonCodeOccurrence":1},"pumpup":false,"pumpupComplete":false,"heartbeat":false}
Contact Details	{"dimension":{"segmentid":"segmentid"},"realtimeData":{"measure":value…},"pumpup":true | false,"pumpupComplete":true | false, heartbeat":true | false } dimension : Map<String, String>	{"dimension":"{\"segmentId\":\"dab814ea-316d-439e-9364-8652b2cb3aa0\"}","realtimeData":"{\"initialDisposition\"

Producer name	Message Format	Example
	realtimeData : Map<String, Object> pumpup : Boolean pumpupComplete : Boolean heartbeat : Boolean	:\"UNKNOWN\",...,\"workType\":\"DEFAULT\"}","pumpupIdentifier":null,"pumpup":false,"pumpupComplete":false,"heartbeat":false,"extra":null,"sourceId":"Oceana_Streams_127.0.0.1_3.5","producerId":"ContactDetail_CDR_3.5","eventAction":"UPSERT"}
Routing Service	{"dimension":{"routingServiceName":"routingServiceName"},"realtimeData":{"measure":value...},"pumpup":true | false,"pumpupComplete":true | false, heartbeat":true | false } dimension : Map<String, String> realtimeData : Map<String, Object> pumpup : Boolean pumpupComplete : Boolean heartbeat : Boolean	{"dimension":{"routingServiceName":"ChatRoutingService"},"realtimeData":{"contactsWaiting":0,"active":0,"heldContacts":0,"contactsAtAgent":1,"alerting":1},"pumpup":true,"pumpupComplete": true, heartbeat": false }
Routing Service By Agent	{"dimension":{"agentId":"agentId","routingServiceName":"routingServiceName"},"realtimeData":{"measure":value...},"pumpup":true | false,"pumpupComplete":true | false, heartbeat":true | false } dimension : Map<String, String> realtimeData : Map<String, Object> pumpup : Boolean pumpupComplete : Boolean heartbeat : Boolean	{"dimension":{"agentId":"8881001","routingServiceName":"ChatRoutingService"},"realtimeData":{"offered":1,"alertDuration":109},"pumpup":false,"pumpupComplete":false, heartbeat": false }
Site	{"dimension":{"siteName":"siteName","siteId":"siteId"},"realtimeData":{"measure":value...},"pumpup":true | false,"pumpupComplete":true | false, heartbeat":true | false } dimension : Map<String, String> realtimeData : Map<String, Object> pumpup : Boolean pumpupComplete : Boolean heartbeat : Boolean	{"dimension":"{\"siteName\":\"traffic2\",\"siteId\":\"4981\"}","realtimeData":"{\"replied\":101206}","pumpupIdentifier":null,"pumpup":false,"pumpupComplete":false,"heartbeat":false,"extra":null,"sourceId":"Oceana_Streams_127.0.0.1_3.5","producerId":"Site_SoD_3.5","eventAction":"UPSERT"}

Producer name	Message Format	Example
VDN	{"dimension":{"routingPointId":"routingPointId "},"realtimeData":{"measure":value...},"pumpup":true | false,"pumpupComplete":true | false, heartbeat":true | false } dimension : Map<String, String> realtimeData : Map<String, Object> pumpup : Boolean pumpupComplete : Boolean heartbeat : Boolean	{"dimension":"{\"routingPointId\":\"5003\"}","realtimeData":"{\"routePointDisconnects\":1,\"routingDuration\":1,\"routePointHandled\":1}","pumpupIdentifier":"cplabagentone@async.galwaylab.avaya.com_2_ROUTINGPOINTMEASURES_1613054564915","pumpup":true,"pumpupComplete":false,"heartbeat":false,"extra":null,"sourceId":"Oceana_Streams_127.0.0.1_3.5","producerId":"VDN_SoD_3.5","eventAction":"UPSERT"}
Routing Service Group	{"dimension":{"groupid":"groupid"},"realtimeData":{"measure":value…},"pumpup":true | false,"pumpupComplete":true | false, heartbeat":true | false } dimension : Map<String, String> realtimeData : Map<String, Object> pumpup : Boolean pumpupComplete : Boolean heartbeat : Boolean	{"dimension":"{\"groupId\":\"10037\"}","realtimeData":"{\"shortAcw\":3}","pumpupIdentifier":null,"pumpup":false,"pumpupComplete":false,"heartbeat":false,"extra":null,"sourceId":"Oceana_Streams_127.0.0.1_3.5","producerId":"RoutingServiceGroup_SoD_3.5","eventAction":"UPSERT"}

Sample Measure Stream Message Data

For the full list of Measures available please refer to the *Avaya Analytics™ Data Dictionary 4.1.2.0*

Agent goes Ready/Not Ready

The following example contains the AGENTMEASURES Measure Stream messages/events produced when an agent goes READY/NOT READY. Includes

- AGENTMEASURES Messages for pumpup of data
- Followed by Agent 'scollins' going READY/NOT READY

PUMPUP

1. {"dimension":{"agentId":"scollins"},"realtimeData":{"consultsAccepted":0,"transferred":0,"abandonedFromAlerting":0,"adHocDuration":0,"disconnectsFromHold":0,"bargedIn":0,"abandonTimeDuration":0,"observed":0,"transferredAcceptedFromService":0,"blendedActiveDuration":0,"agentLogonDuration":0,"acw":0,"additionalWorkDuration":0,"blendedAlertDuration":0,"observedDuration":0,"conferencedInitiated":0,"blendedAlert":0,"abandonedFromQueue":0,"coaching":0,"adHoc":0,"abandoned":0,"shortEngagements":0,"consultedDuration":0,"totalDuration":0,"ringTimeDuration":0,"transferredInitiated":0,"conferencedAccepted":0,"longHolds":0,"transferredInitiatedToAgent":0,"completed":0,"additionalWork":0,"answeredAfterThreshold":0,"consults":0,"blendedActive":0,"longAcw":0,"idleTimeDuration":62,"abandonedAfterThreshold":0,"acwDuration":0,"bargedOutDuration":0,"alertDuration":0,"totalActiveTimeDuration":0,"contacthandlingDuration":0,"transferredInitiatedToService":0,"observingDuration":0,"totalHoldTimeDuration":0,"conferenced":0,"coached":0,"acwExtended":0,"longEngagements":0,"consultsInitiated":0,"transferredToService":0,"agentNotReady":4,"handlingDuration":0,"activeTimeDuration":0,"transferredAcceptedFromAgent":0,"coachingDuration":0,"offered":0,"bargedInDuration":0,"shortNotReady":0,"transferredToAgent":0,"holdDuration":0,"agentReady":3,"consultingDuration":0,"consultDuration":0,"notAnswered":0,"answered":0,"agentNotReadyTimeDuration":4025,"transferredAccepted":0,"shortAcw":0,"bargedOut":0,"coachedDuration":0,"totalWaitTime":0,"holds":0,"observing":0},"pumpup":true,"pumpupComplete":false,"heartbeat":false }

2. {"dimension":{"agentId":"scollins"},"realtimeData":{"active":0},"pumpup":true,"pumpupComplete":false,"heartbeat":false }

3. {"dimension":{"agentId":"scollins"},"realtimeData":{"agentLogoutTimeStamp":"UNKNOWN","agentFirstName":"Siobhan","supervisorFirstName":"Morris","agentLastName":"Collins","lastStateChangeTimestamp":"00000001496919128431","supervisorId":"tmorris","supervisorLastName":"Tommy","nrReasonCode":"0","agentId":"scollins","lastStateReasonTimestamp":"00000001496919128431","loginTimeStamp":"UNKNOWN","accountAddress":"UNKNOWN","workLimit":"1","agentDisplayName":"Siobhan Collins","agentState":"NOT_READY","workState":"UNAVAILABLE","lastWorkCodeChangeTimestamp":"00000001496919128117","nrReasonCodeName":"DEFAULT","lastStateReasonTimestamp":"00000001496919128431","activeWorkCount":"0"},"pumpup":true,"pumpupComplete":false,"heartbeat":false }

4. {"dimension":{},"realtimeData":{},"pumpup":true,"pumpupComplete":true,"heartbeat":false}

AGENT STATE CHANGE FROM READY TO NOT READY

1. {"dimension":{"agentId":"scollins"},"realtimeData":{"agentNotReadyTimeDuration":4352,"agentReady":4},"pumpup":false,"pumpupComplete":false,"heartbeat":false }

NOTES:

- **agentId:** This is a generic unique agent ID that is *not* media specific. Underneath this ID lies multiple "account" ID's, one for each media type. So there can be a voice account ID, chat account ID, email account ID and SMS account ID. Note that specifically in the case of voice, the voice account ID will always be the same as a CC-Elite numerical agent ID.
- **agentNotReadyTimeDuration:** The amount of time in seconds the agent was in a not ready state during the reporting period.
- **agentReady:** Indicates the number of times the agent is in a ready state during the reporting period.
- **pumpup:** Indicates if this is a pumpup message or not. In this example, it is false so no pump up message.
- **pumpupComplete:** Indicates if this is a pumpup is complete message. In this example, it is false so no pump up in this event.

2. {"dimension":{"agentId":"scollins"},"realtimeData":{"lastStateChangeTimestamp":"0000000149691945647 8","nrReasonCodeName":"NOT_AVAILABLE","agentState":"READY","workState":"IDLE","lastWorkCodeChangeTimestamp":"00000001496919456497","lastStateReasonTimestamp":"000000001496919456495","nrReasonCode":"NOT_AVAILABLE"},"pumpup":false,"pumpupComplete":false,"heartbeat":false }

NOTES:

- **lastStateChangeTimestamp:** This is the UTC timestamp of when this state change happened which caused this event to be sent.
- **nrReasonCodeName:** This is a string value that represents the Not Ready Reason Code. The string values for this can be determined by the customer/user – depending on business needs (e.g. Lunch, Busy, Break, etc).
- **agentState:** Self explanatory - the agent's state
- **workState:** State of the agent relevant to their work; Available, Unavailable, Busy, Idle
- **lastWorkCodeChangeTimestamp:** This is UTC timestamp of the last change to the work code.
- **lastStateReasonTimestamp:** This is UTC timestamp of when the last reason code was entered. For example if the agent goes NOT READY with reason code X, they could subsequently change it to reason code Y. This revised reason code would not constitute a state change, but would cause a new event to be fired, and this field will contain the UTC timestamp of when that revised reason code was entered.
- **pumpup:** Indicates if this is a pumpup message or not. In this example, it is false so no pump up message.
- **pumpupComplete:** Indicates if this is a pumpup is complete message. In this example, it is false so no pump up in this event.

3. {"dimension":{"agentId":"scollins"},"realtimeData":{"idleTimeDuration":70},"pumpup":false,"pumpupComplete":false,"heartbeat":false }

NOTES:

- **idleTimeDuration:** The amount of time the agent was waiting to be offered an engagement within the reporting period (Example e.g., the amount of time in seconds that the agent was idle with no active engagement).
- **pumpup:** Indicates if this is a pumpup message or not. In this example, it is false so no pump up message.
- **pumpupComplete:** Indicates if this is a pumpup is complete message. In this example, it is false so no pump up in this event.

4. {"dimension":{"agentId":"scollins"},"realtimeData":{"lastStateChangeTimestamp":"0000000149691946504 5","nrReasonCodeName":"DEFAULT","agentState":"PENDING_NOT_READY","workState":"UNAVAILABLE ","lastWorkCodeChangeTimestamp":"00000001496919464696","lastStateReasonTimestamp":"00000001 496919465049","nrReasonCode":"0"},"pumpup":false,"pumpupComplete":false,"heartbeat":false }

NOTES:

- **lastStateChangeTimestamp:** This is UTC timestamp of when this state change happened which caused this event to be sent.
- **nrReasonCodeName:** This is a string value that represents the Not Ready Reason Code. The string values for this can be determined by the customer/user – depending on business needs (e.g. Lunch, Busy, Break, etc).
- **agentState:** Self explanatory - the agent's state.
- **workState:** State of the agent relevant to their work; Available, Unavailable, Busy, Idle.
- **lastWorkCodeChangeTimestamp:** This is UTC timestamp of the last change to the work code.
- **lastStateReasonTimestamp:** This is UTC timestamp of when the last reason code was entered. For example if the agent goes NOT READY with reason code X, they could subsequently change it to reason code Y. This revised reason code would not constitute a state change, but would cause a new event to be fired, and this field will contain the UTC timestamp of when that revised reason code was entered.
- **pumpup:** Indicates if this is a pumpup message or not. In this example, it is false so no pump up message.
- **pumpupComplete:** Indicates if this is a pumpup is complete message. In this example, it is false so no pump up in this event.

5. {"dimension":{"agentId":"scollins"},"realtimeData":{"agentNotReady":5},"pumpup":false,"pumpupComplete":false,"heartbeat":false }

NOTES:

- **agentNotReady:** The amount of time that the agent was in a not ready state during the reporting period.
- **pumpup:** Indicates if this is a pumpup message or not. In this example, it is false so no pump up message.
- **pumpupComplete:** Indicates if this is a pumpup is complete message. In this example, it is false so no pump up in this event.

6. {"dimension":{"agentId":"scollins"},"realtimeData":{"lastStateChangeTimestamp":"000000014969194651
93

NOTES:

- **lastWorkCodeChangeTimestamp:** This is UTC timestamp of the last change to the work code.

Agent goes Ready / Not Ready Measures

The following table contains the AGENTBYACCOUNTMEASURES Measure Stream messages produced when an agent goes READY/NOT READY

{"dimension":{"accountId":"8501014","agentId":"jsmith", "channelId":"Voice"},
"realtimeData":{"accountReady":1},"pumpup":false,"pumpupComplete":false,"heartbeat":false }

{"dimension":{"accountId":"8501014","agentId":"jsmith", "channelId":"Voice"},
"realtimeData":{"accountState":"READY","loginTimeStamp":"00000001496242479928","lastStateChangeT
imestamp":"00000001496242479928","lastStateReasonTimestamp":"00000001496242479928"},"pumpu
p":false,"pumpupComplete":false,"heartbeat":false }

{"dimension":{"accountId":"jsmith_OCPChatRoutableAddress","agentId":"jsmith", "channelId":"Chat"},
"realtimeData":{"accountReady":1},"pumpup":false,"pumpupComplete":false,"heartbeat":false }

{"dimension":{"accountId":"jsmith_OCPChatRoutableAddress","agentId":"jsmith", "channelId":"Chat"},
"realtimeData":{"accountState":"READY","loginTimeStamp":"00000001496242479726","lastStateChangeT
imestamp":"00000001496242479726","lastStateReasonTimestamp":"00000001496242479882"},"pumpu
p":false,"pumpupComplete":false,"heartbeat":false }

{"dimension":{"accountId":"jsmith_OCPShortMessageServiceRoutableAddress","agentId":"jsmith",
"channelId":"ShortMessageService"},
"realtimeData":{"accountReady":1},"pumpup":false,"pumpupComplete":false,"heartbeat":false }

{"dimension":{"accountId":"jsmith_OCPShortMessageServiceRoutableAddress","agentId":"jsmith",
"channelId":"ShortMessageService"},
"realtimeData":{"accountState":"READY","loginTimeStamp":"00000001496242479857","lastStateChangeT
imestamp":"00000001496242479857","lastStateReasonTimestamp":"00000001496242479882"},"pumpu
p":false,"pumpupComplete":false,"heartbeat":false }

{"dimension":{"accountId":"jsmith_OCPChatRoutableAddress,"agentId":"jsmith", "channelId":"Chat"},
"realtimeData":{"loginTimeStamp":"00000001496242479928","lastStateChangeTimestamp":"000000014
96242479928","lastStateReasonTimestamp":00000001496242479928},"pumpup":false,"pumpupComplet
e":false,"heartbeat":false }

{"dimension":{"accountId":"jsmith_OCPShortMessageServiceRoutableAddress,"agentId":"jsmith",
"channelId":"ShortMessageService"},
"realtimeData":{"loginTimeStamp":"00000001496242479928","lastStateChangeTimestamp":"000000014
96242479928","lastStateReasonTimestamp":"00000001496242479928"},Pumpup:false,"pumpupComplet
e":false,"heartbeat":false }

{"dimension":{"accountId":"8501014","agentId":"jsmith", "channelId":"Voice"},
"realtimeData":{"accountLogonDuration:6,"accountNotReady":1},"pumpup":false,"pumpupComplete":fal
se,"heartbeat":false }

{"dimension":{"accountId":"8501014","agentId":"jsmith", "channelId":"Voice"},
"realtimeData":{"accountState":"NOT_READY","lastStateChangeTimestamp":"00000001496242486588","l
astStateReasonTimestamp:00000001496242486588},"pumpup":false,"pumpupComplete":false,"heartbea
t":false }

{"dimension":{"accountId":"jsmith_OCPChatRoutableAddress","agentId":"jsmith", "channelId":"Chat"},
"realtimeData":{"accountLogonDuration":6,"accountNotReady":1},"pumpup":false,"pumpupComplete":fa
lse,"heartbeat":false }

{"dimension":{"accountId":"jsmith_OCPChatRoutableAddress","agentId":"jsmith", "channelId":"Chat"},
"realtimeData":{"accountState":"NOT_READY","lastStateChangeTimestamp":"00000001496242486368","l
astStateReasonTimestamp":"00000001496242486380"},"pumpup":false,"pumpupComplete":false,"heart
beat":false }

{"dimension":{"accountId":"jsmith_OCPShortMessageServiceRoutableAddress","agentId":"jsmith",
"channelId":"ShortMessageService"},"realtimeData":{"lastStateChangeTimestamp":"00000001496242486
096","lastStateReasonTimestamp":"00000001496242486285"},"pumpup":false,"pumpupComplete":false,
"heartbeat":false }

{"dimension":{"accountId":"jsmith_OCPChatRoutableAddress","agentId":"jsmith",
"channelId":"ShortMessageService"},"realtimeData":{"lastStateChangeTimestamp":"00000001496242486
588","lastStateReasonTimestamp":"00000001496242486588"},"pumpup":false,"pumpupComplete":false,
"heartbeat":false }

{"dimension":{"accountId":"jsmith_OCPShortMessageServiceRoutableAddress","agentId":"jsmith",
"channelId":"ShortMessageService"},"realtimeData":{"accountLogonDuration":6,"accountNotReady":1},"
pumpup":false,"pumpupComplete":false,"heartbeat":false }

{"dimension":{"accountId":"jsmith_OCPShortMessageServiceRoutableAddress","agentId":"jsmith",
"channelId":"ShortMessageService"},"realtimeData":{"accountState":"NOT_READY","lastStateChangeTim
estamp":"00000001496242486588","lastStateReasonTimestamp":"00000001496242486588},"pumpup":f
alse,"pumpupComplete":false,"heartbeat":false }

Third-party client coding considerations

Eventing considerations

The events dispatched by Avaya Analytics™ are based on events received from the Avaya Breeze® Reliable Eventing Framework.

The distributed nature of the Avaya Oceana® solution components means that multiple events can be sent when a user transitions from one state to another. Consequently, third-party clients consuming events sent by the Avaya Analytics™ Open Interface might require some logic in order to follow these state transitions.

For example, when an agent logs in, their state changes from LOGGED OUT to PENDING_LOGIN and then finally LOGGED IN. The event for PENDING_LOGIN may or may not contain other information such as the “nrReasonCode”. If the “nrReasonCode” is not present in the PENDING_LOGIN event, it will be part of the LOGIN event. It is up to the third-party client to build an accumulative view of the state of the agent.

The events sent over the WebSocket interface are “delta updates” which means that only updated values are sent. This is to limit the associated network bandwidth and CPU activity. As a result, if a third-party client is consuming AGENTMEASURES it will only receive a JSON message if an agent state triggered the update of a measure.

A heartbeat JSON message is sent every 8 seconds.

Troubleshooting

System health checks

To check if your Avaya Analytics™ solution is healthy, connect to CCM over SSH and run this command:

```
kubectl get pods --all-namespaces
```

All the pods should have a status set to Running.

Configuration check

Workspaces uses the REST & WebSocket interface to consume real-time data. Therefore, if real-time data is showing in the Workspaces dashboards, then your system is fully configured, and third-party applications should be able to receive real-time data as well.

If no real-time data is shown in Workspaces please refer to the Documentation section for the installation and troubleshooting guides.

Logging

The log files for the REST and WebSocket interfaces can be obtained by connecting to CCM over SSH and calling the `kubectl logs` command on the `orca-streams-rest` and `orca-streams-data-publisher` pods.

On HA systems, there are multiple instances of the `orca-streams-rest` and `orca-streams-data-publisher` pods. There is a log file for each pod instance.

The default log levels for these applications is suitable for debugging connectivity issues and we are not expecting users to change it. However, it is possible to update the log level by editing the ConfigMap of the pods if desired. Please be aware that increasing the log levels will results in higher I/O and may affect performance.

Debugging tools

Below are a few suggestions that will help you with debugging your application.

1. We recommend deploying the REST & WebSocket Sample Client application on the CCM to validate that your system is working as expected. Java is already available on CCM, so all you need to do is configure the `avaya.oceanalytics.stream-server.hostname` field in

the `application.properties` with the FQDN or IP address of your ingress gateway.

2. Use curl or Postman (https://www.postman.com/) to exercise the REST endpoints. In the examples below, replace `<ingress-gw-ip>`, `<username>`, `<password>` and `<token>` with your own data.

   Login

   ```
   curl -k --location --request GET \
   'https://<ingress-gw-ip>:443/orca-streams-
   rest//users/login?tenant=0&username=<username>&password=<password>'

   Example of username: oceanaagent1@cec.avaya.com
   ```

   Get sources

   ```
   curl -k --location --request GET \
   'https://<ingress-gw-ip>:443/orca-streams-rest/sources?locale=en&tenant=0' \
   --header 'Accept: application/json' \
   --header 'Authorization: Bearer <token>'
   ```

   Get producers

   ```
   curl -k --location --request GET \
   'https://<ingress-gw-ip>:443/orca-streams-
   rest/sources/Oceana_Streams_127.0.0.1_3.5/producers?tenant=0&locale=en' \
   --header 'Accept: application/json' \
   --header 'Authorization: Bearer <token>'
   ```

3. You can take inspiration from the REST requests issued by Workspaces by turning on Developer Tools in your browser.

Documentation

To access the relevant documentation, go to https://support.avaya.com

Applicable Documentation for Avaya Analytics™

Deploying Avaya Analytics™ for the Avaya Oceana® Solution
Maintaining and Troubleshooting Avaya Analytics™ for Avaya Oceana® Solution
Avaya Analytics™ Data Dictionary

Applicable Documentation for the Avaya Oceana® Solution

Avaya Oceana® Solution Description
Administering Avaya Oceana® Solution
Using Avaya Workspaces for Avaya Oceana® SolutionDeploying Avaya Oceana® Solution
Troubleshooting Avaya Oceana® Solution

Frequently Asked Questions

Functionality Questions

When registering a client, what information do we need?

- To subscribe to a measure producer, you need to know the IP Address and Port number of the Ingress Gateway as well as the Source and Producer names.

How do we get agent specific events? Are they sent individually, or as a list of multiple agent events?

- Each agent event is sent individually.

Are we filtering out specific events or reason codes or we will be taking in all the events?

- There is no filtering of events. For every state change of an agent, the event is sent to the client. The client will have to filter out events that they are not interested in.

Will the Timezone used for event time stamps be constant?

- All events will have the UTC timestamp.

How to determine if an agent is logged in or logged out?

- The agent/account state will show Logout if the agent is LOGGED_OUT, otherwise it will show READY or NOT_READY.

What happens if the client loses connection with the WebSocket interface?

- The client must monitor the network connection. If the connection goes down an `onClose` event is sent to the client and the client should then try to reconnect and resubscribe for all producers the client is interested in.
  Also the client must keep an eye on heartbeats that it receives. Heartbeats take the form of a real-time message and also a server client heartbeat “--heartbeat--”.
  On receiving the “--heartbeat--” message from the publisher the client should reply back with the same “--heartbeat--” message. This “--heartbeat--” message keeps the WebSocket connection alive even if there are no producers producing real-time messages or real-time heartbeats to the publisher.

Will the order of the events sent through change?

- The order of events received is dependent on the call scenario. You may receive a different sequence of events depending on the current state of the agent.

Last Page of Document