Virtualized Conversations
Internet-Draft
Intended status: Standards Track
Expires: 19 April 2026

D. G Petrie
SIPez LLC
J. Rosenberg
Five9

16 October 2025

The JSON vCon - Contact Center Extension
draft-ietf-vcon-cc-extension-01

Abstract

A vCon is container for data and information relating to a human conversation. This document defines an extension for the JSON vCon schema in support of call, support or contact center application of the vCon conversational data exchange format.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://ietf-wg-vcon.github.io/draft-ietf-vcon-cc-extension/draft-ietf-vcon-cc-extension.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-vcon-cc-extension/.

Discussion of this document takes place on the Virtualized Conversations Working Group mailing list (mailto:vcon@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/vcon/. Subscribe at https://www.ietf.org/mailman/listinfo/vcon/.

Source for this draft and an issue tracker can be found at https://github.com/ietf-wg-vcon/draft-ietf-vcon-cc-extension.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 19 April 2026.

Table of Contents

1. Introduction

This document adds a number of new parameters to the Party Object and the Dialog Object defined as part of the JSON vCon schema in [VCON-CORE]. The vCon parameters defined in this document have been determined to be need and are specific to the contact center uses of vCon. The general framework and requirements for defining an extension to the JSON vCon schema are defined in [VCON-CORE].

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2.1. Terminology

See section 2.1 of [VCON-CORE] for terminology used in this document.

2.2. JSON Notation

This document uses the same JSON notation that is used in [VCON-CORE]. For the ease of documentation, the convention for JSON notation used in this document is copied from section 2.2 of [VCON-CORE].

* "String" - a JSON string type

* "A[]" and array of values of type A.

All parameters are assumed to be mandatory unless other wise noted.

Objects or arrays with no or null values MAY be excluded from the vCon.

3. vCon JSON Object

This vCon extension adds a new extensions parameter name value token. The string token "CC" should be included in the extensions array of the vCon Object. It is not required that consumers of vCons with the *CC* extension content support this extension. It does not change the semantics or remove any parameters form the core vCon schema. There is no need to list the CC extension name in the *must_support* parameter.

3.1. Party Object

This vCon extension adds the following new parameters to the Party Object in support of Contact Center use cases.

3.1.1. role

The role that the participant played in the conversation. In a call center there are roles: such as: agents, customer, supervisor and specialist. In conferences there are roles: host, cohost, speaker, panelist, participant and other roles. The role parameter provides the ability to label the role that the part played in the conversation.

* role: "String" (optional)

The following values for the role parameter MAY be used:

* "agent"

* "customer"

* "supervisor"

* "sme" (for subject mater expert)

* "thirdparty"

Other values for the role parameter MAY also be used.

3.1.2. contact_list

In a contact center scenario, the conversation with this party may be part of a larger effort of contacting a group of parties, individually or perhaps in groups. It is sometimes useful to reference the list from which this party was included. The contact_list may be used as a label for foreign key reference to the contact list that this party was on.

* contact_list "String" (optional)

3.2. Dialog Object

This vCon extension adds the following new parameters to the Dialog Object in support of Contact Center use cases.

3.2.1. campaign

In a contact center scenario, a dialog may be initiated as part of a campaign or set of dialogs initiated with a common goal or focus or to be handled or treated in a specific way. The campaign parameter is string that may be used as a label or foreign key in reference to an external specification for how the communication is to be initiated, handled or treated. In some case it may be appropriate to attached the campaign data as an Attachment Object.

* campaign: "String" (optional)

3.2.2. interaction_type

* interaction_type "String" (optional)

TODO: add enumerated values from JDR

3.2.3. interaction_id

TODO: Is this different from RFC7989 session ID (session_id in core)?

In a contact center scenario, interactions with a party are often labeled with an identifier. In some case the interaction is contained in a single dialog. In others there may be multiple dialogs (e.g. messages or calls) that are all part of a single interaction. There may also be many interactions for a single conversation or vCon. The interaction parameter is used as a label or foreign key in reference to the interaction ID.

* interaction_id "String" (optional)

3.2.4. skill

A contact center may service multiple purposes or customers. In this scenario it is important to label the conversation segment or dialog. The agent or automata which services the dialog are required to have a specific skill. To facilitate this in a vCon dialog, the skill parameter is provided. The string values of the skill parameter are contact center specific.

* skill "String" (optional)

4. Security Considerations

Security considerations are covered in the [VCON-CORE] document. This extension to vCon adds no additional security concerns.

5. IANA Considerations

5.1. vCon JSON Registry Additions

5.1.1. vCon Extensions Names Registry

The following extension name is added to the vCon Extensions Names Registry.

Extension Name	Extension Description	Change Controller	Specification Document(s):
CC	Contact Center	IESG	Section 3 RFC XXXX

Table 1

5.1.2. Parties Object Parameter Names Registry

The following defines additional values for the vCon Parties Object Parameter Names Registry.

Parameter Name	Parameter Description	Change Controller	Specification Document(s)
role	agent party role	IESG	Section 3.1.1 RFC XXXX
contact_list	contact_list including this party	IESG	Section 3.1.2 RFC XXXX

Table 2

5.1.3. Dialog Object Parameter Names Registry

The following defines the initial values for the vCon Dialog Object Parameter Names Registry.

Parameter Name	Parameter Description	Change Controller	Specification Document(s)
campaign	campaign to which dialog is part of	IESG	Section 3.2.1 RFC XXXX
interaction_type	dialog interaction type	IESG	Section 3.2.2 RFC XXXX
interaction_id	dialog interaction id	IESG	Section 3.2.3 RFC XXXX
skill	required skill	IESG	Section 3.2.4 RFC XXXX

Table 3

6. Contact Center Use Cases

6.1. Types of Applications

In the contact center, there are several different types of applications which require consumption of recordings. These typically go under the moniker of Workforce Optimization (WFO). This section describes the main ones.

6.1.1. Recording

Call Recording applications receive call recordings from the core, and then provide long term storage, playback, and search functionality. Recording storage is needed for archival purposes, and is often a requirement to meet compliance regulations in certain industries.

6.1.2. Quality Management (QM)

Quality Management (QM) applications are used by contact center managers to make sure agents are following guidelines on proper handling of calls. Many consumers are familiar with the greeting played in voice response systems which say, "This call may be monitored for quality and training purposes". That greeting refers specifically to QM applications.

QM applications allow a user - an employee in the quality management group typically - to playback recordings for a particular agent, and then based on that recording, rate them on how they performed. These ratings are made against a questionnaire that defines the rubric against which agents are scored. This rubric will often include questions like, "Did the agent thank the customer for calling and ask them how they can help"? Or, "Did the agent upsell the customer on the newest product?". These scorecards are then shared with the agents and their managers (the supervisors), along with coaching and training materials to handle cases where the agent didn't do well. Originally, scoring was done entirely by humans, and as a consequence, only a handful of calls for each agent could possibly be scored. These were often done by sampling calls at random.

It is also common for QM applications to use speech recognition technology to transcribe calls into text. This allows a call to be scored more quickly, and enables search functions for selection of specific calls that would be good candidates for scoring.

A part of the agent role involves usage of corporate applications, such as ordering, billing, shipping, to handle the customer inquiry. To determine whether agents are using these tools correctly, it is common in the contact center for agents to have desktop recording applications installed. These record the screen content as a video file. Typically, the vendor of the QM software provides the desktop screen recording and backend applications which receive and store the recording. These are then combined with the audio, email, or chat recordings that come from the core. The following shows the flow of recordings in this use case:

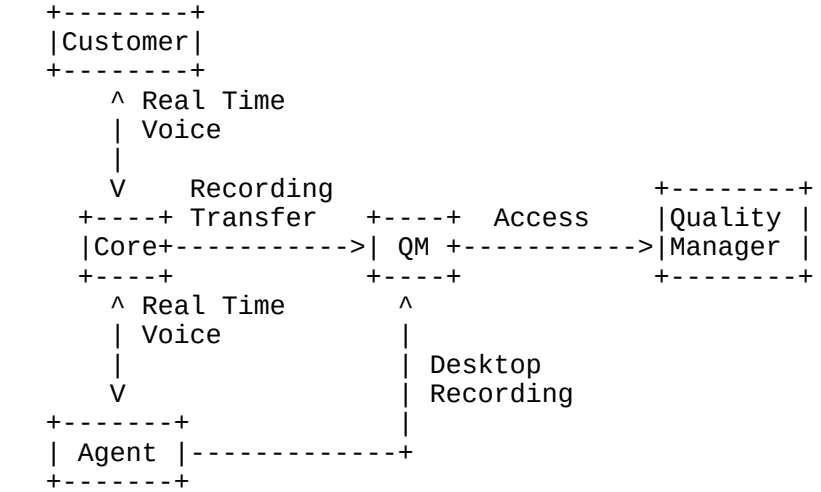

Figure 1: QM Recording Exchanges

In this flow, the customer calls into the contact center, and is connected to the core. Typically this is done through the Session Initial Protocol (SIP) [RFC3261] and the Real-Time Transport Protocol (RTP) [RFC3550]. The call is delivered to the agent, also typically using SIP and RTP. The core will record the call, and then at the end of the call, the recording is transferred to the QM system. During the call, the agent desktop is recorded, and this recording is transferred to the QM system. At a later time, the Quality Manager can log into the QM application, and access the recording, inclusive of the audio, the transcript and the desktop recording.

In practice, there are many variations on this basic exchange. Sometimes, the ACD sends the audio portion of the call to the QM system using real-time streaming, sometimes using SIPREC [RFC7866]. This is then augmented with meta-data using proprietary REST APIs. In other cases, the audio is sent post-call, and similarly, meta-data is obtained using proprietary REST APIs. When transcription takes place, it is most often done by the QM system but not always. In some cases, a transcript is sent from the core to the QM system instead of, or in addition to, the audio recording.

In a similar way, the transfer of the desktop recording from the agent’s computer to the QM system can happen in real-time or post-call. Post-call systems will often upload the recording in chunks, sometimes doing so after hours or when the agent is not on a call.

A key considering for this use case is the concept of recording stitching.

In a typical call in the contact center, there are multiple segments, each of which represents a phase of the call. There will be a segment that contains the customer’s interaction with the voice response system, where no agents were present. When the customer is connected to an agent, there will be a segment representing the portion of the call where the customer talks to the agent. As the call is conferenced, transferred or held, each corresponds to an additional segment.

The process of assembling together these segments into a complete recording is referred to as stitching. Different stitches are needed depending on the use case. In a QM use case, the quality manager is rating the agent, and thus what matters is the call as seen by that agent. In the case where a call was handled by multiple agents (a common use case in the contact center), a single call would result in two separate stitched recordings - one representing the customer's time with the first agent, and the second with the second agent. This is different than recording use cases as described above, where what matters is the entire call as seen by the customer.

6.1.3. Speech Analytics

Speech analytics applications provide graphs and dashboards on the content of conversations. For voice calls, this includes metrics like cross-talk, silence durations, and anger, which are computed directly from the voice. Voice calls are often transcribed to text, and further analysis is provided on the text. This might include customer sentiment, frequency of common reasons for call, and so on. These tools will also often provide discovery features, such as word clouds and clustering.

Speech analytics tools are often used to help companies decide which calls should be reviewed for quality management. This is an improvement over pure random based sampling. They are also used to help companies improve their processes in the contact center, identifying areas where agents are inefficient. For example, speech analytics can be used to determine that there has been a spike in customer refund requests, and the agents are taking too long to handle these types of calls.

Architecturally, speech analytics look a lot like recording. At the end of the call, a transcript is sent from the core platform to the speech analytics platform for processing. Meta-data is then fetched.

6.2. PII and PCI Redaction

A common requirement in contact center use cases is the redaction of payment card information (PCI) and personally identifiable information (PII) from recordings and transcripts. This happens in several ways.

For payment cards, it is common for the agent to transfer the call to dedicated voice response systems whose job is to collect the credit card numbers and process them. This way, the agent never hears this information. Furthermore, the system can be configured to pause the recording so that this particular segment is not recorded. For cases where the agent does collect the credit card information, it is

common for systems to have a "pause recording" button that can be triggered manually by the agent to ensure that this content is not recorded. Another common solution is to instrument the website where credit card information is entered, so that when the agent places their mouse into this form, the recording is paused. It would be useful in the vCon to indicate that this particular section of the recording was absent for PCI reasons.

It is also a common request to remove PII information, such as first and last name, street address, email address, and phone numbers, from recordings and from transcripts. In such cases, it is desirable to clearly indicate in the transferred recording that this has happened, so that downstream analytics applications function properly. Just replacing a first name with "XXX" is likely to confuse a word cloud tool in a speech analytics application, and make it think that "XXX" is a common word in the transcript. At the same time, just removing the PII entirely results in transcripts that are improperly formed language, making it harder to process by natural language understanding (NLU) tools.

6.3. Omni Channel

In contact center, the term "omni channel" is used to refer to the usage of non-voice communications with a customer. Sometimes, this means an email exchange or web chat from a widget on a web page. In other cases, it can involve a combination of voice with these other technologies. For example, a customer might call into the contact center, and then the agent uses SMS to send the customer links to information, or collect information from the customer. In that case, the overall interaction is composed of a voice segment and an SMS segment combined together.

In some cases, video is used in contact center applications. Mostly, this is in support of the "see what I see" case, where the customer uses the front camera on their mobile phone to show something to the contact center agent. For example, a customer might show the agent a part that is broken and needs to be replaced, to help the agent identify which part to send. In other use cases, traditional person-to-person video is used, in high touch support or sales use cases.

Co-browsing is also used in contact center applications. This is sometimes used in support situations, where a customer is having trouble navigating the website. The agent can take control over the browsing experience and get the customer where they need to be. This is different than screen sharing use cases common in meetings.

As it relates to recording, all of these additional channels need to be included in the vCon.

6.4. Deployment Topologies

As one might imagine, there are a variety of deployment topologies for these applications, mixing and matching on-premise vs cloud delivery. The core platform can be delivered on-premise, or via cloud. The supporting applications can also be delivered on-premise or via cloud. In the cloud delivery model, they can be co-resident with the core application (meaning, the vendor of the core service also deploys and operates the supporting application), or be delivered via different cloud services.

7. Example vCons

TODO

Acknowledgments

* Thank you to Thomas McCarthy-Howe for inventing the concept of a vCon and the many discussions that we had while this concept was developed into reality.

* Thank you to Jonathan Rosenberg and Andrew Siciliano for their input to the vCon container requirements in the form of I-D: draft-rosenberg-vcon-cc-usecases.

* Thank you to Rohan Mahy for his help in exploring the CDDL schema and CBOR format for vCon.

* The examples in this document were generated using the command line interface (CLI) from the py-vcon [PY-VCON] python open source project.

* Thank you to Steve Lasker for formatting and spelling edits.

References

Normative References

[JSON] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/rfc/rfc8259>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[UUID] Peabody, B. and K. R. Davis, "New UUID Formats", Work in Progress, Internet-Draft, draft-peabody-dispatch-new-uuid-format-04, 23 June 2022, <https://datatracker.ietf.org/doc/html/draft-peabody-dispatch-new-uuid-format-04>.

[VCON-CORE] Petrie, D., "The JSON format for vCon - Conversation Data Container", Work in Progress, Internet-Draft, draft-ietf-vcon-vcon-core-01, 15 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-vcon-vcon-core-01>.

Informative References

[CDR] ITU, "Recommendation Q.825: Specification of TMN applications at the Q3 interface: Call detail recording", n.d., <https://www.itu.int/rec/T-REC-Q.825>.

[PY-VCON] "Python open source vCon command line interface, library and workflow server", n.d., <https://github.com/py-vcon/py-vcon>.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <https://www.rfc-editor.org/rfc/rfc3261>.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <https://www.rfc-editor.org/rfc/rfc3550>.

[RFC7866] Portman, L., Lum, H., Ed., Eckel, C., Johnston, A., and A. Hutton, "Session Recording Protocol", RFC 7866, DOI 10.17487/RFC7866, May 2016, <https://www.rfc-editor.org/rfc/rfc7866>.

[vCon-white-paper] Howe, T., Petrie, D., Lieberman, M., and A. Quayle, "vCon: an Open Standard for Conversation Data", n.d., <https://github.com/vcon-dev/vcon/blob/main/docs/vCons_%20an%20Open%20Standard%20for%20Conversation%20Data.pdf>.

Authors' Addresses

Daniel G Petrie
SIPez LLC
Email: dan.ietf@sipez.com

Jonathan Rosenberg
Five9
Email: jdrosen@jdrosen.net