

Virtualized Conversations
Internet-Draft
Intended status: Standards Track
Expires: 26 July 2026

D. G Petrie
SIPEZ LLC
22 January 2026

The JSON format for vCon - Conversation Data Container
draft-ietf-vcon-vcon-core-02

Abstract

vCon is a standardized framework for the exchange of conversational data. Conversations, which may involve one or more participants, occur across a wide variety of modes and application platforms. This document defines a JSON format for representing conversational data, encompassing metadata, conversation media, related documents, and analysis. The goal of this standard is to provide an abstracted, platform-independent data format for conversations, regardless of the mode or application platform. By doing so, it facilitates the integration and seamless exchange of conversational data across application platforms, enterprises, and trust boundaries.

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

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-vcon-core>.

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 26 July 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	5
1.1. Purpose of vCon	5
1.2. Extending the vCon Schema	5
1.3. Benefits of vCon	6
1.4. *Scope of the Document*	6
2. Conventions and Definitions	6
2.1. Terminology	6
2.2. JSON Notation	8
2.3. Inline Files	9
2.3.1. body	9
2.3.2. encoding	9
2.4. Externally Referenced Files	10
2.4.1. url	10
2.4.2. content_hash	10
2.5. Extending vCon	10
3. vCon JSON Object	11
4. Unsigned Form of vCon Object	12
4.1. vCon JSON Object Keys and Values	12
4.1.1. vcon	13
4.1.2. uuid	13
4.1.3. extensions	13
4.1.4. critical	14
4.1.5. created_at	14
4.1.6. updated_at	14
4.1.7. subject	14

4.1.8.	redacted	15
4.1.9.	amended	17
4.1.10.	parties Objects Array	19
4.1.11.	dialog Objects Array	20
4.1.12.	analysis Objects Array	20
4.1.13.	attachments Objects Array	20
4.2.	Party Object	20
4.2.1.	tel	21
4.2.2.	sip	21
4.2.3.	stir	21
4.2.4.	mailto	21
4.2.5.	name	21
4.2.6.	did	22
4.2.7.	validation	22
4.2.8.	gmlpos	23
4.2.9.	civicaddress	23
4.2.10.	uuid	24
4.3.	Dialog Object	24
4.3.1.	type	25
4.3.2.	start	26
4.3.3.	duration	26
4.3.4.	parties	26
4.3.5.	originator	27
4.3.6.	mediatype	28
4.3.7.	filename	28
4.3.8.	Dialog Content	29
4.3.9.	disposition	29
4.3.10.	session_id	30
4.3.11.	party_history Objects Array	30
4.3.12.	Dialog Transfer	31
4.3.13.	application	33
4.3.14.	message_id	34
4.4.	Attachment Object	34
4.4.1.	purpose	34
4.4.2.	start	34
4.4.3.	party	34
4.4.4.	dialog	35
4.4.5.	mediatype	35
4.4.6.	filename	35
4.4.7.	Attachment Content	35
4.5.	Analysis Object	36
4.5.1.	type	36
4.5.2.	dialog	37
4.5.3.	mediatype	37
4.5.4.	filename	37
4.5.5.	vendor	37
4.5.6.	product	38
4.5.7.	schema	38

4.5.8. Analysis Content	38
5. Security Considerations	39
5.1. Signing Externally Referenced Files	41
5.2. Signed Form of vCon Object	41
5.2.1. Signature Object	42
5.2.2. Header Object	42
5.2.3. Long Term Archiving of Signed vCons	43
5.3. Encrypted Form of vCon Object	44
5.3.1. Unprotected Object	44
5.3.2. Recipient Object	45
5.3.3. Header Object	45
5.4. Differentiation of unsigned, signed and encrypted forms of vCon	45
6. IANA Considerations	46
6.1. JSON Format vCon Mediatype	46
6.2. gzip Format vCon Mediatype	47
6.3. vCon JSON Registry	48
6.3.1. Object Registry Template	48
6.3.2. vCon Object Parameter Names Registry	49
6.3.3. Party Object Parameter Names Registry	50
6.3.4. Dialog Object Parameter Names Registry	51
6.3.5. party_history Object Parameter Names Registry	54
6.3.6. Attachment Object Parameter Names Registry	56
6.3.7. Analysis Object Parameter Names Registry	57
6.3.8. Redacted Object Parameter Names Registry	58
6.3.9. Amended Object Parameter Names Registry	59
6.4. vCon Extensions Names Registry	60
6.4.1. Extensions Registration Template	60
6.5. JSON Web Signature and Encryption Header Parameters	60
6.6. JSON Web Signature and Encryption Unprotected Parameters	61
7. Non-Upward Compatible Changes to the vCon Container	61
7.1. Version 0.3.0 to 0.4.0	61
7.2. Version 0.0.2 to 0.3.0	61
7.3. Version 0.0.1 to 0.0.2	61
8. References	62
8.1. Normative References	62
8.2. Informative References	64
Appendix A. Example vCons	65
A.1. Two Party Call vCon With Inline Recording	65
A.2. Email Thread Multipart vCon	70
A.3. Email Thread Text vCon	73
A.4. Two Party Call vCon With Externally Referenced Recording	75
A.5. Two Party Call vCon with Analysis	76
A.6. Signed vCon	96
A.7. Encrypted vCon	101
A.8. Redacted vCon	108

A.9. Amended Signed vCon	110
Appendix B. vCon JSON Schema	112
Acknowledgments	124
Author's Address	125

1. Introduction

This document specifies the **vCon** (Conversational Data Exchange) standard, which defines a unified format for representing and exchanging conversational data across different communication platforms and modalities. The goal of **vCon** is to provide a standardized, platform-agnostic structure for capturing and sharing conversational information, allowing for seamless integration between systems, enterprises, and trust boundaries.

1.1. Purpose of vCon

The **vCon** standard enables the abstraction of conversation data, making it easier to integrate communication platforms with external services, including post-conversation analysis tools, customer support systems, and business intelligence platforms. By defining a consistent data format, **vCon** ensures interoperability across various platforms, regardless of the communication mode - whether text, audio, video, or other formats.

The **vCon** format captures the essential components of a conversation, including metadata, media content, related documents, and post-conversation analysis. This abstraction simplifies data exchange, apply AI-driven analysis, as well as model training and testing.

1.2. Extending the vCon Schema

In addition to the core **vCon** schema, this document provides a framework for extending the **vCon** format to support specialized use cases. The **vCon** schema can be customized to address the needs of specific industries or applications, such as:

- * **Contact Centers**: Adding specific metadata related to customer service interactions, case management, and priority tracking.
- * **Messaging Platforms**: Supporting the integration of different message formats, including SMS, email, and chat, group concepts, along with platform-specific information.
- * **ECRIT (Emergency Communication) Centers**: Extending the schema to capture critical data such as location, urgency, and incident details for emergency communications.

The ability to add extensions ensure that *vCon* remains flexible and adaptable, enabling its use in a wide range of domains while maintaining a consistent format for data exchange.

1.3. Benefits of vCon

The *vCon* standard offers several key benefits:

- * *Interoperability*: A consistent data format for conversational data facilitates integration across diverse platforms and systems.
- * *AI-Driven Analysis*: By capturing data in a structured format, *vCon* enables the application of AI analysis such as sentiment analysis, categorization, transcription, and decision-making.
- * *Data Automation*: The *vCon* format simplifies the automation of processes involving conversational data, improving efficiency and reducing the risk of errors.
- * *Enterprise Integration*: The ability to integrate conversation data with third-party tools and services enhances the value of conversations by providing deeper insights and driving business decisions.

1.4. *Scope of the Document*

This document defines the core structure and schema for *vCon* as a JSON format. Detailed specifications for the various parameters and the extension framework will be provided in subsequent sections. The primary focus is on creating a flexible, extensible standard for the exchange of conversational data that can be easily adopted across industries, use cases, and communication technologies.

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

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

- * conversation - an exchange of communication using text, audio or video medium between at least one human and one or more bots or humans
- * 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
- * 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).
- * 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
- * vCon syntax version - the version for the data syntax used for form a vCon
- * signed form - JWS signed document with the unsigned vCon form contained in the payload

2.2. JSON Notation

For the ease of documentation, the convention for [JSON] notation used in this document is copied from sections 1.1-1.5 of [JMAP].

- * Date - A string that MUST have the form of an [RFC3339] date string as defined for the Date type in section 1.4 of [JMAP].
- * "String" - a JSON string type
- * "UnsignedInt" - a positive JSON integer as defined in section 1.3 of [JMAP].
- * "UnsignedFloat" - a positive JSON floating point number containing a decimal point as defined in section 6 of [JSON].
- * "Mediatype" - A "String" value that MUST be of the following form as defined in section 5.1 of [MIME]:

type "/" subtype

- * "ContentHash" - The string token value is generated using the same approach used in section 6 of [STIR-PASS]. The relevant text is copied here for convenience and to remove the normative dependency. The hash string token values are formed from combining a string that defines the crypto algorithm used to generate the digest along with the Base64Url Encoded value of the SHA-512 hash, as defined in section 6.3 and 6.4 [SHA-512], of the body of the content at the given url. The hash algorithm is identified by "sha512". SHA-512 is part of the SHA-2 set of cryptographic hash functions [SHA-512] defined by the US National Institute of Standards and Technology (NIST). Implementations MAY support additional recommended hash algorithms in [IANA-COSE-ALG]; that is, the hash algorithm has "Yes" in the "Recommended" column of the IANA registry. Hash algorithm identifiers MUST use only lowercase letters, and they MUST NOT contain hyphen characters. The character following the algorithm string MUST be a hyphen character, "-", or ASCII 45. The subsequent characters are the Base64Url encoded (see Section 2 [JWS]) digest of a canonicalized and concatenated string or binary data based on the JSON pointer referenced elements of "rcd" claim or the URI referenced content contained in the claim.

- * "SessionId" - An object with the String parameters: "local" and "remote". The String values of these parameters are UUIDs as defined in section 5 of [SESSION-ID] as "local-uuid" and "remote-uuid". This is designed to also work with the H.323 correlated [H-460-27] which refers to them as "Sender-UUID" and "Recipient-UUID".

```
{ "local": String, "remote": String }
```

- * "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.

2.3. Inline Files

Objects that contain a file or data inline (i.e. within the vCon) MUST have the parameters: body and encoding. JSON does not support binary data values. For this reason inline files MUST be Base64url (see Section 2 [JWS]) encoded to be included as a valid JSON string value if they are not already valid JSON strings.

2.3.1. body

The body parameter contains the payload of the file to be included inline. Depending upon the data in the file, it may require encoding to be used as a valid JSON string value.

- * body: "String"

2.3.2. encoding

The encoding parameter describes the type of encoding that was performed on the string value of the body parameter. If the body parameter is provided and is not an empty string, the encoding parameter MUST be provided. In the interest of being tolerant for interoperability, if the encoding value is not provided, applications can attempt to derive the encoding token value based upon the mediatype parameter or filename extension if either is provided.

- * encoding: "String"

This MUST be one of the following strings:

- * "base64url": The payload of the file has been Base64Url encoded and provided as the string value of the body parameter.

- * "json": The value of the body string is a JSON object.
- * "none": The payload of the file is a valid JSON string and can be included without modification as the string value to the body parameter.

2.4. Externally Referenced Files

Files and data stored externally from the vCon MUST be "signed" to ensure that they have not been modified. Objects that refer to a file which is externally stored from the vCon MUST have the parameters: url, content_hash. These parameters are defined in the following subsections. The values and format of the hashes are defined in Section 2.2 type: ContentHash. The use of [SHA-512] hash for ensuring that the externally referenced data or file has not been modified, is defined in this document. Other methods of ensuring integrity may be added in the future.

2.4.1. url

The url value contains the [HTTPS] URL where the externally referenced file is stored. HTTPS MUST be used for retrieval to protect the privacy of the contents of the file.

- * url: "String"

2.4.2. content_hash

The integrity of externally referenced files are protected using a hash of the external content. The algorithm and hash value are contained in the content_hash parameter. The value of the content_hash can be a single token or an array of tokens if multiple hash algorithms are desired to be provided. The SHA-512 [SHA-512] algorithm MUST be supported. Other algorithms MAY be included. The algorithm used for signing the externally referenced file is defined in section 6.3 and 6.4 of [SHA-512].

- * content_hash: "ContentHash" | "ContentHash[]"

2.5. Extending vCon

vCon provides a means to extend the schema defined in this document. A vCon extension can define new parameters at any level of the schema. It can also redefine the semantics of or deprecate existing parameters. To do so, a new extension definition MUST do the following: * Define a new Section 4.1.3 token value and register it with IANA (see examples) (Section 6.3)) * Define the new parameter(s), their meaning, value types and register them with IANA

- * Define the new semantics and values for change parameters
- * List deprecated parameters, what should be used in its stead and migration approaches
- * Use snake case naming convention for parameter names (as opposed to camel case)

Extensions to the vCon schema can be classified into two categories:

- * ***Compatible***: These 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.
- * ***Incompatible***: These extensions modify existing semantics or schema definitions in ways that render a vCon incompatible with implementations that do not support the extension. Interpreting such a vCon correctly requires explicit awareness of the extension.

Wherever feasible, extensions ***SHOULD*** be designed as ***Compatible*** to preserve compatibility with existing implementations. The question of Compatible vs Incompatible is a little subjective. A transcriber for vCon can be fairly tolerant of changes adding new parameters or objects to a vCon. On the other hand, a redactor must be aware of the implications of all parameters in the vCon to be sure to redact all of the objective information. In these cases a transcriber will likely only need to look at the critical extensions. However the redactor perhaps should reject vCons with any extension that it does not support.

However, when disruptive extensions are necessary, the names of all such extensions ***MUST*** be listed in the critical (Section 4.1.4) parameter of the vCon. This allows implementations to determine whether they are capable of processing the vCon safely and accurately.

Implementations that encounter a vCon containing a disruptive extension listed in the critical parameter, but do not support that extension, ***MUST NOT*** process the vCon except to reject it or notify the user.

3. vCon JSON Object

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

- * unsigned
- * signed

* encrypted

The unsigned form of the vCon has a single top level object. This top level vCon object is also contained as described in the signed (Section 5.2) and encrypted (Section 5.3) forms of the vCon. The selection of the JSON format enables interchange between application and lower layers of the network stack, critical for enablement of analysis of conversations. The signed form of vCon is defined in Section 5.2 using [JWE] to provide integrity and proof of the conversation data. The encrypted form of vCon is defined in Section 5.3 to provide complete privacy of the conversational data. Partial privacy or data minimization can be provide for a vCon using the Section 4.1.8 construct.

4. Unsigned Form of vCon Object

The unsigned form of the top level vCon object is necessary as in many cases, a vCon will be partially constructed and in process as conversation data is collected. This may change while the conversation is in progress or on-going. The vCon may start with only meta data and party information, then progress to contain dialog information. It may then get analysis added or it could be passed to another security domain for analysis.

A vCon may be constructed across several security domains. 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 subsequent domain may have need to redact or amend data to the vCon. Alternatively the originating domain may want to redact the vCon before providing it to an other domain. The second or subsequent domain, MAY modify the prior vCon instance version and when complete or exporting to another security domain, it SHOULD sign or encrypt the new vCon instance version. The new vCon instance version SHOULD refer to the prior vCon instance version via the redacted (Section 4.1.8) or amended (Section 4.1.9) parameters.

4.1. vCon JSON Object Keys and Values

The keys and values for the top level vCon JSON object are defined in the following subsections.

4.1.1. vcon

The "vcon" parameter is DEPRECATED as of the publication of this document as an RFC. The "vcon" parameter was used to differentiate schema changes to the vCon core container when incompatibilities were created in Internet-Draft versions. The extension mechanism replaces the need for schema versioning. The vcon parameter was used to contain the syntactic version of the JSON format used in the vCon. (see Non-Upward Compatible Changes to the vCon Container (Section 7))

* vcon: "String"

For syntax defined in this document, the string MUST have the value: "0.4.0"

4.1.2. uuid

The [UUID] for the vCon is used to refer to it when privacy or security may not allow for inclusion or URL reference to a vCon. The UUID MUST be globally unique.

* uuid: "String"

The value of the string SHOULD be generated using the version 8 UUID defined in [UUID] which is generated identically to a version 7 UUID with the exception that:

- * rand_b/custom_c is generated from the high 62 bits of the SHA-1 hash of the FQHN string
- * the variant and version fields are set as described for version 8 UUID

The DNS name string used in generating the uuid value SHOULD be the same FQHN, or a subdomain to allow for more distributed generation of UUIDs, as would be used in the signing certificate as they are the same domains of uniqueness.

4.1.3. extensions

The vCon schema can be extended through the definition of new extensions. The extensions parameter SHOULD contain the list of names of all vCon extensions for any parameters used that are not defined in this core vCon schema document.

* extensions: "String[]"

4.1.4. critical

Implementations that include extensions which are incompatible with the core vCon schema MUST list the names of those extensions in the critical parameter. A vCon that includes a critical parameter indicates that correct interpretation of the vCon requires explicit support for the listed extensions. An implementation that does not recognize or support the extensions listed in the critical parameter MUST NOT attempt to process or operate on the vCon, except to reject it or report unsupported content.

* critical: "String[]"

4.1.5. created_at

The created_at parameter provides the creation time of this vcon, which MUST be present, and should not changed once the vcon object is created.

* created_at: "Date"

4.1.6. updated_at

The updated_at parameter provides the last modified time of this vcon, which MAY be present. For unsigned forms, the updated_at of the vcon object SHOULD be updated as it is modified. For signed forms of the vCon object, the updated_at contains the time when this vcon was last signed. Future updates MUST first set the updated_at to the new signing time as the vcon object is signed again.

* updated_at: "Date"

4.1.7. subject

The subject or the topic of the conversation is provided in the subject parameter. This parameter is optional as not all conversations have a defined subject. Email threads and prescheduled calls and video conferences typically have a subject which can be captured here. The subject may also be derived from analysis of the dialog.

* subject: "String" (optional)

The string value of the subject parameter is a free formed JSON string with no constrained syntax.

4.1.8. redacted

A redacted vCon SHOULD provide a reference to the unredacted or prior, less redacted, vCon instance version of itself. The purpose of the Redacted Object (Section 4.1.8.1) is to provide the reference to the unredacted or less redacted version of the vCon from which this vCon was derived. For privacy reasons, it may be necessary to redact a vCon to construct another vCon without the PII. This allows the non-PII portion of the vCon to still be analyzed or used in a broader scope. The Redacted Object (Section 4.1.8.1) SHOULD contain the uuid parameter and MAY include the url, content_hash parameters (see Externally Referenced Files (Section 2.4)). If a reference to the unredacted vCon is provided in the url parameter, the access to that URL MUST be restricted to only those who should be allowed to see the identity or PII for the redacted vCon.

The method(s) for redaction of text, audio and video can be done with existing post processing of media. The method of redaction is out of scope of this document. The assurance of the accuracy of the redaction is made by the entity that creates the redaction which SHOULD sign the redacted version of the vCon.

All data and parameters in the prior, less redacted, vCon instance version are either:

- * Removed entirely in the redacted version
- * Copied and partially redacted
- * Copied as is

Data which is to be completely removed from the redacted version, that is contained in a JSON array in the unredacted vCon, SHOULD create an empty placeholder such that object array indices do not change for the rest of the elements of the array.

- * redacted: "Redacted" (optional, mutually exclusive with amended and group parameters)

4.1.8.1. Redacted Object

A Redacted Object contains the following parameters:

- * uuid: "String"

The value contains the uuid string value (Section 4.1.2) of the unredacted/prior vCon instance version. The absence of a value for uuid indicates that the less redacted version of this vCon is not available or does not exist.

* type: "String"

The value of the redacted type parameter is used to indicate the type of redaction that was performed on this vCon relative to the less redacted version referenced by the redacted uuid parameter. This should indicate the type of information that was redacted.

The the location of the referenced vCon MAY be provided as defined in Externally Referenced Files (Section 2.4) by including url and content_hash. content_hash MUST be included if url is provided.

* url: "String"

* content_hash: "ContentHash" | "ContentHash[]"

The following diagram illustrates an example partial JSON object tree for a redacted vCon. The top level object is a JWS signed vCon which contains a vCon in the unsigned form in the payload parameter. The second level object is the redacted vcon which refers to the encrypted unredacted vCon in it's redacted parameter. Note that the redacted vCon references the JWE encrypted vCon by UUID and may reference it by URL or direct inclusion. The JWE encrypted unredacted vCon contains the signed version of the unredacted vCon in the ciphertext parameter. The signed unredacted vCon contains the unredacted vCon in the unsigned form in it's payload parameter.

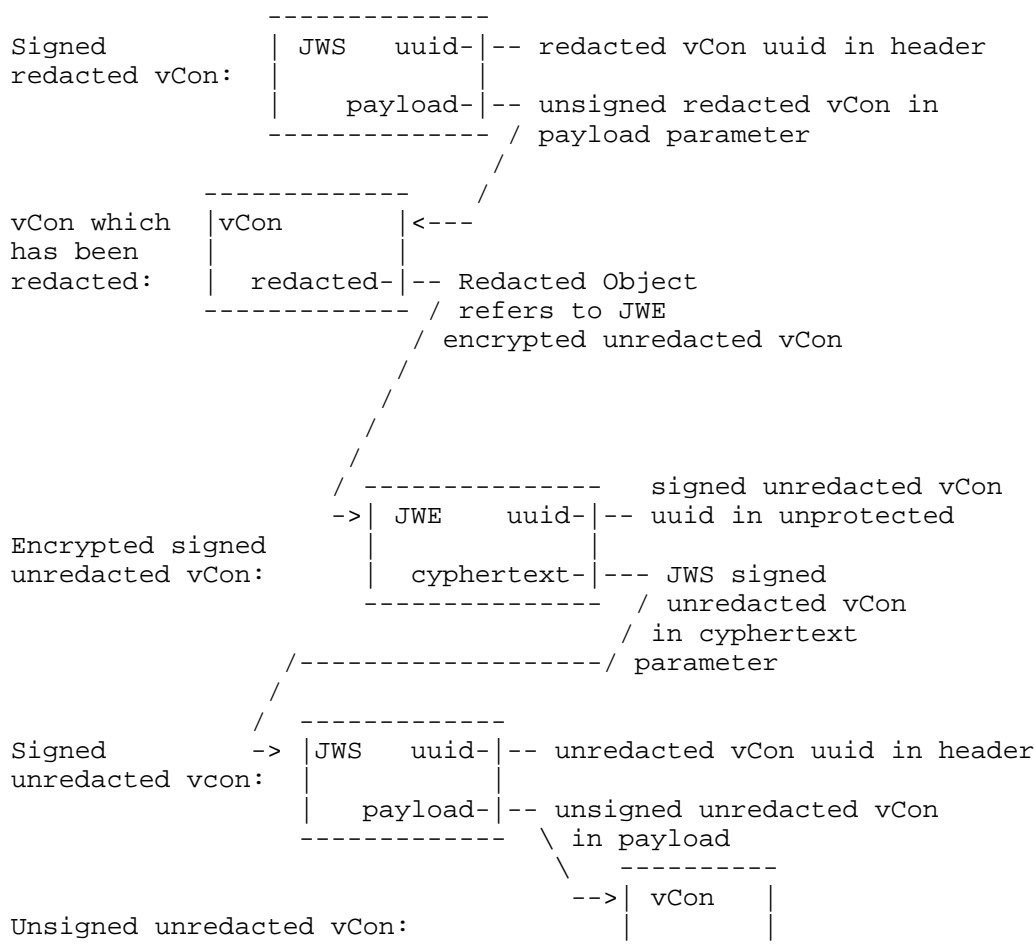


Figure 1: redacted vCon object tree

4.1.9. amended

A signed or encrypted vCon cannot be modified without invalidating it. In these cases, to allow for adding of additional information a new vCon instance version MUST be created. The prior vCon instance version is referenced by the Amended Object (Section 4.1.9.1). The vCon with amended data contains all of the data that is in the referenced vCon with the exception of data that is amended. That is to say that the newer version of the vCon is a deep copy of the prior version with the amended and additional data added to it.

The prior vCon instance version SHOULD be referenced via the uuid of the prior vCon instance version, and MAY include the url and content_hash parameters (see Externally Referenced Files (Section 2.4)).

- * amended: "Amended" (optional, mutually exclusive with redacted and group parameters)

4.1.9.1. Amended Object

The Amended Object contains the following parameters:

- * uuid: "String" (optional if inline or external reference provided)

The value contains the uuid string value (Section 4.1.2) of the unredacted/original vCon instance version.

The location of the referenced vCon MAY be provided, as defined in Externally Referenced Files (Section 2.4) by including url and content_hash. content_hash MUST be provided if url is provided.

- * url: "String"
- * content_hash: "ContentHash" | "ContentHash[]"

The following figure illustrates an example partial JSON object tree for an amended vCon. The top level object is the JWS signed amended vCon which contains the unsigned form of the vCon in its payload parameter. The second level object is the amended vCon with additional conversational data (e.g. analysis data). It refers to its original parent (or prior vCon instance version) of the vCon in its amended parameter. Note: the amended parameter may refer to the prior version of the vCon via URL. The amended vCon in this figure refers to the JWS signed version of the vCon, which in turn contains the original vCon in unsigned form in its payload parameter.

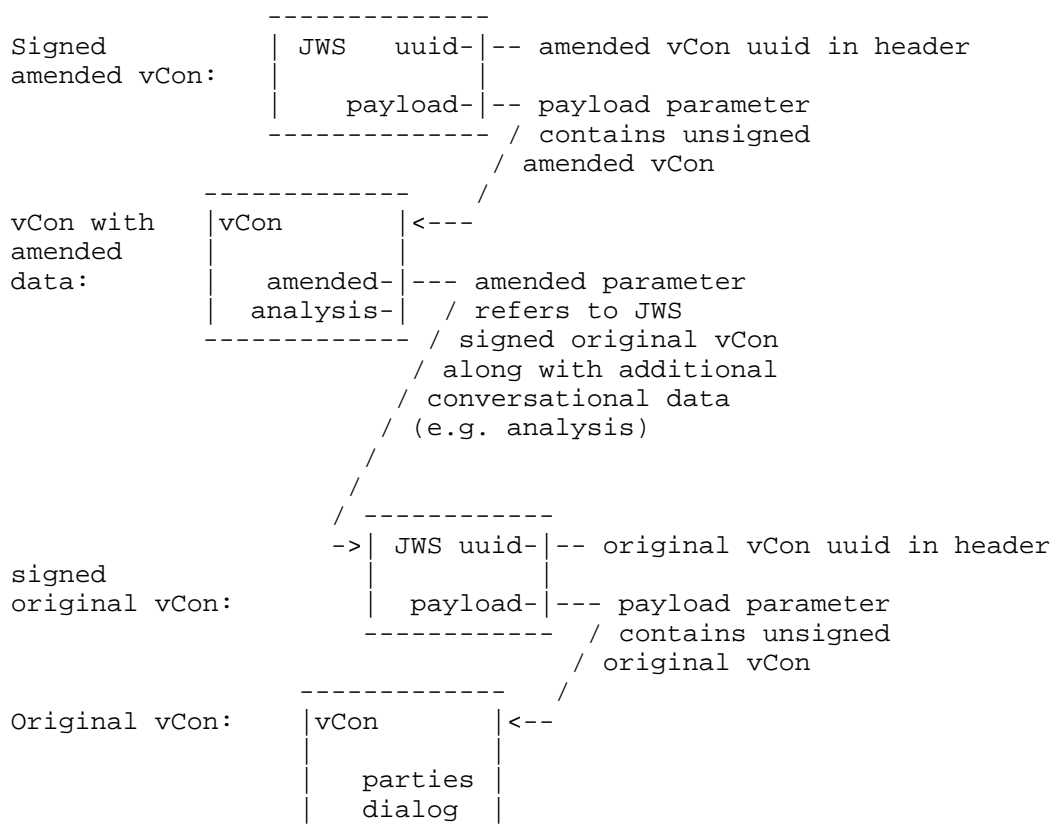


Figure 2: amended vCon object tree

4.1.10. parties Objects Array

The name, identity or contact information of all of the parties involved with the conversation are included in the parties object array. Whether the parties were observers, passive or active participants in the conversation, they each are included as a Party Object in the parties array.

* parties: "Party[]"

The value of the parties parameter is an array of Party Objects (Section 4.2).

4.1.11. dialog Objects Array

The actual conversation which occurred over text, audio or video that was captured, is contained in the dialog Objects Array.

* dialog: "Dialog[]" (optional)

The value of the dialog parameter is an array of Dialog Objects (Section 4.3). It should be noted that the Dialog Objects may not be sorted in chronological order. They are in the order that they were added. It is the applications responsibility to sort dialog and transcriptions if they are needed in chronological order.

4.1.12. analysis Objects Array

Analysis, which is performed on the conversational data, is stored in the analysis Objects array.

* analysis: "Analysis[]" (optional)

The value of the analysis parameter is an array of Analysis Objects (Section 4.5).

4.1.13. attachments Objects Array

Ancillary documents, discussed, presented, referenced or related to the conversation may be stored in the attachments array. It should be noted that the Attachment Objects may not be sorted in chronological order. They are in the order that they were added.

* attachments: "Attachment[]" (optional)

The value of the attachments parameter is an array of Attachment Objects (Section 4.4).

4.2. Party Object

A Party Object contains information about a specific party or participant in the conversation. The information that is available about the party varies with the context, the communication mode and platform. There are times when no information is available for a party either initially or over the entire life of the vCon. In such situations, it is possible to have a Party Object with no parameters in it. A Party Object SHOULD be created for each participant. This may result in more than one empty Party Object. They are distinct by the order or index in the Party Object array.

4.2.1. tel

If the tel URL for the party is known, it can be included in the tel parameter.

* tel: "String" (optional)

The value of the tel parameter SHOULD be a valid [TEL] URL. The URL scheme prefix (i.e. "tel:") is optional.

4.2.2. sip

The [SIP] URL for the party can be provided in the sip parameter.

* sip: "String" (optional)

The value of the sip parameter is the addr-spec as defined in section 25.1 of [SIP].

4.2.3. stir

If the STIR [PASSporT] was provided to identify the party, the PASSporT can be included in the stir parameter for the party.

* stir: "String" (optional)

The string value of the stir parameter contains the [PASSporT] in the JWS Compact Serialization form.

4.2.4. mailto

If the mailto URL is known for the party, it can be included in the mailto parameter.

* mailto: "String" (optional)

The value of the mailto parameter is a string of the format of a valid [MAILTO] URL. The URL scheme prefix (i.e. "mailto:") is optional.

4.2.5. name

If the party's name is known, it can be included in the name parameter.

* name: "String" (optional)

The string value of the name parameter is a free form JSON string in which part or all of the party's name can be included. It may be necessary, for privacy reasons, to not identify any information about the party. This MAY be indicated by creating a Party Object with only the name parameter and setting its value to "anonymous". A Party Object SHOULD be create for each anonymous party in the conversation. This is necessary to capture the number of parties and distiction between which party contribued what data or dialog in the conversation. Hence it is possible to have multiple anonymous Party Objects in the vCon. They are distinct by the order or index in the Party Object array.

4.2.6. did

If a Decentralized Identifier [DID] is known for the party, it can be included in the did parameter.

* did: "String" (optional)

The value of the did parameter MUST be a valid DID URI that enables verifiable, decentralized digital identity for the party. DIDs are designed to be decoupled from centralized registries and identity providers, allowing the controller of the DID to prove control without requiring permission from other parties. The DID can refer to any subject (person, organization, thing, etc.) as determined by the controller of the DID.

4.2.7. validation

Proof of authorization of the communication channel through STIR, login or possession of a device, is often not sufficient proof of the identity of the person at the other end of the communications channel. It is common in call centers to validate the identity of the person on the communication channel through verification of some sort of personal identification information. The methods used, often varies with the situation and is business practices specific. The purpose of the validation parameter, is to allow the validator to save a label or token which identifies the method of identity validation used to identify the person at the other end of the communication channel. For confidentially reasons, it SHOULD NOT contain the data used to validate the name. However it MAY name the data used to validate the name (e.g. "SSN", "DOB", "user ID and password"). It is up to the domain creating the vCon to define the set of tokens or values to be used for the validation parameter.

* validation: "String" (SHOULD be provided if name parameter is provided)

The value of the validation string MAY be "none" or enterprise or domain defined token or string values.

4.2.8. gmlpos

If the geolocation of the party is known, it can be added in the gmlpos parameter. The gmlpos parameter contains the latitude and longitude of the location of the party at the time of the conversation start.

* gmlpos: "String" (optional)

The gmlpos parameter value contains a string of the same format as the gml:pos element defined in section 3.2 of the [PIDF-LO] PIDF.

4.2.9. civicaddress

The the civicaddress parameter value contains a Civicaddress Object which is optional and contains civic address information about the location for the party.

* civicaddress: "Civicaddress" (optional)

The Civicaddress Object MAY contain any or all of the following parameters: country, a1, a2, a3, a4, a5, a6, prd, pod, sts, hno, hns, lmk, loc, flr, nam, pc as defined in section 2.2.1 of [GEOPRIV]. The values of all of these parameters are of type String. Note that the parameter names MUST be in lower case when contained in the Civicaddress Object even though they are in upper case in [GEOPRIV].

* country: "String" (optional)

* a1: "String" (optional)

* a2: "String" (optional)

* a3: "String" (optional)

* a4: "String" (optional)

* a5: "String" (optional)

* a6: "String" (optional)

* prd: "String" (optional)

* pod: "String" (optional)

- * sts: "String" (optional)
- * hno: "String" (optional)
- * hns: "String" (optional)
- * lmk: "String" (optional)
- * loc: "String" (optional)
- * flr: "String" (optional)
- * nam: "String" (optional)
- * pc: "String" (optional)

4.2.10. uuid

The uuid is a unique identifier for the participant. In a contact center, this is particularly important for the call agent participant, and must be static across interactions to allow correlation with the actual agent configuration provisioned into the systems.

- * uuid: "String" (optional)

4.3. Dialog Object

The Dialog object references or contains text, audio or video captured from the conversation. Currently four types of dialog objects are defined in this document:

- * Text based media communications
- * Audio or Video recorded media communications
- * Metadata for failed or incompletd communications
- * Metadata for providing relationships between other Dialog Objects in transfer scenarios

Media-based Dialog Objects (type text and recording) SHOULD contain only media that is transcribable. Media that is transcribable can be converted into a textual representation of sequential communication between parties. While images such as sunsets, cows, photographs of documents, or application forms may be describable, they are not transcribable in a concise manner and SHOULD NOT be included as Dialog Objects. They SHOULD be included as Attachment Objects.

Text and audio may be subject to interpretation, but they are transcribable to a precise dialog representation. Although transcriptions may not always be accurate, there exists a single correct transcription for any given content.

Media that is not transcribable SHOULD be included in the vCon as Attachment Objects rather than Dialog Objects.

For example, a fax or photograph of a text-based letter is transcribable and MAY be included as a Dialog Object. Most other faxes and photographs SHOULD be included as Attachment Objects and SHOULD NOT be included as Dialog Objects. Similarly, a musical recording, even if it is in MP3 audio format, SHOULD be included as an Attachment and SHOULD NOT be included as a Dialog Object.

It may not always be known at the time of vCon construction whether an image or recording is transcribable. When uncertain, the default SHOULD be to include images as Attachment Objects and audio or video recordings as Dialog Objects.

This distinction is important for interoperability. If it is ambiguous as to what belongs in a Dialog Object versus an Attachment Object, interoperability cannot be achieved, as vCon constructors will not be able to consistently determine where content should be placed and users of vCons will not know where content can be found within the vCon.

There are situations when no information is available for a dialog either initially or over the entire life of the vCon and yet it is known that the dialog occurred. For example this may occur in some call transfer cases where there is nothing known about the consultative call. In such situations, it is possible to have a Dialog Object with no parameters in it. There may even be more than one empty Dialog Object. They are distinct by the order or index in the Dialog Object array.

4.3.1. type

* type: "String"

The string MUST have the value of either "recording", "text", "transfer" or "incomplete". A dialog of type "recording" has Dialog Content that either contains a body or refers to via url, which is a recording of the video and/or audio of a segment of the conversation. A dialog of type "text" has Dialog Content that either contains a body or refers to via url, which contains the text from one of the parties for a segment of the conversation. A dialog of type "transfer" does not capture actual conversation exchange, but rather

captures operations, parties and relations between dialog segments. A dialog of type "incomplete" or "transfer" MUST NOT have Dialog Content. In the "incomplete" case the call or conversation failed to be setup to the point of exchanging any conversation. Incomplete Dialog Objects MUST have a disposition parameter which indicates why the call or conversations failed. In the "transfer" case, the conversation is recorded in other Dialog Objects. The Dialog Transfer parameters, are used to show the roles and relationships between the parties and other Dialog Objects as the transfer process occurred.

4.3.2. start

The start parameter contains the date and time for the beginning of the captured piece of dialog. For text it is the time that the party started typing or if not available, then it is the time the text was sent. For audio and video recordings, it is the time which corresponds to the beginning of the recording. It should be noted that Dialog Objects may not be ordered by the value of the start parameter. Dialog Objects in the dialog array are in order that they were added to the vCon and cannot be reordered without correcting the dialog indices which occur in other Objects in the vCon.

* start: "Date"

4.3.3. duration

The duration parameter contains the duration in seconds of the referenced or included piece of dialog. For text, if known, it is the time duration from when the party started typing to when they completed typing and the text was sent. For recordings, it is the duration of the recording.

* duration: "UnsignedInt" | "UnsignedFloat" (optional)

The value MUST be the dialog duration in seconds.

4.3.4. parties

The party(s) which generated the text or recording for this piece of dialog are indicated in the parties parameter.

* parties: "UnsignedInt" | "UnsignedInt[]" | ("UnsignedInt" | "UnsignedInt[]")[]

Single channel recordings should have a parties value of the form: "UnsignedInt" or "UnsignedInt[]" where the integer value or array of integer values are the indices to the Party Object(s) in the parties

array that contributed to the mix for the single channel recording. The index for the Party Object SHOULD be included even if the indicated party was silent the entire piece of dialog.

Multi-channel recordings MUST have a parties value that is an array of the same size as the number of channels in the recording. The values in that array are either an integer or an array of integers which are the indices to the parties that contributed to the mix for the associated channel of the recording. The index for Party Objects SHOULD be included even if the party was silent the entire conversation.

It is implied that the first party in the dialog Object parties list, is the originator of the dialog. However, in some situations, it is difficult to impose the constraint that the first channel of a recording is the originator. If ensuring that the first channel and party listed is the originator is not possible, the optional originator parameter indicates the originator party. In other cases, there may be a separate recording file for each party in a conversation, where only one party is recorded per file. In this situation, it is necessary to indicate the originator as the dialog Object parties parameter may contain only one party and may not be the originator. It should be noted that the recordings may be shorter than the collective conference when there is a separate recording per speaker/party. For example the recording(s) of the individual speaker may be trimmed to only the parts where they are actually speaking.

In the email context, [SMTP] messages have the headers From, To, Cc and Bcc that all correspond to parties to the text Dialog Object. The From header should be the first party index in the parties parameter for the text Dialog Object. They are the implied originator. The parties listed in the To, Cc and Bcc [SMTP] headers (if present) should all be added in that order to the parties parameter in the text Dialog Object. The sender SHOULD be allowed to set a policy, through out of band means, as to whether they would like their application or server to include the Bcc parties in the vCon. A text Dialog Object should only contain a single [SMTP] message. Each may have a different set of parties.

4.3.5. originator

The originator party for this dialog. For email and text, this is the sender of the text. For audio and video, this is the caller or host of the call or conference. The originator parameter is only provided if the first party of the dialog Object parties list is NOT the originator.

- * `originator`: "UnsignedInt" (optional)

The `originator` value is the index into the `parties` Objects Array, to the party that originated the dialog.

4.3.6. `mediatype`

The media type for the piece of dialog included or referenced is provided in the `mediatype` parameter. The `mediatype` parameter MUST be provided for inline dialog files and MUST be provided if the `Content-Type` header in the [HTTPS] response for the externally referenced URL is not provided.

- * `mediatype`: "Mediatype" (optional for externally referenced files, if absent, this is provided in the [HTTPS] `Content-Type` header)

The media types SHOULD be one of the following strings:

- * "text/plain"
- * "audio/x-wav"
- * "audio/x-mp3"
- * "audio/x-mp4"
- * "audio/ogg"
- * "video/x-mp4"
- * "video/ogg"
- * "multipart/mixed"

Note: multi-part will also likely include the `boundary` parameter

4.3.7. `filename`

It is sometimes useful to preserve the name of the file which originally contained this piece of dialog. This can be done in the `filename` parameter.

- * `filename`: "String" (optional)

4.3.8. Dialog Content

The Dialog Object SHOULD contain the body and encoding parameters or the url and content_hash parameters for all dialog types other than "incomplete" and "transfer", these parameters MUST NOT be present for "incomplete" or "transfer" dialog types (see Inline Files (Section 2.3) and Externally Referenced Files (Section 2.4)). The exception to this is that the body or url MAY be absent if it is redacted.

For inline included dialog:

- * body: "String" (optional in an a redacted vCon)
- * encoding: "String"

Alternatively, for externally referenced dialog:

- * url: "String" (optional in an a redacted vCon)
- * content_hash: "ContentHash" | "ContentHash[]"

4.3.9. disposition

If the dialog type is "incomplete", it must have a disposition parameter. The value of the disposition parameter provides the reason that the "call control" failed. The term: "call control" is used in a loose sense, as there is not always a call involved, to differentiate from a call disposition that an agent may assign to a call to indicate the reason, issue addressed or outcome of a conversation. This latter definition of call disposition is not dialog, but analysis of the conversation and is not included in the dialog portion.

- * disposition: "String" (required for incomplete type Dialog Objects, SHOULD NOT be present for other dialog types)

The value of the disposition parameter MUST be one of the following string:

- * "no-answer" - a call or connection was attempted, but no one answered or accepted the connection
- * "congestion" - a call or connection was attempted, but was unable to be completed due to system load
- * "failed" - a call or connection was attempted, but failed

- * "busy" - a call or connection was attempted, but the party was busy with another conversation
- * "hung-up" - a call or connection was made, but the party hung-up before any conversation occurred
- * "voicemail-no-message" - a call or connection was made, the voicemail system answered, but no message was left

Note: if a message was left with the voicemail system this is no longer an "incomplete" type dialog, it is a "recording" type and the conversation SHOULD be included in the Dialog Content.

4.3.10. session_id

The [SESSION-ID] MAY be included for the dialog. If the same [SESSION-ID] applies to all the parties in a dialog, than a single instance of the SessionId Object is provided as the value of session_id. If some parties have a different or no associated [SESSION-ID] for this Dialog Object, then the array structure of the session_id objects MUST correlate to the array structure of the Dialog Object's parties index array. Parties which do not have a [SESSION-ID] associated with this dialog, MUST have an empty object ({}) for the SessionId Object. That an empty Object and not an Object with two "nil UUIDs" as defined in section 7 of [SESSION-ID]. It is also possible that the SessionId Object will not have values for both the local and remote parameters in some conferencing situations (See [SESSION-ID]). In the case that one end of the [SESSION-ID] does not have a UUID, the "nil UUID" String value SHOULD be used as defined in section 7 of [SESSION-ID].

- * session_id: "SessionId" | "SessionId[]" | ("SessionId" | "SessionId[]")[] (optional)

4.3.11. party_history Objects Array

Participants in a dialog may not all join and leave at the same time. To support the capturing of the information when parties join, drop, go on hold or mute and unmute, the party_history array may be added to the Dialog Object.

- * party_history: "Party_History[]" (optional)

4.3.11.1. Party_History Object

The Party_History Object contains the following parameters:

- * party: "UnsignedInt"

The index of the party for this event.

* time: "Date"

The time at which this event occurred.

* event: "String"

The string token for the event which MUST be one of the following:

- "join" - when the party joins the dialog
- "drop" - when the party drops out of the dialog
- "hold" - when the party is put on hold
- "unhold" - when the part is taken off hold
- "mute" - when the party is muted
- "unmute" - when the part is taken off mute
- "keydown" - when a DTMF or application key/button was pressed
- "keyup" - when a DTMF or application key/button was released

* button: "String" - (required for keydown and keyup events)

The button parameter value is the String value of the DTMF digit, character or string label for the button that was pressed or released.

4.3.12. Dialog Transfer

A dialog of type "transfer" documents the rolls of three parties and the relationship between 2 or three dialog segments. In a transfer operation, the roles of the three parties to a transfer are defined in [SIP-XFER] as:

* Transferee

* Transferor

* Transfer Target

There are two or three calls in which the parties are connected:

* original call

- * consultative call (optional as this call may not get created)
- * target call

To capture the above roles and dialog segments, the following parameters are defined and SHOULD be present in the "transfer" type dialog and MUST NOT be present in other dialog types.

- * transferee: "UnsignedInt"

The value of the transferee parameter is the index into the parties Object array to the party that played the role of the Transferee.

- * transferor: "UnsignedInt"

The value of the transferor parameter is the index into the parties Object array to the party that played the role of the Transferor.

- * transfer_target: "UnsignedInt" | "UnsignedInt[]"

The value of the transfer_target parameter is the index into the parties Object array to the party that played the role of the Transfer Target.

The consultation, target_dialog and transfer parameters all refer to the Dialog Objects that correspond to the 2 to 3 calls that are part of a transfer. These calls may end up spread across multiple Dialog Objects due to the nature of how the calls are recorded. For example each party may be recorded in a separate file which will result in a dialog for each. Alternatively a call may go on hold where recording is stopped and back off again resulting where recording is started again in a separate recording file, resulting in multiple Dialog Objects. For this reason, the values for the consultation, target_dialog and transfer parameters MAY have a single UnsignedInt or an array of UnsignedInt.

There are scenarios where we know that a transfer has occurred, but we have no Dialog Object information for one or two of the consultation, target or transfer calls. In this case an empty Dialog Object is created and its index is used for the consultation, target_dialog or transfer parameter. A unique Dialog Object SHOULD be referenced for each role in the transfer. However a Dialog Object may be referenced in more than one transfer dialogs when multiple transfers occur.

- * original: "UnsignedInt" | "UnsignedInt[]"

The value of the original parameter is the index/indices into the dialog Object array to the "recording" or "text" type Dialog Object for the original dialog between the Transferee and the Transferor.

* consultation: "UnsignedInt" | "UnsignedInt[]" (optional)

The value of the consultation parameter is the index/indices into the Dialog Object array to the "recording", "text" or "incomplete" type Dialog Object for the consultative dialog between the Transferor and the Transfer Target. It is also possible for there to be more than one consultation. This may occur for a number of reasons. Call attempts may fail. The caller may decide the consultation with a party is not the desired transfer target.

* target_dialog: "UnsignedInt" | "UnsignedInt[]"

The value of the target_dialog parameter is the index/indices into the Dialog Object array to the "recording", "text" or "incomplete" type dialog for the target dialog between the Transferee and the Transfer Target.

A "transfer" type dialog MUST NOT contain the parties, originator, mediatype, filename or Dialog Content parameters.

The "transfer" type dialog only captures the roles, operations and events of the parties and the dialog setup. It does not capture the purpose or reason for the transfer as that is analysis to be captured in the analysis section of the Vcon after the conversation has occurred.

4.3.13. application

The application parameter is used to capture the application, communication channel or context over which the conversation was held. The conversation mode can be identified by looking at the dialog type. However that does not differ between different communication platform service providers or hosting service. For example, the application parameter can be used to identify the web conference hosting service.

* application "String" (optional)

4.3.14. message_id

Some messaging systems assign a unique message identifier to each message. The message_id parameter may be used to label the message for either cross referencing back to the messaging system or to prevent the addition of duplicate messages to the vCon. For example SMTP [SMTP] messages have a message-id in the SMTP header. In the SMTP case a message may exist in multiple mailboxes or tags. When exporting these SMTP messages, the message_id parameter may be used to avoid adding the same message, to the vCon containing the email thread, more than once.

* message_id: "string" (optional)

4.4. Attachment Object

Ancillary documents to the conversation are included or referenced in the Attachment Object. There is no constraint on the types files which can be included or referenced. As most modes of communication, that allow the exchange of files, do not constrain the file type, any file type may be included here.

4.4.1. purpose

The purpose parameter may be used to provide a text base description of the purpose or point of the attachment. There is no restricted set of values for this description.

* purpose: "String" (optional)

4.4.2. start

The start parameter contains the date and time that the Attachment Object was sent/exchanged. It should be noted that Attachment Objects may not be ordered by the value of the start parameter. Attachment Objects in the attachments array are in order that they were added to the vCon and cannot be reordered without correcting the attachment indices which may occur in other Objects in the vCon.

* start: "Date"

4.4.3. party

In most conversations, ancillary documents originate from one of the parties to the conversation. This is not necessarily the author, but the person who distributed the document. This party is identified by the party parameter in the Attachment Object.

- * party: "UnsignedInt"

The value of the party parameter is the index into the Party Object array to the party that contributed the attachment.

4.4.4. dialog

Attachments are added as data related to the dialog. The dialog parameter is used to identify the dialog that the attachment is part of. The dialog is identified by the index to the dialog Object in the dialog objects array.

- * dialog: "UnsignedInt"

4.4.5. mediatype

The media type for the included or referenced attachment file is provided in the mediatype parameter.

- * mediatype: "Mediatype" (optional for externally referenced files, if absent, this is provided in the [HTTPS] Content-Type header)

The mediatype string contains the media type or [MIME] type of the attached file.

4.4.6. filename

It is sometimes useful to preserve the name of the file which originally contained this attachment file. This can be done in the filename parameter.

- * filename: "String" (optional)

The file name string contains an optional name for the attachment file referenced in this Attachment Object.

4.4.7. Attachment Content

The Attachment Object SHOULD contain the body and encoding parameters or the url and content_hash parameters (see Inline Files (Section 2.3) and Externally Referenced Files (Section 2.4)). The exception to this is that the body or url MAY be absent if it is redacted.

For inline included attachments:

- * body: "String"

- * encoding: "String"

Alternatively, for externally referenced attachments:

- * url: "String"

- * content_hash: "ContentHash" | "ContentHash[]"

4.5. Analysis Object

Analysis is a broad and in some cases developing field. This document does not attempt to suggest a SHOULD support list of types. Nor are formats for analysis data defined in this document. That is for research and specification outside the scope of this document. For this reason the Analysis Object provides multiple ways to define the media type of the analysis file. If a well known media type is defined, it SHOULD be used. For analysis data or files types for which a media type is not defined, the vendor and schema parameters SHOULD be used.

4.5.1. type

The type parameter is used to label the semantic type of analysis data.

- * type: "String"

TODO: this list is not sufficient. Do we enforce a specific set of tokens. Seems like analysis has a pretty huge set of possibilities.

The string value SHOULD be one of the following:

- * "report"

- * "sentiment"

- * "summary"

- * "transcript"

- * "translation"

- * "tts"

4.5.2. dialog

Analysis typically pertains to one or more of the Dialog Objects in the dialog array. The dialog parameter is used to indicate which Dialog Objects this analysis was based upon.

- * dialog: "UnsignedInt" | "UnsignedInt[]" (optional only if the analysis was not derived from any of the dialog)

The value of the dialog parameter is the index to the dialog or array of indices to the Dialog Object array to which this analysis object corresponds.

4.5.3. mediatype

The media type for the included or referenced analysis file is provided in the mediatype parameter.

- * mediatype: "Mediatype" (optional for externally referenced files, if absent, this is provided in the [HTTPS] Content-Type header)

The mediatype string contains the media type or [MIME] type of the analysis file.

4.5.4. filename

It is sometimes useful to preserve the name of the file which originally contained this analysis data. This can be done in the filename parameter.

- * filename: "String" (optional)

The file name string contains an optional name for the analysis data file referenced in this Analysis Object.

4.5.5. vendor

There may not be a IANA registered media type for the file format containing the analysis data. Even if a media type is defined, it is often useful to keep a record of the vendor that produced the software that produced the analysis. Different implementations perform differently and knowing the implementation can be useful in interpreting the analysis. For this reason, the vendor or product name is provided in the vendor parameter.

- * vendor: "String"

The vendor string value contains the vendor or product name of the software which generated this analysis.

4.5.6. product

As the vendor may have more than one product, the product parameter may be used to differentiate it from other products which may have different schemas of analysis data.

- * product: "string" (optional)

4.5.7. schema

The same vendor or software product may produce different formats or data for the same analysis. The schema parameter allows the data format, schema or configuration used to generate the analysis to be saved with the included or referenced analysis data.

- * schema: "String" (optional)

The schema string contains a token or label for the data format or schema for the analysis data. As the vendor name may not be specific enough to identify the format of the analysis, the schema value is provide to differentiate from potentially multiple data formats for analysis provided by the same vendor or software.

4.5.8. Analysis Content

The Analysis Object SHOULD contain the body and encoding parameters or the url and content_hash parameters (see Inline Files (Section 2.3) and Externally Referenced Files (Section 2.4)). The exception to this is that the body or url MAY be absent if it is redacted.

For inline included analysis:

- * body: "String"

- * encoding: "String"

Alternatively, for externally referenced analysis:

- * url: "String"

- * content_hash: "ContentHash" | "ContentHash[]"

5. Security Considerations

The security concerns for vCons can put into two categories: making the conversation immutable through integrity verification and protecting the confidentiality of privacy of the parties to the conversation and/or their PII. These requirements along with need to evolve a vCon (e.g. adding analysis, translations and transcriptions) conflict in some ways. To enable this, multiple versions of a vCon may be created. Versions of a vCon may add information (e.g. analysis added to a prior vCon referenced by the amended (Section 4.1.9)) and versions that remove information (e.g. redactions of privacy information removed from the vCon referenced in the redacted (Section 4.1.8)). Redactions may be at different levels for example:

- * PII masked to remove PII data in the text, audio, video or transcripts
- * De-identified to remove segments or whole recordings that might be used for voice printing or facial recognition

Different parts and versions of a vCon may be created in different security domains over a period of time. In addition, some conversation data may be referenced externally through an HTTPS URL as opposed to completely contained within the vCon. Typically a conversation of one mode, will be hosted or observed in a single domain. This will likely fall into one of the following hosting situations:

- * Enterprise Hosted Communications
- * Software as a Service (SaaS) Hosted Communications
- * Service Provider Hosted Communications

The distinction among these has gotten clouded over recent years. The import consideration is that each is a different security domain. Information about a conversation captured in an enterprise communications system (e.g. meta data and Dialog Object(s) recorded in an IP PBX) is a different security domain from a SaaS transcription service (i.e. an Analysis Object). Before a vCon leaves a security domain, it SHOULD be signed to prevent it from being altered. If the new security domain needs to alter it, a new vCon is created with the removed or added data and the prior version is referenced (i.e. via the redacted (Section 4.1.8) or amended (Section 4.1.9)). See the redacted vCon object tree figure-1 and amended vCon object tree figure-2. If information is redacted for privacy reasons, the vCon referenced in the redacted (Section 4.1.8), if inline, SHOULD be encrypted to protect the privacy information in the unredacted version of the vCon.

The secure storage and access of externally referenced conversation data is considered out of scope from this document. Secure mechanisms for HTTPS access and storage of files are well defined. Identity and credentials for accessing externally stored data will be exchanged out of band from the vCon. The one requirement for externally referenced data from the perspective of this document, is proof of integrity of that data.

Using the above described approach for redaction and amending of data, we can reduce the security operations on a vCon to signing and encryption. Two approaches to signing are needed as we have data, in JSON format, that is contained within the vCon and may have data (typically media and file formats, often binary) not contained, inline in the vCon, that is externally referenced.

Externally referenced data will be "signed" using [SHA-512] hash which along with the URL of the externally referenced data is included in the vCon. [SHA-512] was chosen due to the relatively low cost to generate and verify the signature for what could be very large externally referenced media files. As the hash for each externally referenced file is contained in the vCon which will be signed, the chain of authentication is provided via the signature on the vCon itself.

This document specifies the JSON format for vCons. So it seemed the logical solution for signing vCons, is JOSE [JWS] JSON Serialization and likewise for encrypting vCons is JOSE [JWE] JSON Serialization. The solutions are well documents, implementations are readily available and tested.

Methods of redaction exist for text, audio and video using post processing of the media. The method of redaction used is out of the scope of this document. A redacted vCon SHOULD reference it's non-redacted version. The non-redacted version of the vCon referenced from the redacted vCon MUST be encrypted such that only those with permission to view the non-redacted content can decrypt it.

Any time a vCon is shared outside its original security domain, it SHOULD be signed and optionally encrypted. Files externally referenced by a vCon SHOULD always be signed with the verification information included in the vCon that references the external file as defined in Externally Referenced Files (Section 2.4) and Signing Externally Referenced Files (Section 5.1). Externally referenced files SHOULD only be transported over [HTTPS] and SHOULD be access controlled to those who are permitted to read the contents of that non-redacted vCon. vCons transported over non-secure channels such as email MUST be in the encrypted form.

5.1. Signing Externally Referenced Files

In some deployments, it is not practical to include all of the file contents of a vCon inline. In support of that, a file may be externally referenced. When external files are referenced, the signature on the vCon does not secure the file contents from modification. For this reason any externally referenced files SHOULD also have a signature. vCons use the [SHA-512] hash method for integrity checking of externally referenced file content and include its url and content_hash in the vCon which are included in the integrity signature for the whole vCon.

After retrieving externally referenced files, before using its content, the payload of the HTTPS request should be verified using the signature parameter value for the hash for the url body using the procedure defined in section 6.3 and 6.4 of [SHA-512].

5.2. Signed Form of vCon Object

A signed vCon uses [JWS] and takes the General JWS JSON Serialization Syntax form as defined in section 7.2.1 of [JWS]. The vCon General JWS JSON Serialization MUST include x5c or x5u in the unprotected header.

* payload: "String"

The value of the payload is the Base64Url Encoded string containing either: the unsigned form of the JSON vCon; or the gzipped [GZIP] unsigned form of the JSON vCon. The general construction of the payload string value is described in section 7.2.1 of [JWK]

- * signatures "Signature[]"

The value of signatures is an array of Signature Objects (Section 5.2.1)

5.2.1. Signature Object

The Signature Object MUST contain a header, protected and signature parameter as defined in section 7.2.1 of [JWS].

- * header: "Header"

The value of header is defined in Header Object (Section 5.2.2)

- * protected: "String"

- * signature: "String"

5.2.2. Header Object

The Header Object and its contents are defined in section 4 of [JWS]. The Header Object for a signed vCon MUST include the alg and either the x5c or x5u arrays. The x5c or x5u requirement makes the management and use of vCons easier, allowing the certificate chain to be found as the vCon is moved.

- * alg: "String"

The string value of alg is defined in section 4.1.1 of [JWS]. For a signed vCon this value SHOULD be "RS256" to maximize interoperability.

- * x5c: "String[]" (MUST provide x5c or x5u)

The array of string values for x5c are defined in section 4.1.6 of [JWS].

- * x5u: "String" (MUST provide x5c or x5u)

The string value of x5u MUST contain an [HTTPS] URL as defined in section 4.1.5 of [JWS].

- * uuid: "String" (SHOULD be provided)

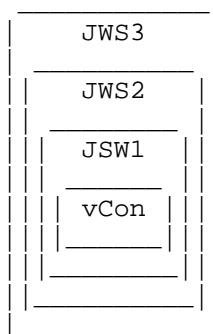
The string value of the UUID for the vCon contained in the payload parameter. This is added for convenience to not require verification of the signed vCon to get it's UUID. When the vCon is verified, the value of this uuid parameter SHOULD be verified against the UUID in the signed vCon.

5.2.3. Long Term Archiving of Signed vCons

In some use cases, it may be necessary to store or archive signed vCons for compliances or records purposes for as long as 10 years or more. In most cases the keys and certificates used to sign the vCon and construct the [JWS] or signed form of the vCon, will have an expiration on the order of a few years at best. Below is an approach on how integrity may be maintained on the archived vCon. There may be other approaches to this as well.

For our example illustrated below, we have a vCon which was signed (labeled as JWS1). The certificate used to sign JWS1 has an expiration date. Before the expiration of the JWS1 signing certificate, the content of JWS1 (the vCon) is verified and then JWS1 is signed as the payload to construct JWS2. Before the expiration of the JWS2 certificate, the contents of JWS2 (JWS1) is verified and then JWS2 is signed as the payload to construct JWS3. The process is continued for the life of the vCon archiving.

When it is desired to access the data in the archived vCon, first the outer JSW (JWS3 in diagram) is verified (payload of JWS2), then the vCon (payload in JWS1) is assumed to have valid integrity.



5.3. Encrypted Form of vCon Object

A vCon MUST be signed first using JWS as defined in Signed Form of vCon Object (Section 5.2), then encrypted using JWE as opposed to just encrypted with integrity protection. The rationale is that meta data and dialog will typically be collected in one security domain, then may be stored or exported to another. The signing is likely for the lifetime of the vCon, where the encryption may be shorter term or domain specific. vCons may be stored in unencrypted form, but the signed form MUST be maintained to ensure its integrity.

A encrypted vCon uses [JWE] and takes the General JWE JSON Serialization Syntax form as defined in section 7.2.1 of [JWE].

- * unprotected: "Unprotected"

- * recipients: "Recipient[]"

- * iv: "String"

The string value of iv is the Initialization Vector as constructed as defined in section 7.2.1 of [JWE].

- * ciphertext: "String"

The string value of ciphertext is constructed as defined in section 7.2.1 of [JWE] using the signed form of the vCon as the plaintext input for encryption.

- * tag: "String"

The string value of tag is the the Authentication Tag as defined in section 7.2.1 of [JWE].

5.3.1. Unprotected Object

- * cty: "String"

The string value of cty SHOULD be either "application/vcon" or "application/vcon+gzip"

- * enc: "String"

The string value of enc SHOULD be "A256CBC-HS512"

- * uuid: "String" (SHOULD be provided)

The string value of the UUID for the vCon contained in the ciphertext parameter. This is provided for identification of the encrypted vCon without the need of the key to decrypt the vCon. The value of this uuid parameter SHOULD be verified when the vCon is decrypted.

5.3.2. Recipient Object

- * header: "Header"
- * encrypted_key: "String"

The string value of encrypted_key is defined in section 7.2.1 of [JWE].

5.3.3. Header Object

- * alg: "String"

The string value of alg SHOULD be "RSA-OAEP".

5.4. Differentiation of unsigned, signed and encrypted forms of vCon

vCons in the unsigned, signed and encrypted form are all valid JSON. It is necessary to distinguish which form a vCon is in order to properly handle it. For example a signed vCon needs to be verified. The following is the recommended approaches to identifying which form a particular vCon is in:

- * unsigned

The presence of one or more of the following parameters indicates that the vCon is in the unsigned form:

- parties
- dialog
- analysis
- attachments

- * signed (JWS)

The presence of both of the following parameters indicates that the vCon is in the signed (JWS) form:

- payload

- signatures
- * encrypted (JWE)

The presence of both of the following parameters indicates that the vCon is in the encrypted (JWE) form:

- ciphertext
- recipients

6. IANA Considerations

This section includes the information required for IANA to register the application/vcon media type per [MEDIATYPE].

6.1. JSON Format vCon Mediatype

The JSON format for vCon is mandatory to implement.

- * Type name: application
- * Subtype name: vcon
- * Required parameters: N/A
- * Optional parameters: N/A; unrecognized parameters should be ignored.
- * Encoding considerations: 8bit
- * Additional information: [JSON] format
- * Security considerations: See Security Considerations section of this document.
- * Interoperability considerations: The vCon schema is defined in RFC XXXX. The schema may be extended through extensions as defined there.
- * Published specification: [VCON], this document: RFC XXXX
- * Applications that use this media type:
- * Fragment identifier considerations: N/A
- * Additional Information:

- Deprecated alias names for this type: N/A
- Magic number(s): N/A
- File extension(s): "vcon"
- Macintosh file type code(s): N/A
- Windows Clipboard Name: VCON
- * Author: See the Author's Addresses section of this document.
- * Change controller: IETF

6.2. gzip Format vCon Mediatype

A gzip format vCon is a JSON format vCon that has been compressed using the gzip [GZIP] compression algorithm. The gzip format is used to reduce the size of potentially very large vCons containing audio or video media in dialog objects. Applications SHOULD support the receiving of gzip format vCons, but do not need to use the gzip format when sending.

- * Type name: application
- * Subtype name: vcon+gzip
- * Required parameters: N/A
- * Optional parameters: N/A; unrecognized parameters should be ignored.
- * Encoding considerations: binary
- * Additional information: [GZIP] compression of [JSON] format
- * Security considerations: See Security Considerations section of this document.
- * Interoperability considerations: The vCon schema is defined in RFC XXXX. The schema may be extended through extensions as defined there.
- * Published specification: [VCON], this document: RFC XXXX
- * Applications that use this media type:
- * Fragment identifier considerations: N/A

* Additional Information:

- Deprecated alias names for this type: N/A
- Magic number(s): N/A
- File extension(s): "vcon"
- Macintosh file type code(s): N/A
- Windows Clipboard Name: VCON

* Author: See the Author's Addresses section of this document.

* Change controller: IETF

6.3. vCon JSON Registry

This document defines a number of new IANA registries for the JSON vCon schema. Each Object (i.e. dict, map, dictionary) has its own registry of parameter names. All of those registries have the same registry template which is defined in the following subsection. When registering new entries, the following template should be used.

6.3.1. Object Registry Template

Parameter Name:

The string name of a parameter to be used within the Object. The name is case sensitive. Names may not match other registered names in a case-insensitive manner unless the Designated Experts state that there is a compelling reason to allow an exception.

Parameter Description:

Brief description of the parameter

Change Controller:

For Standards Track RFCs, list the "IESG". For others, give the name of the responsible party. Other details (e.g., postal address, email address, home page URI) may also be included.

Specification Document(s):

Reference to the document or documents that specify the parameter, preferably including URIs that can be used to retrieve copies of the documents. An indication of the relevant sections may also be included but is not required.

6.3.2. vCon Object Parameter Names Registry

The following defines the initial values for the vCon Object Parameter Names Registry. Use the template in Object Registry Template (Section 6.3.1) when registering additional entries to this table.

Parameter Name	Parameter Description	Change Controller	Specification Document(s)
vcon	Schema version number	IESG	Section 4.1.1 RFC XXXX
uuid	vCon instance UUID	IESG	Section 4.1.2 RFC XXXX
extensions	list of extensions used	IESG	Section 4.1.3 RFC XXXX
critical	list of incompatible extensions used	IESG	Section 4.1.4 RFC XXXX
created_at	creation date	IESG	Section 4.1.5 RFC XXXX
updated_at	modification date	IESG	Section 4.1.6 RFC XXXX
subject	conversation subject	IESG	Section 4.1.7 RFC XXXX
redacted	Redacted Object	IESG	Section 4.1.8 RFC XXXX
amended	Amended Object	IESG	Section 4.1.8 RFC XXXX
group	reserved for future extension	IESG	RFC XXXX
parties	Party Objects array	IESG	Section 4.1.10 RFC XXXX

dialog	Dialog Objects array	IESG	Section 4.1.11 RFC XXXX
analysis	Analysis Objects array	IESG	Section 4.1.12 RFC XXXX
attachments	Attachment Objects array	IESG	Section 4.1.13 RFC XXXX

Table 1

6.3.3. Party Object Parameter Names Registry

The following defines the initial values for the Party Object Parameter Names Registry. Use the template in Object Registry Template (Section 6.3.1) when registering additional entries to this table.

Parameter Name	Parameter Description	Change Controller	Specification Document(s)
tel	tel URL	IESG	Section 4.2.1 RFC XXXX
sip	SIP URL	IESG	Section 4.2.1 RFC XXXX
stir	STIR token	IESG	Section 4.2.3 RFC XXXX
mailto	mailto URL	IESG	Section 4.2.4 RFC XXXX
name	participant name	IESG	Section 4.2.5 RFC XXXX
validation	validation method	IESG	Section 4.2.7 RFC XXXX
gmlpos	participant geolocation	IESG	Section 4.2.8 RFC XXXX
civicaddress	civic address	IESG	Section 4.2.9 RFC XXXX
uuid	participant UUID	IESG	Section 4.2.10 RFC XXXX

Table 2

6.3.4. Dialog Object Parameter Names Registry

The following defines the initial values for the Dialog Object Parameter Names Registry. Use the template in Object Registry Template (Section 6.3.1) when registering additional entries to this table.

Parameter Name	Parameter Description	Change Controller	Specification Document(s)
type	dialog type name	IESG	Section 4.3.1 RFC XXXX

start	dialog start time	IESG	Section 4.3.2 RFC XXXX
duration	dialog duration period	IESG	Section 4.3.3 RFC XXXX
parties	dialog parties list	IESG	Section 4.3.4 RFC XXXX
originator	dialog originator party	IESG	Section 4.3.5 RFC XXXX
mediatype	dialog content media type	IESG	Section 4.3.6 RFC XXXX
filename	dialog content filename	IESG	Section 4.3.7 RFC XXXX
body	dialog inline content	IESG	Section 4.3.8 RFC XXXX
encoding	dialog inline content encoding	IESG	Section 4.3.8 RFC XXXX
url	dialog referenced content URL	IESG	Section 4.3.8 RFC XXXX
content_hash	dialog content hash	IESG	Section 4.3.8 RFC XXXX
disposition	dialog disposition	IESG	Section 4.3.9 RFC XXXX
session_id	RFC7989 session ID	IESG	Section 4.3.10 RFC XXXX
party_history	dialog party events history	IESG	Section 4.3.11 RFC XXXX
transferee	transferee party index	IESG	Section 4.3.12 RFC XXXX
transferor	transferor party index	IESG	Section 4.3.12 RFC XXXX

transfer_target	transfer target party index	IESG	Section 4.3.12 RFC XXXX
original	original transfer dialog index	IESG	Section 4.3.12 RFC XXXX
consultation	consultation transfer dialog index	IESG	Section 4.3.12 RFC XXXX
target_dialog	target_dialog transfer dialog index	IESG	Section 4.3.12 RFC XXXX
application	dialog source application	IESG	Section 4.3.13 RFC XXXX
message_id	dialog message id	IESG	Section 4.3.14 RFC XXXX

Table 3

6.3.4.1. Dialog Type Name Registry

This document defines a new registry for the token name values that are permitted as values to the dialog type (Section 4.3.1) parameter.

6.3.4.1.1. Dialog Type Name Registration Template

Dialog Type Name:

The string token name of a Dialog Object Type The name is case sensitive. Names may not match other registered names in a case-insensitive manner unless the Designated Experts state that there is a compelling reason to allow an exception.

Dialog Type Description:

Brief description of the Dialog Object type

Change Controller:

For Standards Track RFCs, list the "IESG". For others, give the name of the responsible party. Other details (e.g., postal address, email address, home page URI) may also be included.

Specification Document(s):

Reference to the document or documents that specify the parameter, preferably including URIs that can be used to retrieve copies of the documents. An indication of the relevant sections may also be included but is not required.

6.3.4.1.2. Dialog Type Name Registration Initial Values

The following table defines the initial values for the Dialog Object Types Registry.

Dialog Type Name	Dialog Type Description	Change Controller	Specification Document(s)
text	text dialog	IESG	Section 4.3.1 RFC XXXX
recording	recording dialog	IESG	Section 4.3.1 RFC XXXX
transfer	transfer dialog	IESG	Section 4.3.1 RFC XXXX
incomplete	incomplete dialog	IESG	Section 4.3.1 RFC XXXX

Table 4

6.3.5. party_history Object Parameter Names Registry

This initial values for the party_history Object Parameter Names Registry are listed in the following table. Use the template in Object Registry Template (Section 6.3.1) when registering additional entries to this table.

Parameter Name	Parameter Description	Change Controller	Specification Document(s)
party	event party index	IESG	Section 4.3.11.1 RFC XXXX
time	time event occurred	IESG	Section 4.3.11.1 RFC XXXX
event	event type	IESG	Section 4.3.11.1 RFC XXXX

Table 5

6.3.5.1. party_event Event Name Registry

This document defines a new registry for the token name values that are permitted as values to the party_event Object event (Section 4.3.11.1) parameter.

6.3.5.1.1. party_event Event Name Registration Template

party_event Event Name:

The string token name of a party_event Object event type. The name is case sensitive. Names may not match other registered names in a case-insensitive manner unless the Designated Experts state that there is a compelling reason to allow an exception.

party_event Event Description:

Brief description of the party_event Object Event

Change Controller:

For Standards Track RFCs, list the "IESG". For others, give the name of the responsible party. Other details (e.g., postal address, email address, home page URI) may also be included.

Specification Document(s):

Reference to the document or documents that specify the parameter, preferably including URIs that can be used to retrieve copies of the documents. An indication of the relevant sections may also be included but is not required.

6.3.5.1.2. party_event Type Name Registration Initial Values

The following table defines the initial values for the Dialog Object Types Registry.

party_event Event Name	party_event Event Description	Change Controller	Specification Document(s)
join	party join event	IESG	Section 4.3.11.1 RFC XXXX
drop	party drop event	IESG	Section 4.3.11.1 RFC XXXX
hold	party on hold event	IESG	Section 4.3.11.1 RFC XXXX
unhold	party off hold event	IESG	Section 4.3.11.1 RFC XXXX
mute	party on mute event	IESG	Section 4.3.11.1 RFC XXXX
unmute	party off mute event	IESG	Section 4.3.11.1 RFC XXXX
keydown	DTMF/app key/ button press event	IESG	Section 4.3.11.1 RFC XXXX
keyup	DTMF/app key/ button release event	IESG	Section 4.3.11.1 RFC XXXX

Table 6

6.3.6. Attachment Object Parameter Names Registry

The following defines the initial values for the Attachment Object Parameter Names Registry. Use the template in Object Registry Template (Section 6.3.1) when registering additional entries to this table.

Parameter Name	Parameter Description	Change Controller	Specification Document(s)
purpose	attachment object purpose description	IESG	Section 4.4.1 RFC XXXX
start	attachment start time	IESG	Section 4.4.2 RFC XXXX
party	sending party index	IESG	Section 4.4.3 RFC XXXX
dialog	associated dialog index	IESG	Section 4.4.4 RFC XXXX
mediatype	attachment body media type	IESG	Section 4.4.5 RFC XXXX
filename	attachment content filename	IESG	Section 4.4.6 RFC XXXX
body	attachment inline content	IESG	Section 4.4.7 RFC XXXX
encoding	attachment inline content encoding	IESG	Section 4.4.7 RFC XXXX
url	attachment referenced content URL	IESG	Section 4.4.7 RFC XXXX
content_hash	attachment content hash	IESG	Section 4.4.7 RFC XXXX

Table 7

6.3.7. Analysis Object Parameter Names Registry

The following defines the initial values for the Analysis Object Parameter Names Registry. Use the template in Object Registry Template (Section 6.3.1) when registering additional entries to this table.

Parameter Name	Parameter Description	Change Controller	Specification Document(s)
type	analysis type	IESG	Section 4.5.1 RFC XXXX
dialog	associated dialog index	IESG	Section 4.5.2 RFC XXXX
mediatype	analysis body media type	IESG	Section 4.5.3 RFC XXXX
filename	analysis content filename	IESG	Section 4.5.4 RFC XXXX
vendor	vendor producing content	IESG	Section 4.5.5 RFC XXXX
product	vendor product producing content	IESG	Section 4.5.6 RFC XXXX
schema	product schema of content	IESG	Section 4.5.7 RFC XXXX
body	analysis inline content	IESG	Section 4.5.8 RFC XXXX
encoding	analysis inline content encoding	IESG	Section 4.5.8 RFC XXXX
url	analysis referenced content URL	IESG	Section 4.5.8 RFC XXXX
content_hash	analysis content hash	IESG	Section 4.5.8 RFC XXXX

Table 8

6.3.8. Redacted Object Parameter Names Registry

The following defines the initial values for the Redacted Object Parameter Names Registry. Use the template in Object Registry Template (Section 6.3.1) when registering additional entries to this table.

Parameter Name	Parameter Description	Change Controller	Specification Document(s)
uuid	less redacted vCon UUID	IESG	Section 4.1.8.1 RFC XXXX
type	redaction type or reason	IESG	Section 4.1.8.1 RFC XXXX
url	referenced less redacted vCon URL	IESG	Section 4.1.8.1 RFC XXXX
content_hash	less redacted vCon hash	IESG	Section 4.1.8.1 RFC XXXX

Table 9

6.3.9. Amended Object Parameter Names Registry

The following defines the initial values for the Amended Object Parameter Names Registry. Use the template in Object Registry Template (Section 6.3.1) when registering additional entries to this table.

Parameter Name	Parameter Description	Change Controller	Specification Document(s)
uuid	prior vCon version UUID	IESG	Section 4.1.9.1 RFC XXXX
url	referenced prior version vCon URL	IESG	Section 4.1.9.1 RFC XXXX
content_hash	prior version vCon hash	IESG	Section 4.1.9.1 RFC XXXX

Table 10

6.4. vCon Extensions Names Registry

This document defines and new registry for the token name values that are permitted as values to the [extensions]s(#extensions) parameter. However it does not define any values for this registry. Values for this registry are to be defined in subsequent documents which define extensions to the vCon schema.

6.4.1. Extensions Registration Template

Extension Name:

The string token name of an extension to the vCon JSON schema. The name is case sensitive. Names may not match other registered names in a case-insensitive manner unless the Designated Experts state that there is a compelling reason to allow an exception.

Extension Description:

Brief description of the the vCon extension

Change Controller:

For Standards Track RFCs, list the "IESG". For others, give the name of the responsible party. Other details (e.g., postal address, email address, home page URI) may also be included.

Specification Document(s):

Reference to the document or documents that specify the parameter, preferably including URIs that can be used to retrieve copies of the documents. An indication of the relevant sections may also be included but is not required.

6.5. JSON Web Signature and Encryption Header Parameters

This document registers the following new parameter in the [JWS] JSON Web Signature and Encryption Header Parameters registry.

- * Header Parameter Name: "uuid"
- * Header Parameter Description: UUID of the signed object
- * Header Parameter Usage Location(s): JWS
- * Change Controller: IESG

- * Specification Document(s): Header Object (Section 5.2.2) of RFC XXXX

6.6. JSON Web Signature and Encryption Unprotected Parameters

TODO: cannot find this registry

This document registers the following new parameter in the [JWS] JSON Web Signature and Encryption Unprotected Parameters registry.

- * Header Parameter Name: "uuid"
- * Header Parameter Description: UUID of the encrypted object
- * Header Parameter Usage Location(s): JWE
- * Change Controller: IESG
- * Specification Document(s): Header Object (Section 5.3.1) of RFC XXXX

7. Non-Upward Compatible Changes to the vCon Container

7.1. Version 0.3.0 to 0.4.0

- * The "appended" Object was renamed to "amended"
- * The "must_support" parameter was renamed to "critical" to be consistent with other IETF JSON schema extension mechanisms.
- * The "session_id" Dialog Object parameter was changes from a String to a SessionId Object.

7.2. Version 0.0.2 to 0.3.0

- * "transfer-target" parameter was renamed to "transfer_target" to be consistent with other compound names that use the underscore ("_").
- * "target-dialog" parameter was renamed to "target_dialog" to be consistent with other compound names that use the underscore ("_").

7.3. Version 0.0.1 to 0.0.2

- * "mimetype" parameters were renamed to "mediatype"
- * "alg" and "signature" were combined into "content_hash"

8. References

8.1. Normative References

- [DID] W3C, "Decentralized Identifiers (DIDs) v1.0", July 2022, <<https://www.w3.org/TR/did-core/>>.
- [GEOPRIV] Peterson, J., "A Presence-based GEOPRIV Location Object Format", RFC 4119, DOI 10.17487/RFC4119, December 2005, <<https://www.rfc-editor.org/rfc/rfc4119>>.
- [GZIP] Deutsch, P., "GZIP file format specification version 4.3", RFC 1952, DOI 10.17487/RFC1952, May 1996, <<https://www.rfc-editor.org/rfc/rfc1952>>.
- [H-460-27] "End-to-End Session Identifier for H.323 Systems", H.460.27, November 2015, <<https://www.itu.int/rec/T-REC-H.460.27>>.
- [HTTPS] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <<https://www.rfc-editor.org/rfc/rfc9110>>.
- [IANA-COSE-ALG]
"COSE Algorithms", n.d., <<https://www.iana.org/assignments/cose/cose.xhtml>>.
- [JMAP] Jenkins, N. and C. Newman, "The JSON Meta Application Protocol (JMAP)", RFC 8620, DOI 10.17487/RFC8620, July 2019, <<https://www.rfc-editor.org/rfc/rfc8620>>.
- [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>>.
- [JWK] Jones, M., "JSON Web Key (JWK)", RFC 7517, DOI 10.17487/RFC7517, May 2015, <<https://www.rfc-editor.org/rfc/rfc7517>>.
- [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>>.

- [MAILTO] Duerst, M., Masinter, L., and J. Zawinski, "The 'mailto' URI Scheme", RFC 6068, DOI 10.17487/RFC6068, October 2010, <<https://www.rfc-editor.org/rfc/rfc6068>>.
- [MEDIATYPE] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <<https://www.rfc-editor.org/rfc/rfc6838>>.
- [MIME] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, DOI 10.17487/RFC2045, November 1996, <<https://www.rfc-editor.org/rfc/rfc2045>>.
- [PASSporT] Wendt, C. and J. Peterson, "PASSporT: Personal Assertion Token", RFC 8225, DOI 10.17487/RFC8225, February 2018, <<https://www.rfc-editor.org/rfc/rfc8225>>.
- [PIDF-LO] Winterbottom, J., Thomson, M., and H. Tschofenig, "GEOPRIV Presence Information Data Format Location Object (PIDF-LO) Usage Clarification, Considerations, and Recommendations", RFC 5491, DOI 10.17487/RFC5491, March 2009, <<https://www.rfc-editor.org/rfc/rfc5491>>.
- [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>>.
- [RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002, <<https://www.rfc-editor.org/rfc/rfc3339>>.
- [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>>.
- [SESSION-ID] Jones, P., Salgueiro, G., Pearce, C., and P. Giralt, "End-to-End Session Identification in IP-Based Multimedia Communication Networks", RFC 7989, DOI 10.17487/RFC7989, October 2016, <<https://www.rfc-editor.org/rfc/rfc7989>>.
- [SMTP] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, DOI 10.17487/RFC5321, October 2008, <<https://www.rfc-editor.org/rfc/rfc5321>>.

- [TEL] Schulzrinne, H., "The tel URI for Telephone Numbers", RFC 3966, DOI 10.17487/RFC3966, December 2004, <<https://www.rfc-editor.org/rfc/rfc3966>>.
- [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>>.

8.2. 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>>.
- [ISOBMFF] "Information technology -- Coding of audio-visual objects -- Part 12: ISO base media file format", ISO/IEC 14496-12:2022, January 2022, <<https://www.iso.org/standard/83102.html>>.
- [JWT] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <<https://www.rfc-editor.org/rfc/rfc7519>>.
- [PY-VCON] "Python open source vCon command line interface, library and workflow server", n.d., <<https://github.com/py-vcon/py-vcon>>.
- [SHA-512] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, DOI 10.17487/RFC6234, May 2011, <<https://www.rfc-editor.org/rfc/rfc6234>>.

- [SIP] 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>>.
- [SIP-XFER] Sparks, R., Johnston, A., Ed., and D. Petrie, "Session Initiation Protocol (SIP) Call Control - Transfer", BCP 149, RFC 5589, DOI 10.17487/RFC5589, June 2009, <<https://www.rfc-editor.org/rfc/rfc5589>>.
- [STIR-PASS] Wendt, C. and J. Peterson, "PASSport Extension for Rich Call Data", Work in Progress, Internet-Draft, draft-ietf-stir-passport-rcd-26, 5 June 2023, <<https://datatracker.ietf.org/doc/html/draft-ietf-stir-passport-rcd-26>>.
- [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>.

Appendix A. Example vCons

This appendix contains example vCons in the unsigned, signed and encrypted form. The JSON has been indented for readability. The vCons have been formatted such that long text lines have been wrapped with a left hand white space indent for readability purposes. Links for unformatted versions of the vCons are provided for each of the examples. The example vCons in this document were generated using the [PY-VCON] command line interface.

A.1. Two Party Call vCon With Inline Recording

This example vCon is for a simple 2 party PSTN call. It has a single Dialog Object which contains a single channel wav format recording with the two parties audio mixed into the single channel.

The unformatted version of the following example can be downloaded from: https://raw.githubusercontent.com/ietf-wg-vcon/draft-ietf-vcon-vcon-core/refs/heads/main/examples/ab_call_int_rec.vcon

```
{
  "vcon": "0.0.2",
  "parties": [
    {
      "tel": "+12345678901",
      "name": "Alice"
    },
    {
      "tel": "+19876543210",
      "name": "Bob"
    }
  ],
  "dialog": [
    {
      "type": "recording",
      "start": "2022-06-21T17:53:26.000+00:00",
      "duration": 4.72,
      "parties": [
        0,
        1
      ],
      "mediatype": "audio/x-wav",
      "filename": "ab_call.wav",
      "encoding": "base64url",
      "body": "UklGRioeAABXQVZFZm10IBQAAAAxAAEAQB8AAfKGAABBAAAAAG
        BAAWZhY3QEAAAAgJMAAGRhdGH2HQAaoxjXHYXCRekuye4gK4Tsiv4wZeN
        htuvu3RMkgyPXme4NeahQLG0yeLayx5HCIiRWK7psbQrQTFm57NwQJvIb
        qlaNxk6VpMYAXSm5BblrgStIStvEzmCaOIabxs2gjReJ02Rk0BQtOxHkh
        UTIZd4VGKoqlKuoCGhrZTxlvRZICuZEcGkGjdH2jiCU1a0NMW6BE81VHN
        -OYFPHcZspr0CSl1aeo2hOWUx3CbovpGqKsApWteunk8gMPModynRVFWR
        5o012BvaQDcWugGRHrQa9NcEbG1LWQq_AWLpsXKdMQefGrZu4ZE2ZSDIR
        GJLTkPIzDybqSyZishQmjuVGztMKNlcyt2vW546JsutgllpyppQWroAtHb
        kyrrGAx3bIiX81glSR-oVaBbtKobA08fpJYscQSLHHCgm7hDriuWTRm8A
        6GKQlFbcnl0M30DcHctnEchwaAI7dxI2knYCQ5biNHB2Djxm3bFmyU5TA
        3ALZt3DaSeABiKiRjcm0AxpEkSZI7AEiSJEmSwUYZlmQLYNu2bdumPWDb
        tm3b9gOAJEmSJfKfGCRJkiQZbHRhTccDtm3btm0qAEaOJEmSPABiKiRjK
        jAASJIKSZLBRhfw9CKAJEmS49gCgCRJkiQ5P2Dbtm3bZgSAJEmSJClkcm
        VNLgBIKiRjkm8ASJIKSZIvAkiSJEmOswM2btu2bcMVmd3kIoDkyJEkySO
        AJEmS3AgGgCRJkhyPC2DjuG3j9mxOWQiyEUQ1DAtyvQJS9aTWWc4KuFqL
        O9F4CCJX4yefKMaSUWSLYMkYinzGa4HzyrEi6-pgHVdXe6mugKzW7VqHY
        OzcDDsXWnJkeVXHFKpuLDZSrAroZOQohtASpE4dyZSRxozJQC6hJE2OLG
        iNYLMlruIWzsCcRhEW6c3AJhq6bmp9cZ0IvAy2a5xJVcUMNE8Qq3G0CEJ
        y30SNxwrUqUy5NczWkI3S7UCk4iyrBM3AzCvJqkisQPQkcekWjYDoJk4D
        pWiPmUzXCtKFMkuylg7MtWnrqLwMSqpjR8vXDipSKzk1D-bOjdLzgNQol
        lapqqCUMjChmctA6jRmYPOLYJtHbatocTAZCfAQ3LHs6O2zDuCC_LyStA
        gklLFHZesUspXsSr0S5lCRYg-hXc2tKqMsgeqYkNzSVmLc9mk08zbhzUK
        O3fhkzJQsvBTCcNk2s-oUxO5zpatkETyFn-xtrxRaTR0hZk3WDo5ELiFW
        Oq7hmMxgKjdcY0SO4Fk7ciJHC6EYNXEZI6oDUQz2EeJMMhlorxSOup04u
```

UUjOtyjwVQ3HdhpuSRyJv0WzprTo-so7pMpdgKWUk0jleWjiqBJAawcuw5
lca39VznGdMLAg6o-EEenbKEQxgkeS0xxEteISQcQUNPm8KxaV1xKKIq-
BFepsm_bjIFPLiJwpRoEoSZEuqWuBJLuuloxZC8ksQB0Wkt_IqrwaSMq8
p4mzEPrFksdx9i9KZVv36AvGkMmyKsHI864U2eThk-iNKRsvgRsQl1vMT
UHbPK2ZqVvNHW3oEIay3EaN7xJGgV4mmK8Q1k004jHPCFpRzHYxBtbOEZ
FMQeMoOVQ2F8GTOXvh9-qARmvOaH9MYaOpdTIFaS-ZJO4OGmZOvo3fEig
5M8lQvwxinFXKhc4QrPGM0pIJthKKkixBYjvlZJlNwXTljSW8VWFex6ts
hg7BUiVuU8VoDtlMwRBekqXknPYWWNFMRtW_FMhsVIPV4RDESZlqtFXHV
I3GSmHZyNFMV24h48qshAdLYeumkSKhTYG1Qp0LpWhrlUitEmaJlDxuXw
gGEe62VbkIUkKbdkv2CFhIZMdnkebSjRKzoCG7Uc32ywDkqkauEsaAzOq
K1vnNoKFotloMaa5YBWwbZI_aPtKzFMpyqx7WNBXEqt43TasSiaAKW8U
2NSFXmuA6htzZ8ougZAbtgLVamHcwpHZgq6B40DSEtuviaVpKyFIUpxog
TQvwrBFay5MSrRE6GFMFGqYp5L26r4Eo2kooM941UsSSRC6NVuXVfmol
lpueTN7sLdPY-uiHbuECmzJFAvywh-KivWrRRtmK4u4p1Va666LNhrFMT
F3tZKlTDvgdOfIqN-zUGbWLWajishIrY5DTNLYovSDJJ2co6MCCwrxxmd
6u0wU-CtiptRtUQohlesWT1X0I4kR-NrCEmIIFRGelq51TpDpN2jnuXFr
UWfaop44QqkpsixOkeSsFwpyiIoHV4IN-NatFJboRbtIqoILbeS0SJeli
2_t18Xi4xizaGMTbvG3w4C6biz8LrTot8bmQK5TWIKQWqUWloIyQhHJaA
MTQaq4SQWNzTeTaoc5MHEkWrWHEY8moxcmLeQkQStoV9lsim7S6HlKE6n
LKfAJK3GVq0WQeK8fmKNcascCXELWJjd9iwwNSi9mNvGsRwsvvxyHsw00
qqptnxsWafZoJyKIYkTLXobrtrgcAoryw9tNE7TVstKaCBBmEJ2LMUsXJ
KCCBklkWpsul1ZnZRgWoSnleyJ9njK9VrZXCrYFoJg6ofqqAam-jOFbEv
QIIcLJF9f-quNDsQis9ac_ljMEE300jQXKrmVeCxG7ie3UmIcL04g0zVP
FXAN5biakMaMydRuQRlJ2NuaTOEqKabgJg8hc8KNG6PLgDxn1fDGZAZRL
OsMhnEGF6--CEZJU6ly4hJmqiXpha4Wumlis26U5hAVtMxgmpVUIzPOoF
gpsVVLLOHiSOrKysoBq0Zx4xZZTVFJOwqozvXM7WEQtPmkq7GqF-ZKnqo
2YBGysVvsSJpM0i1kDMHODHEkN09hm8NtJWnmga1HDhVoL2ENW6odeXkN
2SjJCMjxotduUws-13OyDu8KtGUMMc7ICrJu6XpMSsdQkSTuWEC3ddtk1
sBUQyktnMqA8Ti6G-crYsSVVlmFXE5VLeQOtC5kuFCrCEaqTeuJWQu662
W5sr4KQGcrRVjLflHVdK1AV8Wt23ivgJmLsuU8TceJg_ZdREvBliaHKH
czlXJmXPOahlFhvcK8lOaKlnSFjSPk8iRxghUySulw230olezwCfOKfz
WqlgbFMH82iPYAk3rlNhZICdTIYblWwvYSnWDlin3TRvLwcmQhMcPLYJ2
G0NuYpJD0KtVEqpSuaUlflJLgdp0aRynlkAzM6_jW-6Ab1vxovqtWJTZzm
Z3YM0gTVgRVm-zNm6lDvaQaqVr1A4IUsQ8LlKmjPjXarHRIxlLSNK3Akqh
uGv2uQA6pcuSkrmCpl1ETvY4g4sgt3IZgbhUtSQ3orOouknUJ1taUPXRI
Dbq5EVeuvgzEsQQ5jgvmTk10zaCg2G2Tm9ZazCJO3ZSLwN3M9XWerUD9y
motaWDvHem_DMyQrMou9w7OsB05t1MLNLZrvBlRCyiKjkSNicYUzuazIJ
WzclQ3ziAtZVVs5a6Azhg6GYPKYGHFPaTKYO5ULdIIuIVksabFDJrRRdw
9tAxEiUvZpcIOmrAj69TTtpaJsMpgneyUu9ZtweRYk6l2zKDS5JJsq0xh
JBfulMtcL5kscBlAcSNHUsIK9qirs87yELpvrLnOzBCklpo5lYvG1JFUj
IBTKTKkcU2BGhePlKqrgF1TrSljrYDcpm5KK2RSYUlnAERu3LaNXQDIce
RIjMASHIjsZIOAkiS5EiSgUcXlgQDgCRJkiR5B2Akx5EkqQOAJEmSJk
FgOQ4kiQZcJQhMbwENm7bxnHnBLZx3MarbwBIkiTJcTcAOHIkSZLBNhvW
xiSAJEmS5DgFYBxJkiRpB4AkSZIkeTxg27Zt28ZoLB1FqA84qpop2DMPx
B3yyHWxCEq-fmsSlw4YUjo9rZTHlFH0q2AdUZiptdIgy0RuGddqgdyotU
Kb6kDbE4ukV4FunQyxDEpOHKgiUSHekezIka8Kys0TR3PlCFqPHrR10te
OkYsNQKolepQdzQALJTqbhIqAomiNxIiNQFTXZWWneXHhSPESVm4lyc10
EahpnMKFSBQojions9MISiQ9u9dX59iNgu2gpb5mlatoGmpjpyqjmBt5

mp0Za5gpVZzpUR17lgp0gqgLVIIIdvMIpMbktErBDJqS65hy7ApUjiojsZ
jXVtnitEDxJq5aG66ApGvxHWMugByp0ds4LmGdNmmEymntVCmzCtSY5M1
qUBO8kizbWc8NqLsw9nW0Gs7tY1vtFaiWTVDSOGVNdffvy7EFD0zJMJMuh
I8etAzstQSK9TqvpdEzdBMUeNm5USi3YDuxszZiSWggyWeoovtoMyHWWq
2ZV9lRVoMpggpZpQ1zgMgaS7rZxLzEDtpK2trKygm6OGJjhtTtkkxQyorh
vXclcNUm6s1VLtDlhq1joy3Qo2RpWU-ZK3zs1CNSH1RJqjWGsho6Zx2oS
sgGJIcZs6ziDMOLLqmo20XU34CDxWFA1OzAwIclQXst8MptWj3tLfCOz1
7dx5VMYMKhLXgCU1Tg1GjMabNW7bZkNg47Zt2-bugBtDMS63bTCdKbESG
k7cNHK5EMBNJbmW4hLKcq0pfkQL41CtJpZcZ51RYExhk0XS7spt4R0nLe
TaHEJbJykFmkoBhLVwfG11DRkpRBIW0aM6dXgKONwlc6IpD-5YTcW9ySC
6cTwpkhfWUFbUi2AfdLNotAyhFTuqiuqzQF43RreH1EHc2G2famXv3GQz
EbaFWyWKvA54ay1HF-0WaJMjq906C-JtlpSGDtaOkSiTYVtKktMmrGCT1
qEbaSxhlDZOJObUoCBLiW2dhTBZSboKOn4bOVnLCsim9Vyu6wioklaqLM
wMRpELrXJT15RJUHwB4zRiZbPWgFzGEW1m7GCjCpoi6w7ArGdeWR1xUdV
E4AhQkiPNbtYLqG2e6JS1Ckrf_tqZwgo4U6xWhpT3lokWzIBrsy3tViSg
WTSNDUusYBTn6eN4rWCcrBCVxGDNVEXBCm6R5sZ2KBO2buNHk98WYK7DX
HwWcsQ1pNVVY-YUSnKsYVypCiTXrKdsqC4TR61A2aJ02vjOIJ2ujquIZO
-cSL4MSpGOSs3HCiZxVDdx8hbGiZS5qekUStXiHI5Plo5N0sqA6vpMc_u
tIDWYcuJI7GB2uf5jpa2A1k7O2hJ1S50MvwjebFu4sckMmFXwNDJdC2gP
GbeqYB24bSA5bg7XWI0ky2DTiDPkIu6A4WapZRtoJu8rhJy4CktZkdm
2QuWUj0CnjWslanShG4kiTXnKkQFnggS42-DhS-8spVTOYSjjIvgZtOSi
QV7IC9NHYYg8yAy-Sq4mdyYeU4rsSaXE-dSdsI1MmiVVS8ErZp3kBO5wj
EUZ4litUKhlyLm5QIFtPFgkxBVCWnInXNoIVYldbH66DrRk768I6gWceU
HbhsbZ1MuBLKqaPWZtUSumrm0pKsEkSW8bZx9gzKVdvYjBkIn41UzoBfK
3LyJsuA6kZz3MUuQeU0DhvnbyEcW5bdqWANOuQrDbibYg2RxwykcQtFxu
oKIuEcw871DDhunD4tzhWRiZKuwORirt6cHWGatkIUN7RA5GRSnbYtgA2
mUh5lce5QSPA0Vily5IjtErKQGbd1vwXm0p1FrctMZJIdtTLs1s6NgKxA
tkKwi3lMgWQ3umtJ7EAdaSMlRayAXbdarDp5LxEpxAgMd4zJkKwMhp4iu
TKzDKaQSkOh3w7Gkde4udP2zowSzGDUzMPuSMvAJU1VLASPgOSocix9s4
DzYfbqOmlN2Si5CDBPTUeN3AyisASTj_gKQrGctqdxE0QupVatDbemTDK
KQCOVhgu562BiGc8kqZSgbmtJFpWswNs2N6N8cTEdLcAIqLKKN2nHDKay
FL-VSAtmrgY5caoKppCbXnRPxkzJ4IxAzLamG0dPweK2jp0WzCAUF4lmu
GuB5PqpZGWxi-Tdu83IkRu58bvpIpTjyo49xVau11BOPq_Mbl1I30pTKhu
LbE-qCpVr1bMoG5MWofRNa09FTtUULKkcXK1se60u4PnCSpYeriVaxor
WtStbT3ZK-p8kPe30UiL2Y0mSV6vGnamuo-xkacV8jGIsCdadKA_ipsBK
XQamggPJeduCoXCctcQSaILpxDhsIvSyalmXrwzUEtw0g9UKyjCDNMmYF
xWalG6BBUWGlK_WoJooLVmiBGHikq3TOO9AEUt8yriBEt1f5gjCrW17l1
gN6E7bRoi2FDau7PaxcRMsvt5Wr9go19VEzaBf2C4UUY5gmj1lSF2MQJU
7ZWVKrUDpFujiln1R4UivHsaQmzSNxgpGtiM9t9kSpn4bzbHvCnaKZmWq
2AfTjebrYBCpdao7zaCoRVZr8-pgwzgwjVDtAJ0mjdKkiRbRbsKSrXlN
KtbD-aKFDR0cwlEhiq5jtAIKHooV3bV9lCVdKsge8N14oiuIJm0ccukra
Ajr32iF89grKzNdamJcSEltQ7aTZpjxc0KIillx5XHCHysPSmv5gx4W20
7Nxf4EFLybMEchVUCy8lg-kryZoEsorZLT5Iz2DuuJGTo4TWsS7EEqZq
ZPWVzwm8cSfnTlMNZnTdrBWqLFRO0FhyCwrNsgWL4WWmlaQnfAKyp7a_o
iJiLDOWg3KsAU5ZdmQWsYhsm3Ua6JUBian3EvyU9yIn9BBIffVv4gU81SF
F9pZFK-RRuGcuho3tJyUIEIdS4jUsJZ2ElNbAWKI9ApOJ10yWFL91xwha
ia6MmSbQUVIm9V4avEExl5up0g7a8M1AJlYnF9pUb4H7NrZay2vBJrc2
k2XXIB0rbxGVbEHSTPLkdHVP3SiyFkrSfMuqxxSoUywtl7IKlnkqWR3qF

NigHFmJWfeOTZKu4OsYU5RWNWLaxG3QKsxAH0Xr5MYsgRRlrRxdeTFZKP
QOWKXeYsHHFFSVoiuy4xLmktxOiccSTLEdt06UeF2KVMlgaJRx2oivgNS
oT9Uo1WEamTIm9CohXLeNQMbUdBiexp2WflvG1TgayFeymZpJEhaXJbvJ
bhY-6rEiNolq35r2NWEeGLEiFZUh7qBmzS6rAtsoEfP2qgIqN07jV62KJ
L5mJ8xvQMXyuSqOmKTZmr0gILnRjZnWLNy3E6apVcqEmXur4e1Ezc0Mhk
FCmGVjYezBbFiQqsRKOZqj4hX5pCtSP8sQhnUaqWlRJSqSNgF3q6VIkts
6gbI6OvI3h4nQmlz1ozQkau0juIS0Qiw7auE3tEEvYXRszuaB2jJr6uOY
LyWaxR4Y1iR7T2sS6u4Z3ULmGNjN21t07hS4nJs8rxUoT9HUI6Ds6-5mx
ylBJLdNZmZPgC7IbpV61UCqx7FkV4FwUSi1CpI5kpRt8Agwbx11iMIYg
vVuNbBCKTsXsSQmQeVyYQ0IV2XedKKrSAvabkyS-5gMV0WNe3twGuVgdy
kiZNZTb8MQkvFX_GvDiqZ7Jqx6Qy4kc0YieoUxlCMptKYB1WRQO_AZunS
7XqsYBZOkdyUruCy6A2gRc6A21apH0MB9Slu2wbkbbbohLvQWuEnhBE2qF
qZQ1XpJsBRARZ4qzkyZ5SIjVEHJMknKsq1BI7RtG_eKABbHbKrc7kCkWI
dGQ4GyoUnzCjItYBipvAq8yFbt7WcMKI5j6s3iDkRSmwxGV-cOlub0oNv
Y0poFzYBrV5kx60thYpml7IgtARLGrKsJfdFYkcIUJrrlmHrzCMLzKzV9
xgy6Vh1Hpe0SuDYd125Nm9hzUw-hHj1S3JiNQJ1Ijdu2fWbj2CcZ14pk8
msp48awbTU3Vhr-TrTvKFicJPoJf333KkjyGUaSZTV6aQ3L1U5roOuWHE
eaE3BkySiiTaBS7LYsgZnfdoSZBqHdpK1LJ6nQXJpwFchnZ-vv8xp4cqV
Lqa0cGpWl9XayEpCRW4nNE_pKXYuXQZRLUfXLKmFox06kAmxhaUULY2rP
QBndZc5KgVBZVccIYtqqtA31CDpUHcdw6hTWrcLilSsKUoKsXoqS1xKRs
q3AEDV1YXdNQVsLcavky2BUuV3TLK2A5BI-god9seFE9A6yU9tW2-oKUq
Z0RZLgCjw5p0125BC4kZnYadDW1I1grKAlt9LbprSAbMalGvUuQSAhbhP
JagGLRHEktWixXCXhDOQRDBlSVh0mU7LYCU0RTgM_Q7Y6EtCuUUuyS_a0
ldRNYTI30c12jsDe9orzdy2Bs1qyY11X4TtVsuuqWM4YTasUyMbarHYtE
byJ3Bo12BWYivAUrq0sG1IcSajNGBOaV83iXSaO2orMgR1LTvmbqIkV1
bqhycBO6edpup4cqVNSBKorPfkK-kQVj4uOb2vFDR71CiSzRK2yQu9bRk
2U5GA74Bmmc-a3Duh2wt71OqKId1YaORWTKF70pIrx41S3SjKDFAtog5N
RBOoUKRmaXYWRJGeyNbGFNphZDhQSOeQyWBtYUOViaKmq4Ds5tTb0i1hV
D-ObJWqoF3EMeuQZM0kSbIUxka8to7uDEI22iqLLA1GVqU1kdkKUM7juL
HMxtBJwoygoxXHHceqgNI0coPGksAkukX7kCphGrTuXkWBENVizxI-MSU
nUrAIukf691HJDDYve0teUQ1Y1V9IUo3mzkVUz6AUtQccy8pgLSQ5BXnL
IBYzmirr7WBhyUVr9WjuHE0rCwhyWueNtw5IbfpWqrgUNnoiaXLDCiy50
lvsjtdQkWKLAkNekZZirKCnaDVvVu3gHN1VYX3MQLIoSq6KgTCdRPiO0I
2jusa2CmbG5uautQ5Ipl3WT64IOLVbW9PQKNcZZ8-A7Dp3pswbQdQ6PYG
NK6HaxUDiVs2AZRnOzRaRUJ1Z6RRI0iThYfYKRI4cOjrxEqgqospuKyO4
VZK_1lMokw2JQ2EolrXkqIzAI9wynWjn4FHvUqacrIC0RnZac4WPmUzNE
CZOpSggWw2qbCREedIMNFu0NS7rDCQRm5iOledUzULLgCLLkIOJzyDaI6
6hO--g3DQi01zLYCJhtJ2biVhHKPYMTkKxqW7hDEp1K9Uhlg6aTE04qTY
AYJ3fXApcCBWS5MWA30SpIoePwJXFTVo2rGAVRlHst8tgbmeSpDZxUJUk
sA6kcd41kuUIso2UtZG9ClrW79RZ6QpmbuLiclb31E2k7CAqw45ki60A4
6rW27Yq4D2fkhOizUDkok4SxYkyWSmyCFauWgc2xgT6dKJu7dAIqpENKQ
bnCDROpEfX1vfUiXK1YHY3fvRXzGD8yJUHCY0A3TJtU4XvgNpMip1KbQ-
VLPQMwoxUJ1rIDDQuKyN5qg5Ek_28smIPSHotx9XY91JF5MxAqi4uJffq
QLS4qSZr12BVG31b7q1gmRdtnJiNcF0kyAqGNkw5KuoOyG3XXJbpCMxxZ
zuZ9A7IjBzXzRQHk4Xi66Ciak0sa9Zak7uQJmtIBrDQhIZDCCisukrYW
2QUWj1CrhUGrmp9grYb-oYUuWMOglkzJLUCMht7lzQHpubADSzYJ1ftv4
KrIGVNW4YtUphGyv65IpeYkaXigzfwRxbJK4eULosuofVTeKhI0V7rVx6
syVN58RFipMWLcXb7Hwa0ool1tVSbRerw1U13_R970Yst2rTUDSHiqJEk

```

    dDB3u8tLXXEZRi3iLuKOJJsq89gTZ4TTUs9wXJSZIUvXR-7sZUYe8uDt1
    KkJhPJMc6uUfVsadMGMupeZmqFtXHdstEeH4X3qL6u60iO3kZOOfwyo_K
    eppGj8RXzk2zV00dWW80n0QTMxLbTfSB8jmdFp02aKlNlK1l97XheSYTX
    kOLGwR4faduDnqHCOE2_ZLbx6rq35oJe0p26lepuXOWTt7Ie_DmLni4ID
    SZom4YzSEtD47tmVUra4GERBpUJACE0Uua5njuKvpK2ddy599W22r4k_U
    2rbBguUz2dY6tWrvTZuZr9sdZE7-olvZu1lUrIGj9QIfauaCtaeXRZr4k
    fW5Hlxb0fr3U2lWqt6sOSaYeyZJEIdkWdSu89fZvqdnrVNRmum_xp2jTn
    h2ISAYJiKwN0oNFtnSyD5Eiyw3dPBAY5diql0b9ecXVDBo2Rs63nWNr6t
    fuyXTmu8nIabShdNojSNGobq52IxMKBgybmG60GieX0pUNWh0AA9KxbRY
    yBJnluZH2Jc4hI2iigbuXGkTgINqqeKLVUDKTR2dQt6gpWMhzXTRd4B2B
    ULCDiSHXDgy8AnTuMqiROgCsJsYs4tGAjtlWcVdE2SATnCLSKbNhtyAJQ
    ciPHaL0Eyo0bWcX0DiZu29hwFntXxIIy4KuVckrtDCD4tGYSHctAFDmNz
    VbPQCvDauJGqTgafrCINDGcNGnZR7Zt3PoALxSCUqY0wbyrSHmTIZc"
  }
],
"analysis": [],
"attachments": [],
"uuid": "0199decd-7187-8ef1-b9a2-279e0d16bc46"
}

```

A.2. Email Thread Multipart vCon

The following is an unsigned form of an vCon for a 2 message email thread between 2 parties. The email messages are multipart MIME message bodies.

The unformatted version of the following example can be downloaded from: https://raw.githubusercontent.com/ietf-wg-vcon/draft-ietf-vcon-vcon-core/refs/heads/main/examples/ab_email_acct_prob_thread.vcon

```

{
  "vcon": "0.0.2",
  "group": [],
  "parties": [
    {
      "mailto": "b@example.com",
      "name": "Bob"
    },
    {
      "mailto": "a@example.com",
      "name": "Alice"
    }
  ],
  "dialog": [
    {
      "type": "text",
      "start": "2022-09-23T21:44:25.000+00:00",
      "duration": 0,

```

```
"parties": [
  0,
  1
],
"mediatype": "multipart/mixed; \n\tboundary=\"-----_Part_38
8314_1108189513.1663969465698\"",
"encoding": "none",
"body": "-----_Part_388314_1108189513.1663969465698\r\nCo
ntent-Type: multipart/alternative;\r\n\tboundary=\"-----_
Part_388313_805631480.1663969465697\"\r\n\r\n-----_Part
_388313_805631480.1663969465697\r\nContent-Type: text/pla
in; charset=UTF-8\r\nContent-Transfer-Encoding: 7bit\r\n\r
\r\nAlice: Please find the image attached.\r\n\r\nRegards, B
ob\r\n\r\n-----_Part_388313_805631480.1663969465697\r\n
Content-Type: text/html; charset=UTF-8\r\nContent-Transfe
r-Encoding: 7bit\r\n\r\n<html><head></head><body><div cla
ss=\"yahoo-style-wrap\" style=\"font-family:Helvetica Neu
e, Helvetica, Arial, sans-serif;font-size:16px;\"><div di
r=\"ltr\" data-setdir=\"false\"><div><div dir=\"ltr\" sty
le=\"font-family: &quot;Helvetica Neue&quot;; Helvetica,
Arial, sans-serif; font-size: 16px;\">Alice:</div><div di
r=\"ltr\" style=\"font-family: &quot;Helvetica Neue&quot;
, Helvetica, Arial, sans-serif; font-size: 16px;\">Please
find the image attached.<br></div><div dir=\"ltr\" style
=\"font-family: &quot;Helvetica Neue&quot;; Helvetica, Ar
ial, sans-serif; font-size: 16px;\"><br></div><div dir=\"
ltr\" style=\"font-family: &quot;Helvetica Neue&quot;; He
lvetica, Arial, sans-serif; font-size: 16px;\">Regards,</
div><div dir=\"ltr\" style=\"font-family: &quot;Helvetica
Neue&quot;; Helvetica, Arial, sans-serif; font-size: 16p
x;\">Bob</div></div><br></div></div></body></html>\r\n---
-----_Part_388313_805631480.1663969465697--\r\n\r\n-----
_Part_388314_1108189513.1663969465698\r\nContent-Type: im
age/png\r\nContent-Transfer-Encoding: base64\r\nContent-D
isposition: attachment; filename=\"=?UTF-8?B?c2NyZWVuc2hv
dC5wbmc=?\" \r\nContent-ID: <c4c28087-6fa1-395e-e50f-dcef
d3d6bf96@yahoo.com>\r\n\r\niVBORw0KGgoAAAANSUheUgAAACYAAA
AeCAIAAADPdvS5AAAAA3NCSVQICAjb4U/gAAAAGXRFWHRT\r\n\r\nb2Z0d2F
yZQBnbm9tZS1zY3JlZW5zaG907wO/PgAAALpJREFUSInl1t0NgzAMBOC7
qPvPlgW6Qjdo\r\n\r\nj4f0J4qAJI6NROvniE8+mwCvtzv6imTnyf1K/UclH
UFWjIvaIL3CHCABSPKKtJckWfY6zw+sJ5dq\r\n\r\nISdVIzmj2klzTZG2Rm
e7NKgOwY6qPrMcUs+2PqchL14P+oyz+fHx77K5Sv8xyxAyZ0uBaxn/Spc
U\r\n\r\n8NxcosAut1Y3NtjVX7UjZlmpgWTSFyvVQPKB16tSqW53bFnKtyyZ
vYT3rSsAWADWAlOjseYFfwAA\r\n\r\nAABJRU5ErkJggg==\r\n\r\n-----
-_Part_388314_1108189513.1663969465698--\r\n",
"message_id": "<1126698042.388315.1663969465698@mail.yahoo.
com>"
},
```

```

{
  "type": "text",
  "start": "2022-09-23T21:44:59.000+00:00",
  "duration": 0,
  "parties": [
    1,
    0
  ],
  "mediatype": "multipart/alternative; \n\tboundary=\"-----_Part_390234_811845464.1663969499935\"",
  "encoding": "none",
  "body": "-----_Part_390234_811845464.1663969499935\r\nContent-Type: text/plain; charset=UTF-8\r\nContent-Transfer-Encoding: 7bit\r\n\r\nHi Bob:Got it. Thanks.\r\n\r\nOn Friday, September 23, 2022, 9:44:40 PM UTC, Bob <b@example.com> wrote: \r\n\r\nAlice:Please find the image attached.\r\n\r\nRegards,Bob\r\n\r\n\r\n-----_Part_390234_811845464.1663969499935\r\nContent-Type: text/html; charset=UTF-8\r\nContent-Transfer-Encoding: 7bit\r\n\r\n<html><head></head><body><div class=\"ydp89e479a0yahoo-style-wrap\" style=\"font-family:Helvetica Neue, Helvetica, Arial, sans-serif;font-size:16px;\"><div></div>\r\n<div><div><div dir=\"ltr\" style=\"font-family: Helvetica Neue, Helvetica, Arial, sans-serif; font-size: 16px;\">Hi Bob:</div><div dir=\"ltr\" style=\"font-family: Helvetica Neue, Helvetica, Arial, sans-serif; font-size: 16px;\">Got it. Thanks.</div></div><br></div><div><br></div>\r\n\r\n</div><div id=\"ydpdc5150a3yahoo_quoted_4829810715\" class=\"ydpdc5150a3yahoo_quoted\">\r\n  <div style=\"font-family:'Helvetica Neue', Helvetica, Arial, sans-serif;font-size:13px;color:#26282a;\">\r\n    \r\n    <div>\r\n      On Friday, September 23, 2022, 9:44:40 PM UTC, Bob &lt;b@example.com> wrote:\r\n\r\n        </div>\r\n        <div><br></div>\r\n        <div><br></div>\r\n        <div><div id=\"ydpdc5150a3yiv4578005613\"><div><div style=\"font-family:Helvetica Neue, Helvetica, Arial, sans-serif;font-size:16px;\" class=\"ydpdc5150a3yiv4578005613yahoo-style-wrap\"><div dir=\"ltr\"><div><div dir=\"ltr\" style=\"font-family:Helvetica, Arial, sans-serif;font-size:16px;\">Alice:</div><div dir=\"ltr\" style=\"font-family:Helvetica, Arial, sans-serif;font-size:16px;\">Please find the image attached.<br></div><div dir=\"ltr\" style=\"font-family:Helvetica, Arial, sans-serif;font-size:16px;\"><br></div><div dir=\"ltr\" style=\"font-family:Helvetica, Arial, sans-serif;font-size:16px;\">Regards,</div><div dir=\"ltr\" style=\"font-family:Helvetica, Arial, sans-serif;font-size:16px;\">Bob<

```



```

        /div></div><br></div></div></div></div></div>\r\n
        </div>\r\n        </div></body></html>\r\n-----_Par
        t_390234_811845464.1663969499935--\r\n",
        "message_id": "<2031327240.390235.1663969499936@mail.yahoo.
        com>"
    }
],
"analysis": [],
"attachments": [],
"created_at": "2025-03-02T00:39:02.990+00:00",
"redacted": {},
"subject": "Account problem",
"uuid": "0195544a-cc55-8d85-b9a2-279e0d16bc46"
}

```

A.3. Email Thread Text vCon

The following is an unsigned form of an vCon for a 3 message email thread between 2 parties. The email messages are plain text message bodies.

The unformatted version of the following example can be downloaded from: https://raw.githubusercontent.com/ietf-wg-vcon/draft-ietf-vcon-vcon-core/refs/heads/main/examples/ab_email_prob_followup_text_thread.vcon

```

{
  "vcon": "0.0.2",
  "group": [],
  "parties": [
    {
      "mailto": "a@example.com",
      "name": "Alice"
    },
    {
      "mailto": "b@example.com",
      "name": "Bob"
    }
  ],
  "dialog": [
    {
      "type": "text",
      "start": "2022-09-23T23:24:59.000+00:00",
      "duration": 0,
      "parties": [
        0,
        1
      ],
    },
  ],
}

```

```
"mediatype": "text/plain",
"encoding": "none",
"body": "Hi Bob:\nI just wanted to follow up to see that yo
ur account is ok now.\n\nRegards,\nAlice\n\n",
"message_id": "<2031327240.390235.1663969499936@mail.yahoo.
com>"
},
{
  "type": "text",
  "start": "2022-09-23T23:34:32.000+00:00",
  "duration": 0,
  "parties": [
    1,
    0
  ],
  "mediatype": "text/plain; charset=UTF-8",
  "encoding": "none",
  "body": "Hi Alice:\nAll is fine. Thanks.\n\n    On Friday,
September 23, 2022, 23:24:59 PM UTC, Alice <a@example.com
> wrote: \n \n Hi Bob:\n I just wanted to follow up to
see that your account is ok now.\n \n Regards,\n Alic
e\n",
  "message_id": "<1126698042.388315.1663969465698@mail.yahoo.
com>"
},
{
  "type": "text",
  "start": "2022-09-23T23:38:12.000+00:00",
  "duration": 0,
  "parties": [
    0,
    1
  ],
  "mediatype": "text/plain",
  "encoding": "none",
  "body": "Awesome!\n\n    On Friday, September 23, 2022, 23:
34: PM UTC, Bob <b@example.com> wrote: \n\n Hi Alice:\n
All is fine. Thanks.\n \n    On Friday, September 23
, 2022, 23:24:59 PM UTC, Alice <a@example.com> wrote: \n
\n Hi Bob:\n I just wanted to follow up to see t
hat your account is ok now.\n \n Regards,\n Alic
e\n",
  "message_id": "<2031327240.390235.1663969499936@mail.yahoo.
com>"
}
],
"analysis": [],
"attachments": [],
```

```
"created_at": "2025-03-02T00:39:04.591+00:00",  
"redacted": {},  
"subject": "Account issue followup",  
"uuid": "0195544a-d292-8cda-b9a2-279e0d16bc46"  
}
```

A.4. Two Party Call vCon With Externally Referenced Recording

This example vCon is for a simple 2 party PSTN call. It has a single Dialog Object which reference a single channel wav format recording with the two parties audio mixed into the single channel.

The unformatted version of the following example can be downloaded from: https://raw.githubusercontent.com/ietf-wg-vcon/draft-ietf-vcon-vcon-core/refs/heads/main/examples/ab_call_ext_rec.vcon

```
{
  "vcon": "0.0.2",
  "parties": [
    {
      "tel": "+12345678901",
      "name": "Alice"
    },
    {
      "tel": "+19876543210",
      "name": "Bob"
    }
  ],
  "dialog": [
    {
      "type": "recording",
      "start": "2022-06-21T17:53:26.000+00:00",
      "duration": 33.12,
      "parties": [
        0,
        1
      ],
      "url": "https://github.com/ietf-wg-vcon/draft-ietf-vcon-vcon-container/raw/refs/heads/main/examples/ab_call.mp3",
      "mediatype": "audio/x-mp3",
      "filename": "ab_call.mp3",
      "content_hash": "sha512-GLy6IPaIUM1GqzZqfIPZlWjaDsNgNvZM0iC
        ONNThnH0a75fhUM6cYzLZ5GynSURREvZwmOh54-2lRRieyj82UQ"
    }
  ],
  "analysis": [],
  "attachments": [],
  "uuid": "0199e054-ed10-8035-b9a2-279e0d16bc46"
}
```

A.5. Two Party Call vCon with Analysis

TODO: fix diarization in transcript

The unformatted version of the following example can be downloaded from: https://raw.githubusercontent.com/ietf-wg-vcon/draft-ietf-vcon-vcon-core/refs/heads/main/examples/ab_call_ext_rec_analysis.vcon

```
{
  "vcon": "0.0.2",
  "parties": [
    {
      "tel": "+12345678901",
      "name": "Alice"
    }
  ]
}
```

```
    },
    {
      "tel": "+19876543210",
      "name": "Bob"
    }
  ],
  "dialog": [
    {
      "type": "recording",
      "start": "2022-06-21T17:53:26.000+00:00",
      "duration": 33.12,
      "parties": [
        0,
        1
      ],
      "url": "https://github.com/ietf-wg-vcon/draft-ietf-vcon-vcon-container/raw/refs/heads/main/examples/ab_call.mp3",
      "mediatype": "audio/x-mp3",
      "filename": "ab_call.mp3",
      "content_hash": "sha512-GLy6IPaIUM1GqzZqfIPZlWjaDsNgNvZM0iC
        ONNThnH0a75fhUM6cYzLZ5GynSURREvZwmOh54-2lRRieyj82UQ"
    }
  ],
  "analysis": [
    {
      "type": "transcript",
      "dialog": 0,
      "body": {
        "metadata": {
          "transaction_key": "deprecated",
          "request_id": "277eb8c5-86a9-4333-968e-243c77bc9a63",
          "sha256": "04dc074100461f5082f2a7a286d0161f0e2728025e83
            c20592aa7ea3724c2933",
          "created": "2025-10-14T01:32:14.393Z",
          "duration": 33.119938,
          "channels": 1,
          "models": [
            "2187e11a-3532-4498-b076-81fa530bdd49"
          ],
          "model_info": {
            "2187e11a-3532-4498-b076-81fa530bdd49": {
              "name": "general-nova-3",
              "version": "2025-07-31.0",
              "arch": "nova-3"
            }
          }
        }
      },
      "results": {
```

```
"channels": [
  {
    "alternatives": [
      {
        "transcript": "Hello. This is example.com. My name is Bob. How can I help you? Hi. I'd like to add Fu to my service. Okay. Can you give me your name and phone number so I can look up your account? My name is Alice, and my number is +1 234567890. Thank you. Okay. I've added two to your service. Is there anything else that I can help you with? No. Thank you. Goodbye. Goodbye. Have a nice day.",
        "confidence": 0.9996748,
        "words": [
          {
            "word": "hello",
            "start": 1.1999999,
            "end": 1.76,
            "confidence": 0.9944868,
            "speaker": 0,
            "speaker_confidence": 0.13938093,
            "punctuated_word": "Hello."
          },
          {
            "word": "this",
            "start": 1.76,
            "end": 2.24,
            "confidence": 0.9994715,
            "speaker": 0,
            "speaker_confidence": 0.13938093,
            "punctuated_word": "This"
          },
          {
            "word": "is",
            "start": 2.24,
            "end": 2.3999999,
            "confidence": 0.99721825,
            "speaker": 0,
            "speaker_confidence": 0.13938093,
            "punctuated_word": "is"
          },
          {
            "word": "example.com",
            "start": 2.3999999,
            "end": 3.4399998,
            "confidence": 0.906518,
            "speaker": 0,
```

```
"speaker_confidence": 0.13938093,
"punctuated_word": "example.com."
},
{
  "word": "my",
  "start": 3.4399998,
  "end": 3.6,
  "confidence": 0.9998653,
  "speaker": 0,
  "speaker_confidence": 0.50997114,
  "punctuated_word": "My"
},
{
  "word": "name",
  "start": 3.6,
  "end": 3.6799998,
  "confidence": 0.99983394,
  "speaker": 0,
  "speaker_confidence": 0.50997114,
  "punctuated_word": "name"
},
{
  "word": "is",
  "start": 3.6799998,
  "end": 3.84,
  "confidence": 0.9993299,
  "speaker": 0,
  "speaker_confidence": 0.50997114,
  "punctuated_word": "is"
},
{
  "word": "bob",
  "start": 3.84,
  "end": 4.3199997,
  "confidence": 0.99975526,
  "speaker": 0,
  "speaker_confidence": 0.50997114,
  "punctuated_word": "Bob."
},
{
  "word": "how",
  "start": 4.3199997,
  "end": 4.48,
  "confidence": 0.99988914,
  "speaker": 0,
  "speaker_confidence": 0.50997114,
  "punctuated_word": "How"
},
},
```

```
{
  "word": "can",
  "start": 4.48,
  "end": 4.72,
  "confidence": 0.99987936,
  "speaker": 0,
  "speaker_confidence": 0.50997114,
  "punctuated_word": "can"
},
{
  "word": "i",
  "start": 4.72,
  "end": 4.7999997,
  "confidence": 0.99978393,
  "speaker": 0,
  "speaker_confidence": 0.50997114,
  "punctuated_word": "I"
},
{
  "word": "help",
  "start": 4.7999997,
  "end": 5.04,
  "confidence": 0.9999826,
  "speaker": 0,
  "speaker_confidence": 0.50997114,
  "punctuated_word": "help"
},
{
  "word": "you",
  "start": 5.04,
  "end": 5.52,
  "confidence": 0.99984753,
  "speaker": 0,
  "speaker_confidence": 0.50997114,
  "punctuated_word": "you?"
},
{
  "word": "hi",
  "start": 5.6,
  "end": 6.3199997,
  "confidence": 0.9995427,
  "speaker": 0,
  "speaker_confidence": 0.57726574,
  "punctuated_word": "Hi."
},
{
  "word": "i'd",
  "start": 6.3199997,
```



```
"end": 6.72,  
"confidence": 0.995689,  
"speaker": 0,  
"speaker_confidence": 0.57726574,  
"punctuated_word": "I'd"  
},  
{  
  "word": "like",  
  "start": 6.72,  
  "end": 6.8799996,  
  "confidence": 0.99973685,  
  "speaker": 0,  
  "speaker_confidence": 0.57726574,  
  "punctuated_word": "like"  
},  
{  
  "word": "to",  
  "start": 6.8799996,  
  "end": 6.96,  
  "confidence": 0.9985936,  
  "speaker": 0,  
  "speaker_confidence": 0.57726574,  
  "punctuated_word": "to"  
},  
{  
  "word": "add",  
  "start": 6.96,  
  "end": 7.2799997,  
  "confidence": 0.98127615,  
  "speaker": 0,  
  "speaker_confidence": 0.57726574,  
  "punctuated_word": "add"  
},  
{  
  "word": "fu",  
  "start": 7.2799997,  
  "end": 7.68,  
  "confidence": 0.37009612,  
  "speaker": 0,  
  "speaker_confidence": 0.57726574,  
  "punctuated_word": "Fu"  
},  
{  
  "word": "to",  
  "start": 7.68,  
  "end": 8,  
  "confidence": 0.9966299,  
  "speaker": 0,
```

```
"speaker_confidence": 0.57726574,
"punctuated_word": "to"
},
{
  "word": "my",
  "start": 8,
  "end": 8.16,
  "confidence": 0.9992099,
  "speaker": 0,
  "speaker_confidence": 0.57726574,
  "punctuated_word": "my"
},
{
  "word": "service",
  "start": 8.16,
  "end": 8.799999,
  "confidence": 0.9892741,
  "speaker": 0,
  "speaker_confidence": 0.57726574,
  "punctuated_word": "service."
},
{
  "word": "okay",
  "start": 9.2,
  "end": 9.839999,
  "confidence": 0.9997375,
  "speaker": 0,
  "speaker_confidence": 0.57726574,
  "punctuated_word": "Okay."
},
{
  "word": "can",
  "start": 10.16,
  "end": 10.4,
  "confidence": 0.9997745,
  "speaker": 0,
  "speaker_confidence": 0.57726574,
  "punctuated_word": "Can"
},
{
  "word": "you",
  "start": 10.4,
  "end": 10.559999,
  "confidence": 0.9999038,
  "speaker": 0,
  "speaker_confidence": 0.57726574,
  "punctuated_word": "you"
},
},
```

```
{
  "word": "give",
  "start": 10.559999,
  "end": 10.719999,
  "confidence": 0.9992379,
  "speaker": 0,
  "speaker_confidence": 0.57726574,
  "punctuated_word": "give"
},
{
  "word": "me",
  "start": 10.719999,
  "end": 10.88,
  "confidence": 0.99977034,
  "speaker": 0,
  "speaker_confidence": 0.57726574,
  "punctuated_word": "me"
},
{
  "word": "your",
  "start": 10.88,
  "end": 11.04,
  "confidence": 0.99950397,
  "speaker": 0,
  "speaker_confidence": 0.2667445,
  "punctuated_word": "your"
},
{
  "word": "name",
  "start": 11.04,
  "end": 11.2,
  "confidence": 0.9999646,
  "speaker": 0,
  "speaker_confidence": 0.2667445,
  "punctuated_word": "name"
},
{
  "word": "and",
  "start": 11.2,
  "end": 11.44,
  "confidence": 0.9985328,
  "speaker": 0,
  "speaker_confidence": 0.2667445,
  "punctuated_word": "and"
},
{
  "word": "phone",
  "start": 11.44,
```

```
"end": 11.599999,
"confidence": 0.9995678,
"speaker": 0,
"speaker_confidence": 0.2667445,
"punctuated_word": "phone"
},
{
  "word": "number",
  "start": 11.599999,
  "end": 11.84,
  "confidence": 0.9994261,
  "speaker": 0,
  "speaker_confidence": 0.2667445,
  "punctuated_word": "number"
},
{
  "word": "so",
  "start": 11.84,
  "end": 12.08,
  "confidence": 0.89335036,
  "speaker": 0,
  "speaker_confidence": 0.2667445,
  "punctuated_word": "so"
},
{
  "word": "i",
  "start": 12.08,
  "end": 12.16,
  "confidence": 0.999236,
  "speaker": 0,
  "speaker_confidence": 0.2667445,
  "punctuated_word": "I"
},
{
  "word": "can",
  "start": 12.16,
  "end": 12.32,
  "confidence": 0.99919015,
  "speaker": 0,
  "speaker_confidence": 0.6548568,
  "punctuated_word": "can"
},
{
  "word": "look",
  "start": 12.32,
  "end": 12.48,
  "confidence": 0.9993882,
  "speaker": 0,
```

```
    "speaker_confidence": 0.6548568,
    "punctuated_word": "look"
  },
  {
    "word": "up",
    "start": 12.48,
    "end": 12.639999,
    "confidence": 0.99802244,
    "speaker": 0,
    "speaker_confidence": 0.6548568,
    "punctuated_word": "up"
  },
  {
    "word": "your",
    "start": 12.639999,
    "end": 12.799999,
    "confidence": 0.99812704,
    "speaker": 0,
    "speaker_confidence": 0.6548568,
    "punctuated_word": "your"
  },
  {
    "word": "account",
    "start": 12.799999,
    "end": 13.12,
    "confidence": 0.9978804,
    "speaker": 0,
    "speaker_confidence": 0.6548568,
    "punctuated_word": "account?"
  },
  {
    "word": "my",
    "start": 13.735,
    "end": 14.135,
    "confidence": 0.9996507,
    "speaker": 0,
    "speaker_confidence": 0.6548568,
    "punctuated_word": "My"
  },
  {
    "word": "name",
    "start": 14.135,
    "end": 14.295,
    "confidence": 0.9998702,
    "speaker": 0,
    "speaker_confidence": 0.6548568,
    "punctuated_word": "name"
  },
  },
```

```
{
  "word": "is",
  "start": 14.295,
  "end": 14.455,
  "confidence": 0.99962306,
  "speaker": 0,
  "speaker_confidence": 0.6548568,
  "punctuated_word": "is"
},
{
  "word": "alice",
  "start": 14.455,
  "end": 15.174999,
  "confidence": 0.93576753,
  "speaker": 0,
  "speaker_confidence": 0.6548568,
  "punctuated_word": "Alice,"
},
{
  "word": "and",
  "start": 15.174999,
  "end": 15.335,
  "confidence": 0.9998963,
  "speaker": 0,
  "speaker_confidence": 0.6548568,
  "punctuated_word": "and"
},
{
  "word": "my",
  "start": 15.335,
  "end": 15.575,
  "confidence": 0.9999398,
  "speaker": 0,
  "speaker_confidence": 0.6548568,
  "punctuated_word": "my"
},
{
  "word": "number",
  "start": 15.575,
  "end": 15.8949995,
  "confidence": 0.9999269,
  "speaker": 0,
  "speaker_confidence": 0.6548568,
  "punctuated_word": "number"
},
{
  "word": "is",
  "start": 15.8949995,
```

```
"end": 16.455,
"confidence": 0.9997197,
"speaker": 0,
"speaker_confidence": 0.6548568,
"punctuated_word": "is"
},
{
  "word": "+1",
  "start": 16.455,
  "end": 16.775,
  "confidence": 0.99967515,
  "speaker": 0,
  "speaker_confidence": 0.6548568,
  "punctuated_word": "+1"
},
{
  "word": "234567890",
  "start": 16.775,
  "end": 20.615,
  "confidence": 0.99982315,
  "speaker": 0,
  "speaker_confidence": 0.65485674,
  "punctuated_word": "234567890."
},
{
  "word": "thank",
  "start": 20.775,
  "end": 21.095,
  "confidence": 0.9998648,
  "speaker": 0,
  "speaker_confidence": 0.6548568,
  "punctuated_word": "Thank"
},
{
  "word": "you",
  "start": 21.095,
  "end": 21.654999,
  "confidence": 0.999934,
  "speaker": 0,
  "speaker_confidence": 0.6548568,
  "punctuated_word": "you."
},
{
  "word": "okay",
  "start": 22.935,
  "end": 23.335,
  "confidence": 0.9996748,
  "speaker": 0,
```

```
"speaker_confidence": 0.6548568,
"punctuated_word": "Okay."
},
{
  "word": "i've",
  "start": 24.134874,
  "end": 24.534874,
  "confidence": 0.995402,
  "speaker": 0,
  "speaker_confidence": 0.6548568,
  "punctuated_word": "I've"
},
{
  "word": "added",
  "start": 24.534874,
  "end": 24.854876,
  "confidence": 0.9885041,
  "speaker": 0,
  "speaker_confidence": 0.6548568,
  "punctuated_word": "added"
},
{
  "word": "two",
  "start": 24.854876,
  "end": 25.094875,
  "confidence": 0.80214345,
  "speaker": 0,
  "speaker_confidence": 0.6548568,
  "punctuated_word": "two"
},
{
  "word": "to",
  "start": 25.094875,
  "end": 25.254875,
  "confidence": 0.8400144,
  "speaker": 0,
  "speaker_confidence": 0.6548568,
  "punctuated_word": "to"
},
{
  "word": "your",
  "start": 25.254875,
  "end": 25.414875,
  "confidence": 0.9990602,
  "speaker": 0,
  "speaker_confidence": 0.6548568,
  "punctuated_word": "your"
},
},
```



```
{
  "word": "service",
  "start": 25.414875,
  "end": 25.734875,
  "confidence": 0.9951705,
  "speaker": 0,
  "speaker_confidence": 0.6548568,
  "punctuated_word": "service."
},
{
  "word": "is",
  "start": 25.734875,
  "end": 25.894875,
  "confidence": 0.9998394,
  "speaker": 0,
  "speaker_confidence": 0.4939385,
  "punctuated_word": "Is"
},
{
  "word": "there",
  "start": 25.894875,
  "end": 26.054874,
  "confidence": 0.9998031,
  "speaker": 0,
  "speaker_confidence": 0.4939385,
  "punctuated_word": "there"
},
{
  "word": "anything",
  "start": 26.054874,
  "end": 26.294874,
  "confidence": 0.9997297,
  "speaker": 0,
  "speaker_confidence": 0.4939385,
  "punctuated_word": "anything"
},
{
  "word": "else",
  "start": 26.294874,
  "end": 26.534874,
  "confidence": 0.999869,
  "speaker": 0,
  "speaker_confidence": 0.4939385,
  "punctuated_word": "else"
},
{
  "word": "that",
  "start": 26.534874,
```

```
"end": 26.694874,
"confidence": 0.6311746,
"speaker": 0,
"speaker_confidence": 0.4939385,
"punctuated_word": "that"
},
{
  "word": "i",
  "start": 26.694874,
  "end": 26.774876,
  "confidence": 0.9994104,
  "speaker": 0,
  "speaker_confidence": 0.4939385,
  "punctuated_word": "I"
},
{
  "word": "can",
  "start": 26.774876,
  "end": 26.934875,
  "confidence": 0.9998919,
  "speaker": 0,
  "speaker_confidence": 0.4939385,
  "punctuated_word": "can"
},
{
  "word": "help",
  "start": 26.934875,
  "end": 27.094875,
  "confidence": 0.9998467,
  "speaker": 0,
  "speaker_confidence": 0.4939385,
  "punctuated_word": "help"
},
{
  "word": "you",
  "start": 27.094875,
  "end": 27.174875,
  "confidence": 0.9998344,
  "speaker": 0,
  "speaker_confidence": 0.4939385,
  "punctuated_word": "you"
},
{
  "word": "with",
  "start": 27.174875,
  "end": 27.734875,
  "confidence": 0.9998601,
  "speaker": 0,
```

```
"speaker_confidence": 0.4939385,
"punctuated_word": "with?"
},
{
  "word": "no",
  "start": 28.054874,
  "end": 28.534874,
  "confidence": 0.99971354,
  "speaker": 0,
  "speaker_confidence": 0.4939385,
  "punctuated_word": "No."
},
{
  "word": "thank",
  "start": 28.534874,
  "end": 28.774876,
  "confidence": 0.99988055,
  "speaker": 0,
  "speaker_confidence": 0.5919341,
  "punctuated_word": "Thank"
},
{
  "word": "you",
  "start": 28.774876,
  "end": 29.334873,
  "confidence": 0.99990463,
  "speaker": 0,
  "speaker_confidence": 0.5919341,
  "punctuated_word": "you."
},
{
  "word": "goodbye",
  "start": 29.334875,
  "end": 29.974874,
  "confidence": 0.99884653,
  "speaker": 0,
  "speaker_confidence": 0.5919341,
  "punctuated_word": "Goodbye."
},
{
  "word": "goodbye",
  "start": 30.374874,
  "end": 31.014874,
  "confidence": 0.9991617,
  "speaker": 0,
  "speaker_confidence": 0.5919341,
  "punctuated_word": "Goodbye."
},
},
```

```
{
  "word": "have",
  "start": 31.014874,
  "end": 31.334875,
  "confidence": 0.9997665,
  "speaker": 0,
  "speaker_confidence": 0.5919341,
  "punctuated_word": "Have"
},
{
  "word": "a",
  "start": 31.334875,
  "end": 31.494875,
  "confidence": 0.9997255,
  "speaker": 0,
  "speaker_confidence": 0.5919341,
  "punctuated_word": "a"
},
{
  "word": "nice",
  "start": 31.494875,
  "end": 31.654875,
  "confidence": 0.99926084,
  "speaker": 0,
  "speaker_confidence": 0.5919341,
  "punctuated_word": "nice"
},
{
  "word": "day",
  "start": 31.654875,
  "end": 31.974874,
  "confidence": 0.9997565,
  "speaker": 0,
  "speaker_confidence": 0.007671833,
  "punctuated_word": "day."
}
],
"paragraphs": {
  "transcript": "\nSpeaker 0: Hello. This is example.com. My name is Bob. How can I help you? Hi.\n\nI'd like to add Fu to my service. Okay. Can you give me your name and phone number so I can look up your account? My name is Alice, and my number is +1 234567890. Thank you.\n\nOkay. I've added two to your service. Is there anything else that I can help you with? No. Thank you.\n\nGoodbye. Goodbye. Have a nice day.",
```

```
"paragraphs": [
  {
    "sentences": [
      {
        "text": "Hello.",
        "start": 1.1999999,
        "end": 1.76
      },
      {
        "text": "This is example.com.",
        "start": 1.76,
        "end": 3.4399998
      },
      {
        "text": "My name is Bob.",
        "start": 3.4399998,
        "end": 4.3199997
      },
      {
        "text": "How can I help you?",
        "start": 4.3199997,
        "end": 5.52
      },
      {
        "text": "Hi.",
        "start": 5.6,
        "end": 6.3199997
      }
    ],
    "speaker": 0,
    "num_words": 14,
    "start": 1.1999999,
    "end": 6.3199997
  },
  {
    "sentences": [
      {
        "text": "I'd like to add Fu to my ser
          vice.",
        "start": 6.3199997,
        "end": 8.799999
      },
      {
        "text": "Okay.",
        "start": 9.2,
        "end": 9.839999
      }
    ]
  }
]
```

```
      "text": "Can you give me your name and  
        phone number so I can look up your  
        account?",  
      "start": 10.16,  
      "end": 13.12  
    },  
    {  
      "text": "My name is Alice, and my number  
        is +1 234567890.",  
      "start": 13.735,  
      "end": 20.615  
    },  
    {  
      "text": "Thank you.",  
      "start": 20.775,  
      "end": 21.654999  
    }  
  ],  
  "speaker": 0,  
  "num_words": 37,  
  "start": 6.3199997,  
  "end": 21.654999  
},  
{  
  "sentences": [  
    {  
      "text": "Okay.",  
      "start": 22.935,  
      "end": 23.335  
    },  
    {  
      "text": "I've added two to your service.",  
      "start": 24.134874,  
      "end": 25.734875  
    },  
    {  
      "text": "Is there anything else that  
        I can help you with?",  
      "start": 25.734875,  
      "end": 27.734875  
    },  
    {  
      "text": "No.",  
      "start": 28.054874,  
      "end": 28.534874  
    }  
  ],  
}
```

```

        "text": "Thank you.",
        "start": 28.534874,
        "end": 29.334873
    }
],
    "speaker": 0,
    "num_words": 20,
    "start": 22.935,
    "end": 29.334873
},
{
    "sentences": [
        {
            "text": "Goodbye.",
            "start": 29.334875,
            "end": 29.974874
        },
        {
            "text": "Goodbye.",
            "start": 30.374874,
            "end": 31.014874
        },
        {
            "text": "Have a nice day.",
            "start": 31.014874,
            "end": 31.974874
        }
    ],
    "speaker": 0,
    "num_words": 6,
    "start": 29.334875,
    "end": 31.974874
}
]
},
    "encoding": "json",
    "vendor": "deepgram",
    "schema": "deepgram_prerecorded",
    "product": "transcription"
}
],
"attachments": [],

```

```
"uuid": "0199e054-ed10-8035-b9a2-279e0d16bc46"
}
```

A.6. Signed vCon

This example vCon is the signed form of the Two Party Call vCon With Externally Referenced Recording (Appendix A.4) example. The private key used to sign this can be found at:

https://raw.githubusercontent.com/vcon-dev/vcon/main/certs/fake_grp.key

The certificate chain is included in the `x5c` parameter of the header Object.

The unformatted version of the following example can be downloaded from: [https://raw.githubusercontent.com/ietf-wg-vcon/draft-ietf-vcon-vcon-core/refs/heads/main/examples/ab call ext rec signed.vcon](https://raw.githubusercontent.com/ietf-wg-vcon/draft-ietf-vcon-vcon-core/refs/heads/main/examples/ab%20call%20ext%20rec%20signed.vcon)

[illegible]

75BGsl4rB9txFQeef3Q5SJIzeqffJDssXDSyREuwQ2iTcOrZaZxs8
h1NX9NjVD6MMAwx7tw0Cgmu8rj6+/Kjs0KI8OzCuDXZGTGf/bH6EQ
W85qYh9IF0Pvi jf3uq+HCke6WApusx76LpUp36IzvYEuTbWBFeoLo
+dyUdWmXpj0ghHiWZtT5zxJn9VrUhJX5KlaI0palWd0fOnr7mkdq7
9sFsKjE0cr1llanceAicCpD3/zRW+7AS8atGKAMfe2CcXEABucI7Dt
tLMXcDIJM6EOkyW7n4duelRXGuNgnjeQIDAQABMA0GCSqGSib3DQE
BCwUAA4IBAQAQLjhKomSx10qThhcO+aK0rzX6RjKh0eM0X7O7pPC8
lHxZKCDLhDvnUO9++17U6ba0bTnznKYKKG7HnDcB4dKBbH8XP2u3E
nDskyoMrKpYRSRKwwJa8+RwOS7JEG+OtXYCRbWZvPjAC/z6o6OZK+
QPTyK5Grw2Px2mTr6BAdBQv7XsYWBWRw0ZyvSGnKyK60jcPlYUnu6Q
t0fYXiYp6qF5l956fuT0d2UyDkM7S/Z7K+OC9+aIBNKH32B3nfXqZ
PuP8WuiTOxcMwnefPwjxaqndFUkyk2Y+i6nXU8oN/DGUOTfY4xbm3
s3q7Dq5QPtq5kljB9yYkvcOULCSAmWBZl+G",
"MIIEyTCCArECAQEwDQYJKoZIhvcNAQELBQAwgagxCzAJBgNVBAYTA1
VTMQswCQYDVQQIDAJNQTERMA8GA1UEBwwIRmFrZXRVd24xFTATBgN
VBAoMDEZha2VWY29uIEluYzESMBAGA1UECwwJQ29ycG9yYXRlMRQw
EgYDVQQDDAtnYmVldmNvbi5pbzEhMB8GCSqGSib3DQEJARYSYWRta
W5AZmFrZXZjb24ub3JnMRUwEwYDVR0RDAXmYWtldmNvbi5vcmcwHh
cNMjIwNTI1MjEzOTMxWhcNMzIwNTI1MjEzOTMxWjCBqzELMAKGA1U
EBhMCVVMxXzAJBgNVBAGMAklBMREwDwYDVQQHDAhGYWtldG93bJER
MA8GA1UECgwIRmFrZXZjb24xETAPBgNVBAsMCERpdmlzaW9uMRgwF
gYDVQQDDA9kaXYuZmFrZXZjb24uaW8xITAFBgkqhkiG9w0BCQEWEm
FkbWluQGZha2V2Y29uLm9yZzEZMBcGA1UdEQwQZG12LmZha2V2Y29
uLm9yZzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAM81
SKielS78VAOBdC3LflYNSL9phecmjQs/9P+iqDXSvm3xpoqsZGoLn
7w21aUAxl++F9Z80Tc70URN2qqUcVTYcx3SmgHuYgspEmJsmRor/m
n3HpHr2ajh+26RKgDGPBrtebeHlsi8mX15a/ICzHxrUaE2Y/6DZ/2
wGTxRAE4bYAOuaYft5GGSZl6yP6dXg3EJy2bK2cUTdMIZG/Y7Ny1D
3N4V7MBirDd5FU7k6pMbyr3d/xoi8VL8F1mRCf6zxfFOjamFT0atE
iTp3M0rw6iS08hy0cDFsPJzIcPgXEitUgGPo/NLgeb8ZmL1kg9B2v
MgaanbuZB6il79A4GgDIUCAwEAATANBgkqhkiG9w0BAQsFAAOCAgE
AWOJlVlSQDsDROQZNdx3kMkAH3HYyAlE7LHntIYHVhyt7I7WuqS6Z
GvSoxSfkt0GdAN7cv+jchdZpAnpCXGX+yHuSvrw4oSqigdSLeSUMx
2zaSv4wLAvydC/zDkHIRObyuDEKyy/pa4Na0aunPU3E1Bg0V+JqQ3
RUJiYlv6AGLrliwaEigP3Clzi/u640dWBHABOEjxFj9bX73+FRWhi
t4EzOJYKvtileXlrmjMb3SmsrcbGMrpu4WipeG5S05VlMhXXSYqq
9IScFho5CftamDgI3jL49lyrx0+CcjDkw0nW9fxdAtB3BhSYEXdLu
JAvLRQixUQFwH23doupPiluW8euEhHTRolpEFWQ5U59NOTUA5Ghuu
po0aM2CnDD5Re8uO3ChilCUB0FlwgMD/7ABSd1LChU8iluhKy9CEB
R2zc8uTqW43Gcv/psBaejgOKJlxZuusGVvuVlnTHRa4hmFYxxPel7
ogeZzGQqfjBlb35CE+BcGYY0ViaHefy+fmi/fN6NrrN6pBOPjVZbp
CFn+wFTUyrMU5Vq/FaPe0us7i3paQcwIz9wyide5OztpxebY8UnXm
y22X+6PkHB0jnL35Qe8xl7Z0nMZNdy4xkzGgaa6talRDPHkY1lId
w+KdCXJuUc7rkLKa5po9SPdvVLhEPu26SHkI0I+d5hz+gMp8=",
"MIIGMzCCBBugAwIBAgIUHk/6QbzjDZ9AONAFvwYJqB/jbvowDQYJKo
ZIhvcNAQELBQAwgagxCzAJBgNVBAYTA1VTMQswCQYDVQQIDAJNQTE
RMA8GA1UEBwwIRmFrZXRVd24xFTATBgNVBAoMDEZha2VWY29uIElu
YzESMBAGA1UECwwJQ29ycG9yYXRlMRQwEgYDVQQDDAtnYmVldmNvb

```
i5pbzEhMB8GCSqGSib3DQeJARYSYWRtaW5AZmFrZXZjb24ub3JnMR
UwEwYDVR0RDAXmYWtldmNvbi5vcmcwHhcNMjIwNTI1MjEzOTMxWhc
NMzIwNTI1MjEzOTMxWjCBqDELMAkGA1UEBhMCVVMxCzAJBgNVBAGM
Ak1BMREwDwYDVQQHDAhGYWtldG93bjEVMBMGAlUECgwmRmFrZVZjb
24gSW5jMRIwEAYDVQQQLDA1Db3Jwb3JhdGUxZDASBgNVBAMMC2Zha2
V2Y29uLmlvMSEwHwYJKoZIhvcNAQkBFhJhZG1pbkBMYWtldmNvbi5
vcmcwFTATBgNVHREMDGZha2V2Y29uLm9yZzCCAiIwDQYJKoZIhvcN
AQEBBQADggIPADCCAgoCggIBAMIEaqkm8pn6mwUzWBgnQiY9Jr3Vy
VamxBc9/duT8eCN9mmANGzX5zGj86KtGjldCoAzANrtsr2Dpz4zra
29HWbwboAlZTOOAr93QgifikS802zp+zHGGRaKlq3euDqDVzyXinhs
vXOVlEwkOrAvV6kUcTIzcxVibrFfHttzv/hXUNxjCouAE7MIG6wO8
y+E2bBkg7R5ydfE4Q9npBefwtiQ8oWf0C46mDYFQ8l3lgoael7Vp
iyS3t4h5HL+lyqdYJSWms5Qh0vp9req5AjuwHIDbxNlp02gY1SQmS
d003oLcZwBglGKlpQSwYgr9J5uo2B+NV9aYKE051bySinSF8T4D/o
4cq6oyJLXBwmSsLbjjVIQBoSdRBmWh8zE2ZzFNaaBlyO9ikIaWXwb
tkbsE+4QpOaOj/kUIHLFI9iOhOidsInaOfgYzw12FtGE1Wq5x5Xsz
wWpPWU8NQw8gzTzFzV0bxrfcsXrs0s6feuTCDMt5Zht2HCWFr9TmY
DFAfOB05kFWTi5sE5KYaP54iJvJAFmuTDjliF7Z8ZWVal9XB+g3sM
yQ8DVEj/0dXS+nRVuJQlUH4q19SUIwK63RKdedALYC6X1j2lGK9aT
uL+cyNJ5SbuNA95CxNgvWtBrc06HxlbmIDxKrAAo5aj4hnlQcY4RR
bNwppCAjK+zcASlpyvRAGMBAAGjUzBRMB0GA1UdDgQWBBSDDtfqU6
+DEyYPCar08yVnnZSFzTAFBgNVHSMEGDAWgBSSDtfqU6+DEyYPCar
08yVnnZSFzTAPBgNVHRMBAf8EBTADAQH/MA0GCSqGSib3DQeBCwUA
A4ICAQC8JDxTdmc86YFdKElWQnWxoPMsPeRyDYQoVTC2wydZsdJ5d
GbkAhYlPPxK+zypP18fGry5PHAH19mZYz+FdgJNTwllcfN9/G0Sdo
y4kU9eN9KfC9OUIkBIq8D0r7ANT3l3XmGbnIxWJEtB0SyTcqk3dwW
cNTqGguhQ03jUKW5yUpVoQjN4TtmHyweoYTNi+HKsCUvEXGdA9wdh
4jVOqYeepmsax5eTUJskUCKQ3+xB7JONw85imfVotySCXsoMe7lu
SRYuldn8SBIpKYUGJlpWii6Gr0mLwHarrYoi9bwtcKAXRNXXZiGQgh
iUw2Zsw+jmgBrEtI4Iarg8wWqJ5Mov7MsTVpnoZ5Xywr0GznDlgcN
tyqMibOULEm8yLPQRhn/tQjSiP3eUPz6+Zb9FqzQJCCi2UiauD4mO
gzBSCCbew9PDYb2QeMZqNZmN3eVNMV4hh7t9jTc0zqPF5jluddVmp
dHxLx/T9XZuSV21p8clNprJNQxprrtHpxDxjhNMircV39rzi9tNikw
mzRMf0r40f/JWnaV668sekDXtvIj8pZwyE4FNuaiJwojkQxDfjqk6
7Qs5NUQsFcaIwMes8It74ASHm8DUKEhDT2ASgMtUDwKWROBefrSPq
jY7+EMh2yVZnQhMwpx4s7CH0za32Yq+KC8GotHdI6jisDEom7aHgl
g=="
],
"alg": "RS256",
"uuid": "0199e054-ed10-8035-b9a2-279e0d16bc46"
},
"signature": "qy0bjh14wxAddJl8inxxUDmNPd6IG6dD69ToSSdLVxtR
V6w09ciK33s9WmlQdKoTvDdtN_qpnK8H58ZbZgobWKA_OLyWkw3k-oqvY
JHoqBiPJEEaXdpA0c85TmS9mIQepCMSeo3geIfP00MaCtcMYMwlnVuw6a
uiTk32-NqW3prIoxQeVb2Ev9ZpDX8S8j-emyYpukq00z4qR9kTAlarltN
qvQRrMdl-OFv8mFKD4CZ4m07Oyet5WkVTOfs_Kqaazl2xMj2ZKie2VooL
nOL4YzaNav5pGC4_IDIjEVhfr5Xr8VADoChaIIeFwv8iHzPIitRlsaaCO
F6-4hkmN3e8w",
```

"protected": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsIng1YyI6W
yJNSU1EMFRDQ0Fya0NBUEV3RFFZSktvWklodmNOQVFFTEJRQXdnYXN4Q3
pBSk5JnTlZCQVlUQWxwVE1Rc3dDUVlEVlFRSURBSk5RVEVSTUE4R0ExVUV
Cd3dJUm1GclpYUnZkMjR4RVRBUEJnTlZCQW9NQ0VaaGEyVldZMj11TVJF
d0R3WURWUVMRFoRWFYWnBjMmx2YmpFWU1CWUdBMVVFQXd3UFpHbDJMb
VpoYTJWMLkyOXVMbWx2TVNFD0h3WUpLb1pJaHZjTkFRa0JGaEpoWkcxcG
JrQm1ZV3RsZG1OdmJpNXZjbWN4R1RBWEJnTlZlUkVNRUdScGRpNW1ZV3R
sZG1OdmJpNXZjbWN3SGhjTk1qSXdOakEwTURFME16STFXaGNOTXpJd05q
QXhNREUwTXpJMVdqQ0JzREVMTUFR0ExVUVCaE1DVlZNeEN6QUoPCZ05WQ
kFntUFRMUJNUKv3RHdZRFZRUUHEQWhHWVd0bGRHOTNiakVSTUE4R0ExVU
VDZ3dJUm1GclpWWmpimjR4RGpBTUJnTlZCQXNNQlVkeWlZVndNund3R2d
ZRFZRUUREQk5uY25BdVpHbDJMbVpoYTJWMLkyOXVMbWx2TVNFD0h3WUpL
b1pJaHZjTkFRa0JGaEpoWkcxcGJrQm1ZV3RsZG1OdmJpNXZjbWN4SFRBY
k5JnTlZlUkVNRkdkeWNDNWthWF11Wm1GclpYWmpimjR1YjNkYj1JSUJJa
kFOQmdrcWhraUc5dzBCQVFFRkFBT0NBUThtBTU1JQkNnS0NBUEVBde10VEN
4ZU1ST2IxRm1XSWV4ZTdPcDk1cVlJVk5FcW4rZkRnaHhXcW9ZbTZUNzVC
R3NsNHJC0XR4RlFlZWYzUTVTsk16ZXFmZkpEc3NYRHN5UkVld1EyaVRjT
3JaYVp4czhoMU5YOU5qVqQ2TU1Bd3g3dHcwQ2dtdThyaJYrL0tqcZBLST
hPekN1RFhaR1RHzi9iSDZFUVc4NXFZad1JRjBQdmlqZjN1cStIQ2t1Nld
BchVzeDc2THBVcDM2Sxp2WUV1VGJXQkZlb0xvK2R5VWRXbVhwaJbNaEhp
V1p0VDV6eEpuOVZyVWhKWDVLbGFJMBHbFdkMGZPbnI3bWtkcTc5c0ZzS
2pFT2NyMWxhbmNlQW1jQ3BEMy96U1crN0FTOGF0R0tBTWZlMkNjWEVBQn
VjSTdEdHRMTVhJRElKTTZFT2t5VzduNGR1ZTFSWEd1TmdumVRsSURBUUF
CTUEwR0NTcUdTSWlZrFFFQkN3VUFBNElCQVFBUEUxqaEtvbVN4MU9xVGho
Y08rYUswcnpYn1JqS2gwZU0wWDDPN3BQQzgxSHhaS0NETGhEdm5VTzkrK
ze3VTZiYTBiVG56bktZS0tHN0huRGNCNGRLQmJlOFhQMnUzRW5Ec2t5b0
lyS3BZUnNSS3d3Sme4K1J3T1M3SkVnK090WF1DUmJXWnZQakFDL3o2bzZ
PWksrUVBUeUs1R3J3M1B4Mm1UcJZCQWRCUXY3WHNZV0JSd09aeXZTR25L
eUs2MGpJUGxZVW51N1F0MGZZWG1ZcDZxRjVsOTU2ZnVUMGQyVX1Ea003U
y9aNoSrT0M5K2FJQk5LSDMyQjNuZlhxWlB1UDhXdWlUT3hJTXduZWZQd2
p4YXFuZEZVa3lrMlkrATZuWfU4b04vREDVT1RmWTR4Ym0zczNxn0RxnVNF
QdHE1a2xqQj15Wwt2Y09VTENTQW1XQlpsK0ciLCJNSU1FeVRDQ0FyRUNB
UUV3RFFZSktvWklodmNOQVFFTEJRQXdnYwd4Q3pBSk5JnTlZCQVlUQWxwV
E1Rc3dDUVlEVlFRSURBSk5RVEVSTUE4R0ExVUVCd3dJUm1GclpYUnZkMj
R4RlRBVEJnTlZCQW9NREVAaGEyVldZMj11SUVsdVl6RVNNQkFHQTFVRUN
3d0pRMj15Y0c5eVlYUmxNULF3RWDZRFZRUUREQXRtWVd0bGRtTnZiaTVw
YnpFaE1COEDU3FHU01iM0RRRUpBU11TWVdSdGFxNUFABUZYWlhaamIyN
HVim0puTVJvD0V3WURWUjBSREF4bVlXdGxkbU52Ymk1dmNtY3dIaGNOTW
pJd05USTVNakV6T1RNeFdoY05Nek13T1RJMk1qRXpPVE14V2pDQnF6RUx
NQWtHQTFVRUJoTUNWVkl4Q3pBSk5JnTlZCQWdNQWsxQk1SRXded11EVlFR
SERBaEdZV3RsZEc5M2JqRVJNQThHQTFVRUNnd01SbUZyWlZaamIyNHhFV
EFQQmdOVk5Bc01DRVJwZG1semFXOXVNUmd3RmdZRFZRUUREQTlrYVhZdV
ptRnJaWFPqYjI0dWFXOHhJVEFmQmdrcWhraUc5dzBCQ1FFV0VtRmtiV2x
1UUDaaGEyVjJZMj11TG05eVp6RVpNQmNHQTFVZEVRdlFaR2wyTG1aaGEy
VjJZMj11TG05eVp6Q0NBUE013RFFZSktvWklodmNOQVFFQkJRQURNZ0VQQ
URDQ0FRb0NnZ0VCQU04MVNLaWUxUzc4VkfFPQkRjM0xmbF1OU0w5cGhlY2
1qUXMvOVARaXFEWFNwbTN4cG9xc3pHb0xuN3cyMWFVQXhsKytGOVo4MFR
jNzBVUm4ycXFVY1ZUWwN4M1NtZ0h1WwdzcEVtSnNNUm9yL21uM0hwSHIy

YWpoKzI2UktnREdQQnJ0ZWJlSDFzaThtWDElYS9JQ3pIeHJVYUUYWS82R
FovMndHVFhyQUU0Yl1BT1VhWWZ0NUdHU1psNnlQNmRYZzNFSnkyYksyY1
VUZE1JWkcvWTD0eTFEM040VjdNQmlyRGQ1RlU3azZwTWJ5c jNkL3hvaTh
WTDhGMWlSQ2Y2enhmRk9qYU1GVDBhdEVpVHAzTTBydzZpUzA4aHkwY0RG
c1BKekljUGdYRWl0VWdHUG8vTkxnZWI4WmlMMWtnOUiYdklnYWFuYnVaQ
jZpbDc5QTRHZ0RJVNBD0VBQVRBTkJna3Foa2lHOXcwQkFRc0ZBQU9DQW
dFQVdPSkpsdlNRRHNEUk9RWk5kWDNRtTWtBSDNIWXlBbEU3TEhOdElZSFZ
oexQ3STdXdXFTNlpHdlNveFNma1QwR2RBTjdjVItqY2hkWnBBbnBDWEdY
K3lIdVN2cnc0b1NxaWdkU0xlU1VNeDJ6YVN2NHdMQXZ5ZEMvekRrSElST
2J5dURFS3l5L3BhNE5hMGF1b1BVM0UxQmcwVitKcVEzUlVKSXkxdjZBR0
xybG13YUVJZlAzQ2x6aS91NjQwZFdCSGFiT0VqeEZqOWJYNzMrRlJXaGl
0NEV6T0pZS3Z0aTFFWGxtUmpNYmozU2lzcMniR0lycHU0V2lwRUclUzA1
VmxNaFhYU3lRcTlJU2NGaG8lQ0Z0YW1EZ0kzakw0OWx5cngwK0NjakRrd
zBuVzlmWGRBdEiZQmhtWUVYZExlSkf2TFJRaXhVUUZ3SDIzZG91bVBpMX
VXOGVlRwhIVFJvMXBFRldRNVU1OU5PVFVBNUdodXVwbzBhTTJDbkRENVJ
lOHVPM0NoaWxDVWIwRmx3Z01ELzdBQlNkMUxDaFU4aWxlaEt5OUNFQlIy
emM4dVRxVzQzR2N2L3BzQmFlamdPS0pseFpldXNHVnZlVjFuVEhSYTRob
UZZeHhQZWw3b2dlWnpHUXFGakJsYjMlQ0UrQmNHwVkwVmlhSGVmeStmbW
kvZk42TnJyTjZwQk9QalZaYnBDRm4rd0ZUVXlyTVU1VnEvRmFQZTB1czd
pM3BhUWN3SXo5d3lpZGU1T3p0cHhlylK4VW5YbXkyMlgrNlBrSEIwam5M
MzVRZTh4bDdaMG5NWk5keTR4a3pHZ2FhNnRhbfJEUghLa1kxbElkdytLz
ENYSnVYyZdya0xLYTVwbz1TUGR2VkoRVB1MjZTSGtJMEkrZDVoeitnTX
A4PSIsIk1JSUDNekNDQkJ1Z0F3SUJBZ0lVSGsvNlFiempEWj1BT05BRnZ
3WUpXQi9qYnZvd0RRWUpLb1pJaHZjTkFRRUxCUUF3Z2FneEN6QUUpCZ05W
QkFZVEFsVlRNUXN3Q1FZRFRZRUU1EQUpOUVRfUk1BOEdBMVVFQnd3SVJtR
nJaWfJ2ZDI0eEZUQVRCZ05WQkFvTURFWmhMlZXWTI5dU1FbHVZekVTTU
JBR0ExVUVDd3dKUTI5eWNHOXlZWfJstVJRd0VnWURWUVFEREF0bVlXdGx
kbU52Ymk1cGJ6RWhNQjhHQ1NxR1NJYjNEUUVKQVJZU1lXUnRhVzVBWm1G
clpYWmpimjR1YjNkbnk1SVXdfDl1EVlIwUkRBeG1ZV3RsZG10dmJpNXZjb
WN3SGhjTklqSXdOVEk1TWpFek9UTXhXaGNOTXpJd05USTJNakV6T1RNeF
dqQ0JxREVMTUFR0ExVUVCaE1DVlZNeEN6QUUpCZ05WQkFnTUFrMUJNUkV
3RHdZRFZRUUheQWhHWvd0bGRHOTNiakVWTUJNR0ExVUVDZ3dNUM1GclpW
WmpimjRnU1clak1SSXdfQVlEVlFRTERBBERiM0p3YjNkaGRHVXhGREFTQ
mdOVk1JBTU1DMlpoYtJWMLkyOXVmbWx2TVNFd0h3WUpLb1pJaHZjTkFRA0
JGaEpoWkcxcGJRQm1ZV3RsZG10dmJpNXZjbWN4RlRBVEJnTlZlUkVNREd
aaGEyVjJZMj1lTG05eVp6Q0NBaU13RFFZSkvtWklodmNOQVFFQkJRQURn
Z0lQURDQ0Fnb0NnZ0lCQU1JZWfXa204cG42bXdVeldCZ25RaVks5SnIzV
nlWYw14QmM5L2R1VDhlQ045bW1BTkd6WDV6R2o4Nkt0R2psRGNvQXpBTn
J0c3IyRHB6NHpyYTI5SFdid2JPYTFaVE9PYVI5MlFnaWZrUzgwMnpwK3p
IR0dSYUtscTNldURxRFZ6eVhpbmhzdlhPVmxFd2tpckf2VjZrVWNUSXpj
eFZpYnJGZkh0dHp2L2hyVU54akNvdUFFN01JRzZ3Tzh5K0UyYkjrZzdSN
XlkZmVNF5b5nBCZWZ3dGlROG9XZjBDNDZtRFlGUThsMzFnb2F1MTdWcG
l5UzN0NGglSEwrMXlxZF1KUldtczVRaDB2cDlyZXE1QWp3dUhJREJ4Tkx
wTzJnWTFtUW1tZE8wM29MY1p3QmdsR0tscFFTd1lncj1KNXVvMkIrTlY5
YVlLRU81MWJ5U2luU0Y4VDREL280Y3E2b3lKTFhCd21Tc0xiampWSVFCb
lNkUkJtV2g4ekUyWnpGTmFhQmxZTzlpa0lhV1h3YnRrYnNFKzRRcE9hT2
ovalVJSExGSTlpt2hPaWRzSW5hT2ZnWXP3MTJGdEdFMVdxNXglWHN6d1d
wUFDVOE5RdzhnelR6RnpWT2J4cmZjc1hyczBzNmZldVRDRE10NVpodDJI

```

Q1dGcjlUbVlERkFmT0IwNWtGV1RpNXNFNUtZYVA1NGlKdmpBRm11VERqb
GlmN1o4WldWYWw5WEIrZzNzTXlROERWRWovMGRYUytuU1Z1SlFsVUg0cW
w5U1VJd0s2M1JLZGVkQUxZQzZYMWoybGdLOWFUdUwryY3lOSjVTYnVOQTk
1Q3hOZ3ZXdeJyY082SHhsYm1JRHhLckFBbzVhajRobmxRY1k0U1JiTndw
cENBaksremNBU2xweXZSQWdNQkFBR2pVekJSTUIwR0ExVWREZ1FXQkJTU
0R0ZnFVNitERXlZUENhcjA4eVZublpTRnpUQWZCZ05WSFNNUdeQVdnQ1
NTRHRmcVU2K0RFeVlQQ2FyMDh5Vm5uWlNGelRBUEJnTlZIUk1CQWY4RUJ
UQURBUUgvtUEwR0NTcUdTswIZRFFFQkN3VUFBNELDQVFDQEpEeFRkbWM4
N1lGZEtFbFdrBld4b1BNclBlUnleWVFvVlRDMnd5ZFpzZEolZedia0FoW
WxQUHhLK3p5cFAxOGZHcnk1UEhBSGw5bVpZeitGZEdqTlR3bGxjZk45L0
cwU2RveTRrVTl1tJlLZkM5T1VJa0JJcThEMHI3QU5UM2wzWG1HYm5JeFd
KRXRiMFN5VGNxazNkdldjTlRrX2dlaFEwM2pVS1cleVVwVm9Rak40VHRt
SHl3ZW9ZVE5pK0hLc0NVdkVYR2RBOXdkaDRqVk9xWWVlcG1zYXg1ZVRVS
nNLdUNLcVEzK3hCN0pPTnc4NWltZlZvdH1TQ1hzb01lN2x1U1JZdWxkbj
htQklwS1lVR0pscFdJaTZhCjBtTHdIYXJyWW9pOWJ3dGNLQXhSTlhaaUd
RZ2hpVXcyWnN3K2ptZ0JyRXRJNElhcm4d1dxSjVNb3Y3TXNUNbBub1o1
WHl3cjBHem5EbGdjTnR5cUlpYk9VTGVNOHlMUFFySG4vdFFqU2lQM2VVU
Ho2K1piOUZxelFKQ0NpMlVpYXVENG1PZ3pCU0NDYmV3OVBEWWIyUWVNWn
FOWm1OM2VWtk1WNGhoN3Q5a1rjMHPxUEY1amx1ZGRWbXBkSHhMcC9UOVh
adVNWMjFwOGNsTnBySk5ReHBydEhweER4amhOTUlyY1YzOXJ6aTl0Tmlr
d2l6UklmMHI0MGYvSlduYVY2NjhZzWtEWHR2SWo4cFp3eUU0Rk5lYw1kd
29qalF4RGZqcWs2N1FzNU5VUXNGY2Fjd01lczhJdDc0QVNibThEVUtFaE
RUMkFTZ010VUR3S1dyT0JlZnJlTUHFqWTcrRUlOMn1WWM5RaE13cHg0czd
DSDB6YTMjYXErS0M4R290SGRJNmppc0RFb203YUhnbgc9PSJdfQ"

```

```

    }
  ]
}

```

A.7. Encrypted vCon

This example vCon is the encrypted form of the Signed vCon (Appendix A.6) example. The private key to decrypt it can be found at:

https://raw.githubusercontent.com/vcon-dev/vcon/main/certs/fake_grp.key

The unformatted version of the following example can be downloaded from: https://raw.githubusercontent.com/ietf-wg-vcon/draft-ietf-vcon-vcon-core/refs/heads/main/examples/ab_call_ext_rec_encrypted.vcon

```

{
  "protected": "eyJhbGciOiJSU0EtT0FFUCIsImVuYyI6IkJyNTZDQkMtSFM1M
    TIifQ",
  "iv": "QdiLBqkZBsZmvzgyes9iSw",
  "ciphertext": "09tPHYpiiLOjSc8vi0KzhJ0lp4e5g00tGJ483F64f_ZsMuAo
    kfnWiCgThUfrUpk070zS867rchBaUoFbPnD6CNWBg5s9BG6zx375uKxPf2piI
    Jf5gsRfYDGF5PUKMD27v97TYP8paitB8JuE2pSjMl1MSobU7-uxWGiZJ_BA8
    Ens-yn_CJCFpscld-MxPixbbBz78lRlmAwz72NoSZbIBtV6BCiVqoQZ5Yd5o7

```

KXA_F9265nhe9To-6DFk8qLuKV9dJP1ftjemSP9wiYCPIaSpBNG8JWkZo2GWy
NJreQY9j_3YtldZ40sFkRFFdNJ12N-2BtCGceAfMvDD7Oyr7uSUFgRBZKqooH
BbWR6gIzyW5f-nD2VOjDGxOcXuUZi53COyqAzPdRCIXz_-rmACI5KmydLZRTX
HCgeQYOaEfEdP_wB0PF8j-GjoynP_gHzFZ8YgmHy8upRi5hUqGV7qzGC6t0Zv
O9fkqdH5mfmyXDPmhnZYtRFqol0MCl3OMQTt jLtJdw0s4hn30uzKhVwWlJsXK
EwuwlKcYdEYDa10dfsd4DgJGxqcZwyEXHqg24fIYwEQDHFcw8RTnQV9E2SCDJ
YpydLVFff0MszhFW_WQHlgT4i_869LWQDk2XMzqEHQZEjSKcyZAMVaQ8FLiTF
GhRclvyt5Wlql_oFdTxB-Jl9xyYdsLubbhDwVPKhmlz0qLePFEEluDJoZmjzbz
0KCpdMoP5AW8l0wLIp511AtLoIpX7UiL7uly1-9xGk0SnuiRmDqN01LBfJleR
UXSu5qOBdhY4dCKZu2tU2nLASqVMJUZYFxCoxWEct3wPKTR50N_g6KIG_t3Qc
tNqzhL2i9ineA0-1ESWXG-so3NPEeDRpxl7-dtlzmVBVrTVUeaqPnlxdraA0A
QeUn9MiIOYhFUJoJ_CwUEASCqd-GJ6T_teOizoYRDxBxiHfXH3YlFMxGe9XIV
Qzli8EvIfRpdG2NT15xM4BKvt0pjfCR0yZJX5h7sOxe8sFREz26DXXrcI_1Ew
02XCqQFhXEvDmp0HHrGr-ThXSo9Hp3nkFr0-nAQXXLcyqdLTtuCQRLVEuV5F1
OONlkqbOyMQK7WLAjXKXTLcDswerOYrTv92_VDln4YjmGoy3spGeezbSqyRFi
6tY9284ZEXERDwCF2kbPJFncVuXM5ldrBtL3Qyvwmko1qKKvPT2gEaaCQQ4b
r1QnVYOfEiMtylN6uEyAzmlN6QFAjwAnyAPKbN2XTlC_6HeOXeMzfbGa47qZp
KO5lhoXtWuoiFSOP1-jNqFQ9mlugQ1IGTOjrltCUds_gqAL2XRNvzLFozsuqi
HW9S5MvWk8ZcOKCNSrAD2zxIZAQQkbsVmMrzrwQpNtr7ZA3Fb4KT3o5sEXusD
K-pJD0wn7Qgor5HIEOGQZlfZ08H-HoXkowT1ExFehiYX4_JsoMxCz-RO9oIJf
F7FFxwLX8esIebFsCoA0oWueschZ2Zg2AKeIzbOWUGJZKyCPbSW-p9EglnUy
86sOWLQ4EX5Z02zUwP6Wrv3BH_vjEofQ2OX4C3l49Zu-j-k-L9-BYOmD5LUl9
2iUhj0hDyJAxayTPpJCEukCyvg63j9Utd9rlze5mhMUuCby-a73fJNA6ldDFb
v8Hdt-W_6xYEKXXRhuTQGTENZP6DGY3vKCMHsqXxTRAcPyreldMUuxpDU9eyd
qX25Y0_HUDD8zDXadqfdDf0nYvfQtmHA3bDBXDdSHwMlc3KznTJkHnTuYli_
hMKOSbuOWEO5Ur5BS-OiorcWlpPd25c9kW0YzmGrvN_-TUAgcGWat4voGZYxi
bqyk3j6_tTSuPiIjHmWucF9GjztOfH7xK5jffjOA98YuEzbQLJMBwcyU1LPUI7
2KUPyA73Dg05z7b5lK_b7qmC3idmPYU10ve4ussJGmUlgVFTF-F4o7MgzWw_N
a9a1jkHNldddzkQUgCYNLualnr33gUvR_3r2lJBUM-6K5jQvNiQbGsgh-U2Q7
6ZRU98cluVqMmnV4wNxQgXgykd2U1MtS_VgNtwL6ettjmSKl9wGd8RbYTqR64
3R_cFOevRsmuq7emmsr_UWUcUleCSiZeWg-i99Z7u_-tv3CqRRKnF3rd0QJi8
zpCnmJq6HDW_akORAQdRzb6vJqnY_3h6qI8LdA2XwnlXXAlThUiXInM_p0mG4
bAVNJ00wk1-T075FwXJOzFtYqn4hd19emtBiKdcJcIyxLdRc6hG7xJM-be-L8
tocqpvvxupcolXIsQxDcBKpmp6UgrvlVufn0ZOhuElEMcv6n8UcaSoxWh2-qy
xqAsfsp3MT5pjQBLzokAn0S3e9ZnId3B80sN10n9b8CfubBoFoznRdduTlY9y
6qnOYnJc67tOewohfqznn2D-9JO5FtzaCRNzVB7sAhidimdIzI-WB059PjwIG
Y2ybqH9iR7AwqnWeuZSHtQjhJbCr9N7NWcf-IhtzFIPBMCzB_hU8nBvEGpepk
z08-gIKwjhtQUCoMifSQJZma_eGtjff43kKdK_K4cknWB3njaOyDskdaLLHV90
4mT5EC5sQCcfDxE29wFWDvOZx5T8t5ncnc475hD-j8WsNjMolKNV4-ldSslzM
TR-Cggqt65tWb15THRrumeQN_tp88Da6OsHTOHL8w_TfE3dNKBAWPETvYS4CV
oDYRuI4NsW9_MWTvw_be6R2MRN86nFn9IQKcAMfUGL_uFCcjU9CHMBbsMlk8N
HIMTTd9xzWWZEWVW2CMLJWzOxb3ZN_VDhkJIIOtIUReOgl4uk65Afr9i_T57x
jsnJCKZMNUUHvmEHYfQTX74INhGZlGstnKFQynwu_en_CX_Z1luo1G0EXf0YX
TGUpp0QOPywuN5u-WVgMcA67pIigWlq6s8GVO3j7GmT5I49b0928lLw0gcpBR
bqW9lOKMlAhxpemnvK_1X_9iRdfxPteOWUWNBeddAYr8gIS-AVZrU3AbUJWbE
Pwadeqgjcjh7lxocbm0lB5BzOERRzn5xRCgs7J_jTsgZtxLpoj6mVGGvart8
p62TtLjFOCKnnkUN_4QxhrcPAPy5-E5ebBtAxUDV2w0bpiKxvLf5WPO_3m2NI
XtOjyPOxGjatHolGx1qboKhJmb-d_LGPzDEYzghZG0nyrX5qmMR_R_1roXJiJ

ouPCBq4ifysGg_cSEPvV-Hnct5BxQPCeVFeDjmJ4yLDaPobyRXbDYzilxyMoR
6dmRo_8EdicO4wCVjCYGSS2oCcKdX0ZAKhNegmDgKY3xIyapxP0-98QAGYtuy
GiSAY4QxEWGYkntNCeTftLSlu57mMz2FxiTeJNI8n6Ks32--TP_QGawxkz9CR
FDFlW7CNlvdHhGZa4MYPBxB-b_BAYZzO-PL9pOPNDhmOefgChGN6nm8cNIaIn
4m0TAXXRYg8VKcJsHawF9d3XBnClXVkuB0hYSrpoy0uZJ6zsGZ_KGQF1VlMqw
Tist-enp2xbU8X4MtZDQVqeBAsiBPXKJliU141_GoD6edChaYkMQZL0psQOuS
Vp3fO--zAPVqfpe08qwnVVCbPrKlBftW9EcRelVNoO8_HASHKGmlgdIzzdPlm
h4gxoe0JJlQyP3cNfzGFVTvXEYb0uZGzJB_wPdP73aNAuQuZlDmOqELzG0KUU
fpZ8sc2XorA-2gIZDdL_ZBznCDH82CNpFz_5l_FSkJH3JBWbsWgt6FX6WWHWw
mS6-T3pulZYupudqvmqI6UzP60JHvUxZjeBqhmFtlLBfdsmS2_Im-wAw14Yo5
iO4n6aFOWlqZfQc-Jr5oz9lCuyg2fhj9uEl-NFnFD0JP-b3EX5FYPqJ2B6kVS
ICmkXCmRcuEMCBPocUdg11j0ZMvII3KP2fj1meo0ZMlJTPd9F72pXCiTrqbtI
DtI1Hm9_002wAVEcCgh5GSX1tbOfUqcEuo4rRuKdpHqOFvYgYNIdD06zd_Cjd
g8laeI2j9j19Qq_2hUM76oWoiFMeFEI2Z7PIEe2US4wUKpzLPvMl06hsLJTcM
N8rXrXsVMX5vB1LbGhvnni6H_iULaXpaHtWyHwo2HNtsqNYNy3-FLtgFpElG1
gwUVz-Byk6ZURFFcDpfGkm6VXm1VMnLY_JNXuAJi5y1LCVpHeWy_HZ2vRVj80
poPD4beVHzdQnQPEkkcpwcv0MPmFPCmNjhPdMfBE0rCuvSfmj-aYZyVLBcieR
sJd0xBXx45swlD-bCS2jN7FHnFfjzoM4U7-PKXSeCuiowlwyWipncrDyJoPX_
WbKgesKDXqF3q78hASKic25FdhomluNULfFduo09WBKdYq50vxY7SI88jbah
Gob9JELz3iIrmPAGZ4v7sy9Y4ZMCG_bi06iVDRU239Ip8ZLlrg74BhGETxJH2
EeUrBiPBINATLzG6Fe5c3YmohXEQQtyHUSyBPEfqN8OPSSMoAdTLJp7s0xnrF
2hCqU_n-Nky7MQS0SylpQ7fGAPSOl8g9j9iRL9PNpM86xCDDWpHmuYp_uvnB-5
pnHo7w0fvBCsV24pPAUICthovbaf5LEBJ85ESiQIs_RBPNVpai2v_9eaIz06
LGKK4-GzOESd2s6DoND023H0XhvOM1sl7qPACEpnqrB-dN3jALauGALiSEQk4
FKh67W_-oV7oJx5ZbF29npYr0m79mhtOqlGPMwvL_jNdDrCHDciixpUqOEShv
Cf77Zi423ED3QHWPwPReU-DkipffmPKO4M9k8rH3_V3KNPLQ6Pek20XExrZ87
ENAQNLOwTXxmCn8eJV8f1QZTb0L8HqpzLQYhXAQIh_YHyK-QyW4Gec5jwJhnV
_z3MvYQMiHWWExbelN3urIjffZ90QaOrWCYfwCBRT5in6Try7XNVH9ZzYjDjq
bxjLQMFnBUNmL8wRpDUUDy2VJeqZYfN3i7JzJ_uzNO1GEmmLREa24SOXmBVt
rukI6rUph3-xXvndo4gZJLsgc74xFBSSHIFylG1z57aVJoQTKXsTa0Yr6-yr6
93BEeMCT8-Znol7uLdFInoaHyYRgr6aqH5es7H7ywZJ4ZSoOGR5fw6diRa6Wp
apzo3pSFu-teoVqknHT7ImKjv8lo2V_AvdeYaUKamlgMD3T9Y77nrmgtY6kfj
jJ04nHG6hqImeyuAOuq107tTXOGjEPMqne2jBSmPly5iuxj3G9BYQcup2WIGS
WSt_i8ZCM_ojAh3fx_feJvLB0awvR5-LJ3g-pUHjNJffdqQj6N1tAdwlzh_i
0fuvj5ZfjXmji6MveyVmHt-kCzU9J0lMoVvg-daPxITJeQ5wUzVgNHfylH1
h-sUPgJr0fE4r7pPEJce2SrMpeJlp-rZA-TRjju29_z38dZT5FczGNBtccmWJ
PPDqVRpV2lr-KVM5sKSWiVtU6Man-UivF3JXUqRBjLwlngh-kCc1LOpJwq_bZ
2tqonm0J0MF4rV0ogld3shVOqPmyH23sYmFYBJ4lSCozfEpUTZK4HaYpJLgT
hr8cK0fSV2haSOHiabNeOxZT8n446OFNumYMwW-CYCd5u5CL870Js3gbmzOFK
44qhyD6v8SsGPD0C4S3SM4my22bHMFcnkcnzBLdRBR_09SRNREL3l07GIG2sX
6SMwTDrafU2nzRvKv5gJ9CAyLg7YXyW6h1femtdFG_vOisSrvLausKJKNaMww
LRD-zpMAB5SG2Y3BDZZEoEBpOcuJ9TomJRiss4xgAunG_50xtWjQP1BDBzi6b
RTDnbOa94muc8nN4-ndWqVeMtxeBKTPAe8N0wMrKwOSk_vw-OfjSR-mURxg_T
-aj_MrpXZC1X5lgrW71tRgFGV4w5sJH5goEfuI-u9QweH-xj-vqIHhTkZSCnt
kiUoVBRZBekSVwimAF4RH_L7oMNCr7SHXB_LNlmjLqldFtfIY9EL92BUgXBD
GVLugxWtnfTl-CSCUaKNAVKrcU2xahYXzhZ2GEBINibptWL2jyGwmv_SzJZjo
7NmM6oKrecps0z6lq_uBnQkvfXd5U5zcHgJB_LVfMFcXDpR39MlUnZfYmFTZ
5mOhKVe6Xyh5tPlPwrfSXSA-ykkoqysB1XfSDRcA0oMdYTMYqKUGawTSOSfyn

HQ0BRKT4-B47EA2NsANh756dT_EtBXs6wqsp9EgPf2aurHPecYx_zbXXCrjfc
wWalnvp_3KR4yujKYfGoQTNVn3XzcvcB04Wo5lgRozjrPaoMKbocNn6Sth5xI
SiUMxViseqVOWFd2SljgnQHvHwn16i3vxQIvev0WAdiLnnEiRMY-0D8kw-33u
9i1-cK45Pb3nE-1F0XLHkJ_k-ocejvudOrDSt8pOgXKEgni-PMbeYDf1CcW-G
DiI-JOM2X2sNZl3XE4D_VL5YsrG4i3K2OdFDly5XaG8ts6sAl2vp_B1DS-_Ld
ehg3lin3whMc11eSCHA_-cUkWif9JNowvX5euI8ogJoYaAhIza0lmdgCK57g_
RS5iWRclcJSLMR3koOCh0Is0q-Yc1YK4lJ2DVLjkQXIRXz1DjHxaIuSE4pwZP
R9dbI4qsE8jfyZzdnVfKPHl0y4-WJmTk3eulkiY7lk8OSvQQNUQDoCJCdK0yW
WgnybTNUKxisfwnLUKmb1retkx8mnBegdMkgrqOw02qGlwTCYxUX_XM6uJsfYy
jO2R03TzCfkipGmn3jTYpTGXvPJNlt1cDYryc5FOW2tKoBqWmXPKiVlNhgRt5A
HSdre4ONxsqgnby7aFMrM9eHHvloDgyDdVagaOokEingADEa7VRKUilIDNP0w
PECxFSR65nuLj9kGGrVL4SPHzZSjDcysIcutdYCMqlwtP84qhOgoNjBNTSpJR
NZN8RHGnTxfiyXVNqozo3nJjQjsNl6dgvzvVl17cExKl3w72Yi_kMlqX4IrUR
gGYY3TEV9NfHxi4lHaEK0UaZxewOxMO6VyPqnkhVy4QPhiwtDmJsd76P-qKeD
rMK8yyFOqD0u8BM2j9tnELpUb3ikptYWPAut9hSSpJscTQrX02KiLNhepfHe2
TRsisxYeKhUCFnrrjNQM6q0xaSdjEiYdq5J6UANXP19MPSVJa-ZrNS2c06SWd
sMXfdfCCCM3SCQQM2gAs2krY3wxlnrQeQwinii-MROTSKT3cMeaL_wqlOfMlg
Qd7d9Eg0MgOU3J4YrdH4hccwuDTVI6AZ5ch8fenHkzYslruIy6Z7hFalIWlco
CrQHxRuXXF5zgzShR-TZ4UhcLGdHPFKBCq0OfJ8Zmn831slgEt1EAlF7on4Is
HqPoHSLF4cBku-T_ZTEdfyhH0Jj-nTkvarSWcorhAB1HCHnLE_u2myUggj0q_
6TTlcWC7umESpZ7eYPCiArQ0FtXukmjo12AMqOPHYSem7grN-Qu9RDJ9ILiMa
YmFYlODC7n3rkbeVH8stbSJ0luTlKfZbX6-CCSVBuJgtfzTA-pFa3rP_od8MK
odrraV59aVDglH867MW64pJXZMb8NmSQs7PzcL0lKBiB_P6Nt60MAkmEKDgRD
H4W8xKYQFl_bKZ__zoPqyDB-JNMqbtJWI2ZlL0h4B6qAltSMZcLG-3o67HcBz
dqJwXf5-KZVbtsiqhzQzCRr-0RG7W8bH_BvfSlfoxQneGIiIQS5ZWGDPhc-x1
B63rTWxYMRzQE7ufBOvwUFvcV7o03mkkgNnSZeyoSsi6HlLZEydoXkqcKxiz
qa30jAaWBXoG3E_AoiQI113GAV4GDc61pbQQhrhgkcP8l_4-PrHgTy_jjnk9I
Dk8blRMztN-v8Bsazl2fgTwKzvObyq4hfJSedDXOJY-lXttsGV34Lzh7Wha4V
G3d9995Cxa4McW_854ZhUmK3jVH7mhrGjrHWSMUuUjkmdfvWTifosI7s9L-jm
bWj7n0jvmXRBB57kbvl2YYGFIRyN699cTObslJ4o2J5qoVAznYrxSOVXWQE_K
q9vpnQBzu3QcvcRSLBowAT-OwDlW9STP8TMgJtMHDluAC12NGCj7Com2FZto
C_t_W3b-4HyAs8JvBOnR83XSknaqx1c56F6GB8BnWuK77yIDQce65bMNCELx
AESwOPzUA7HHsjZm2N85M7B8sqfYsWo0q0xvavrcPVxQYlBgatUq0hruvA6KS
uFRmXtb329SmWQdlKf3kxkhRNgDeTxgnMoSgx69kiqemlbedvJcAUajbD8vCM
6lyF0XeCgcMZ3cvVQS8ECuaozxt5Il3MOLhsL75Z59k2hkyIfVM6tzJDjb17G
1lZLOC07dFUZSJcRIBPpE2NVf5DRlifu8KmateGNeDgds13_Ctwfb7trrEdEG
ELhYgbMo_aKeGHiUiNapNj0qe4FWbxafwzvFYjnpioJtmdBao02_r6I5bGTc
djAl9KHuD05W-Yz3zP1f41NQ8ZDRWpf2W5qU_phnPDV6ooxskPC3i60cfS6GH
WG8r_yMGRR0-Gv6-m0waXl-hf-Yslqvw2vSwUmkjV4vmWu0v8Ih-MfeQx_in1
RIgZ-jtYRKe-ejUAEOK55Fek8z09y_NWRQsfFU6Kogh9Keu8p9ccnJgX5rEii
2KNMw44uqgwAJlSAAUHEetnsIAGKku8JLxZinyaIwfxX7TusHR-UKz5v4TYuO
OOh34vLLoTXVMnNjvQ6WjLBNRuWnDvdvJPXMcIZdjdbcdq58M569x0noc4PM
XWJUHAwiqls8khdDgRy8qEgi56GhP_ncDd2ZzaIErJO0lmd5SfmLgecyHF1BL
a2Pb2_nSImQ3HT-UFyFjAh2yltv33MJ95FvKX-cz_apt0vXG3L70th-Pl7mCm
3-Nu88c96uTRgt5mmFu0-1zt7NFmSf2J860Oe4oMUG5N0srYwQsgnoEGXCRit
eV-2LnWGUMlEkp3UWvmK-gbK2CAEzeWZ5gwTZwn6VGA2FyrxxHmirpB3cs7mk
yFq7iU_UP5RC5Y4Vboq7d03kpKUAJcoBNXvamlO-AloCIzakxAWFKTKmhKouq
DOSzJb0pci0bmH96g2OifH0xQIewNQAmD7vzicRleBhXdmGurDAjmanpWHQj

yP_Ud00_Z-zsOqwdF9iG3hKQdBpt7fOmByEBE0MM58qnOYsLGMeSJNfeM3sBf
_z6T8MenZuq-lIIu4OUOwkGLnDpV82Je_iOwXSe02y9sZWB0rWFeT32ea8xja
T6oKqN2fpzIrH5LMMzu78OEx21DTgR8Wy8MVH_ToVln1Kv3Q333Hx5yquydOt
NBGqf9ncmUuLNMJVJ8jDx3WzZCwU87DCBm8L_uMRV0gPvxITu5lMNdVDqST0Q
o-4We8NktoYxgDJ2WYUSBLceiLFqhyQXULtc248SXYEkeDSswSNAZA487in9m
5RM4kTRulhxEAqiqQ2Gk9IQtsQgAORxToZtkxHK3j5o5rgZcFT0Z92OJCR9VW
rRzsPstqcJpuCoBLd4nPDM3HMfiRXjSYBObxJlV9hsSKzj_M4pilVOJ627S2B
4FNqa5FKsGMgMdSYq5WUBO2YbGeGdDlksdKFZ9oC1y1IAcYAg6hpTbdLESYL_
3PUSat39AdiMVo9gB8xL0gZT7QmDGeCqdlNBdWvER7VZBfmlVlhU5NpFsFYyB
4wUB3QUdGLe6lYWE2vIXs1G33i28RWJywSjrwqM0Y5xwp2hJ9WKsdlonU_kuF
mxYf2OjOr7vXgFAjGz8hlC5nhZcRnG8vz2bKtu5zG7WI3Ab2k221U4zjDsQzM
gtcjyEky1Lh9GfzMHwhza2cfl7j4RrnuKe5mRZYDFctiEHDU115nJz_5vpwEM
vNPQJo-JOCsyupGMx27S3Wzs0QV9GkWVIpTn05K9CFXNvt3bWuuRyAgtgIVlJ
fba6U0aBrVRC7E8kdxUKXvHf9RxjeIX0pqLyRDvv9m4088Djib6g5F1jAY5iW
OwTLNbCmNkRkH8BiuI8QwpBA-JVSJ2s5LFcGpNT7EM0HObRzpmV3TDrYWRWA2
0pP_kQ8FsEu9ZCKctTdRiUvou4wMlx6INXp3OPQZneU66Hs9IcUOLBUetwmju
hwxpTRPOGnqlioDdLZRHP-hLKvYFGjkqDr1qM0Plm9ff53RIBUYFts6AnKvQ5
lbyqX6yX_fij_HinMTW_709ObtaQLibSRvuB7n6tpq6neSw9M8-T8qTZFGcO1
vmAUri5Qa84KnHFG6I0CpDrBSiAGP43jT0wWqExGBn62lX5aAnjpc2YkptPARC
bkxP-u5ycDUBoxSi55Gil0nv848Y5IW-LsIZps78D70066Px3aZAGnulbYjnB
cOBjy0qrd5RmOdHUII38ObXrxwp01TDUoESeJC3gxBgpz4-ogMPos1GvQp1Yd
_vstQNd5SP-_3jsKGoWvQ1Vmlt1odmFPYb87hAyYWcTxoQKrNwD85zdp8qDfM
0DVXgm7OZCyqG3Kw-0SHqOVWwRlhmFBI01RIMEM88L427QpkVHV1qTQZdtQ9N
5dWSz68xh40oC_sEDKuORnQlE9cDGWcLVJN4cSwH0b0pKN2aC-6sheBP_kKP2
McMjM1i9KzJ63U1KkeekGHw_dJM8Fg1X-AfUPkScqZPkVXQIKurjwdf0J1agI
q_oIkCbpvoq8l1l2yUttfOdS2bIx6lRIUVdJWI_RXF-7-SwCGP0iKp2v05SvG
H3hMAGbWwr7lLwZCV5uX-BCom-ci_fBRN-1GInRzpt1262WJnhSY6qK7PptdE
raViIbfOUbJUvVKA48xNLU_DSaEPBH-rTNaDZS202Xc35vDincrUlGqhDUSMB0
i6VHoAJE3qRharkmgzNUEzGo9ZDBvCGvGMDJ2kq8JBIq-ty4JBmsH1Fww1x83
DB4_qR_J3ESP7bavNiyZt-gKImURXBTjB1tFBNvoAygrls9ak_0hp9MhkTQuC
-bzGC3ao9W7i2Fp8jDpjhhK1llgI0boea00JIM2ihblmAyfipRZHMbZCEvIkT
Y8uWFvOUVTatdXRUBQ5UvIqX0qXG6flhakz83ZFVv_56L228jxtr4XlRDksqE
D9rjSEXP3XlNSVq2VOD_GZufvZvFVVPnPvB9-AI86ixQPv4MKLOm9ZtguKmdJ
ZT3vUUtOBHVMaOSXFSKYXNZsgyuDNY6bXvxxIjn8OCFqg09R0P8u5Rw7JZ0Nu
QjYebWD-F53gRikMix2kdTLH2v_7Qo6dCzS8kyj_GXlwRrNDlfy96CpYewqfy
ki56d8dVCodeE3yKhpenbb2JkmlfzngMsjsvYBPnUBYPpaOlNyRoWE67Z5IlnJ
JRktq4rDJ_urR1eVwTrga_J4c4Jjy5ZyObc_bLYHqzs1F3_ocowgKpMe0OaPO
ylrsoFmnlkYrGePcDRvUv16gaEVfBvRuYuJ1Pq1TJ4y2maW6XtvYIc6cZUHTm
fZ7fwkfNs7awJ6pOgF-XIBhz4qGdRXnNabOatcbolubblmWjoUrI5BrI6ITl1
U6x9e2Y_rYlRkaMmYKY2jDxHKuc53STeNqKlNK1NVawBjWSANEJZk_PbJ6Gvz
ioQ7bY3lUMYBBH-rddKxNZdaX02Z5WPvyTTA8Xhl7UPa05XQ0CMGHZ8HWjhaF
yvj3XKLkUXXenoqUqxzquc6pOFKSQMAfV0xPqwLuitr7ScG8VogAH7ARTddvm
monhN9Ehq9XqC67jPwjVIGynBQ58xYUre015GVgTf_FrBXLtR5tn85_GRRG91
88dJL-TWZQ_XNQMPaFeL6K5RThc4gnRr_zrljU0LAPmZsd86dlFhzHoqcmWx
BE0ln6zKf4E3ARMcDODHvwGueTYooUU6Lg9nTFYDmd14vJn20ebwmmtn0MzqC
94HlOf_PbYgLdAG8EWZjQuNrSOix_d4g3fdHl3LjHnOoCysvXs2f63sbEii7v
J-4y5wljTgY6P4y26yA5YXQw-LA_H-sLuH8pgcvCD9FzvovrAbnaXyOpQojpr
cKS4peMz7iXsxdIVnrySmileClqpg9foc-xaptN3C7G7RyXuQ4fWT0LZzUxr

BQFzOCQqbbn3TOa33Z3HrXH0g-2QFFfAXJH4uFn97dgsE20Ompq5Q4ZhHlOj6
jn7rGR2EaldThf1iYKnRTuNLEyQsMWndWYITZRZ1AyboeKIpf7v3wEavl0-pi
KB3A0aq26fMk7lSbjHzPnWepZE6IB9CVIAqZDYGTKqjeeuM4rdfXhKJIMOxGP
OLXr-_pXlDaW62xqSlNZRDqodJETnEHBWJi9kxogk2xINW17SbkZNQxg3TuaD
QNyrZlQ0QowXCJQ6rt5tIsotRyfkTL-r5Rwdf8lcPTkMbrkjA1DGJH18H9x0U
h4YdMbXp548vFdHZpoATIBhVGgl3NX_n5fkLlPvtRH6N97ZXTxEKVS5l4Bkat
qc8rWgCpalvYwTQSipEISm4mlr13Ar_s2xxANPSWJgSRglRfvAuzfkXtFSrfK
Cq-MdKQXhWig22pInKOstIldIerQUdH3rFN-YfUp-UgeiYARfQNa2WpQFIpPH
Varm5WMkj-r-z6-ZYox231A2JIAwNHKQUvUrgQKR30jcOYyXCQ_VScMhNx7H
RV6ptQCovr4hkfSJ6s6VX04XYbUbmK2C3TgOUhM15cx4r8odzHzYNVKn7hg3
XL9yat5nnzcMD9xiqXeBw5cXxV2gDGU6e4v4TN0stfey_cXa3UH2Rt6pMfRjO
EN4j10LgoTnJ6bEmrmu3FV6yDEPckphmjbguZgZG3K8TwZeONy2GldNVJiXN
tPL7sLL409e6B-QxxZfBe-LrDQXLBg0rk8Nu8YtDf45mjpTTGpOvyfSk-G_A9S
3m_o3tnGqMeuZIPvpr9SiAckapqXpGZV5sHcTxYRGyDhNLxxg7haFeEVAFMCO
0g-K2Cggu2iI8UKXz5DoXZ_vwBANjcbdiKRSpyYYSRmlBkIE7dhMBBEKf97ef
yTtdD6ZGDkYIbkj-iLwq0szXOIpnQaEo-9R2ivdg0UN1Ijswv74xSImpPOrp
3C_Izplakf0X8njo7RFBBrP2R-ni3OAQKFTbN1IzJlWn4MAim6vsD0UGjtdHrq
98Bgah89q4i-eRocVtk300SxLKSklGOLQQZ__ELUf6G2Cg9o2l85k7bodr4I
iISGY60JnkXtez3g46hw8syN7Kz4osW8QZpOK2tRLEJLYOWkuB4TnOcYnNWME
_DftaJi2QY3KXebfJmdGxNOaXena5hV5D_zU4uDXlvXwXoYNEqb0-xPkdD_u
hiUHXddK0m55V49IayS6slcNdtPVLlo2F878LxL04XdSgj3eIkJExsTGNx84i
GhjLn2wALklwkIXJz-nhEalHgBfwOi10Jm7KuvgpdUiyjeld9RSITWZn-r7km
V6ZkrGiOtnPLTluCkFr0KSvtpJKocD9j8UVH5UnQznVRoQ-tqSVzr-eVefjKp
y_GNfyMdz2mAQ6jgFS1cJ-TvmWDiKbORQilsmTSktXHYDEwxIreb-PTai0JBz
P8KYDYFBhkzSKlmy1cUfwbZdg1lKK5yNlHea3XEjndZSHEnh8nCeXXFhG56ZB
cAKUObjn-seqA-u_WZIVKntbFqC5A8YQBOKvKbifrgYZH72Xf_LB_EllfJrn0
vrpyumQV5AUjhIKp3pLCmkTrmlUArkxqBAe349tMnegR5rlg0q1CbCEAGp4FY
gXd7hratrLgccN6ArM79OR__nWNTImj3cf3n82qM8UNCDFH415rBRyn4TsQbi
vUgHfJEymDGhJaDKoHEa5l1ttZka3jzMMoiGFK6H7QbhJ0fUCDKpbIvf34EQnG
2jjIVYMzmXZjjUDONoOD-NlwpCBbCT0IsxMo2OUra8Jux5nI8romnC4QUf96G
HeTtsfge_aTqgIC_0HOSGUWyl87lvh9ZxXyPcNvv68LgQUlmqHzox6XyInsdX
Coy5ys-EOLxmJ8U4CwbeimnQu5TooYilYKsv_nnm_F6o5ViR4WBa9PoNMKffU
wPNfwTsTbL8jrLpNqOnxICxbe9TgDn7b3F3Gne42DOZzTy7GaiPTV3ece8PrK
kGch8478UpRnq7CynildYeEFkXmUpMrxU-7VU6uMLtw7y4E_u94mKTnKVzq6
PdUa0S_JNVLfEsyHPU2FJ-6N_6Cfz3l1fGPOqDWpVHZc_jeYlcDp8EAEqZXFvr
8Dwzx4ior_UpBBlnWxmRT2V8Dajb_u5gAkN2KO3CSXxzV9cswiefvSeo7PJ3r
H-VzPNkO-qSfQRlGhlHFwyK-joADaYtcfURyAXTWIVpw6pQbx0588M_A0FAyl
vwMmmv9xWCil-az5926bzBQmLytp4-vSF_tIm81uhHYwswzt8sETwq2JTP9yI
McdiYz4MzPlQUbzpME9cMfONGwU3g2y90KrWyYg-Ul6PYUkcW2Muctxorak9i
ZClpXMM4ls9apw4wwqvJsdm5jzxcUfmrCPvbsHokKdyNbbt0BNkZhLLEhhrHB
WF5ovhYJhWlCdmCR0vaCYlB-mz9iWXBfp7OyrLsvQgeksoTXvAEBLUi-NZArO
6HCpiu3l6d0EoVuyENjWfHSNODoVKY8vzEKRV8QA4KZXJN2EpEFCnr2ly0ee-
Gh0uKec_11_eoQ7o5tYa5Z4nZgq6m8KX_bzJiZturOUhI2af8tPUzlaLgqKdZ
R_60_dZZ9Rs59GLuCjnzjDu6yzCur7eJRCKyNI0kPdZGkIWt2tYWpqW3K05M9
K4rMT40cGTTBKcyclqnmDtoASKT--CogAer4A-NC-8KyE-5SB-nmmY4Lelr3Y
qf_JpUzF13zt7Wg1l3pog6NR_AXqLJMs0baGiA70mgpM1WKjmhsBz18kYV4uh
lZbRQtOqC1pLuzizr4Y-r8vLKvs2I_K8XEhB3Og544qRJFeIHltCCyCXUnP8Z
jPNTuQ_ruxn9vR2DxbmoWPYrGd-kgDtPn_v12Ix1zZXtd3dbpgB8Lc0X_xqZA

```

WgJB3-sDdTow4K5hItjMt6bYTGgNtUJJ_jHvvfd6Sj0zu4X-IuoJPI0n8nf9g
8P3Hxx6bbOftbelIoJLgJkvxo9SfGRi_G5zPbLqGc4DmWHOT-Qx6K4zM9qdez
Hhwpf6DBY6J0HTZiXni7JnU0CsAIziGLhdmQOObglR-lQKtoQvimKTNTc97t0
iNTOR4MOQ97zQZhd4EHeIPO4kbV-fnFQP3SfSLLE2aT7ZhnWQkAaxtafThvqD
XTSwud2WoF98tM0I7m0hBKtE8YO35C7-jJ3MeKj0vkPvOlorUAeyf8gca3-Tl
uG4m2XGJHqb7sPRIrgaKx_IdI4qnBQeu7vpnfalGkS6rEBhhEVIpRRmjekVBw
O2pa3dm_X5JNCoR5-fgVaKTGAolcORYMO7zYdLO_XvOL1ha8DC1GdDuk6kSti
XRdd45o3sDmxEOqWCQEsZESTPNzBrRrmnR21N9SlNkZciTN6UNsZL1800qTLH
kbeDzix4C6H8laiFA6lIClBccqmoYSdawjduHqIYfdZ_XNWSg7lhpTfoa9s4V
obEbFi9_FUghPEAMB2F4eGVN4FtY23edNQBWFUyclBEDBUzoaQU-G22bmioJ
yFHUdxrXhrzhgQd0DKad9Q5VomhLFkfyBftplnsehD4DY18kp5O5R032HuMEk
o4buCqb68Kc2v6vs6DVXPEk0KIzTHA4tPlI3h7S54_Au8Z30Gz0cY0OgF7qWa
lJTSfNl5WmqSwllFs01mI3ftWlFWEE5KvNXw_AidhZMrM_VR1nljRozkmZHub
Mw6RUweJX4AIH3sO482apLTj-rmHAh15MLxzJ7pJzgnZQmERKPrv2zV12U3N1
wZ9z3Efa4zoQx9yq698kg9chTdUmdsi3vVetchoWNM86ONBrelpJ3x2Zscuqe
B-CD7eg5m-2qlQmMvANBCBV7-gb641341yDidnFUuL3lKvFBj17AIHFRQbA90
k-xzsRwKXgRGVvLJoYa0HUYZWc3_14gUyQGqF8m-aZLvKNlQUbwvgAVUlbWDD
iSW78PQRCTzIXPYUxrOjA24g2-LXHoItArKL70MJJfstiRD3XXF73ISRjOhBN
dq7r74ajJN-e7jl_fAUcwKgeCbk00wMcagPwSXR4hZtC_HSDT_wmtEiossj_T
4VheVn-AkhxakMjThlb6CmgohUV-zU40VoHEcPQZ_ztPzVT6U_Wb8O4i5ZoaI
lf597GgZzYKqZmaLpt_rlg5jYLZuCrFOSJOWlyOKU91Ny3JdVxF3ijynRzD9j
ZUR_HnHler2R2sm8hCKiGRivRW0HuaHnxADAGN8HOfrwggF2PsskgC3mF9UoF
_YL3nJl_Ig9imUs294lJOxz9aPUZ-DRBEKoZQLXGdVww0OwzaCHIlp9y2Y7Hf
64ZFJmO8Slk3TSrAWLI5QKCDJzFYxtWov_ovpe9qnhW3pIbDogy0zFnaWixhg
Zxd9MKz_BGufYODPA6aKi4CaKdi2ca9p-eYAKoQqIONHndwBbQOg3JNovnPLs
lKyFK54VBVWKOlf4BhWaX-gnF-l-inRpl9i1EO2PdpzyFuG_aGpzdfr2Gs--
TfB40hmdOaKrY8xwZieLgihOPznTK_amur9lucVJ7HMi5XcQPqRQ5CQPCOWna
WS7NHJOW_38OQIyhBXZ5YqA8WbJrEbASTY4-oIbGvqm53WU28Us7dPxUppUeQ
gik6P2uB18o-Ka6gNIB1pFBMN23-CNXG6FPiRw_oceIUTg7RiSPxfPwpny24M
pRqzVbhlwd7qkAkjUMM_jdQLPnKhBzWs5t5-WGr1TFsocmz4b8c-cMNAeGb8R
N3J2Uk0XBNpO0wpYshJx95KMg2Ilzxq5XZft59kc7uedvmj9SQiRSIb4lZ4Q0
k8UaHB0KbuI6s45poBR4uh8xT-RCOAot1TU2D3RNT0pXlmHg-rvtZ-yjA_zEy
3b6iQl-ir-P_S2fKSbfsaVrfdWUCAqt-ai0qIKH7QHyUc25hGKQQVYd4nqhqd
tTspiHlIl_wRg8qo25k7APssM-hvdAw0ohvNofSEoEHMPJJIGZVfv7uMa24c4
UbdZJ0z2bHwGnhT-968wqcWEYDPDlap7m5LwQElQyfhLUk56eekNPwo83n-JB8
FKPDkLtZoY4SGRxfDqtT_ZVDcXj_67WgRDhoQyEFJmnpS0kLLtBcvxLE_ovRM
RCdfXS3-7dCxA1URm3y267OnU1WG6yuExrVasn5g4mhRq6PJd0h_d5CaVufJ7
vumdVotbBIWzGOC0kE8ndxsuGjPsKmElH5H1MMVxA61R86fokAalLtC6BgB4B
ExIkWATu90761lhO_M6c9t1Dw-QYI6lxIUHNJyfg9v6_92hUufu0WSboA-GsK
U7IF6Kw5R218wgCiCzp2fRWDa2w5wNywg23MkPlgylzlqKhyDu3DHLSu-VaX8
gZng8XZdV3ycFsSSRTQBgyTfOp79gKEOPAU7UwmVNKupKaFjId_-lycI6XhJP
ZOB5vPc2QfKLj-KMWAUG3DdXmLoifDyX9111pIA2Pj3vpKYjWLYld-ZeMU2gd
61wiwlK100n9JpXIq03MkftuM15lnwMA56Gzdfp4rBy_UdNjVg3HNDVMOGBaa
7YQERuspbew9p5Eglwr5lIQTK6H4T-hfBduVl-mAE",
"tag": "XaPjTWohDPGfCkaO2OKwaONEbi9DF59PXEsbxZlmj4w",
"recipients": [
{
  "header": {

```

```

    "enc": "A256CBC-HS512"
  },
  "encrypted_key": "auU5-pz75ZkIRxnVef7wBJSnhMIIPPyVKO8XCYHcJ
    lpyj4yC8BONX8NUVMLSgUhPlVAeDgirvWEGvIU5YmwBE0fQkmwXyhBRgo
    q3fkTR_5___7y1UkcsDxXpO5WHmyG8xeMeJDubkv0Cf0JNyJSGhgB0XL8-
    VmikI3x2ZSAZuCaxUZlRofBMQy4v9WvKEV9oEY-qu0RygffndDH550gQV
    MeePyRpDdBnDbB2VmSAGL8lRde2lIfyxPpH6h0RkerBAkU4TN2cx1QmcZ
    0kXdrhd1_pevI1QU9aeO2N-JrB296yIIn33NHyHQewJR0btC7Z1Z9oX3w
    k_KnOvabU7tgip-g"
  }
],
"unprotected": {
  "uuid": "0199e054-ed10-8035-b9a2-279e0d16bc46",
  "cty": "application/vcon+json",
  "enc": "A256CBC-HS512"
}
}

```

A.8. Redacted vCon

This is an example redaction of the vCon in the example Two Party Call vCon With Externally Referenced Recording (Appendix A.4). Most notable is the Redacted object (Section 4.1.8) which references the lesser redacted version of this vCon. In addition the **url** parameter has been redacted from the Dialog Object (Section 4.3), but the rest of the Dialog Object (Section 4.3) was left in the redaction.

The unformatted version of the following example can be downloaded from: https://raw.githubusercontent.com/ietf-wg-vcon/draft-ietf-vcon-vcon-core/refs/heads/main/examples/ab_call_ext_rec_redacted.vcon

```

{
  "vcon": "0.0.2",
  "parties": [
    {
      "name": "Alice"
    },
    {
      "name": "Bob"
    }
  ],
  "dialog": [
    {
      "type": "recording",
      "start": "2022-06-21T17:53:26.000+00:00",
      "duration": 33.12,
      "parties": [

```

```
    0,
    1
  ],
  "mediatype": "audio/x-mp3",
  "filename": "ab_call.mp3",
  "content_hash": "sha512-GLy6IPaIUM1GqzZqfIPZlWjaDsNgNvZM0iC
    ONNThnH0a75fhUM6cYzLZ5GynSURREvZwmOh54-2lRRieyj82UQ"
}
],
"analysis": [
  {
    "type": "transcript-redacted",
    "dialog": 0,
    "body": [
      {
        "parties": 0,
        "start": "2022-06-21T17:53:27.200000+00:00",
        "duration": 5.1199998,
        "text": "Hello. This is {{URL}}. My name is Bob. How
          can I help you? Hi."
      },
      {
        "parties": 0,
        "start": "2022-06-21T17:53:32.320000+00:00",
        "duration": 15.3349993,
        "text": "I'd like to add Fu to my service. Okay. Can
          you give me your name and phone number so I can look
          up your account? My name is Alice, and my number is
          {{INTEGER}} {{SSN}}. Thank you."
      },
      {
        "parties": 0,
        "start": "2022-06-21T17:53:48.935000+00:00",
        "duration": 6.3998730000000003,
        "text": "{{US_STATE}}. I've added two to your service.
          Is there anything else that I can help you with? N
          o. Thank you."
      },
      {
        "parties": 0,
        "start": "2022-06-21T17:53:55.334875+00:00",
        "duration": 2.6399989999999995,
        "text": "Goodbye. Goodbye. Have a nice day."
      }
    ]
  },
  {
    "encoding": "json",
    "vendor": "CapitalOne",
    "schema": "data_labeler_schema",
  }
]
```

```
    "product": "dataprofiler"
  },
  "attachments": [],
  "uuid": "01928e10-193e-8231-b9a2-279e0d16bc46",
  "redacted": {
    "uuid": "0199e054-ed10-8035-b9a2-279e0d16bc46",
    "type": "PII Redaction"
  }
}
```

A.9. Amended Signed vCon

This is an example vCon with an Amended Object (Section 4.1.9) referencing the vCon that it amends or adds additional content. In this case it is referencing the signed vCon example above. An example text Dialog Object (index 1) is the amendment.

The unformatted version of the following example can be downloaded from: https://raw.githubusercontent.com/ietf-wg-vcon/draft-ietf-vcon-vcon-core/refs/heads/main/examples/ab_call_ext_rec_amended.vcon

```
{
  "vcon": "0.0.2",
  "parties": [
    {
      "tel": "+12345678901",
      "name": "Alice"
    },
    {
      "tel": "+19876543210",
      "name": "Bob"
    }
  ],
  "dialog": [
    {
      "type": "recording",
      "start": "2022-06-21T17:53:26.000+00:00",
      "duration": 33.12,
      "parties": [
        0,
        1
      ],
      "url": "https://github.com/ietf-wg-vcon/draft-ietf-vcon-vcon-container/raw/refs/heads/main/examples/ab_call.mp3",
      "mediatype": "audio/x-mp3",
      "filename": "ab_call.mp3",
      "content_hash": "sha512-GLy6IPaIUM1GqzZqfIPZlWjaDsNgNvZM0iC
        ONNThnH0a75fhUM6cYzLZ5GynSURREvZwmOh54-2lRRieyj82UQ"
    },
    {
      "type": "text",
      "start": "2022-06-21T17:53:26.000+00:00",
      "parties": [
        1,
        0
      ],
      "mediatype": "text/plain",
      "body": "Good day!",
      "encoding": "none"
    }
  ],
  "analysis": [],
  "attachments": [],
  "uuid": "0199e054-f029-8e87-a20f-d188127fad29",
  "amended": {
    "uuid": "0199e054-ed10-8035-b9a2-279e0d16bc46"
  }
}
```

Appendix B. vCon JSON Schema

This JSON schema is provided as informational. The text in sections 1-6 of this document define the normative definition of the vCon schema. If the JSON schema provided below differs from that in the text of the above sections, the text should be taken as the definitive definition and the JSON Schema below should be considered incorrect.

The JSON Schema definition below has been formatted with line breaks to print nicely on the page. The unformatted version of the JSON schema can be downloaded from: https://raw.githubusercontent.com/ietf-wg-vcon/draft-ietf-vcon-vcon-core/refs/heads/main/vcon_json_schema.json

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "https://ietf.org/vcon/schemas/unsigned-vcon.json",
  "title": "vCon - Unsigned Form",
  "description": "JSON schema for the unsigned form of vCon (Con
    versational Data Container) as defined in draft-ietf-vcon-vco
    n-core-01, sections 1-4",
  "type": "object",
  "required": ["uuid", "created_at", "parties"],
  "properties": {
    "vcon": {
      "type": "string",
      "description": "DEPRECATED: Syntactic version of the JSON
        format. For this document, must be '0.4.0'",
      "const": "0.4.0"
    },
    "uuid": {
      "type": "string",
      "description": "Globally unique identifier for the vCon. S
        HOULD be a version 8 UUID",
      "format": "uuid"
    },
    "extensions": {
      "type": "array",
      "description": "List of names of all vCon extensions for a
        ny parameters not defined in the core schema",
      "items": {
        "type": "string"
      }
    },
    "critical": {
      "type": "array",
      "description": "List of extension names that are incompati
```



```
    ble with the core vCon schema and require explicit support",
    "items": {
      "type": "string"
    }
  },
  "created_at": {
    "type": "string",
    "description": "Creation time of this vCon in RFC3339 format",
    "format": "date-time"
  },
  "updated_at": {
    "type": "string",
    "description": "Last modified time of this vCon in RFC3339 format",
    "format": "date-time"
  },
  "subject": {
    "type": "string",
    "description": "Subject or topic of the conversation"
  },
  "redacted": {
    "type": "object",
    "description": "Reference to the unredacted or less redacted vCon instance version",
    "required": ["uuid", "type"],
    "properties": {
      "uuid": {
        "type": "string",
        "description": "UUID of the unredacted/prior vCon instance version",
        "format": "uuid"
      },
      "type": {
        "type": "string",
        "description": "Type of redaction performed"
      },
      "url": {
        "type": "string",
        "description": "HTTPS URL where the referenced vCon is stored",
        "format": "uri"
      },
      "content_hash": {
        "oneOf": [
          {"type": "string"},
          {"type": "array", "items": {"type": "string"}}
        ]
      }
    }
  }
}
```

```

    ],
    "description": "Hash(es) of the external content using
                    format: algorithm-base64url_encoded_hash"
  }
},
"amended": {
  "type": "object",
  "description": "Reference to the prior vCon instance versi
                  on that this vCon amends",
  "properties": {
    "uuid": {
      "type": "string",
      "description": "UUID of the prior vCon instance versio
                      n",
      "format": "uuid"
    },
    "url": {
      "type": "string",
      "description": "HTTPS URL where the referenced vCon is
                      stored",
      "format": "uri"
    },
    "content_hash": {
      "oneOf": [
        {"type": "string"},
        {"type": "array", "items": {"type": "string"}}
      ],
      "description": "Hash(es) of the external content (requ
                      ired if url is provided)"
    }
  }
},
"group": {
  "type": "array",
  "description": "Array of related vCons that aggregate into
                  this conversation",
  "items": {
    "type": "object",
    "description": "Group Object - details not fully specifi
                    ed in sections 1-4"
  }
},
"parties": {
  "type": "array",
  "description": "Array of Party Objects representing all pa
                  rties involved in the conversation",
  "items": {

```

```
    "$ref": "#/definitions/Party"
  },
  "dialog": {
    "type": "array",
    "description": "Array of Dialog Objects containing the cap
      tured conversation content",
    "items": {
      "$ref": "#/definitions/Dialog"
    }
  },
  "analysis": {
    "type": "array",
    "description": "Array of Analysis Objects containing analy
      sis performed on the conversational data",
    "items": {
      "$ref": "#/definitions/Analysis"
    }
  },
  "attachments": {
    "type": "array",
    "description": "Array of Attachment Objects for ancillary
      documents related to the conversation",
    "items": {
      "$ref": "#/definitions/Attachment"
    }
  }
},
"definitions": {
  "Party": {
    "type": "object",
    "description": "Represents a party involved in the convers
      ation",
    "properties": {
      "tel": {
        "type": "string",
        "description": "TEL URL (RFC3966) for the party"
      },
      "sip": {
        "type": "string",
        "description": "SIP URL for the party"
      },
      "stir": {
        "type": "string",
        "description": "STIR PASSport in JWS Compact Serializa
          tion form"
      },
      "mailto": {
```

```
    "type": "string",
    "description": "MAILTO URL (RFC6068) for the party",
    "format": "email"
  },
  "name": {
    "type": "string",
    "description": "Name of the party"
  },
  "did": {
    "type": "string",
    "description": "Decentralized Identifier (DID) URI for
      the party"
  },
  "validation": {
    "type": "string",
    "description": "Label or token identifying the method
      of identity validation used"
  },
  "gmlpos": {
    "type": "string",
    "description": "Geographic location in GML pos format
      (latitude longitude)"
  },
  "civicaddress": {
    "$ref": "#/definitions/Civicaddress"
  },
  "uuid": {
    "type": "string",
    "description": "Unique identifier for the participant",
    "format": "uuid"
  }
}
},
"Civicaddress": {
  "type": "object",
  "description": "Civic address information for a party's lo
    cation",
  "properties": {
    "country": {"type": "string"},
    "a1": {"type": "string", "description": "National subdiv
      ision (state/province)"},
    "a2": {"type": "string", "description": "County/parish/d
      istrict"},
    "a3": {"type": "string", "description": "City/township"},
    "a4": {"type": "string", "description": "City division/b
      orough"},
    "a5": {"type": "string", "description": "Neighborhood/bl
      ock"},
  },
}
```

```

    "a6": {"type": "string", "description": "Street"},
    "prd": {"type": "string", "description": "Leading street
      direction"},
    "pod": {"type": "string", "description": "Trailing stree
      t suffix"},
    "sts": {"type": "string", "description": "Street suffix"
      },
    "hno": {"type": "string", "description": "House number"},
    "hns": {"type": "string", "description": "House number s
      ufix"},
    "lmk": {"type": "string", "description": "Landmark"},
    "loc": {"type": "string", "description": "Additional loc
      ation info"},
    "flr": {"type": "string", "description": "Floor"},
    "nam": {"type": "string", "description": "Name/occupant"
      },
    "pc": {"type": "string", "description": "Postal code"}
  }
},
"Dialog": {
  "type": "object",
  "description": "Represents a segment of captured conversat
    ion",
  "required": ["type", "start"],
  "properties": {
    "type": {
      "type": "string",
      "enum": ["recording", "text", "transfer", "incomplete"
        ],
      "description": "Type of dialog"
    },
    "start": {
      "type": "string",
      "format": "date-time",
      "description": "Start time of the dialog in RFC3339 fo
        rmat"
    },
    "duration": {
      "anyOf": [
        {"type": "integer", "minimum": 0},
        {"type": "number", "minimum": 0}
      ],
      "description": "Duration in seconds"
    },
    "parties": {
      "anyOf": [
        {"type": "integer", "minimum": 0},
        {"type": "array", "items": {"type": "integer", "mini

```

```

        "mum": 0}},
    {
      "type": "array",
      "items": {
        "anyOf": [
          {"type": "integer", "minimum": 0},
          {"type": "array", "items": {"type": "integer",
            "minimum": 0}}
        ]
      }
    }
  ],
  "description": "Index/indices of parties in the parties array",
},
"originator": {
  "type": "integer",
  "minimum": 0,
  "description": "Index of the originating party if first party is not the originator"
},
"mediatype": {
  "type": "string",
  "description": "Media type of the dialog content"
},
"filename": {
  "type": "string",
  "description": "Original filename of the dialog content"
},
"body": {
  "type": "string",
  "description": "Inline content of the dialog (for inline files)"
},
"encoding": {
  "type": "string",
  "enum": ["base64url", "json", "none"],
  "description": "Encoding type for inline content"
},
"url": {
  "type": "string",
  "format": "uri",
  "description": "HTTPS URL for externally referenced content"
},
"content_hash": {
  "oneOf": [

```

```

        {"type": "string"},
        {"type": "array", "items": {"type": "string"}}
    ],
    "description": "Hash(es) of external content"
},
"disposition": {
    "type": "string",
    "enum": ["no-answer", "congestion", "failed", "busy",
        "hung-up", "voicemail-no-message"],
    "description": "Reason for incomplete dialog (required
        for incomplete type)"
},
"session_id": {
    "oneOf": [
        {"$ref": "#/definitions/SessionId"},
        {"type": "array", "items": {"$ref": "#/definitions/S
            essionId"}},
        {
            "type": "array",
            "items": {
                "oneOf": [
                    {"$ref": "#/definitions/SessionId"},
                    {"type": "array", "items": {"$ref": "#/definit
                        ions/SessionId"}}
                ]
            }
        }
    ]
},
    ],
    "description": "Session ID(s) for the dialog"
},
"party_history": {
    "type": "array",
    "items": {"$ref": "#/definitions/PartyHistory"},
    "description": "History of party join/drop/hold/mute e
        vents"
},
"transferee": {
    "type": "integer",
    "minimum": 0,
    "description": "Party index of the transferee (for tra
        nsfer type)"
},
"transferor": {
    "type": "integer",
    "minimum": 0,
    "description": "Party index of the transferor (for tra
        nsfer type)"
},

```

```

    "transfer_target": {
      "oneOf": [
        {"type": "integer", "minimum": 0},
        {"type": "array", "items": {"type": "integer", "minimum": 0}}
      ],
      "description": "Party index/indices of the transfer target (for transfer type)"
    },
    "original": {
      "oneOf": [
        {"type": "integer", "minimum": -1},
        {"type": "array", "items": {"type": "integer", "minimum": -1}}
      ],
      "description": "Dialog index/indices of original conversation (for transfer type)"
    },
    "consultation": {
      "oneOf": [
        {"type": "integer", "minimum": -1},
        {"type": "array", "items": {"type": "integer", "minimum": -1}}
      ],
      "description": "Dialog index/indices of consultation (for transfer type)"
    },
    "target_dialog": {
      "oneOf": [
        {"type": "integer", "minimum": -1},
        {"type": "array", "items": {"type": "integer", "minimum": -1}}
      ],
      "description": "Dialog index/indices of target dialog (for transfer type)"
    },
    "application": {
      "type": "string",
      "description": "Application, communication channel or context of the conversation"
    },
    "message_id": {
      "type": "string",
      "description": "Unique message identifier from the messaging system"
    }
  },
},

```



```
"sessionId": {
  "type": "object",
  "description": "Session identifier with local and remote U
    UUIDs",
  "required": ["local", "remote"],
  "properties": {
    "local": {
      "type": "string",
      "description": "Local UUID for the session"
    },
    "remote": {
      "type": "string",
      "description": "Remote UUID for the session"
    }
  }
},
"partyHistory": {
  "type": "object",
  "description": "Records party events during the dialog",
  "required": ["party", "time", "event"],
  "properties": {
    "party": {
      "type": "integer",
      "minimum": 0,
      "description": "Index of the party"
    },
    "time": {
      "type": "string",
      "format": "date-time",
      "description": "Time of the event in RFC3339 format"
    },
    "event": {
      "type": "string",
      "enum": ["join", "drop", "hold", "unhold", "mute", "un
        mute", "keydown", "keyup"],
      "description": "Type of event"
    },
    "button": {
      "type": "string",
      "description": "DTMF digit, character or string (requi
        red for keydown/keyup events)"
    }
  }
},
"attachment": {
  "type": "object",
  "description": "Represents an ancillary document related t
    o the conversation",
```

```
"required": ["start", "party", "dialog"],
"properties": {
  "purpose": {
    "type": "string",
    "description": "text description of what the attachment is for"
  },
  "start": {
    "type": "string",
    "format": "date-time",
    "description": "Time the attachment was sent/exchanged in RFC3339 format"
  },
  "party": {
    "type": "integer",
    "minimum": 0,
    "description": "Index of the party that contributed the attachment"
  },
  "dialog": {
    "type": "integer",
    "minimum": 0,
    "description": "Index of the dialog this attachment is part of"
  },
  "mediatype": {
    "type": "string",
    "description": "Media type of the attachment"
  },
  "filename": {
    "type": "string",
    "description": "Original filename of the attachment"
  },
  "body": {
    "type": "string",
    "description": "Inline content of the attachment (for inline files)"
  },
  "encoding": {
    "type": "string",
    "enum": ["base64url", "json", "none"],
    "description": "Encoding type for inline content"
  },
  "url": {
    "type": "string",
    "format": "uri",
    "description": "HTTPS URL for externally referenced attachment"
  }
}
```

```
    },
    "content_hash": {
      "oneOf": [
        {"type": "string"},
        {"type": "array", "items": {"type": "string"}}
      ],
      "description": "Hash(es) of external content"
    }
  },
  "Analysis": {
    "type": "object",
    "description": "Represents analysis performed on the conversational data",
    "required": ["type", "vendor"],
    "properties": {
      "type": {
        "type": "string",
        "description": "Semantic type of analysis (e.g., report, sentiment, summary, transcript, translation, tts)"
      },
      "dialog": {
        "oneOf": [
          {"type": "integer", "minimum": 0},
          {"type": "array", "items": {"type": "integer", "minimum": 0}}
        ],
        "description": "Index/indices of dialog objects this analysis is based on"
      },
      "mediatype": {
        "type": "string",
        "description": "Media type of the analysis file"
      },
      "filename": {
        "type": "string",
        "description": "Original filename of the analysis data"
      },
      "vendor": {
        "type": "string",
        "description": "Vendor or product name that generated the analysis"
      },
      "product": {
        "type": "string",
        "description": "Product name to differentiate from other vendor products"
      }
    }
  },
```

```

    "schema": {
      "type": "string",
      "description": "Token or label for the data format/schema of the analysis"
    },
    "body": {
      "type": "string",
      "description": "Inline content of the analysis (for inline files)"
    },
    "encoding": {
      "type": "string",
      "enum": ["base64url", "json", "none"],
      "description": "Encoding type for inline content"
    },
    "url": {
      "type": "string",
      "format": "uri",
      "description": "HTTPS URL for externally referenced analysis"
    },
    "content_hash": {
      "oneOf": [
        { "type": "string" },
        { "type": "array", "items": { "type": "string" } }
      ],
      "description": "Hash(es) of external content"
    }
  }
}

```

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 and testing out the extension framework with MIME.

- * 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.
- * Thank you to Mike Jones for input and help media types, version and helping to form the extension framework.
- * Thank you to Jonathan Lennox for numerous inputs including separate single channel recordings per party and the extension framework.
- * Thank you to Marc Petit-Huguenin for sorting out session_id.
- * Thank you to Rob Sliwa for heling to review and test the JSON Schema.
- * Thank you to Mike Jones, Richard Barnes and Mike Ounsworth for their help with providing an approach to maintaining integrity for long term storage of signed vCons.

Author's Address

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