Virtualized Conversations
Internet-Draft
Intended status: Informational
Expires: 3 September 2026

T. McCarthy-Howe
Strolid

2 March 2026

The vCon - Conversation Data Container - Overview
draft-ietf-vcon-overview-01

Abstract

A vCon is the container for data and information relating to a real-time, human conversation. It is analogous to a [vCard] which enables the definition, interchange and storage of an individual’s various points of contact. The data contained in a vCon may be derived from any multimedia session, traditional phone call, video conference, SMS or MMS message exchange, webchat or email thread. The data in the container relating to the conversation may include Call Detail Records (CDR), call meta data, participant identity information (e.g. STIR PASSporT), the actual conversational data exchanged (e.g. audio, video, text), realtime or post conversational analysis and attachments of files exchanged during the conversation. A standardized conversation container enables many applications, establishes a common method of storage and interchange, and supports identity, privacy and security efforts (see [vCon-white-paper])

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-overview/draft-ietf-vcon-overview.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-vcon-overview/.

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-overview.

Status of This Memo

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

Copyright Notice

Table of Contents

1. Introduction

The generation of conversational data, contained in transcripts and multi-media files, is common in business, especially in customer facing organizations. However, the storage, analysis and sharing of the data they contain is not currently a standard, hampering efforts for both system interoperation and responsible data handling. Standardizing a container for conversation data (vCon) has numerous advantages, and enables the management of the conversation's content.

Often the system providing the communications service, the consumer and/or owner of the communications data and the communications analysis services are distinct systems and in many case separate business entities. vCons provide a standard means of exchanging communications data between these systems and services. The use of vCons can ease service integration by using a common container and format for enterprise communications, becoming the standardized input to communication analysis tools and machine learning and categorization.

* For organizations in dialog with customers or citizens, a vCon can be the container of where conversations are stored and personal data protections are expressed, managed and governed.

* For conversations of record, the vCon can be a legal instrument, providing a testable expression of conversational fact, while enabling conversational trust and transparency.

* For machine learning efforts, vCons can track what information was used in the training of models. As the result of a customer right to know request, an accurate answer to how their data was processed can be derived and communicated, and as the result of customer correction or deletion request, the responsible organization can properly and ethically respond as required by governing law.

1.1. What’s in a vCon?

A vCon contains four major categories of data (parties, dialog, analysis and attachments), with descriptive identifiers and metadata, inside a JSON container that can be signed and encrypted. The parties portion allows for an expanded set of data from a typical call detail record ([CDR]), with identifications of the participants or parties to the conversation. The dialog portion contains a set of multimedia and media type elements, each representing the actual, physical conversation in original media form: text, audio, video and imagery. The analysis portion contains data derived from the party and dialog portions, intended to carry items like transcripts, translations, summaries, text to speech, sentiment analysis and other semantic tagging. Finally, the attachment portion contains any other documents, such as slide deck or sales lead information, expressions of consent or authenticity, which provides context and support for the conversation itself.

In addition to these four major categories, the vCon itself carries top-level metadata including a globally unique identifier (UUID), creation and last-modified timestamps (created_at and updated_at), an optional subject, and references to other vCons through redaction or amendment. The vCon may also contain integrity checking information such as the issuer of the vCon and tamper-proof features such as signatures. An extensions mechanism allows the vCon schema to be extended for specialized use cases while maintaining interoperability with implementations that support only the core format.

1.2. Data Responsibility: Privacy vs Utility

Since vCons are designed to carry conversational data between systems, they must provide the ability to balance the utility and sensitivity of the information they contain. The transmission of information outside of a security boundary does not release the controller of the data from the responsibility of handing personal data. vCons enable the best practices of personal data management through approaches such as data minimization, consent validation and integrity protection.

The parties section carries significant privacy implications and responsibilities; the very definition of the sensitive biometric data addressed by the GDPR. Each party identified in a vCon represents an individual or entity whose personal information is being captured and potentially shared. The vCon creator and any subsequent processors of the vCon have a responsibility to ensure that the collection, storage, and sharing of party information complies with applicable privacy laws and regulations (such as GDPR, CCPA, or other regional privacy frameworks). This includes obtaining appropriate consent for

data collection, implementing data minimization practices, and providing mechanisms for data subjects to exercise their rights regarding their personal information.

At the same time, the conversations defined by the vCon carry the most authentic and important data in many scenarios from healthcare to commerce; a powerful addition to any data set. To enable adoption, the JSON format implemented by the vCon is the lingua franca of modern software; a frictionless integration to applications that require the human conversation. It is expected that JavaScript handling of vCons in the front end and RESTful interfaces and back end platforms will be used for operations and manipulation of vCons. Many media analysis services which will be used with vCons, such as transcription, already use JSON based interfaces. For these reasons, JSON [JSON] has been chosen for the initial format binding of vCons and the scope of this document. Other bindings (e.g. [CBOR] or [CDDL]) may be considered for vCon in the future in other documents.

For most application architectures, JSON objects are created by applications, for applications. However, most of the initial set of use cases differ from this established pattern, and are expected to be in the interchange between front end and back end application and lower layers of the network stack, critical for enablement of analysis of conversations. Thus, the contents of the vCon, if not the vCon itself, are generated by various and diverse network and communications elements like SIP user agents and SMTP servers, and then delivered across networks, and sometimes across security boundaries. This diversity of conversational data creates difficulty in creating unified views of customer conversations, especially as they traverse conversational modes. By providing a common mechanism to describe conversations, appropriate to the various network elements that create them, enables new scenarios and usage kinds.

1.3. Use Cases and Requirements

1.3.1. Contact Center Use Case

Contact centers typically handle customer interactions across multiple communication channels including voice telephony, web-based chat systems, electronic mail, Short Message Service (SMS), and video conferencing platforms. Each interaction channel generates conversational data that is often stored in disparate systems using incompatible formats, creating operational challenges for organizations seeking to maintain comprehensive customer interaction records, perform cross-channel analytics, or implement consistent privacy management practices.

A vCon-based implementation addresses these challenges by establishing a standardized container format for each interaction while maintaining referential relationships between related conversations. When a customer interaction spans multiple channels (e.g., initial web chat escalated to video conference with email follow-up), each communication system generates a vCon containing the conversation parties, dialog content, automated analysis results, and relevant attachments. These vCons are linked through common case identifiers and sequential references, enabling downstream systems to reconstruct complete customer interaction timelines while preserving the integrity and context of each individual conversation component.

The implementation of vCons in contact center environments provides several operational benefits: unified customer journey tracking across all communication channels, enhanced analytics capabilities through standardized data formats, simplified regulatory compliance through consistent consent tracking and audit trails, efficient privacy rights management with granular data deletion and redaction capabilities, and improved quality assurance processes through comprehensive evaluation of multi-channel customer service interactions. This standardization reduces operational complexity while ensuring compliance with applicable privacy regulations.

1.3.2. Messaging Use Case

Healthcare organizations providing patient communication services across multiple messaging platforms including SMS, secure patient portals, electronic mail, and integrated telehealth systems face significant challenges in maintaining complete medical communication records while ensuring compliance with healthcare privacy regulations such as the Health Insurance Portability and Accountability Act (HIPAA). Patient conversations frequently span multiple communication channels over extended periods, resulting in fragmented medical records that impede clinical decision-making and complicate regulatory compliance efforts.

A vCon implementation in healthcare messaging environments employs privacy-preserving design principles including explicit consent management, data minimization capabilities, healthcare-grade encryption standards, and role-based access controls. Each communication channel creates a vCon instance containing conversation participants, message content, automated analysis results, and relevant medical attachments, while maintaining integration pathways with Electronic Health Record (EHR) systems. This architecture enables authorized healthcare providers to access complete patient communication histories for care coordination purposes while implementing granular privacy controls to protect sensitive health information in accordance with applicable regulations.

The deployment of vCons in healthcare messaging systems delivers measurable improvements including comprehensive patient communication records integrated with clinical data systems, enhanced privacy protection through granular control mechanisms for sensitive health information, improved care coordination between multiple healthcare providers, built-in regulatory compliance capabilities with automated audit trails and consent management, and enhanced clinical decision support through access to complete patient communication context. This standardization enables healthcare organizations to improve patient care delivery while maintaining strict privacy protection and regulatory compliance requirements.

1.3.3. ECRIT (Emergency Context Resolution with Internet Technologies) Use Case

Emergency services organizations require rapid access to comprehensive situational information during crisis response operations. Traditional emergency communication systems create information silos where critical multimedia content, geographic location data, and inter-agency coordination communications are distributed across multiple systems and jurisdictional boundaries. This fragmentation delays access to vital operational information, complicates multi-agency coordination efforts, and produces incomplete documentation for subsequent legal proceedings and incident investigations.

A vCon implementation for emergency services enables real-time creation and maintenance of linked conversation containers that capture initial emergency calls, multimedia updates from incident witnesses, location tracking data, multi-agency coordination communications, and post-incident investigation records. Each vCon contains conversation participants (emergency callers, dispatchers, response personnel), dialog content (voice recordings, text messages, radio communications), automated analysis results (emergency classification, resource requirements, incident reconstruction), and relevant attachments (photographs, videos, location coordinates, official reports). Critical operational features include real-time vCon creation and updates, priority processing mechanisms, cryptographic integrity protection, and secure multi-jurisdictional information sharing capabilities.

The implementation of vCons in emergency services environments provides operational improvements including complete situational awareness for emergency response personnel, reduced response times through expedited access to critical information, enhanced inter-agency coordination through standardized information sharing protocols, regulatory compliance through complete tamper-evident record keeping, and improved public safety outcomes through enhanced

information management capabilities. Integration with existing emergency services infrastructure including Computer-Aided Dispatch (CAD) systems, Geographic Information Systems (GIS), and Next Generation 911 (NG911) platforms ensures that vCon implementation enhances rather than replaces current emergency response capabilities.

1.4. VCON Requirements

* Standardize container for conversational data exchange
* Consolidation of data and information for a conversation
* Multiple modes of communication, changing over time
* Snapshots of conversation during or once completed along with analysis
* Ease of integration of services and analysis
* Better organize conversational data so that it can be handled in a consistent, privacy safer means
* Immutable
* Hiding of PII or entire conversation
* Amendable with additional information and data elements

1.4.1. VCON Communication Modes

Define a standard for exchange of conversational data in a sea of modes, platforms and service offerings for conversations, such as these example conversational modes and protocols:

* SMS
* MMS
* JABBER
* SIMPLE
* Proprietary web chat
* SMTP
* PSTN

* SIP

* WEBRTC

* Proprietary video conferencing

The following are considered not in scope or non-requirements: * Real-time streaming or updating of conversational data * Transport mechanisms * Storage or databases specifications * Methods of redaction of text, audio or video media * Validation of redactions or amended data beyond the signature of the domain making the changes to the conversational data (e.g. Merkle tree like redactions)

Standardization of analysis data formats or file media types is supported by the extensions mechanism, described in section 3.7.

2. Conventions and Definitions

2.1. Terminology

* amended vCon - a new vCon instance version created by adding to or modifying a prior signed vCon, referencing the earlier version through the amended parameter while containing a deep copy of all prior data plus new content

* analysis - analysis, transformations, summary, sentiment, or translation typically of the dialog data

* compatible extension - a vCon schema extension that introduces additional data or fields without altering the meaning of existing elements, allowing implementations that do not recognize the extension to safely ignore it

* consent - explicit permission granted by a party for the collection, processing, or sharing of their conversation data

* conversation - an exchange of communication using text, audio or video medium between at least one human and one or more bots or humans

* critical extension - an incompatible vCon schema extension that modifies existing semantics and must be listed in the vCon’s critical parameter; implementations that do not support a critical extension must reject the vCon

* data minimization - the practice of limiting the collection and processing of personal data to what is necessary for the stated purpose

* de-identification - removal of all information that could identify a party in a conversation. This includes PII as well as audio and video recordings. Voice recordings might be re-vocalized with a different speaker.

* dialog - the captured conversation in its original form (e.g. text, audio or video)

* encrypted form - encrypted [JWE] document with the [JWS] signed vCon form contained in the ciphertext

* extension - a registered addition to the vCon schema that defines new parameters or modifies existing semantics, identified by a unique token value registered with IANA

* file - a data block either included or referenced in a vCon

* object - JSON object containing key and value pairs

* parameter - JSON key and value pair

* party - an observer or participant to the conversation, either passive or active

* payload - the contents or bytes that make up a file

* PII - Personal Identifiable Information

* PII masked - may include voice recordings, but PII is removed from transcripts and recordings (audio and video)

* redaction - the process of removing or obscuring specific content from a vCon while maintaining the overall structure and integrity

* signed form - [JWS] signed document with the unsigned vCon form contained in the payload

* vCon - container for conversational information

* vCon instance - a vCon populated with data for a specific conversation

* vCon instance version - a single version of an instance of a conversation, which may be modified to redact or amend additional information forming a subsequent vCon instance version

2.2. Inline vs Externally Referenced Files

Due to the size and complexity of some portions of a vCon, both inline and externally referenced dialog, analysis, attachments and other vCon reference assets are supported. For instance, vCons may reference a video conference media recording as an external URL with an accompanying content hash of the contents to detect tampering. Alternatively, vCons may directly contain the media of the entire dialog internally, keeping the conversation in one place, and optionally encrypted.

3. vCon JSON Object

3.1. A Conversational Definition

vCons define conversations, and are created by systems during and after the conversation itself. vCons provide ways to express and define the contents, participants and context of a particular conversation. Unlike some measurable physical phenomena, like mass and volume, conversations are heterogeneous, relatively complex and contain relevant information outside of the physical phenomena, such as consent and provenance. Some communication modes, like SMS texting, lack natural session boundaries and require explicit definition. Thus, the definition of a conversation requires more than a simple scalar value, or a series of samples of a time-based waveform. The definition of a conversation enables tools and systems to precisely identify, responsibly manage, efficiently process and accurately govern their use.

vCons also enables the definer of the conversation to express the scope of the conversations. A vCon may contain any combination of content appropriate to the use case:

* A vCon may be a single audio recording, or a complete conversational journey from a text message, to a resulting conversation and a followup email.

* A vCon may represent a conversation between two people, a conversation between a person and a machine, or all of the conversations between customers and a contact center team.

* A vCon may be sent in response to a Right To Know request to a single customer, or to a governance body during an audit

None of the major parts of the vCon (parties, dialog, attachments and analysis) are required to be present, to maximize the conversations that can be expressed. For instance, a recording without a parties definition is a valid expression of a conversation without defining

the people involved, either because it is unknown, to be discovered through the analysis of the recording, or to be hidden for data minimization reasons. vCons may have two or more parties involved, but since a fundamental role of the vCon is to define and protect the data it contains, at least one should be, in the words of the GDPR, a "natural person." For instance, an interaction between a bot and a human is an appropriate scope for vCons, but a conversation between two bots would not.

3.2. Parties

The parties section in a vCon serves as the container for all participant identity information involved in the conversation. Structurally, it is an array of party objects, each of which can include various attributes such as telephone numbers, email addresses, names, and even structured contact information (like civic addresses and geographic coordinates). The purpose of this section is to provide clear attribution of every interaction by documenting who participated in the conversation. This not only supports accurate record-keeping but also enables accountability, context, and subsequent analysis of the conversation data.

Each party object may contain a variety of identification attributes. Traditional contact identifiers include telephone numbers, email addresses, SIP URIs, and participant names. Decentralized Identifiers (DIDs) provide a verifiable, decentralized digital identity that is independent of centralized registries. A validation parameter records how the party’s identity was verified (for example, by social security number, date of birth, or user credentials), capturing the method of validation without exposing the actual verification data. For scenarios requiring cross-conversation correlation, such as contact center agents handling multiple interactions, a persistent UUID can be assigned to a party that remains stable across vCon instances. Geographic context can be captured through geolocation coordinates and civic address information, recording the party’s location at the time of conversation.

The identification of parties serves multiple purposes beyond simple attribution. It enables proper consent management by clearly identifying whose data is being processed, supports data subject rights requests by providing a clear mapping of individuals to their conversation data, and facilitates compliance with privacy regulations that require organizations to track and manage personal data throughout its lifecycle. Additionally, the structured nature of party identification allows for consistent handling of privacy-related operations such as data deletion, anonymization, or redaction requests across different systems and jurisdictions.

3.3. Dialog

The dialog section in a vCon captures the actual conversation content that occurred between parties. This is the core of what makes a vCon valuable - it contains the real communication that took place, whether that was spoken words, text messages, or other forms of interaction. The dialog section serves as the primary record of what was said, when it was said, and who was involved in each exchange. Dialogs contain the "ground truths" of the conversation.

Each dialog entry represents a distinct communication event within the broader conversation. This could be a single text message, a phone call, a video conference session, or any other form of communication. The dialog section maintains the chronological flow and context of the conversation, preserving not just what was communicated, but the timing and sequence of exchanges that give meaning to the interaction.

The identification and tracking of dialog content serves critical privacy and compliance functions. The structured format enables precise identification of which specific conversations contain personal information, allowing for targeted data subject rights fulfillment such as selective deletion of specific dialog segments rather than entire conversation records. The timestamp and party reference metadata provide essential context for privacy impact assessments and data retention decisions. Additionally, the content hash mechanism ensures data integrity while also enabling verification that external content has not been tampered with, which is crucial for maintaining the trustworthiness of conversation records in legal or compliance contexts.

The dialog section supports several types of content. Recording dialogs capture audio or video segments of the conversation. Text dialogs contain written exchanges such as chat messages, SMS, or email content. Transfer dialogs record call transfer events, identifying the transferee, transferor, and transfer target roles along with references to the original and resulting dialog segments. Incomplete dialogs capture failed or unanswered communication attempts, with a disposition parameter indicating the reason for failure (such as no-answer, busy, congestion, or voicemail without a message).

Each dialog entry carries rich metadata beyond the conversation content itself. An originator parameter explicitly identifies the initiating party (the caller, message sender, or conference host). A party_history array tracks temporal events within the dialog, such as when participants join, drop, go on hold, mute, or press DTMF keys, providing a detailed timeline of the interaction dynamics. The

application parameter identifies the communication platform or service provider (for example, distinguishing between different video conferencing services), while a message_id parameter enables cross-referencing with messaging system identifiers such as SMTP message-ids for deduplication and threading. A session_id parameter links the dialog to SIP Session-ID values for cross-system correlation in telephony environments.

The purpose of the dialog section is two-fold:

* *Content Representation*: It accurately captures the details of any conversation exchange -- be it spoken words, text messages, or other communication types. This ensures that the exact sequence and content are archived in a standardized format. The content appropriate to dialogs are any of the times and places where personal data is communicated and recorded: audio, video, email, fax, rich emails as examples.

* *Interoperability and Analysis*: The dialog’s structured format supports further analysis (such as transcription or sentiment analysis) and ensures that conversations can be reliably exchanged between systems. By storing metadata like timestamps and participant references, the dialog section also enables the reconstruction of events (such as when participants join or leave a conversation) and aids in analytic processing.

3.4. Attachments

Attachments carry the context of the conversation; storing supplemental files and data that support the conversation record. In the vCon, attachments are designed as an array of opaque objects. They can be documents, images or any valid mediatype that can serve the proper analysis and comprehension of the conversation by providing context. In them, they may contain expressions of consent, references to content authenticity, authorization receipts and business tracking objects.

In parallel with each opaque object is a set of metadata that enables semantic understanding of the contents without the exposure of the underlying data. Attachment metadata contains information such as the type of data it contains, the party or dialog it refers to, and whether or not the data is contained in the attachment itself, or is referenced by external network identifier. A purpose parameter provides a free-form description of why the attachment is included, enabling downstream systems to understand the relevance of supplemental materials without necessarily inspecting their contents. Each attachment is linked to a specific party (the contributor, not necessarily the author) and a specific dialog segment through index references, maintaining clear provenance within the conversation record.

3.5. Analysis

The analysis section of a vCon contains processed insights and derived information from the original conversation data, serving as a bridge between the raw conversation data and business intelligence applications. The analysis section increases the utility of the conversation record by transforming raw dialog data into meaningful information. This can include automatically deriving summaries, measuring sentiment, extracting key topics, and identifying action items. Common analysis types include reports, sentiment assessments, summaries, transcripts, translations, and text-to-speech renderings. Such insights are pivotal in generating business intelligence, facilitating quality control, and supporting various applications where actionable information from conversations is crucial.

Each analysis entry identifies its provenance through a required vendor parameter, which names the service or organization that produced the analysis. This is important because different analysis implementations can produce significantly different results in quality, format, and interpretation. An optional product parameter differentiates between multiple offerings from the same vendor, while a schema parameter labels the specific data format or configuration used to generate the analysis. Together, these parameters enable consumers of vCon data to understand exactly how analysis was produced and to select appropriate processing logic for different analysis sources.

Analysis data represents a significant privacy consideration as it often contains processed personal information that may reveal insights about individuals beyond what is explicitly stated in the original conversation. This includes inferred characteristics, behavioral patterns, emotional states, and other derived information that could be considered personal data under privacy regulations. The vCon creator and processors must ensure that any analysis

performed complies with applicable privacy laws and that data subjects are informed about what analysis is being conducted on their data.

The "Right to know" principle is particularly important in the analysis section, as individuals have the right to understand what information is being derived from their conversations and how it is being used. This includes transparency about what types of analysis are being performed, what insights are being generated, and how those insights are being applied. Organizations processing vCons must provide clear information about the analytical processes being applied to conversation data, including the purposes of analysis, the types of insights being generated, and how those insights may be used to make decisions about individuals.

The analysis section enables organizations to extract value from conversations while maintaining accountability for how personal information is processed. By clearly documenting what analysis has been performed and linking it back to specific dialog segments, the analysis section supports compliance with data subject rights requests, enables audit trails for analytical processes, and provides transparency about how conversation data is being transformed into business intelligence.

3.6. Relationships between vCons

Relationships between vCons may also be defined, either through grouping, redaction or through amending past vCons. Groups of vCons can be expressed, to indicate general interrelationships. Redactions are at the heart of data minimization, a primary technique of personal data protection. vCons enable the sharing of limited data through redaction, while retaining the ability of systems to guarantee the accuracy of the redaction itself.

3.7. Extensions Mechanism

The vCon schema provides a formal mechanism for extending the core format to address specialized use cases and evolving requirements. Extensions allow new parameters to be defined at any level of the schema, and can also redefine the semantics of or deprecate existing parameters. This replaces the earlier approach of schema versioning through a version parameter, which has been deprecated.

Extensions are classified into two categories:

* *Compatible extensions* introduce additional data or fields without altering the meaning or structure of existing elements. Implementations that do not recognize these extensions can safely

ignore them while maintaining valid processing of the vCon. Wherever feasible, extensions should be designed as compatible to preserve interoperability.

* *Incompatible (critical) extensions* modify existing semantics or schema definitions in ways that require explicit awareness for correct interpretation. The names of all such extensions must be listed in the vCon's critical parameter, allowing implementations to determine whether they can safely process the vCon. An implementation that encounters an unsupported critical extension must reject the vCon rather than risk misinterpretation.

The distinction between compatible and incompatible is somewhat context-dependent. A transcription service can be fairly tolerant of new parameters added to a vCon. A redaction service, on the other hand, must be aware of the implications of all parameters to ensure complete redaction of sensitive information, and may need to reject vCons with any unrecognized extension.

Each extension must define a unique token value registered with IANA, specify its new or modified parameters and their semantics, and use snake_case naming conventions for parameter names. This framework ensures that vCon remains adaptable to industry-specific needs (such as contact centers, messaging platforms, and emergency services) while maintaining a consistent base format for data exchange.

3.8. Amended Use Cases

A vCon will often evolve over time. It is not always created with all of its metadata, conversation media, attachments, and analysis at once. There are several reasons for this:

* Different components of the vCon may be produced by different application platforms or entities.

* The vCon may pass across multiple trust boundaries during its lifecycle, with entities on either side contributing content.

* Each of these entities may wish to sign the vCon to attest to its integrity or to the authenticity of their contributions.

Once a vCon has been signed, it becomes immutable. Any modification will invalidate the signature. To address this, a new vCon can be created containing the updated or additional content. This new vCon can reference the previously signed version, preserving the integrity of the earlier state while allowing the overall conversation record to evolve.

This approach can also be applied even when a vCon is unsigned, for scenarios where maintaining a historical snapshot is important. For example, an application may wish to preserve a point-in-time representation of the vCon at a significant stage in its lifecycle.

There are two competing requirements in this use case:

* *Ease of use and access to the latest version of the vCon*

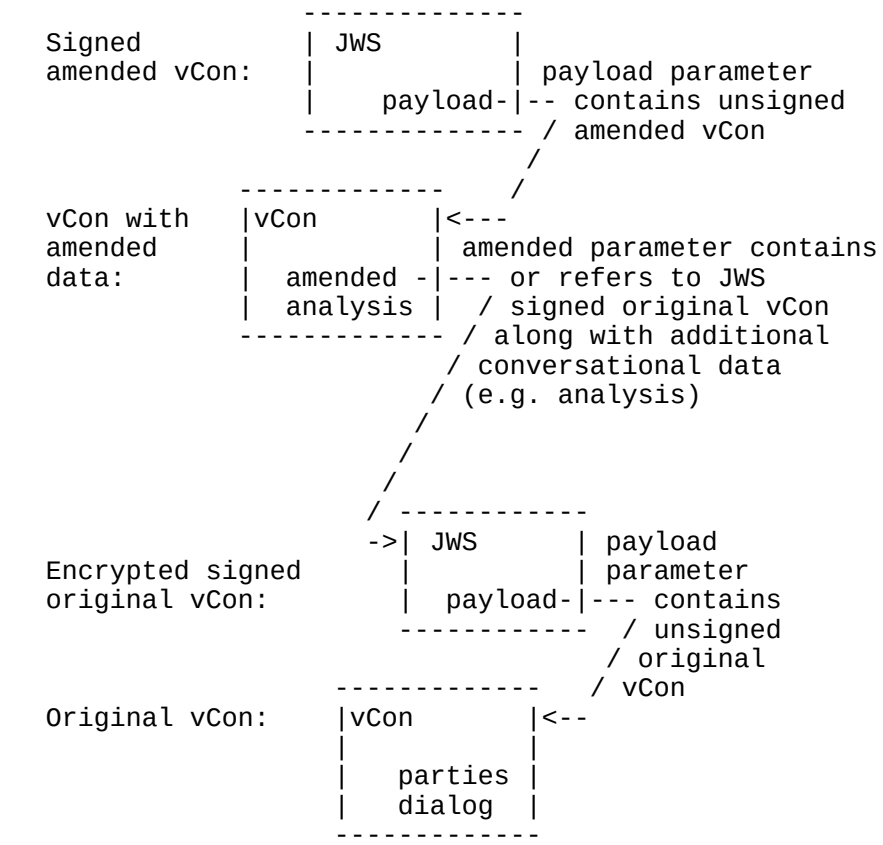
* *Data size minimization and normalization*

For ease of use, it is often desirable to work with a fully composed vCon that contains all prior and newly added or updated content. This has sometimes been referred to in vCon discussions as a "deep copy." Such a vCon requires no special handling, redirection, or reconstruction. It contains all relevant information in a single, self-contained structure.

However, this approach introduces duplication and increases document size. To address concerns around efficiency and data normalization, a more compact representation using _patches_ or _incremental updates_ may be preferred. This method allows changes to be recorded relative to earlier versions, reducing redundancy. Additionally, it enables labeling or referencing specific stages in the vCon’s lifecycle, offering a flexible way to manage changes. In vCon discussions, this method has been referred to as representing _incremental changes_.

3.8.1. Signed vCon Modified for Correction or Addition of Conversational Data

When a signed vCon requires correction or the addition of new information (such as analysis results produced after the original signing), a new vCon instance version is created using the amended parameter. The amended vCon is a deep copy of the prior version: it contains all of the original data plus the new or modified content. The amended object references the prior vCon instance version by UUID, and may optionally include a URL and content hash for direct retrieval and integrity verification of the earlier version.



This mechanism preserves the cryptographic integrity of the original signed vCon while allowing the conversation record to grow. Multiple rounds of amendment can form a chain: each amended vCon references its predecessor, creating a verifiable history of the conversation's evolution.

3.8.2. Capture of vCon in Various Lifecycle Stages

A vCon may be constructed across several security domains. Initially, a vCon exists in unsigned form while conversation data is being collected -- it may start with only metadata and party information, then accumulate dialog content, and later receive analysis results.

When a vCon is to be exported from one security domain to another, it should be signed or encrypted by the domain that constructed it. The receiving domain may then need to add new data (such as its own analysis results or additional metadata). Since the signed vCon is

immutable, the receiving domain creates a new amended vCon instance version containing the prior signed version plus any new content, and signs this new version when complete or before exporting to yet another domain.

This lifecycle pattern supports several practical scenarios:

* A communications platform captures dialog and parties, signs the vCon, and sends it to an analysis service

* The analysis service creates an amended vCon with transcription and sentiment analysis added, signs it, and forwards it to a business intelligence platform

* The business intelligence platform may further amend the vCon with categorization or disposition data

At each stage, the integrity of prior contributions is preserved through the chain of signatures, while the overall conversation record continues to grow with new information.

4. Security Considerations

The JSON form of a vCon is contained in a JSON object in one of three forms:

* unsigned - for internal use or trusted environments where data integrity and authenticity verification are not required

* signed - for scenarios requiring data integrity verification and authenticity confirmation without encryption, enabling tamper detection while maintaining readability

* encrypted - for sensitive conversations requiring confidentiality protection, ensuring that only authorized parties with proper decryption keys can access the conversation content

5. IANA Considerations

This document has no IANA considerations. They will be addressed in other vCon documents.

6. Informative References

[CBOR] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/RFC8949, December 2020, <https://www.rfc-editor.org/rfc/rfc8949>.

[CDDL] Birkholz, H., Vigano, C., and C. Bormann, "Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and JSON Data Structures", RFC 8610, DOI 10.17487/RFC8610, June 2019, <https://www.rfc-editor.org/rfc/rfc8610>.

[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>.

[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>.

[JWE] Jones, M. and J. Hildebrand, "JSON Web Encryption (JWE)", RFC 7516, DOI 10.17487/RFC7516, May 2015, <https://www.rfc-editor.org/rfc/rfc7516>.

[JWS] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <https://www.rfc-editor.org/rfc/rfc7515>.

[vCard] Kewisch, P., "jCard: The JSON Format for vCard", RFC 7095, DOI 10.17487/RFC7095, January 2014, <https://www.rfc-editor.org/rfc/rfc7095>.

[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>.

Acknowledgments

* Thank you to Daniel Petrie for making a concept real, for all the right reasons, and for the many projects we've shared over our careers.

Author's Address

Thomas McCarthy-Howe
Strolid
Email: thomas.howe@strolid.com