

Virtualized Conversations  
Internet-Draft  
Intended status: Standards Track  
Expires: 8 January 2026

D. G Petrie  
SIPEZ LLC  
7 July 2025

The JSON format for vCon - Conversation Data Container  
draft-ietf-vcon-vcon-core-00

## 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 8 January 2026.

## Copyright Notice

Copyright (c) 2025 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
1.3.1. 1.4. *Scope of the Document* . . . . .	6
2. Conventions and Definitions . . . . .	6
2.1. Terminology . . . . .	7
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 . . . . .	9
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 . . . . .	11
4.1. vCon JSON Object Keys and Values . . . . .	12
4.1.1. vcon . . . . .	12
4.1.2. uuid . . . . .	12
4.1.3. extensions . . . . .	13
4.1.4. must_support . . . . .	13
4.1.5. created_at . . . . .	13
4.1.6. updated_at . . . . .	13
4.1.7. subject . . . . .	14

4.1.8.	redacted . . . . .	14
4.1.9.	appended . . . . .	16
4.1.10.	group Objects Array . . . . .	18
4.1.11.	parties Objects Array . . . . .	19
4.1.12.	dialog Objects Array . . . . .	19
4.1.13.	analysis Objects Array . . . . .	19
4.1.14.	attachments Objects Array . . . . .	20
4.2.	Party Object . . . . .	20
4.2.1.	tel . . . . .	20
4.2.2.	sip . . . . .	20
4.2.3.	stir . . . . .	20
4.2.4.	mailto . . . . .	20
4.2.5.	name . . . . .	21
4.2.6.	did . . . . .	21
4.2.7.	validation . . . . .	21
4.2.8.	jCard . . . . .	22
4.2.9.	gmlpos . . . . .	22
4.2.10.	civicaddress . . . . .	22
4.2.11.	timezone . . . . .	23
4.2.12.	uuid . . . . .	23
4.3.	Dialog Object . . . . .	24
4.3.1.	type . . . . .	24
4.3.2.	start . . . . .	24
4.3.3.	duration . . . . .	25
4.3.4.	parties . . . . .	25
4.3.5.	originator . . . . .	26
4.3.6.	mediatype . . . . .	26
4.3.7.	filename . . . . .	27
4.3.8.	Dialog Content . . . . .	27
4.3.9.	disposition . . . . .	28
4.3.10.	session_id . . . . .	28
4.3.11.	party_history Objects Array . . . . .	29
4.3.12.	Dialog Transfer . . . . .	29
4.3.13.	application . . . . .	31
4.3.14.	message_id . . . . .	31
4.4.	Attachment Object . . . . .	31
4.4.1.	type or purpose . . . . .	32
4.4.2.	start . . . . .	32
4.4.3.	party . . . . .	32
4.4.4.	dialog . . . . .	32
4.4.5.	mediatype . . . . .	32
4.4.6.	filename . . . . .	33
4.4.7.	Attachment Content . . . . .	33
4.5.	Analysis Object . . . . .	33
4.5.1.	type . . . . .	33
4.5.2.	dialog . . . . .	34
4.5.3.	mediatype . . . . .	34
4.5.4.	filename . . . . .	34

4.5.5.	vendor	35
4.5.6.	product	35
4.5.7.	schema	35
4.5.8.	Analysis Content	35
4.6.	Group Object	36
5.	Security Considerations	37
5.1.	Signing Externally Referenced Files	39
5.2.	Signed Form of vCon Object	39
5.2.1.	Signature Object	40
5.2.2.	Header Object	40
5.3.	Encrypted Form of vCon Object	41
5.3.1.	Unprotected Object	41
5.3.2.	Recipient Object	42
5.3.3.	Header Object	42
5.4.	Differentiation of unsigned, signed and encrypted forms of vCon	42
6.	IANA Considerations	43
6.1.	JSON Format vCon Mediatype	43
6.2.	gzip Format vCon Mediatype	44
6.3.	vCon JSON registry	45
6.3.1.	Object Registry Template	45
6.3.2.	vCon Object Parameter Names Registry	46
6.3.3.	Party Object Parameter Names Registry	47
6.3.4.	Dialog Object Parameter Names Registry	48
6.3.5.	party_history Object Parameter Names Registry	51
6.3.6.	Attachment Object Parameter Names Registry	53
6.3.7.	Analysis Object Parameter Names Registry	54
6.3.8.	Redacted Object Parameter Names Registry	55
6.3.9.	Appended Object Parameter Names Registry	56
6.3.10.	Group Object Parameter Names Registry	56
6.4.	vCon Extensions Names Registry	57
6.4.1.	Extensions Registration Template	57
6.5.	JSON Web Signature and Encryption Header Parameters	58
6.6.	JSON Web Signature and Encryption Unprotected Parameters	58
7.	Non-Upward Compatible Changes to the vCon Container	58
7.1.	Version 0.0.2 to 0.3.0	58
7.2.	Version 0.0.1 to 0.0.2	59
8.	References	59
8.1.	Normative References	59
8.2.	Informative References	61
Appendix A.	Example vCons	62
A.1.	Two Party Call vCon With Inline Recording	63
A.2.	Text Chat vCon	67
A.3.	Email Thread Multipart vCon	67
A.4.	Email Thread Text vCon	70
A.5.	Two Party Call vCon With Externally Referenced Recording	72

A.6. Two Party Call vCon with Analysis . . . . .	73
A.7. Signed vCon . . . . .	92
A.8. Encrypted vCon . . . . .	98
A.9. Redacted vCon . . . . .	105
A.10. Appended Signed vCon . . . . .	107
A.11. vCon Group . . . . .	107
Acknowledgments . . . . .	107
Author's Address . . . . .	107

## 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.

#### 1.3.1. 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 append 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.
- \* "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.

\* `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.

However, when disruptive extensions are necessary, the names of all such extensions *\*MUST\** be listed in the `must_support` (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 `must_support` 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 append 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 appended (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 the value of vcon parameter contains the syntactic version of the JSON format used in the vCon.

- \* vcon: "String"

For syntax defined in this document, the string MUST have the value: "0.3.0"

##### 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. must\_support

Implementations that include extensions which are incompatible with the core vCon schema MUST list the names of those extensions in the must\_support parameter. A vCon that includes a must\_support 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 must\_support parameter MUST NOT attempt to process or operate on the vCon, except to reject it or report unsupported content.

```
* must_support: "String[]"
```

#### 4.1.5. created\_at

The created\_at parameter provides the creation time of this vcon, which MUST be present, and should not be 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 appended 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.

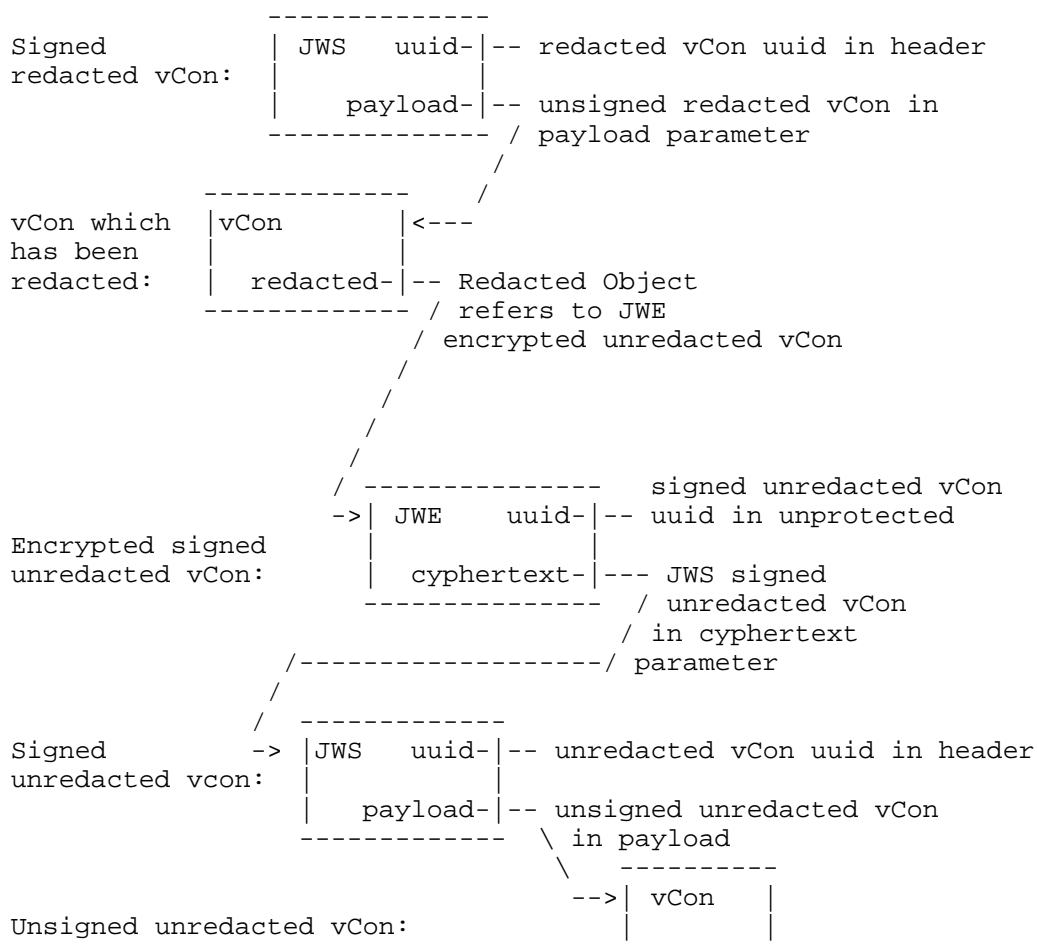


Figure 1: redacted vCon object tree

## 4.1.9. appended

TODO: should the name be appended, updated or 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 Appended Object (Section 4.1.9.1). The vCon with appended or 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)).

- \* appended: "Appended" (optional, mutually exclusive with redacted and group parameters)

#### 4.1.9.1. Appended Object

The Appended 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 appended vCon. The top level object is the JWS signed appended vCon which contains the unsigned form of the vCon in its payload parameter. The second level object is the appended 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 appended parameter. Note: the appended parameter may refer to the prior version of the vCon via URL. The appended 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.

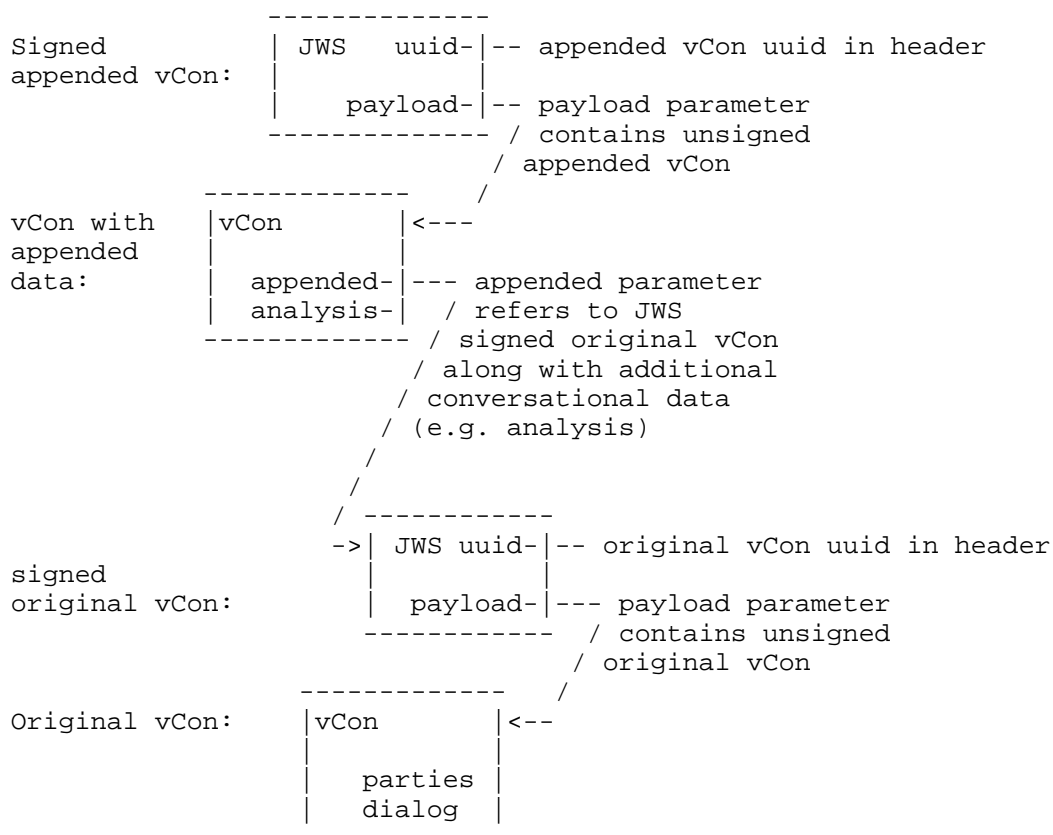


Figure 2: appended vCon object tree

## 4.1.10. group Objects Array

The scope of a conversation is defined by the observer. It may be any of the following in this non-exhaustive list:

- \* a quick text exchange
- \* a simple 2-way call
- \* an evolving group chat
- \* a class lecture and question and answer session
- \* a web chat, evolving to a 2 way call, progressing to a transferred 3-way call transitioning to a video conference

- \* a series of weekly status calls

In support of these constructs, it may be desirable to aggregate a group of vCons as opposed to including all of the dialog in a single vCon. The conversations may be over heterogeneous or homogeneous medium. A vCon MAY aggregated a group of vCon instances in the group array, using a Group Object for each vCon instance.

- \* group: "Group[]" (optional, mutually exclusive with redacted and appended parameters)

The group array contains a Group Object (Section 4.6) for each vCon.

#### 4.1.11. 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.

TODO: Should this be a object not an array to make it easier to append parties (i.e. indices of append vCons change when appended)?

- \* parties: "Party[]"

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

#### 4.1.12. 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. It is the applications responsibility to sort dialog and transcriptions if they are needed in chronological order.

#### 4.1.13. 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).

#### 4.1.14. attachments Objects Array

Ancillary documents, discussed, presented, referenced or related to the conversation may be stored in the attachments array.

\* attachments: "Attachment[]" (optional)

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

### 4.2. Party Object

#### 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.

#### 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. jCard

TODO: Do we want to support including a jCard for the party?

#### 4.2.9. 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.

#### 4.2.10. 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)

TODO: Do we need RFC6848 civic address extensions?

TODO: Is there a need for any temporal location? For example should location be an array, each element with a time stamp?

TODO: Do we just specify for the start of the conversation?

#### 4.2.11. timezone

TODO: timezone for the location of the party?

#### 4.2.12. 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 two types of dialog objects are defined in this document. One for text media and the other for audio or video recording files.

TODO: Is there other signalling data that we want to capture other than start and duration and the media (e.g. from jabber, sms, mms, email, SIP, etc.)?

#### 4.3.1. type

\* type: "String"

The sting 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" had 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 dialogs MUST have a disposition parameter which indicates why the call or conversations failed. In the "transfer" case, the conversation is recorded in other dialogs. The Dialog Transfer parameters, are used to show the roles and relationships between the parties and other dialogs 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.

\* 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.

TODO: For an email thread, To and Cc parties are all passive. Do we just put the sender as the party or do we want to list all of the recipients and by convention the sender is the first party? Note that each dialog/email could have a difference set of recipients.

#### 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

TODO: should this go in CC extensions?

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 dialogs, 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.

- \* session\_id: "String" (optional)

The `session_id` value is the string as defined in section 4.1 of [SESSION-ID].

#### 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

#### 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"

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.

- \* original: "UnsignedInt"

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

- \* consultation: "UnsignedInt" (optional)

The value of the consultation parameter is the index into the dialogs Object array to the "recording", "text" or "incomplete" type dialog for the consultative dialog between the Transferor and the Transfer Target.

- \* target\_dialog: "UnsignedInt"

The value of the `target_dialog` parameter is the index into the `dialogs` 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. type or purpose

TODO: Do we want a semantic type like: contract or presentation? Or a subject or title.

#### 4.4.2. start

The start parameter contains the date and time that the Attachment Object was sent/exchanged.

\* 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:

- \* "summary"
- \* "transcript"
- \* "translation"
- \* "sentiment"
- \* "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 dialogs in the dialog 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[]"
```

#### 4.6. Group Object

A conversation may have take place using different modes (e.g. web chat which evolves to email, which evolves to phone call, which evolves to video conference). A conversation could take place over several calls (e.g. multiple calls regarding a support incident or problem). Each of these examples might be considered a single conversation event though there are multiple sets of dialog in each. What is considered the boundary of a conversation is a business decision. There are situations in the above example, where it is desired to treat these as a single conversation, but each set of dialog is created in a single vCon (e.g. each dialog occurred in a separate communication silo or security domain) which gets signed. For this reason, it may be necessary to aggregate the separate vCon into a single vCon which is considered the whole of a conversation. The Group Object includes or refers to a vCon to be aggregated into the whole of a single vCon conversation.

The Group Object SHOULD contain the uuid and the url and content\_hash parameters (see Externally Referenced Files (Section 2.4)). The referenced vCon SHOULD be via UUID:

```
* uuid: "String"
```

The value of the uuid parameter, is the uuid string value (Section 4.1.2) of the referenced vCon to be aggregated.

The vCon MAY be included in line as the value of the body parameter. The encoding parameter MUST be included with the body parameter, if provided, to describe the encoding of the vCon body.

```
* body: "vCon"
```

The JSON unsigned form of the vCon, the JWS signed form of the vCon or the JWE encrypted form of the vCon.

- \* encoding: "String"

The encoding string MUST have the value: "json".

In addition, the location for the referenced vCon MAY be provided. The url and content\_hash parameters and values are defined in Externally Referenced Files (Section 2.4).

- \* 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 appended (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 appended (Section 4.1.9)). See the redacted vCon object tree figure-1 and appended 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 appending 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.

TODO: How to deal with expired signatures?

### 5.3. Encrypted Form of vCon Object

TODO: Check this terminology:

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: [JSON]
- \* Additional information:
- \* Security considerations: See Security Considerations section of this document.
- \* Interoperability considerations:
- \* Published specification: VCON, this document

- \* Applications that use this media type:
- \* Fragment identifier considerations:
- \* 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: [GZIP]
- \* Additional information:
- \* Security considerations: See Security Considerations section of this document.
- \* Interoperability considerations:
- \* Published specification: [VCON], this document

- \* Applications that use this media type:
- \* Fragment identifier considerations:
- \* 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.

#### 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.

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
must_support	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
appended	Appended Object	IESG	Section 4.1.8 RFC XXXX
group	Group Objects array	IESG	Section 4.1.10 RFC XXXX
parties	Party Objects array	IESG	Section 4.1.11 RFC XXXX

dialog	Dialog Objects array	IESG	Section 4.1.12 RFC XXXX
analysis	Analysis Objects array	IESG	Section 4.1.13 RFC XXXX
attachments	Attachment Objects array	IESG	Section 4.1.14 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.

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
jCard	participant jCard	IESG	Section 4.2.8 RFC XXXX
gmlpos	participant geolocation	IESG	Section 4.2.9 RFC XXXX
civicaddress	civic address	IESG	Section 4.2.10 RFC XXXX
timezone	participant timezone	IESG	Section 4.2.11 RFC XXXX
uuid	participant UUID	IESG	Section 4.2.12 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.

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**

The party\_history Object Parameter Names Registry uses the Object Registry Template (Section 6.3.1) registry template. This initial values for the party\_history Object Parameter Names Registry are listed in the following 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

Table 6

## 6.3.6. Attachment Object Parameter Names Registry

TODO: type or purpose

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

Parameter Name	Parameter Description	Change Controller	Specification Document(s)
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.

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.

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. Appended Object Parameter Names Registry

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

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

#### 6.3.10. Group Object Parameter Names Registry

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

Parameter Name	Parameter Description	Change Controller	Specification Document(s)
uuid	child vCon version UUID	IESG	Section 4.6 RFC XXXX
url	referenced child vCon URL	IESG	Section 4.6 RFC XXXX
content_hash	prior version vCon hash	IESG	Section 4.6 RFC XXXX

Table 11

#### 6.4. vCon Extensions Names Registry

This document defines a 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 0.0.2 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 ("\_").

## 7.2. 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>>.
- [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. Tschafenig, "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](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](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
        BAAWZhY3QEAAAAGJMAAGRhdGH2HQAQoxjXHYXCRekuye4gK4Tsiv4wZeN
        htuvu3RMkgyPXme4NeahQLG0yeLayx5HCIiRWK7psbQrQTFm57NwQJvIb
        q1aNxk6VpMYAXSm5BblrgStIstvEzmCaOIabxs2gjReJ02Rk0BQtOxHkh
        UTIZd4VGKoqlKuoCGhrZTx1vRZICuZEcgkGjdH2jiCU1a0NMW6BE8lVHN
        -OYFPHcZspr0CSl1a02hOWUx3CbovpGqKsApWteunk8gMPM0dynRVFWR
        5o0l2BvaQDcWugGRHrQa9NcEbG1LWQq_AWLpsXKdMQefGrZu4ZE2ZSDIR
        GJLTKpIzDyBqSyZishQmjuVGztMKNlcyt2vW546JsutglllyppQWroAtHb
        kyrrGAx3bIiX8lglSR-oVaBbtKobA08fpJYscQSlHHcGm7hDriuWTRm8A
        6GKQlFbcnl0M30DcHctnEchwaAI7dxI2knYQC5biNHB2Djxm3bFmyU5TA
        3ALZt3DaSeABIKiRjcm0AxpEkSZI7AEiSJEmSwUYZlmQLYNu2bdumPWDb
        tm3b9gOAJEmSJfKfGCRJkiQZbHRhTccDtm3btm0qAEaOJEmSPABIKiRjK
        jAASJIKSZLBRhfW9CKAJEmS49gCgCRJkiQ5P2Dbtm3bZgSAJEmSJClkcm
        VNLgBIKiRjkm8ASJIKSZIvAkisJEmOswM2btu2bcMVmd3kIoDkyJEkySO
```

AJEmS3AgGgCRJkhyPC2DjuG3j9mxOWQiyEUQ1DATyvQJS9aTWWc4KuFqL  
O9F4CCJX4yEfKMaSUWSLYMkYinzGa4HzyrEi6-pgHVdXe6mugKzW7VqHY  
OzcDDsXWnJkeVXHFKpuLDZSrAroZOQohtASpE4dyZSRxozJQC6hJE2OLG  
iNYLMlruIWzsCcRhEW6c3AJhq6bmp9cZ0IvAy2a5xJVcUMNE8Qq3G0CEJ  
y30SNxwrUqUy5NczWkI3S7Uck4iyrBM3AzCvJqkisQPQkcekWjYDoJk4D  
pWiPmUzXCtKFMkuylg7MtWnrqLwMSqpjR8vXDipSKzk1D-bOjdLzgNQol  
lapqqCUmjChmctA6jRmYPOLYJtHbatocTAZCfAQ3LHs6O2zDuCC\_LyStA  
gkklFHZesUspXsSr0S5lCRYg-hXc2tKqMsgeqYkNzSVmLc9mk08zbhzUK  
O3fhkzJQsvBTCcNk2s-oUxO5zpatkETyFn-xtrxrATR0hZk3WDo5ELiFW  
Oq7hmMxgKjdcY0SO4Fk7ciJHC6EYNXEZi6oDUQz2EeJMMhlorxSOup04u  
UUjOtyjwVQ3HdhpuSRyJv0WzprTo-so7pMpdgKWUk0jLEwjiqBJAwcuw5  
lca39VznGdMLAg6o-EEenbKEQxgkes0xxEteISQcQUNPm8KxaVlxKKIq-  
BFepsm\_bjIFPLiJwpRoEoSZEuqWuBJLuuloxZC8ksQB0Wkt\_IqrwaSMq8  
p4mzEPrFksdx9i9KZVv36AvGkMmyKsHI864U2eThk-iNKRsvgRsQl1vMT  
UHBPK2ZqVvNHW3oEIay3Ea7xJGgV4mmK8Q1k004jHPCFpRzHYxBtbOEZ  
FMQeMoOVQ2F8GTOXvh9-qARmvOaH9MYaOpdTIFaS-ZJO4OGmZOvo3fEig  
5M8lQvwxinfXKhc4QrPGM0pIJthKKkixBYjvlZJlNwXTljSW8VWFex6ts  
hg7BUiVuU8VoDtlMwRBekqXknPYWWNFMrTW\_FMhsVIPv4RDESZlqtFXHV  
I3GsmHZyNFMV24h48qsHADLYeumkSKhTYG1Qp0LpWhr1UitEmaJlDxuXw  
gGEe62VbkIUkKbdkv2CFhIZMdnkebSjRKzoCG7Uc32ywDkqkauEsaZ0q  
KlvnNoKFOTloMaa5YBWwbZI\_aPtKzFMpyqx7WNBXEqt43TasSiaAKW8U  
2NSFxMuA6htzZ8ougZAbtgLVamHcwpHZgq6B40DSETuvriVpKyFIUpxog  
TQvwrBFaY5MSrRE6GFMFGqYp5L26r4Eo2kooM941UsSSRC6NVuXVfmo1  
lpueTN7sLdPY-uiHbuECmzJFAvywh-KivWrRRtmK4u4p1Va666LNhrFMt  
F3tZKlTDvgdOFlqN-zUGbWLWajishIrY5DTNLYovSDJJ2co6MCCwrxmmd  
6u0wU-CtiptRtUQohlesWTlX0I4kR-NrCEmIIFRGelq51TpDpN2jnuxFr  
UWfaop44QqksixOkeSsFwpyiIoHV4IN-NATfJboRbtIqoILbeS0SJeli  
2\_t18Xi4xizaGMTbVg3w4C6biZ8LrTot8bmqK5TWIKQWqUWloIyQhHJaA  
MTQaq4SQWNzTeTaoc5MHEkWrWHEY8moxcmLeQkQStoV9lsim7S6HlKE6n  
LKfAJK3GVq0WQeK8fmKNcascCXELWJjd9iwwNSi9mNvGsRwsvvxyHsw0O  
qqptnxsWafZojyKIyKTLXobrtrgcAoryw9tNE7TVstKaCBBmEJ2LMUsXJ  
KCCBklkWpsullZnZRgWoSnleyJ9njK9VrZXCryFoJg6ofqqAam-jOfbEv  
QIIcLJf9f-quNDsQiS9ac\_ljMEE300jQXKrmVeCxG7ie3UmIcLo4g0zVP  
FXAN5biakMaMydRuQRlJ2NuaTOEqKabgJg8hc8KNG6PLgDxn1fDGZAzRL  
OsMhnEGF6--CEZJU6ly4hJmqiXpha4Wumlis26U5hAvtMxgmpVUIzPOoF  
gpsVVLLOHiSOrKysoBq0Zx4xZZTVFJOwqozvXM7WEQtpmkq7GqF-ZKnqo  
2YBGysVvsSJPM0I1kDMHodHEkN09hm8NtJWnmga1HDhVoL2ENW6odeXkN  
2SjJCMjxotduUws-13OyDu8KtGUMMc7ICrJu6XpMSsdQkSTuWEC3ddtk1  
sBUQyktmMqA8Ti6G-crYSsVVlmFXE5VLeQOtC5kuFCrCEaqTeuJWQu662  
W5sr4KQGcrRVjLflHVdK1AV8Wt23ivgJmLsuU8TceJg\_ZdREvBliaAHKh  
czlXJmXPOahlFhvcK8lOaK1nSFjSPk8iRxghUySuliw230olEzwCfOKfz  
WqlgbFMH82iPYAk3rlNhziCdTIYblWwvYSnWdlin3TRvLwcmQhMcPLYJ2  
G0NuYpJD0KtVEqpSuaU1fJLgdp0aRynlkAzM6\_jW-6Ab1vxovqtWJTZzm  
Z3YM0gTVgRvm-zNm61DvaQaqVr1A4IUsQ8LLkMJpXarHRIxlLSNK3Akqh  
uGv2uQA6pcuSkrmCpl1ETvY4g4sgt3IZgbhUtSQ3orOouknUJ1taUPXRI  
Dbq5EVEuvvgEsQQ5jgvmTk10zaCg2G2Tm9ZazCJO3ZSLwN3M9XWerUD9y  
motaWDvHEm\_DMyQrMou9w7OsB05t1MLNLZrvBlRCyikjKSNicYUzuazIJ  
WzclQ3ziAtZVVs5a6Azhg6GYPKYGHFPaTKYO5ULdIIuIVksabFDJrRRdw

9tAxEiUvZpcIOmrAj69TTtpaJsMpgneyUu9ZtweRYk6l2zKDS5JJsq0xh  
JBfulMtcL5kscBlAcSNHUSIK9qirs87yELpvrLnOzBCKlpo5lYvGlJFUj  
IBTKTKkcU2BGhePlKqrgF1TrSljrYDcpm5KK2RSYUlnAERu3LaNXQDIce  
RIjmMASHIjSZIOAkiS5EiSgUcXl9QDgCRJkiR5B2Akx5EkqQOAJEmSJJK  
FgOQ4kiQZcJQHmbwENm7bxnHnBLZx3MarbwBIkiTJcTcAOHIkSZLBNhvW  
xiSAJEmS5DgFYBxJkiRpB4AkSZIkeTxg27Zt28ZoLB1FqA84qpop2DMPx  
B3yyHWxCEq-fmsSlw4YUjo9rZTHlFH0q2AdUZlptdIgy0RuGddqgdyotU  
Kb6kDbE4ukV4FunQyxDEpOHKgiUSHekezIka8Kys0TR3PlCFqPhrR10te  
OkYsnQK0lepQdzQALJTqbhIqAomiNxIiNQFTXZWWneXHhSPESVm4lyc1o  
EahpnMKFsBQojionS9MISi9u9dX59iNgu2gpbdb5mlatoGmpjpyqjmBt5  
mp0Za5gpVZzpUR17l9p0ggqLVIIIdvMIpMbktErBDJqS65hy7ApUjiOjsZ  
jXVtnitEDxJq5aG66ApGvxHwMugByp0ds4LmGdNmmEymntVCmzCtSY5M1  
qUBO8kizbWc8NqLsw9nW0Gs7tYlvtFaiWTVDSOGVNdFvy7EFD0zJMJMuh  
I8etAzstQSK9TqvpdEzdBMUeNm5USi3YDuxszZiSWgyyWeoovtoMyHWWq  
2ZV9lRVoMpgpZpQ1zgMgaS7rZxLzEDtpK2trKygm6OGJjhtTtkkxQyorh  
vXclcNUm6s1VLtDlhq1joy3Qo2RpWU-ZK3zs1CNSH1RJqjWGsho6Zx2oS  
sgGJIcZs6ziDMOLLqmo20XU34CDxWFA1OzAwIclQXst8MptWj3tLfC0z1  
7dx5VMYmkhLXgCU1Tg1GjMabNW7bZkNg47Zt2-bugBtDMS63bTCdKbESG  
k7cNHK5EMBNJbmW4hLKcq0pfkQL4lCtJpZcZ5lRYExhk0XS7spt4R0nLe  
TaHEJbJykFmkoBhLVwfG1lDRkprBIW0aM6dXgKONwlc6IpD-5YTcW9ySC  
6cTwpkhfWUFbUi2AfdLNotAyhFTuqiuzqQF43RreH1EHc2G2famXv3GQz  
EbaFWyWKvA54aylHF-0WaJMjq906C-JtlpSGDtaOkSItYVtKktMmrGCT1  
qEbaSxhlDZOJObUoCBLiW2dhTBZSboKOn4bOVnLCsim9Vyu6wioklaqLM  
wMRpELrXJT15RJUHwB4zRiZbPWgFzGEWlm7GCjCpoi6w7ArGdeWR1xUdV  
E4AhQkiPNbtYLqG2e6JS1CkRf\_tqZwgo4U6xWhpT3lokWzIBRsy3tVIsG  
WTSNDUusYBTn6eN4rWCcrBCVxGDNVEXBCm6R5sZ2KBO2buNHk98WYK7DX  
HwWcsQlpNVVy-YUSnKsYVypCiTXrkDsQ4TR61A2aJ02vjOIJ2ujquiZO  
-cSL4MSPGOSs3HCiZxVDdx8hbGiZS5qekUSTXiHI5Pl05N0sqA6vpMc\_u  
tIDWYcuJI7GB2uf5jpa2Alk702hJ1S50MvwjebFu4sckMmFXwNDJdC2gP  
GbeqYB24bSA5bg7XWI0ky2DTiDPkIu6A4WapZRrtoJu8rhnJy4CktZkdm  
2QuWUj0CnjWslanShG4kiTXnKkQFnggS42-DhS-8spVTOYSjjIvgZtOSi  
QV7IC9NHYYg8yAy-Sq4mduYeU4rsSaXE-dSdsI1MmiVVS8ErZp3kBO5wj  
EUZ4litUKhlYlM5QIFtPFgkxBVCWnInXNoIVYldbH66DrRk768I6gWceU  
HbhsbZ1MuBLKqaPWZtUSumrm0pKsEkSW8bZx9gzKVdvYjBkIn4lUzoBfK  
3LyJsuA6kZz3MUuQeU0DhvnbyEcW5bdqWANOuQrDbibYg2RxwykcQtFxu  
oKIuEcw871DDhunD4tzhWRiZKuWORIrt6cHWGatkIUN7RA5GRSnbYtgA2  
mUh5lce5QSPA0VIly5IjtErKQGbdlvwxM0p1FrtcmZJIdtTlS1s6NgKxA  
tkKwi3lMgWQ3umtJ7EAdasMlRayAXbdarDp5LxEpxAgMd4zJkKwMhp4iu  
TKzDKaQSkOh3w7Gkde4udP2zowSzGDuZMPuSMvAJU1VLASPgOSocix9s4  
DzYfbqOmlN2Si5CDBPTUeN3AyisASTj\_gKQrGctqdxE0QupVatDbemTDK  
KQCOVhgu562BiGc8kqZSgbmtJFpWswNs2N6N8cTEdLcAIqLKKN2nHDKay  
FL-VSAtmrgY5caoKppCbxxnRPxkzJ4IxAzLamG0dPweK2jp0WzCAUF4lmu  
GuB5PqpZGWxi-Tdu83IkRu58bvpIpTjyo49xVau11BOPq\_Mbl130pTKhu  
LbE-qCpPvrlbMoG5MWofRNa09FTtUULKkcxK1se60u4PnCSpYeriVaxor  
WtSTbT3ZK-p8kPe3OUiL2Y0mSV6vGnamuo-xkacV8jGIsCdadKA\_ipsBK  
XQamggPJeduCoXCctcQSaILpxDhsIvSyalmXrwzUEtw0g9UKYjCDNMmYF  
xWalG6BBUWGlK\_WoJooLVmiBGHik3TOO9AEUt8yriBEt1F5gjCrW17l1  
gN6E7bRoi2FDau7PaxcRMsVt5Wr9gol9VEzaBf2C4UUY5gmjI1SF2MQJU

7ZWVKrUDpFujiInlR4UivHsaQmzSNxgpGtiM9t9kSpn4bzbHvCnaKZmWq  
2AfTjebrYBCpdao7zaCoRVZr8-pgwzgwjVDtAJ0mjdKkiRBhRbsKSrXlN  
KtbD-aKFDRocwlEhiq5jtAIKHooV3bV9lCVdKSge8N14oiuIJm0ccukra  
AjR32iF89grKzNdamJcSEltQ7aTZpjxc0KIi1lx5XHCHysPSmv5gx4W20  
7Nxf4EFLybMEchVUCy8lg-kryZoEsoRzLTesIz2DuuJGTo4TWsS7EEqZq  
ZPWVzwm8cSfnTlMNZnTdRBWqLFR00FhyCwrNsgWL4WWmlaQNfAKyp7a\_o  
iJiLDOWg3KsAU5ZdmQWsYhsm3Ua6JUBian3EvYU9yIn9BBIfVv4gU81SF  
F9pZFK-RRuGcuho3tJyUIEIdS4jUsJZ2ElNbAWKI9ApOJ10yWFL91xwha  
ia6MmSbQUVIm9V4avEEExlc5up0g7a8M1AJlYnF9pUb4H7NrZay2vBJrc2  
k2XXIB0rbxGVbEHSTPLkdHVP3SiyFkrSfMuqxxSoUywTl7IKl1nkqWR3qF  
NigHFmJWfeOTZKu4OsYU5RWNWLaxG3QKsxAH0Xr5MYsgRRlrRxdeTFZKP  
QOWKXeYsHHFFSVoiuy4xLmktxOiccSTLEdt06UeF2KVMlgaJRx2oivgNS  
oT9Uo1WEamTIm9CohXLeNQMbUdBiexp2WflvG1TgayFeymZpJEhaXJbvJ  
bhY-6rEiNolq35r2NWEeGLEiFZUh7qBmzS6rAtsoEfP2qgIqN07jv62KJ  
L5mJ8xvQMXyuSqOmKTZmr0gILnRjZnWLNy3E6apVcqEmXur4e1Ezc0Mhk  
FCmGVjYezBbFiQqsRKOZqj4hX5pCtSP8sQhnUaqWlRJSqSNgF3q6VIkts  
6gbI6OvI3h4nQmlzlozQkau0juIS0Qiw7auE3tEEvYXRszuaB2jJr6u0Y  
LyWaxR4Y1iR7T2sS6u4Z3ULmGNjN21t07hS4nJs8rxUoT9HUI6Ds6-5mx  
y1BJLdNZmZPgc7IbpV6lUCqx7FkV4FwUSilCpI5kpRt8Agwbx1liMIYg  
vVuNbBCKTsXsSQmQeVyYQ0IV2XedKKrSAvabkyS-5gMV0WNe3twGuVgdy  
kiZNZTb8MQkvFX\_GvDiqZ7Jqx6Qy4kc0YieoUxlCMptKYB1WRQO\_AZunS  
7XqsYBZOkdyUruCy6A2gRc6A21apH0mB9Slu2wbkbbbohLvQWuEnhBE2qF  
qZQlXpJsBRaRZ4qzkyZ5SIjVEHJMknKsq1BI7RtG\_eKABbHbKrC7kCkWI  
dGQ4GyoUnzCjItYBipvAq8yFbt7WcMKI5j6s3iDkRSmwxGV-cOlub0oNv  
Y0poFzYBrV5kx60thYpml7IgtARLGrKsJfdFYkcIUJrrlmHrzCMLzKzV9  
xgy6Vh1Hpe0SuDYd125Nm9hzUw-hHjls3JiNQJ1Ijdu2fWbj2CcZ14pk8  
msp48awbTU3Vhr-TrTvKFICJPojf333KkjjGUaSZTV6aQ3L1U5roOuwHE  
eaE3BkySiiTaBS7LYsgZnfdoSZBqHdpK1LJ6nQXJpwFchnZ-vv8xp4cqV  
Lqa0cGpWl9XayEpCRW4nNE\_pKXYuXQZRLUfXLKmFox06kAmxhaUU1Y2rP  
QBndZc5KgVBZVccIYtqqtA3lCDpUHcdw6hTWrcLilSsKUoKsXoqS1xKRs  
q3AEDV1YXdNQVsLcavky2BUuV3TLK2A5BI-god9seFE9A6yU9tW2-oKUq  
Z0RZLGcjw5p0l25BC4kZnYadDW1ilgrKAlt9LbprSABMalGvUuQSahbhp  
JagGLRHEktWixCXhDOQRDBlSVh0mU7LYCU0RTgM\_Q7Y6EtCuUUuyS\_aO  
ldRNYTI30cl2jsDe9orzdy2Bs1qyYl1X4TtVsuuqWM4YTasUyMbarHYtE  
byJ3Bo12BWYivAUrq0sG1IcSajNGB0aV83iXSa02orMgR1LTvmbqIkV1  
bqhycBO6edpup4cqVNSBKorPfkK-kQVj4uOb2vFDR71CiSzRK2yQu9Brk  
2U5GA74Bmmc-a3Duh2wt710qKId1YaORWTKF70pIrx41S3SjKDFAtog5N  
RBOoUKRmaXYWRJGeyNbGFNphZDhQSOeQyWBtYUOViaKmq4Ds5tTbOihV  
D-ObJWqoF3EMeuQZM0kSbIUxka8to7uDEI22iqLLa1GVqUlkdkKUM7juL  
HMxtBJwoygoxXHHceqgNI0coPGksAkukX7kCphGrTuXkWBENVizxI-MSU  
nUrAIukf69lHJDDYvE0teUQ1Y1V9IUo3mzkVUz6AUtQccy8pgLSQ5BXnL  
IBYzmirr7WBhyUVr9WjuHE0rCwhyWueNtw5IbfpWqrgUNnoiaXLDCiy50  
lvsjtdQkWKLAKNekZZirKCnaDVvVu3gHN1VYX3MQLIoSq6KgTCdRPI00I  
2jusa2CmbG5uautQ5Ipl3WT64IOLVbW9PQKNcZZ8-A7Dp3pswbQdQ6PYG  
NK6HaxUDiVs2AZRnOzRaRUJ1Z6RRI0iTHyFYKRI4cOjrxEqgqospuKyO4  
VZK\_1lMokw2JQ2EolrXkqIzAI9wynWjn4FHvUqacrIC0RnZac4WPMUzNE  
CZOpSgqWw2qbCREedIMNFu0NS7rDCQRm5iOledUzULLgCLLkIOJzyDaI6  
6hO--g3DQi01zLYCJhtJ2biVhHKPYMtKkxqW7hDEp1K9Uhlg6aTE04qTY

```

    AYJ3fXApcCBWS5MWA3OSpIoePwJXFTVo2rGAVRlHst8tgbmeSpDZxUJUk
    sA6kcd4lkuUIso2UtZG9ClrW79RZ6QpmbuLiclb31E2k7CAqw45ki60A4
    6rW27Yq4D2fkhOizUDkok4SxYkyWSmyCFauWgc2xgT6dKJu7dAIqpENKQ
    bnCDROpEfXlvfUiXKlYHY3fvRXzGD8yJUHCY0A3TJtU4XvgNpMip1KbQ-
    VLPQMwoxUJlrIDDQuKyN5qg5Ek_28smIPSHotx9XY9lJF5MxAqi4uJffq
    QLS4qSZr12BVG31b7q1gmRdtnJiNcF0kyAqGNkw5KuoOyG3XXJbpCMxxZ
    zuZ9A7IjBzXzRQhk4Xi66Ciak0sa9Zak7uQJJmtIBrDQhIZDCCisukrYW
    2QWUj1CrhUGrmp9grYb-oYUuwMOG1kzJLUCMht71zQHpubADSzYJ1ftv4
    KrIGVNW4YtUphGyv65IpeYkaXigzfwRxbJK4eULOsuofVTeKhIOV7rVx6
    syVN58RFipMWLcXb7Hwa0ool1tVSbRerw1U13_R970Yst2rTUDSHiqJEk
    dDB3u8tLXXEZRi3iLuKOJJsq89gTZ4TTUs9wXJSZIUvXR-7sZUYe8uDt1
    KkJhPJMc6uUfVsadMGmupEzmqFtXHdstEeH4X3qL6u60iO3kZOOfwyo_K
    eppGj8RXzk2zV00dWW80n0QTMxLbTfSB8jmdFp02aKlNlK1197XheSYTX
    kOLGwR4faduDnqHCOE2_ZLbx6rq35oJe0p26lepuXOWTt7Ie_DmLNi4ID
    SZom4YzSEtD47tmVUra4GERBpUJACE0Uua5njuKvpK2ddy599W22r4k_U
    2rbBguUz2dY6tWrvTZuZr9sdZE7-olvZu1lUrIGj9QIfauaCtaeXRZr4k
    fW5Hlxb0fr3U21Wqt6sOSaYeyZJEIdkWdSu89fZvqdnrVNRmum_xp2jTn
    h2ISAYJiKwN0oNFtnSyD5Eiyw3dPBAY5diql0b9ecXVDBo2Rs63nWNR6t
    fuyXTmu8nIabShdNojSNGobq52IxMKBgymG60G1eX0pUNWh0AA9KxbRY
    yBjnluzH2Jc4hI2iigbuXGkTgINqqeKLVUDKTR2dQt6gpWMhzXTRd4B2B
    ULCDiSHXDgy8AnTuMqiROgCsJsYs4tGAjtlWcVdE2SATnCLSKbNhtyAJQ
    ciPHaL0Eyo0bWcX0DiZu29hwFntXxIIy4KuVckrtDCD4tGYSHctAFDmNz
    VbPQCvDauJGqTgafrCINDGcNGnZR7Zt3PoALxSCUqY0wbyrSHmTIZc"
  }
},
"analysis": [],
"attachments": [],
"uuid": "0195544a-bd15-8960-b9a2-279e0d16bc46"
}

```

## A.2. Text Chat vCon

TODO: text vCon example

## A.3. 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](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: "Helvetica Neue"; Helvetica,
        Arial, sans-serif; font-size: 16px;\">>Alice:</div><div di
        r=\"ltr\" style=\"font-family: "Helvetica Neue"
        , Helvetica, Arial, sans-serif; font-size: 16px;\">>Please
        find the image attached.<br></div><div dir=\"ltr\" style
        =\"font-family: "Helvetica Neue"; Helvetica, Ar
        ial, sans-serif; font-size: 16px;\">><br></div><div dir=\"
        ltr\" style=\"font-family: "Helvetica Neue"; He
        lvetica, Arial, sans-serif; font-size: 16px;\">>Regards,</
        div><div dir=\"ltr\" style=\"font-family: "Helvetica
        Neue"; 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\nniVBORw0KGgoAAAANSUUEGAAACYAAA
AeCAIAAADPdvS5AAAAA3NCSVQICAjb4U/gAAAAGXRFWHRT\r\nnb2Z0d2F
yZQBnbnm9tZS1zY3JlZW5zaG907wO/PgAAALpJREFUSInl1t0NgzAMBOC7
qPvPlgW6Qjdo\r\nnj4f0J4qAJI6NROvniE8+mwCvtzv6imTnyf1K/UclH
UFWjIvaIL3CHCABSPKKtJckWfY6zw+sJ5dq\r\nISdVizmj2klzTZG2Rm
e7NKgOwY6qPrMcUs+2PqchL14P+oyz+fHx77K5Sv8xyxAyZ0uBaxn/SpC
U\r\nn8NxcosAut1Y3NtjVX7UjZlmpgWTSFyvVQPKB16tSqW53bFnKtyyZ
vYT3rSsAWADWAlOjseYFfwAA\r\nnAABJRU5ErkJggg==\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=\"-----_P
art_390234_811845464.1663969499935\"",
  "encoding": "none",
  "body": "-----_Part_390234_811845464.1663969499935\r\nCon
tent-Type: text/plain; charset=UTF-8\r\nContent-Transfer-
Encoding: 7bit\r\n\r\nHi Bob:Got it. Thanks.\r\n\r\n
On Friday, September 23, 2022, 9:44:40 PM UTC, Bob <b@exa
mple.com> wrote: \r\n\r\nAlice:Please find the image a
ttached.\r\n\r\nRegards,Bob\r\n\r\n\r\n-----_Part_390234_
811845464.1663969499935\r\nContent-Type: text/html; chars
et=UTF-8\r\nContent-Transfer-Encoding: 7bit\r\n\r\n<html>
<head></head><body><div class=\"ydp89e479a0yahoo-style-wr
ap\" style=\"font-family:Helvetica Neue, Helvetica, Arial
, sans-serif;font-size:16px;\"><div></div>\r\n
<div><div dir=\"ltr\" style=\"font-family: Helvetica Ne
ue, Helvetica, Arial, sans-serif; font-size: 16px;\">Hi B
ob:</div><div dir=\"ltr\" style=\"font-family: Helvetic
a Neue, Helvetica, Arial, sans-serif; font-size: 16px;\">Go
t 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', Helvetic

```

```

    a, 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, Bo
b &lt;b@example.com>; wrote:\r\n                </div>\r\n
\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;\n\" clas
s=\"ydpdc5150a3yiv4578005613yahoo-style-wrap\"><div dir=\"
ltr\"><div><div dir=\"ltr\" style=\"font-family:Helvetic
a, Arial, sans-serif;font-size:16px;\n\">Alice:</div><div d
ir=\"ltr\" style=\"font-family:Helvetica, Arial, sans-ser
if;font-size:16px;\n\">Please find the image attached.<br><
/div><div dir=\"ltr\" style=\"font-family:Helvetica, Aria
l, sans-serif;font-size:16px;\n\"><br></div><div dir=\"ltr\"
\" style=\"font-family:Helvetica, Arial, sans-serif;font-s
ize:16px;\n\">Regards,</div><div dir=\"ltr\" style=\"font-f
amily:Helvetica, Arial, sans-serif;font-size:16px;\n\">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.4. 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](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.5. 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](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": "0195544a-b9b1-8ee4-b9a2-279e0d16bc46"
}
```

#### A.6. 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](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": "1a5edb19-c3a3-4a48-9b0a-17da2ee59f3a",
          "sha256": "04dc074100461f5082f2a7a286d0161f0e2728025e83
            c20592aa7ea3724c2933",
          "created": "2025-03-02T00:39:07.666Z",
          "duration": 33.119938,
          "channels": 1,
          "models": [
            "33063c07-ad0b-41de-b393-071dbe2b35da"
          ],
          "model_info": {
            "33063c07-ad0b-41de-b393-071dbe2b35da": {
              "name": "2-meeting-nova",
              "version": "2024-02-06.3141",
              "arch": "nova-2"
            }
          }
        }
      },
      "results": {
```

```
"channels": [
  {
    "alternatives": [
      {
        "transcript": "Hello. This is example.com. My name is Bob. How can I help you? Hi. I'd like to add Sue 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 I can help you with? No. Thank you. Goodbye. Goodbye. Have a nice day.",
        "confidence": 1,
        "words": [
          {
            "word": "hello",
            "start": 1.1999999,
            "end": 1.6999999,
            "confidence": 0.98950195,
            "speaker": 0,
            "speaker_confidence": 0.34591234,
            "punctuated_word": "Hello."
          },
          {
            "word": "this",
            "start": 1.92,
            "end": 2.08,
            "confidence": 0.9980469,
            "speaker": 0,
            "speaker_confidence": 0.34591234,
            "punctuated_word": "This"
          },
          {
            "word": "is",
            "start": 2.08,
            "end": 2.32,
            "confidence": 0.9838867,
            "speaker": 0,
            "speaker_confidence": 0.34591234,
            "punctuated_word": "is"
          },
          {
            "word": "example.com",
            "start": 2.32,
            "end": 3.28,
            "confidence": 0.95963544,
            "speaker": 0,
```

```
"speaker_confidence": 0.34591234,
"punctuated_word": "example.com."
},
{
  "word": "my",
  "start": 3.28,
  "end": 3.4399998,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.66926944,
  "punctuated_word": "My"
},
{
  "word": "name",
  "start": 3.4399998,
  "end": 3.6,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.66926944,
  "punctuated_word": "name"
},
{
  "word": "is",
  "start": 3.6,
  "end": 3.84,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.66926944,
  "punctuated_word": "is"
},
{
  "word": "bob",
  "start": 3.84,
  "end": 4.24,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.66926944,
  "punctuated_word": "Bob."
},
{
  "word": "how",
  "start": 4.24,
  "end": 4.4,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.66926944,
  "punctuated_word": "How"
},
},
```

```
{
  "word": "can",
  "start": 4.4,
  "end": 4.56,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.66926944,
  "punctuated_word": "can"
},
{
  "word": "i",
  "start": 4.56,
  "end": 4.72,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.66926944,
  "punctuated_word": "I"
},
{
  "word": "help",
  "start": 4.72,
  "end": 4.88,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.66926944,
  "punctuated_word": "help"
},
{
  "word": "you",
  "start": 4.88,
  "end": 5.38,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.66926944,
  "punctuated_word": "you?"
},
{
  "word": "hi",
  "start": 5.68,
  "end": 6.18,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.24804765,
  "punctuated_word": "Hi."
},
{
  "word": "i'd",
  "start": 6.3199997,
```

```
"end": 6.56,
"confidence": 0.951416,
"speaker": 0,
"speaker_confidence": 0.24804765,
"punctuated_word": "I'd"
},
{
  "word": "like",
  "start": 6.56,
  "end": 6.7999997,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.24804765,
  "punctuated_word": "like"
},
{
  "word": "to",
  "start": 6.7999997,
  "end": 6.96,
  "confidence": 0.99902344,
  "speaker": 0,
  "speaker_confidence": 0.24804765,
  "punctuated_word": "to"
},
{
  "word": "add",
  "start": 6.96,
  "end": 7.3599997,
  "confidence": 0.9707031,
  "speaker": 0,
  "speaker_confidence": 0.24804765,
  "punctuated_word": "add"
},
{
  "word": "sue",
  "start": 7.3599997,
  "end": 7.68,
  "confidence": 0.33374023,
  "speaker": 0,
  "speaker_confidence": 0.24804765,
  "punctuated_word": "Sue"
},
{
  "word": "to",
  "start": 7.68,
  "end": 7.9199996,
  "confidence": 1,
  "speaker": 0,
```

```
"speaker_confidence": 0.24804765,
"punctuated_word": "to"
},
{
  "word": "my",
  "start": 7.9199996,
  "end": 8.08,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.24804765,
  "punctuated_word": "my"
},
{
  "word": "service",
  "start": 8.08,
  "end": 8.58,
  "confidence": 0.982666,
  "speaker": 0,
  "speaker_confidence": 0.24804765,
  "punctuated_word": "service."
},
{
  "word": "okay",
  "start": 9.2,
  "end": 9.7,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.63695174,
  "punctuated_word": "Okay."
},
{
  "word": "can",
  "start": 10,
  "end": 10.24,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.63695174,
  "punctuated_word": "Can"
},
{
  "word": "you",
  "start": 10.24,
  "end": 10.4,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.63695174,
  "punctuated_word": "you"
},
},
```

```
{
  "word": "give",
  "start": 10.4,
  "end": 10.639999,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.63695174,
  "punctuated_word": "give"
},
{
  "word": "me",
  "start": 10.639999,
  "end": 10.719999,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.63695174,
  "punctuated_word": "me"
},
{
  "word": "your",
  "start": 10.719999,
  "end": 10.96,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.63695174,
  "punctuated_word": "your"
},
{
  "word": "name",
  "start": 10.96,
  "end": 11.12,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.4344439,
  "punctuated_word": "name"
},
{
  "word": "and",
  "start": 11.12,
  "end": 11.28,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.4344439,
  "punctuated_word": "and"
},
{
  "word": "phone",
  "start": 11.28,
```

```
"end": 11.5199995,
"confidence": 1,
"speaker": 0,
"speaker_confidence": 0.4344439,
"punctuated_word": "phone"
},
{
  "word": "number",
  "start": 11.5199995,
  "end": 11.759999,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.4344439,
  "punctuated_word": "number"
},
{
  "word": "so",
  "start": 11.759999,
  "end": 11.92,
  "confidence": 0.89746094,
  "speaker": 0,
  "speaker_confidence": 0.4344439,
  "punctuated_word": "so"
},
{
  "word": "i",
  "start": 11.92,
  "end": 12.08,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.4344439,
  "punctuated_word": "I"
},
{
  "word": "can",
  "start": 12.08,
  "end": 12.24,
  "confidence": 0.99609375,
  "speaker": 0,
  "speaker_confidence": 0.4344439,
  "punctuated_word": "can"
},
{
  "word": "look",
  "start": 12.24,
  "end": 12.48,
  "confidence": 0.99902344,
  "speaker": 0,
```

```
    "speaker_confidence": 0.4344439,
    "punctuated_word": "look"
  },
  {
    "word": "up",
    "start": 12.48,
    "end": 12.559999,
    "confidence": 0.99609375,
    "speaker": 0,
    "speaker_confidence": 0.4344439,
    "punctuated_word": "up"
  },
  {
    "word": "your",
    "start": 12.559999,
    "end": 12.799999,
    "confidence": 0.9980469,
    "speaker": 0,
    "speaker_confidence": 0.4344439,
    "punctuated_word": "your"
  },
  {
    "word": "account",
    "start": 12.799999,
    "end": 13.299999,
    "confidence": 0.99902344,
    "speaker": 0,
    "speaker_confidence": 0.4344439,
    "punctuated_word": "account?"
  },
  {
    "word": "my",
    "start": 13.815,
    "end": 13.974999,
    "confidence": 1,
    "speaker": 0,
    "speaker_confidence": 0.36550593,
    "punctuated_word": "My"
  },
  {
    "word": "name",
    "start": 13.974999,
    "end": 14.054999,
    "confidence": 1,
    "speaker": 0,
    "speaker_confidence": 0.36550593,
    "punctuated_word": "name"
  },
  },
```

```
{
  "word": "is",
  "start": 14.054999,
  "end": 14.455,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.36550593,
  "punctuated_word": "is"
},
{
  "word": "alice",
  "start": 14.455,
  "end": 14.955,
  "confidence": 0.9506836,
  "speaker": 0,
  "speaker_confidence": 0.36550593,
  "punctuated_word": "Alice,"
},
{
  "word": "and",
  "start": 15.014999,
  "end": 15.335,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.36550593,
  "punctuated_word": "and"
},
{
  "word": "my",
  "start": 15.335,
  "end": 15.495,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.36550593,
  "punctuated_word": "my"
},
{
  "word": "number",
  "start": 15.495,
  "end": 15.8949995,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.36550593,
  "punctuated_word": "number"
},
{
  "word": "is",
  "start": 15.8949995,
```

```
"end": 16.395,
"confidence": 1,
"speaker": 0,
"speaker_confidence": 0.36550593,
"punctuated_word": "is"
},
{
  "word": "+1",
  "start": 16.455,
  "end": 16.775,
  "confidence": 0.99902344,
  "speaker": 0,
  "speaker_confidence": 0.36550593,
  "punctuated_word": "+1"
},
{
  "word": "234567890",
  "start": 16.775,
  "end": 20.474998,
  "confidence": 0.9998915,
  "speaker": 0,
  "speaker_confidence": 0.36550593,
  "punctuated_word": "234567890."
},
{
  "word": "thank",
  "start": 20.775,
  "end": 21.015,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.69241047,
  "punctuated_word": "Thank"
},
{
  "word": "you",
  "start": 21.015,
  "end": 21.515,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.69241047,
  "punctuated_word": "you."
},
{
  "word": "okay",
  "start": 23.015,
  "end": 23.515,
  "confidence": 0.99853516,
  "speaker": 0,
```

```
"speaker_confidence": 0.69241047,
"punctuated_word": "Okay."
},
{
  "word": "i've",
  "start": 24.214874,
  "end": 24.454874,
  "confidence": 0.9946289,
  "speaker": 0,
  "speaker_confidence": 0.69241047,
  "punctuated_word": "I've"
},
{
  "word": "added",
  "start": 24.454874,
  "end": 24.854876,
  "confidence": 0.98828125,
  "speaker": 0,
  "speaker_confidence": 0.69241047,
  "punctuated_word": "added"
},
{
  "word": "two",
  "start": 24.854876,
  "end": 25.014875,
  "confidence": 0.41625977,
  "speaker": 0,
  "speaker_confidence": 0.69241047,
  "punctuated_word": "two"
},
{
  "word": "to",
  "start": 25.014875,
  "end": 25.094875,
  "confidence": 0.6845703,
  "speaker": 0,
  "speaker_confidence": 0.54776955,
  "punctuated_word": "to"
},
{
  "word": "your",
  "start": 25.094875,
  "end": 25.334875,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.54776955,
  "punctuated_word": "your"
},
},
```

```
{
  "word": "service",
  "start": 25.334875,
  "end": 25.654875,
  "confidence": 0.9885254,
  "speaker": 0,
  "speaker_confidence": 0.54776955,
  "punctuated_word": "service."
},
{
  "word": "is",
  "start": 25.654875,
  "end": 25.814875,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.54776955,
  "punctuated_word": "Is"
},
{
  "word": "there",
  "start": 25.814875,
  "end": 25.974874,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.54776955,
  "punctuated_word": "there"
},
{
  "word": "anything",
  "start": 25.974874,
  "end": 26.294874,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.54776955,
  "punctuated_word": "anything"
},
{
  "word": "else",
  "start": 26.294874,
  "end": 26.534874,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.54776955,
  "punctuated_word": "else"
},
{
  "word": "i",
  "start": 26.534874,
```

```
"end": 26.614876,
"confidence": 0.8144531,
"speaker": 0,
"speaker_confidence": 0.54776955,
"punctuated_word": "I"
},
{
  "word": "can",
  "start": 26.614876,
  "end": 26.854874,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.25460207,
  "punctuated_word": "can"
},
{
  "word": "help",
  "start": 26.854874,
  "end": 27.014875,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.25460207,
  "punctuated_word": "help"
},
{
  "word": "you",
  "start": 27.014875,
  "end": 27.174875,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.25460207,
  "punctuated_word": "you"
},
{
  "word": "with",
  "start": 27.174875,
  "end": 27.674875,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.25460207,
  "punctuated_word": "with?"
},
{
  "word": "no",
  "start": 28.054874,
  "end": 28.454874,
  "confidence": 1,
  "speaker": 0,
```

```
"speaker_confidence": 0.25460207,
"punctuated_word": "No."
},
{
  "word": "thank",
  "start": 28.454874,
  "end": 28.774876,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.65151024,
  "punctuated_word": "Thank"
},
{
  "word": "you",
  "start": 28.774876,
  "end": 29.254875,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.65151024,
  "punctuated_word": "you."
},
{
  "word": "goodbye",
  "start": 29.254875,
  "end": 29.754875,
  "confidence": 0.9980469,
  "speaker": 0,
  "speaker_confidence": 0.65151024,
  "punctuated_word": "Goodbye."
},
{
  "word": "goodbye",
  "start": 30.374874,
  "end": 30.874874,
  "confidence": 0.9995117,
  "speaker": 0,
  "speaker_confidence": 0.65151024,
  "punctuated_word": "Goodbye."
},
{
  "word": "have",
  "start": 31.014874,
  "end": 31.254875,
  "confidence": 0.99902344,
  "speaker": 0,
  "speaker_confidence": 0.65151024,
  "punctuated_word": "Have"
},
},
```

```
{
  "word": "a",
  "start": 31.254875,
  "end": 31.334875,
  "confidence": 0.99902344,
  "speaker": 0,
  "speaker_confidence": 0.65151024,
  "punctuated_word": "a"
},
{
  "word": "nice",
  "start": 31.334875,
  "end": 31.574875,
  "confidence": 1,
  "speaker": 0,
  "speaker_confidence": 0.65151024,
  "punctuated_word": "nice"
},
{
  "word": "day",
  "start": 31.574875,
  "end": 32.074875,
  "confidence": 0.9995117,
  "speaker": 0,
  "speaker_confidence": 0.20106936,
  "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 Sue 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 I can help you with? No. Thank you.\n\nGoodbye. Goodbye. Have a nice day.",
  "paragraphs": [
    {
      "sentences": [
        {
          "text": "Hello.",
          "start": 1.1999999,
          "end": 1.6999999
        },
        {

```

```
      "text": "This is example.com.",
      "start": 1.92,
      "end": 3.28
    },
    {
      "text": "My name is Bob.",
      "start": 3.28,
      "end": 4.24
    },
    {
      "text": "How can I help you?",
      "start": 4.24,
      "end": 5.38
    },
    {
      "text": "Hi.",
      "start": 5.68,
      "end": 6.18
    }
  ],
  "speaker": 0,
  "num_words": 14,
  "start": 1.1999999,
  "end": 6.18
},
{
  "sentences": [
    {
      "text": "I'd like to add Sue to my service.",
      "start": 6.3199997,
      "end": 8.58
    },
    {
      "text": "Okay.",
      "start": 9.2,
      "end": 9.7
    },
    {
      "text": "Can you give me your name and phone number so I can look up your account?",
      "start": 10,
      "end": 13.299999
    },
    {
      "text": "My name is Alice, and my number is +1 234567890.",

```

```
        "start": 13.815,
        "end": 20.474998
      },
      {
        "text": "Thank you.",
        "start": 20.775,
        "end": 21.515
      }
    ],
    "speaker": 0,
    "num_words": 37,
    "start": 6.3199997,
    "end": 21.515
  },
  {
    "sentences": [
      {
        "text": "Okay.",
        "start": 23.015,
        "end": 23.515
      },
      {
        "text": "I've added two to your servi
          ce.",
        "start": 24.214874,
        "end": 25.654875
      },
      {
        "text": "Is there anything else I can
          help you with?",
        "start": 25.654875,
        "end": 27.674875
      },
      {
        "text": "No.",
        "start": 28.054874,
        "end": 28.454874
      },
      {
        "text": "Thank you.",
        "start": 28.454874,
        "end": 29.254875
      }
    ],
    "speaker": 0,
    "num_words": 19,
    "start": 23.015,
    "end": 29.254875
  }
]
```

```

    },
    {
      "sentences": [
        {
          "text": "Goodbye.",
          "start": 29.254875,
          "end": 29.754875
        },
        {
          "text": "Goodbye.",
          "start": 30.374874,
          "end": 30.874874
        },
        {
          "text": "Have a nice day.",
          "start": 31.014874,
          "end": 32.074875
        }
      ],
      "speaker": 0,
      "num_words": 6,
      "start": 29.254875,
      "end": 32.074875
    }
  ]
}

]
}
]
}
]
}
},
"encoding": "json",
"vendor": "deepgram",
"schema": "deepgram_prerecorded",
"product": "transcription"
}
],
"attachments": [],
"uuid": "0195544a-b9b1-8ee4-b9a2-279e0d16bc46"
}

```

#### A.7. Signed vCon

This example vCon is the signed form of the Two Party Call vCon With Externally Referenced Recording (Appendix A.5) example. The private key used to sign this can be found at:

[https://raw.githubusercontent.com/vcon-dev/vcon/main/certs/fake\\_grp.key](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_call_ext_rec_signed.vcon)

```
{
  "payload": "eyJ2Y29uIjojMC4wLjIiLCJwYXJ0aWVzIjpbeyJ0ZWwiOiIrMTIzNDU2Nzg5MDEiLCJyZW11IjojQWxpY2UifSx7InRlbCI6IisxOTg3NjU0MzIxMCIsm5hbWUiOiJCb2IifV0sImRpYWxvZyI6W3sidHlwZSI6InJlY29yZGluZyIsInN0YXJ0IjojImJAYmI0wNi0yMVQxNz0lMzoyNi4wMDArMDA6MDAiLCJkdXJhdGlvbiI6MzMzMjIiLCJmcmwiOiJodHRwczovL2d0pdGh1Yi5jb20vaWV0Zi13Zy12Y29uL2RyYWZ0LW1ldGYtdmNvbi12Y29uLWNVbnRhaW5lci9yYXcvcmVmcj9oZWZkcy9tYWluL2V4YW1wbGVzL2FiX2NhbgwubXAzIiwibWVkaWF0eXB1IjojYXVkaW8veCltcDMiLCJmaWxlbmFtZSI6ImFix2NhbgwubXAzIiwidGVudF9oYXNoIjoic2hhNTEyLUdMeTZJUGFJVU0xR3F6WnFmSVBabFdqYURzTmdOdlpNMGlDT05OVGhuSDBhNzVmaFVNNmNZekxANud5blNVULJFdlp3bU9oNTQtMmxSUmlleWo4MlVRInldLCJhbmFseXNpcyI6W10sImF0dGFjaG1lbnRzIjpbXSwidXVpZCI6IjAxOTU1NDRhLWI5YjEtOGVlNCliOWEyLTI3OWUwZDE2YmM0NiJ9",
  "signatures": [
    {
      "header": {
        "x5c": [
          "MIID0TCCArkCAQEwDQYJKoZIhvcNAQELBQAwgasxCzAJBgNVBAYTAlVTMQswCQYDVQQLIDAJNQTERRMA8GA1UEBwwIRmFrZXZXRvd24xETAPBgNVBAoMCEZha2VWY29uMREwDwYDVQQLDAhEaXZpc2ljbjeYMBYGA1UEAwwPZG12LmZha2V2Y29uLm1vMSEwHwYJKoZIhvcNAQkBFhJhZG1pbkBMWtldmNvbi5vcmcxGTAXBgNVHREMEGRpdj5mYWtldmNvbi5vcmcwHhcNMjIwMDE0MzI1WcNMZi1wNjAxMDE0MzI1WjCBsDELMAKGA1UEBhMCVVMxCzAJBgNVBAGMAk1BMREwDwYDVQQLDAhGYWtldG93bjERMA8GA1UECgwIRmFrZXZjb24xZjAMBgNVBASMBUdyb3VwMRwwGgYDVQQDDBNncnAuZG12LmZha2V2Y29uLm1vMSEwHwYJKoZIhvcNAQkBFhJhZG1pbkBMWtldmNvbi5vcmcxHTABBgNVHREMFkdycC5kaXYuZmFrZXZjb24ub3JnMIIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAtMtTCxeIROb1FmWlIexe7Op95qYIVNEqn+fDghxWqoYm6T75BGsl4rB9txFQeef3Q5SJIZEqffJDssXDSyREuwQ2iTcOrZaZxs8h1NX9NjVD6MMAwx7tw0Cgmu8rj6+/Kjs0KI8OzCuDXZGTGf/bH6EQW85qYh9IF0PviJf3uq+HCke6WApusx76LpUp36IzvYEuTbWBFeoLo+dyUdWmXpj0ghHiWZtT5zxJn9VrUhJX5KlaI0palWd0fOnr7mkdq79sFsKjEOcrllanceAicCpD3/zRW+7AS8atGKAMfe2CcXEABucI7DttLMXCDIjME0kyW7n4duelRXGuNgnjeQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQAQLjhKomSx1OqThhc0+aK0rzX6RjKh0eM0X707pPC81HxZKCDLhDvnUO9++17U6ba0bTznzKYKKG7HnDcB4dKBbH8XP2u3E nDskyoMrKpYRsRKwwJa8+RwOS7JEg+OtXYCRbWZvPjAC/z6o6OZK+
```

QPTyK5Grw2Px2mTr6BAdBQv7XsYWBRwOZyvSGnKyK60jcPlyUnu6Q  
t0fYXiYp6qF5l956fuT0d2UyDkM7S/Z7K+OC9+aIBNKH32B3nfXqZ  
PuP8WuitOxcMwnefPwjxaqndFUkyk2Y+i6nXU8oN/DGUOTfY4xbm3  
s3q7Dq5QPtq5kljB9yYkvcOULCSAmWBZl+G" ,  
"MIIEyTCCArECAQEwDQYJKoZIhvcNAQELBQAwwgAgCzAJBgNVBAYTA1  
VTMQswCQYDVQQIDAJNQTERTMA8GA1UEBwwIRmFrZXRVd24xFTATBgN  
VBAoMDEZha2VWY29uIEluYzESMBAGA1UECwwJQ29ycG9yYXRlMRQw  
EgYDVQQDDAtmYWtldmNvbi5pbzEhMB8GCSqGSIb3DQEJARYSYWRta  
W5AZmFrZXZjb24ub3JnMRUwEwYDVRODAXmYWtldmNvbi5vcmcwHh  
cNMjIwNTI1MjEzOTMxWhcNMzIwNTI1MjEzOTMxWjCBqzELMAkGA1U  
EBhMCVVMxCzAJBgNVBAGMAk1BMREwDwYDVQQHDAhGYWtldG93bjER  
MA8GA1UECgwIRmFrZXZjb24xETAPBgNVBASMCERpdmlzaW9uMRgwF  
gYDVQQDDA9kaXYuZmFrZXZjb24uaW8xITAFBgkqhkiG9w0BCQEWEm  
FkbWluQGZha2V2Y29uLm9yZzEZMBcGA1UdEQwQZG12LmZha2V2Y29  
uLm9yZzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAM81  
SKie1S78VAOBDC3LflYNSL9phcmjQs/9P+iqDXSVm3xpoqsZGoLn  
7w21aUAxl++F9Z80Tc70URN2qqUcVTYcx3SmgHuYgspEmJsMRor/m  
n3HpHr2ajh+26RKgDGPBrtebeH1si8mX15a/ICzHxrUaE2Y/6DZ/2  
wGTxRAE4bYAOuaYft5GGSZl6yP6dXg3EJy2bK2cUTdMIZG/Y7Ny1D  
3N4V7MBirDd5FU7k6pMbyr3d/xoi8VL8F1mRCf6zxfFOjamFT0ate  
iTp3M0rw6iS08hy0cDFsPJzIcPgXEitUgGPo/NLgeb8ZmL1kg9B2v  
MgaanbuZB6il79A4GgDIUCAwEAATANBgkqhkiG9w0BAQsFAAOCAG  
AWOJlJlvsQDsDROQZNDX3kMkAH3HYyAlE7LHNTIYHVhyt7I7WuqS6Z  
GvSoxSfkt0GdAN7cV+jchdZpAnpCXGX+yHuSvrw4oSqigdSLeSUMx  
2zaSv4wLAvydC/zDKHIRObyuDEKyy/pa4Na0aunPU3E1Bg0V+JqQ3  
RUJIylv6AGLrliwaEIgP3Clzi/u640dWBHABOEjxFj9bX73+FRWhi  
t4EzOJYKvti1EXlrmjMbj3SmsrcbGMrpu4WipeG5S05VlMhXXSYQq  
9IScFho5CftamDgI3jL49lyrx0+CcjDkw0nW9fXdAtB3BhSYEXdLu  
JAvLRQixUQFwH23douwPiluW8euEhHTRolpEFWQ5U59NOTUA5Ghuu  
po0aM2CnDD5Re8u03ChilCub0FlwgMD/7ABSd1LChU8iluhKy9CEB  
R2zc8uTqW43Gcv/psBaejgOKJlxZuusGVvuVlnTHRa4hmFYxxPel7  
ogeZzGQqfjBlb35CE+BcGYY0ViaHefy+fmi/fN6NrrN6pBOPjVZbp  
CFn+wFTUyrMU5Vq/FaPe0us7i3paQcwIz9wyide5OztpxebY8UnXm  
y22X+6PkHB0jnL35Qe8xl7Z0nMZNdy4xkzGgaa6talRDPPhKkY1lId  
w+KdCXJuUc7rkLKa5po9SPdvVLhEPu26SHki0I+d5hz+gMp8=" ,  
"MIIGMzCCBBugAwIBAgIUHk/6QbzjDZ9AONAFvWYJqB/jbvowDQYJKo  
ZIhvcNAQELBQAwwgAgCzAJBgNVBAYTA1VTMQswCQYDVQQIDAJNQTERT  
MA8GA1UEBwwIRmFrZXRVd24xFTATBgNVBAoMDEZha2VWY29uIElu  
YzESMBAGA1UECwwJQ29ycG9yYXRlMRQwEgYDVQQDDAtmYWtldmNvbi  
5pbzEhMB8GCSqGSIb3DQEJARYSYWRtaW5AZmFrZXZjb24ub3JnMRUw  
EwYDVRODAXmYWtldmNvbi5vcmcwHhcnNMjIwNTI1MjEzOTMxWhcNMz  
IwNTI1MjEzOTMxWjCBqDELMAkGA1UEBhMCVVMxCzAJBgNVBAGMAk1BM  
REwDwYDVQQHDAhGYWtldG93bjEVMBMGA1UECgwMRmFrZXZjb24gSW5  
jMRIWEAYDVQQQLDAlDb3Jwb3JhdGUxZDASBgNVBAMMC2Zha2V2Y29u  
Lm9yZzEZMBcGA1UdEQwQZG12LmZha2V2Y29uLm9yZzCCAIwDQYJKoZI  
hvcNAQEBBQADggIPADCCAgoCggIBAMIEaqkm8pn6mwUzWBgnQiY9Jr3V  
yVamxBc9/duT8eCN9mmANGzX5zGj86KtGjldCoAzANrtsr2Dpz4zra

```
29HWbwboalZTOOaR93QgifkS802zp+zHGGRaKlq3euDqDVzyXinhs
vXOVlEwkOrAvV6kUcTIzcxVibrFfHttzv/hXUNxjCouAE7MIG6wO8
y+E2bBkg7R5ydfE4Q9npBefwtiQ8oWf0C46mDYFQ8l3lgoael7Vp
iyS3t4h5HL+lyqdYJSWms5Qh0vp9req5AjwuHIDBxNLpO2gYlSQmS
dO03oLcZwBglGKlpQSwYgr9J5uo2B+NV9aYKE05lbySinSF8T4D/o
4cq6oyJLXBwmSsLbjjVIQBoSdRBmWh8zE2ZzFNaaBLYO9ikIaWXwb
tkbsE+4QpOaOj/kUIHLFI9iOhOidsInaOfgYzw12FtGE1Wq5x5Xsz
wWpPWU8NQw8gzTzFzVObxrfcsXrs0s6feuTCDMt5Zht2HCWFr9TmY
DFAfOB05kFWTi5sE5KYaP54iJvjAFmuTDjlif7Z8ZWVal9XB+g3sM
yQ8DVEj/0dXS+nRVuJQlUH4ql9SUIwK63RKdedALYC6X1j2lgK9aT
uL+cyNJ5SbuNA95CxNgvWtBrC06HxlbmIDxKrAAo5aj4hnlQcY4RR
bNwppCAjK+zcASlpyvRAGMBAAGjUzBRMB0GA1UdDgQWBBSsdTfqU6
+DEyYPCar08yVnnZSFzTafBgNVHSMEGDAWgBSSDtfqU6+DEyYPCar
08yVnnZSFzTAPBgNVHRMBAf8EBTADAQH/MA0GCSqGSIb3DQEBCwUA
A4ICAQC8JDxTdmC86YFdKElWQnWxoPMsPeRyDYQoVTC2wydZsdJ5d
GbKAhYlPPxK+zypP18fGry5PHAH19mZYZ+FdGjNTwllcfN9/G0Sdo
y4kU9eN9KfC9OUiKBIq8D0r7ANT3l3XmGbnIxWJEtB0SyTcqk3dwW
cNTqGguhQ03jUKW5yUpVoQjN4TtmHyweoYTNi+HKsCUvEXGdA9wdh
4jVOqYeepmsax5eTUJskUcKqQ3+xB7JONw85imfVotySCXsoMe7lu
SRyuldn8SBIPkYUGJlpWii6Gr0mLwHarrYoi9bwtcKAXRNXXZiGQgh
iUw2Zsw+jmgBrEtI4Iarg8wWqJ5Mov7MsTVpnoZ5Xywr0GznDlgn
tyqMibOULeM8yLPQRhn/tQjSiP3eUPz6+Zb9FqzQJCCi2UiauD4mO
gzBSCCbew9PDYb2QeMZqNZmN3eVNMV4hh7t9jTc0zqPF5jluddVmp
dHxLx/T9XZuSV2lp8clNprJNqxpRtHpxDxjhNMIRcV39rzi9tNikw
mzRMf0r40f/JWnaV668sekDXtvIj8pZwyE4FNuaiJwojkQxDfjqk6
7Qs5NUQsFcaIwMes8It74ASHm8DUKEhDT2ASgMtUDwKWROBefrSPq
jY7+EMh2yVZnQhMwpX4s7CH0za32Yq+KC8GotHdI6jisDEom7aHgl
g=="
},
"alg": "RS256",
"uuid": "0195544a-b9b1-8ee4-b9a2-279e0d16bc46"
},
"signature": "K4FPWnAsriqGjrvDyIGqHd_L4Sjdjjf1YUFQJcFU79tdK
Ay9fVQ6cHRA_EgkJKgnyMKYtLgn6Gu6vYMPkfOedQuZOWkNyA4R5N15V-
8GRLxZconPJYGpXzEWVl-y_pjBKVGa7VieyLhQlWM2EbqrHLSow6D9mDA
HCUscGTcXN-ocV8DISQNNNiF9biPE5C7hhA3ZQt0g8NasZb5kWPnWTXZ7
zg9lcnO-dUq8lokyjrmDmKODtFsQKwI_IksG4Aeco0YAfi4RcZqHMBid
AafMirA0z_30yVFYVLP6RZiV8fIcCadTmuRjgrDIPjzq_pESkdYEZdgNo
p30x6Jvzn4tw",
"protected": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsIng1YyI6W
yJNSUleMFRDQ0Fya0NBuUV3RFFZSkvtWklodmNOQVFFTEJRQXdnYXN4Q03
pBskJnTlZCQVlUQWxWVE1rc3dDUVlEVlFRSURBSk5RVEVSTUE4R0ExVUV
Cd3dJUm1GclpYUnZkMjR4RVRBUEJnTlZCQW9NQ0VaaGEyVldzMjllTVJF
d0R3WURWUWFMREFoRWfYWnBjMmx2YmpFWU1CWUdBMVVFQXZ3UFpHbDJMb
VpoYTJWMLkyOXVMbWx2TVNFd0h3WUpLb1pJaHZjTkFRa0JGaEpoWkcxcG
JrQmlZV3RsZG10dmJpNXZjbWN4R1RBWEJnTlZiUkVNRUdScGRpNW1ZV3R
sZG10dmJpNXZjbWN3SGhjTk1qSXdoakEwTURFME16STFXaGNOTXpJd05q
QXhNREUwTXpJMVdqQ0JzREVMTUFR0ExVUVCaE1DVlZNeEN6QUpcZ05WQ
```

kFnTUfRmUJNNUkV3RHdZRFZRUUhEQWhHWVd0bGRHOTNiakVSTUE4R0ExVU  
VDZ3dJUmlGclpWWmpimjR4RGpBTUJnTlZCQXNNQlVkeWlZVndNUnd3R2d  
ZRFZRUUREQk5uY25BdVpHbDJMbVpoYTJWMLkyOXVMbWx2TVNFd0h3WUpL  
blpJaHZjTtkFRa0JGaEpoWkcxcGJrQm1ZV3RsZG1OdmJpNXZjbWN4SFRBY  
kJnTlZlIUkVNRkdkeWNDNWthWFl1WmlGclpYWmpimjR1YjNkbk1JSUJJaK  
FOQmdrcWhraUc5dzBCQVFFRkFBT0NBUTHBTUlJQkNnS0NBuUVBdE10VEN  
4ZUlST2IxRmlXSWV4ZTdPcDk1cVlJVk5FcW4rZkRnaHhXcW9ZbTZUNzVC  
R3NsNHJCOXR4RlFlZWYzUTVTskl6ZXfmZkpEc3NYRHN5UkVld1EyaVRjT  
3JaYVp4czhoMU5YOU5qVqQ2TU1Bd3g3dHcwQ2dtdThya jYrL0tqcZBLST  
hPekN1RFhaR1RHzi9iSDZFUVc4NXFZaDlJRjBQdmlqZjNlcStlQ2t1Nld  
BcHVzeDc2THBVcDM2SXp2WUVlVGJXQkZlb0xvK2R5VWRXbVhwa jBnaEhp  
Vlp0VDV6eEpuOVZyVWhKWDVLbGFJMHbhbFdkMGZPbnI3bWtkcTc5c0ZzS  
2pFT2NyMWxhbmNlQWl jQ3BEMy96UlcRn0FTOGF0R0tBTWZlMkNjWEVBQn  
VjSTdEdHRMTVhjRElKTTZFT2t5VzduNGR1ZTFSEWEd1TmduamVRSURBUUF  
CTUEwR0NTcUdTswlZrFFFQkN3VUFBNElCQVFBUUxqaEtvbVN4MU9xVGho  
Y08rYUswcnpYNlJqS2gwZU0wWDDPN3BQZqgxSHaS0NETGhEdm5VTzkrK  
ze3VTZiYtBiVG56bktZS0tHN0huRGNCNGRLQmJlOIFhQMnUzRW5Ec2t5b0  
lyS3BZUnNSS3d3Sme4K1J3T1M3SkVnK090WFlDUmJXWnZQakFDL3o2bzZ  
PWksrUVBUeUs1R3J3MlB4Mm1Uc jZCQWRCUXY3WHNZV0JSd09aeXZTR25L  
eUs2MGp jUGxzVW51NlF0MGZZWGlZcDZxRjVsOTU2ZnVUMGQyVXlEa003U  
y9aNoSrTOM5K2FJQk5LSDMyQjNuZlhxWlB1UDhXdw1UT3hjTXduZWZQd2  
p4YXFuZEZVa3lrMlkrATZuWfU4b04vREdVTlRmWTR4Ym0zczNxn0RxNVF  
QdHEla2xqQj15Wwt2Y09VTENTQWlXQlpsK0ciLCJNSUlFeVRDQ0FyRUNB  
UUV3RFFZSktvWklodmNOQVFFTEJRQXdnYwd4Q3pBSkJnTlZCQVlUQWxWV  
E1Rc3dDUVlEVlFRSURBSk5RVEVSTUE4R0ExVUVcd3dJUmlGclpYUnZkMj  
R4RlRBVEJnTlZCQW9NREVAaGEyVldZMjllSUVsdVl6RVNNQkFHQTFVRUN  
3d0pRMj15Y0c5eVlYUmxNULF3RWdZRFZRUUREQXRtWVd0bGRtTnZiaTVw  
YnpFaElCOEDDU3FHU0liM0RRRUpBUl1TWVdSdGFxNUFabUZyWlhaamIyN  
HVim0puTVJVd0V3WURWUjBSREF4bVlXdGxkbU52YmkldmNtY3dIaGNOTW  
pJd05USTVNakV6T1RNeFdoY05Nek13TlRJMk1qRXpPVE14V2pDQnF6RUx  
NQWtHQTFVRUJoTUNWVkl4Q3pBSkJnTlZCQWdNQWsxQk1SRXdEd1lEVlFR  
SERBaEdZV3RsZEc5M2JqRVJNQThHQTFVRUNnd0lSbUZyWlZaamIyNHhFV  
EFQqmdOVkjbC01DRVJwZG1semFXOXVNUmd3RmdZRFZRUUREQTLrYVhZdV  
ptRnJaWFPqYjI0dWFXOHhJVEFmQmdrcWhraUc5dzBCQ1FFV0VtRmtiV2x  
lUUDaaGEyVjJZMjllTG05eVp6RVpNQMNHQTFVZEVRdlFar2wyTGlaaGEy  
VjJZMjllTG05eVp6Q0NBU013RFFZSktvWklodmNOQVFFQkJRQURnZ0VQQ  
URDQ0FRb0NnZ0VCQU04MVNLaWUxUzc4VkfPQkRjM0xmbFlOU0w5cGhlY2  
lqUXMvOVARaXFEWFNwbTN4cG9xc3pHb0xuN3cyMWFVQXhsKytGOVo4MFR  
jNzBVUm4ycXFVY1ZUWwN4M1NtZ0h1WwdzcEVtSnNNUm9yL2lUm0hwSHIy  
YWpKzI2UktnREDQqNj0ZWJlSDFzaThtWDElYS9JQ3pIeHJVYUUYWS82R  
FovMndHVfhyQUU0Yl1BT1VhWWZ0NUdHU1psNnlQNmRYZzNfSknYkSyY1  
VUZE1JWkcvWTdOeTFEM040VjdNQmlyRGQ1RlU3azZwTWJ5c jNkL3hvaTh  
WTDhGMWlSQ2Y2enhmRk9qYU1GVDBhdEVpVHAzTTBydzZpUzA4aHkwY0RG  
clBkekl jUGdYRWl0VWdhUG8vTkxnZWI4WmlMMWtnOUIydklnYWFuYnVaQ  
jZpbDc5QTRHZ0RJVUNBd0VBQVRBTKJna3Foa2lHOXcwQkFRc0ZBQU9DQW  
dFQVdPSkpsdlNRRHNEUk9RWk5kWDNRtWTBsdNIWx1BbEU3TEhOdElZSFZ  
oeXQ3STdXdXFTNlpHdlNveFNma1QwR2RBTjd jVItqY2hkWnBBbnBDWEdY  
K3lIdVN2cnc0b1NxaWdkU0xlU1VNeDJ6YVN2NHdMQXZ5ZEMvekRrSElST

2J5dURFS3l5L3BhNE5hMGF1blBVM0UxQmcwVitKcVEzUlVKSXkxdjZBR0  
xybG13YUVJZ1AzQ2x6aS91NjQwZFdCSGFiT0VqeEZqOWJYNzMrRlJXaG1  
0NEV6T0pZS3Z0aTFFWGxtUmpNYmozU21zcmNiR01ycHU0V2lwRUclUzA1  
VmxNaFhYU3lRcTlJU2NGaG81Q0Z0YW1EZ0kzakw0OWx5cngwK0NjakRrd  
zBuVzlmWGRBdEizQmhTWUVYZEx1SkF2TFJRaXhVUUZ3SDIzZG91bVBpMX  
VXOGV1RWhIVFJvMXBFRldRNVU1OU5PVFVBNUdodXVwbzBhTTJDbkRENVJ  
lOHVPM0NoaWxDVWIwRmx3Z01ELzdBQlNkMUxDaFU4aWx1aEt5OUNFQ1Iy  
emM4dVRxVzQzR2N2L3BzQmFlamdPS0pseFpldXNHVnZ1VjFuVEhSYTRob  
UZZeHhQZWw3b2dlWnpHUXFGakJsYjM1Q0UrQmNHwVkwVmlhSGVmeStmbW  
kvZk42TnJyTjZwQk9QalZaYnBDRm4rd0ZUVXlyTVU1VnEvRmFQZTB1czd  
pM3BhUWN3SXo5d3lpZGU1T3p0cHhlylk4VW5YbXkyMlgrNlBrSEIwam5M  
MzVRZTh4bDdaMG5NWk5keTR4a3pHZ2FhNnRhbfJEUGhLa1kxbElkdYtLZ  
ENYSnVvYzdyaoxLYTVwbz1TUGR2VxkxRVB1MjZTSgtJMEkrZDVoeitnTX  
A4PSIsIk1JSUDNekNDQkJ1Z0F3SUJBZ0lVSGsvNlFiempEWj1BT05BRnZ  
3WUpXQi9qYnZvd0RRWUpLb1pJaHZjTkFRRUxCUUF3Z2FneEN6QUUpCZ05W  
QkFZVEFsVlRNUXN3Q1FZRfZRUU1EQUpOUVRfUk1BOEdBMVVFQnd3SVJtR  
nJaWfJ2ZDI0eEZUQVRCZ05WQkFvTURFWmhMlZXWTI5dUlFbHVZekVTTU  
JBR0ExVUVDd3dKUTI5eWNHOXlZWfJstVJRd0VnWURWUVFEREF0bv1XdGx  
kbU52Ymk1cGJ6RWhNQjhHQ1NxR1NJYjNEUUVKQVJZU1lXUnRhVzVBWm1G  
clpYWmpimjR1YjNkbnk1SVXdfD1lEVlIwUkRBeG1ZV3RsZG1OdmJpNXZjb  
WN3SGhjTklqSXdoVEk1TWpFek9UTXhXaGNOTXpJd05USTJNakV6T1RNeF  
dqQ0JxREVMtUFR0ExVUVCaE1DVlZNeEN6QUUpCZ05WQkFvTURfMUJNUkV  
3RHdZRFZRUUheQWhHWvd0bGRHOTNiakVWTUJNR0ExVUVDZ3dNUmlGclpW  
WmpimjRnU1clak1SSXdfQVlEVlFRTERBBERiM0p3YjNKAGRHVXhGREFTQ  
mdOVkJBTU1DMlpoYTJWMLkyOXVMBWx2TVNFd0h3WUpLb1pJaHZjTkFRA0  
JGAEpoWkcxcGJRQm1ZV3RsZG1OdmJpNXZjbWN4RlRBVEJnTlZiUkVNREd  
aaGEyVjJZMj1lTG05eVp6Q0NBaUl3RFFZSkvtWklodmNOQVFFQkJRQURn  
Z0lQQURDQ0Fnb0NnZ0lCQU1JZWfXa204cG42bXdVeldCZ25RaVks5SnIzV  
nlWYw14QmM5L2R1VDhlQ045bW1BTkd6WDV6R2o4Nkt0R2psRGNvQXpBTn  
J0c3IyRHB6NHpyYTI5Sfddid2JPYTFaVE9PYVI5M1FnaWZrUzgwMnpwK3p  
IR0dSYUtscTNldURxRFZ6eVhpbmhzdlhPVmxFd2tPckF2VjZrVWNUSXpj  
eFZpYnJGZkh0dHp2L2hYVU54akNvdUFFN01JRzZ3Tzh5K0UyYkRtZzdSN  
XlkZmVNVFE5bnBCZWZ3dG1ROG9XZjBDNDZtRF1GUTHsmZfNb2F1MTdWcG  
15UzN0NGg1SEwrMXlxZF1KU1dtczVRaDB2cdlyZXE1QWp3dUhJREJ4Tkkx  
wTzJnWTFtUW1TZE8wM29MY1p3QmdsR0tscFFTdl1ncj1KNXVvMkIrTlY5  
YVlLRU81MWJ5U2luU0Y4VDREL280Y3E2b3lKTFhCd21Tc0xiampWSVFCb  
1NkUkJtV2g4ekUyWnpGTmFhQmxZTzlpal0lhV1h3YnRrYnNFKzRRcE9hT2  
ovalVJSExGSTlpt2hPaWRzSW5hT2ZnWXP3MTJGdEdFMVdxNXglWHN6dlld  
wUfDVOE5RdzHnelR6RnpWT2J4cmZjc1hyczBzNmZldVRDRE10NVpodDJI  
QldGcjlUbVlERkFmT0IwNwtGV1RpNXNFNUtZYVA1NGlKdmpBRml1VERqb  
GlmN1o4WldWYwW5WEIrZzNzTXlROERWRWovMGRYUytU1ZlSlFsVUg0cW  
w5U1VJd0s2M1JLZGVkQUxZQzZYMwoybGdLOWFUDUwrY3lOSjVtYnVOQTk  
lQ3hOZ3ZXdeJyY082SHhsYmlJRhhLckFBbzVhaJRbmxRY1k0UlJiTndw  
cENBaksremNBU2xweXZSQWdNQkFBR2pVekJSTUIwR0ExVWREZ1FXQkJTU  
0R0ZnFVNitERXlZUENhcjA4eVZublpTRnpUQWZCZ05WSFNRRUdeQVdnQl  
NTRHRmcVU2K0RFeVlQQ2FyMDh5Vm5uWlNGelRBUEJnTlZiUk1CQWY4RUJ  
UQURBUUgvtUEwR0NTcUdTSWIzRFFFQkN3VUFBNE1dQVFD0EpEeFRkbWM4  
N1lGZEtFbFdrBld4b1BNc1BlUnlEWVfVlRDMnd5ZFpzZEolZEdia0FoW

```

WxQUHhLK3p5cFAxOGZHcnk1UEhBSGw5bVpZeitGZEdqTlR3bGxjZk45L0
cwU2RveTRrVTl1Tj1LZkM5T1VJa0JJcThEMHI3QU5UM2wzWG1HYm5JeFd
KRXRiMFN5VGNxazNkd1djTlRrR2d1aFEwM2pVS1c1eVVwVm9Rak40VHrt
SH13ZW9ZVE5pK0hLc0NVdkVYR2RBOXdkaDRqVx9xWWVlcG1zYXg1ZVRVS
nNLdUNLcVEzK3hCN0pPTnc4NWltZlZvdHlTQ1hzb011N2x1U1JZdWxkbj
hTQklwS1lVR0pscFdJaTZHc jBtTHdIYXJyWW9pOWJ3dGNLQXhSTlhaaUd
RZ2hpVXcyWnN3K2ptZ0JyRXRJNElhcmc4d1dxSjVNb3Y3TXNUVnBub1o1
WH13c jBHem5EbGdjTnR5cU1pYk9VTGVNOH1MUFFySG4vdFFqU2lQM2VVU
Ho2KlpiOUZxelFKQ0NpMlVpYXVENG1PZ3pCU0NDYmV3OVBWwWiyUWVNWn
FOWm1OM2VWTk1WNGhoN3Q5a1RjMHpxUEY1amx1ZGRWbXBkSHhMeC9UOVh
adVNWMjFwOGNsTnBySk5ReHBydEhweER4amhOTUlyY1YzOXJ6aTl0Tmlr
d216Uk1mMHI0MGYvSlduYVY2NjhzZWtEWHR2SWo4cFp3eUU0Rk51YW1kKd
29qa1F4RGZqcWs2N1FzNU5VUXNGY2Fjd01lczhJdDc0QVNlBThEVUtFaE
RUMkFTZ010VUR3Sl1dyT0JlZnJTUHFqWTcrRU1oMnlWWm5RaE13cHg0czd
DSDB6YTMwXErS0M4R290SGRJNmppc0Rfb203YUhnbgc9PSJdfQ"

```

```

    }
  ]
}

```

#### A.8. Encrypted vCon

This example vCon is the encrypted form of the Signed vCon (Appendix A.7) example. The private key to decrypt it can be found at:

[https://raw.githubusercontent.com/vcon-dev/vcon/main/certs/fake\\_grp.key](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](https://raw.githubusercontent.com/ietf-wg-vcon/draft-ietf-vcon-vcon-core/refs/heads/main/examples/ab_call_ext_rec_encrypted.vcon)

```

{
  "protected": "eyJhbGciOiJSU0EtT0FFUCIsImVuYyI6IkJEYNTZDQkMtSFM1M
    TIifQ",
  "iv": "_0zAYOxo66NUDUser-5jBA",
  "ciphertext": "jTPmEFYMr8W_Saab3mZAF6QOiqlijl15Y5Yqob0lnINq2OcD
    xddWx-_v_mqPv92-Z8ZGJDgxNZfv9HAT54wbmjfj5SgOrJS6nMFBruFBQU4ew
    EvvTqiG6wGIbnZG5I-KJsn2xVGhBpUkh-eWhrkNxdugy8ekg9ZpggIofIn2AD
    Fjdx9x46wS_YShj5l7EOIv130F-1M3yx9NpICwKW563161TRKKoKfDLZRRDCM
    Wpc860sDUHafPutHXasW5iWjRxq7UFRvc-_BYUoTXrrKz3wMAxwJUKFWdIv3J
    _mUQJMO1U-qq-8cgJWv-1aQWcl1p2OKQRaWy0CVn6IC_u0GQYp3sRHCYa00hu
    zUIxnL0SPJziCLWPVMUsWIEezshwfPa_MH5lJgFjkcIYaq2vkHt4r0oiop52R
    0LnV49njC2KaN_8Q-XB-If-m9tEDBLzkdnEV35HGstgXv-I_UN2FXMRL4KgHH
    -8_hmwH-jVbmFXZBqSddTYulizjQu22bHSzVFc0IV1jM1lxFd9y3Td00vIw-c
    eVWgd1hEhtjV9TmvRZR7dJk7kD1BbtMiLgdSLhYC5kM5aQPHxuklmfCDuimF1
    HmsdXx30zTXoOLQ79POTdf_7TSsR8OkVq9_NenupmpJ4DxnGcx56zPN50rHVJ
    KsjOP19UPrO__jGvpDpolx_PabwUG7TGNUwlr5eQlTXMzwjpb9AubSnrZkXz9
    1m20807cmIsf6T5KijI3rpGFRgFNdxbihrhaynRirw6H5rmLqYYVY97Lb1qt-

```

97LxSotK7orTrRWOKeWYO1dxim2PgCh3mFPKuAqEV4CkvXfOljHflNe5S78DH  
KZUmNTczfNCEs6dColMnmPSgwh2jdy4zP-9hvheDDWoHu6zEdjiP4XYGI-UFj  
zRleNzZeJcIFgjOiNH7nG6iVKp-zdkgQnE5N6U4Jj0mMwyD-gVVbotd-h2wWZ  
XpjhPeB5yuaSDIizeSs9Ndqq5Ev8XpJgu2ndDaxGAEuJNcgFRhvvGZ2rDoHbj  
bjGeiaxC5fCvczH\_zuTXMHiNwQxc4AVV5y\_9BjXfBcr-\_16JUotSvNjPMvmOP  
\_vbYAm9Rnec4\_x76DbPh3X98FQMkzAFZOKhBwQJlwyAcDCh7p9Ee2gwV-LgOm  
OBD22kuoKnf8AqLSEjLWnNdUDcKtr6EKrC7Z1OU2vIFoN85jEs2m-fxLsagwt  
ntrEpoELl19ekSd5JldU85MQfuuIwHit120-YbBAikfD9TDjFJ3YhuIG5Coai  
CHWYGwEB6rOeL0MzEpcnQATK\_HySImse5Z62rd7UL1hZQ7psglIdtMIblmcsZ  
mXc3xqfQer3dorNU3-pQUWi68Ggf0BPGNe4mDtJeNH9f-Un\_nhsT835THalrI  
0pknzTrKoU3ep7gIFdvunlevf4Po664sqVB\_obt7BFmqMfaX3FfoAZpbA5CHT  
JVdKEmAwBm\_pxxJRfEQCzzOLLx-ti5-QRCehXLX98Q7sJtIXvN2wf-WKjsycb  
\_UkXDm\_iRBuaub3lfsj\_VS9ur2oLIh0wqYGfTTEqcGCmz2bHf6yu7o2dK7Y8d  
w4Rc0ViwCU0--C1XI-XMGEPFC-CYdkQMakTpvgeZy8JDbyAZMdeX-e4zleuBQ  
7dr36zFlFA-hqptPv51qswcKZU-7iQd02a9ANKeYVKuXqcB5Iw8x2JlWQf9cA  
4qIBj33YUXfEgo\_DSQLRqPz4-vP97cu0JahoEQDI2rSJghCwiO585Gu7QePSL  
EFRhGQubQIf5iWEa1WZS88yCCcR200kOTQVc\_NNZzVWZzorhkIsjPG-Jzjybp  
bzkbv3AYHrKmBVD5LrnsM7FuHE8Hp8kwa7XcusPjKdMjqp1f2K60udw7iQf5v  
9\_CJ\_vjOTbgdLGxOiwk86LUAM9XLxSvv9CfhDT3WSrCBrsqrEeDTiRqQ3SHI  
cOlTnadub5gKWgXl0VokhpF\_WdoH\_Uv8FWzS5Q-2YPEFL8LfQ2EqAxk8fwVcm  
8dCCFAVt3-H8uQopqinum3X2zMLco8YRO4RMA34s5uHSG6QuPu-6pS0oWBnk6  
59XWkP0-oyC6\_enE-yUCjE3CbnAw-X7MjmtImWffmeTih0X2G1M9V8c-Y6OFR  
8QbHZjP6K9D\_5Xsz8wBmhSvErrHFbH3EPg-0Jssrb2mr\_CvXo6CwljIU74y\_G  
d8xNrvwQWulXw1E2fVyhv4dPI2w9U2y\_bdL\_LLwa2Tvm9myGr6lpZABMpl5G  
jO4H81HMUW46bAereO8v8E7Y2OpmGKuvQJ9sBXMfjHfAiSyEI2M5gpb5qha\_p  
RF6BUmKo0eAhklxaxjul3GcgzNQzWIEJ9WWRJ4pC-Tk\_9egjlxRrsy9ZjKY5E  
T\_wuy0haXOZ0MNKMG99VoreLXW02otihkNV7u6CZqAjX7nyyNyD9dqMSZX2cU  
FLVoJ6GLLOz\_gFv\_xUYomXuViPG8fe07jg3EBYshAuQ-DO5LCpiymtPDIE5Uf  
AYtvH9Ytcz\_sdTrZJKAUiViLEs2-JKq2J49JpST6Iv6Zyyb6K\_bxnegNy5QwB  
MRYsXiYAWm2UdOZfaKilKzNrx\_egOyU4QwtK8DUgvc2obMD5hd35wPC0DNFKa  
SRJkdG1GHDTeiYCy6IsWrjKY4jkAY3jVFmaEsAlBPft\_jwTcvUElZ2QVF-czq  
CfGZM4JvMek\_hp7bcbgOniNdYrxmnlUZQBgfWbjBQG4qcR-eDsx23cdulWhSV  
e39iYoRF3UKfXevQxJraY7Ifwo997ZAD6yWzrKvFqqYAMZaGoIwEvN62CdCL8  
u5mcyGIJ8YiuZT8tdKMVLUK1BES8bXy1CSDF\_vJdbgjPk22f2-a2fmmkzBPjL  
yqhYuSenPhqaTeBbgdyBCr9ITNhQabmX80lKcLFRX\_tAjJTzJ-HG2gYHjGLd1  
LhNobbbUFE2Y4167oOK-d9NX5nOVfwLZa0Oyp7mn5537vKodmkXj\_YjxRG\_db  
oNpH\_xCnXoWKGth3MZsx9yvKaHIBjnbq6z7TC3UxaKIQLKrX4Wieut0aGq3oJa  
dVIgIt5y19rT8Z4gyDevAe3dcfVn-GNC\_ORLh8Vy56mZ6T7PDHFqz74JX79f8  
DVdyCEDJFUiPaOaBcaVcy8AW5aBdVH7ZyyCgfeGpynWgYMFZUUIAG2IThrAy  
kOn9\_dm3BkX9GiyBlMsgWrv-y2VZKtiMIO-V6chQP1HFRXF-\_ZmRZqz7qHtBF  
JQpZPHMOTe\_jBklnjvZc90IzhkM0A2ZxQKAK-TXXUUDj\_e00tyH7berjdJwuQ  
n87\_j8l3Olq3Ph9hFER7\_bwxclFCgzgiYEdD0We\_twn2mmG32bGoThpgAdnY3  
p6KYgWTMeTo92PXC23A2htjUKTqpbji5d1QhOip6mHUHYCnAaSZugWv6zTqlt  
nOIilt2SGyvBP\_Hnh5ilTh2fzPVqiEOFPQehuESBb71eOWqpxgjkRay\_tP7K2  
LKMAo3YTh7vj5GcFldro4hs4PV4x7AL3mrZFUoj7pFnVRrhohilaelSL2YuJA  
wv7s-0NiMlegRrDQUL4AL9k3m75Pcxatzl10ZvN--d6P7An2CMmvUFpQEfwCQ  
5FehmOV8cpsitIujLP8CWHa7PtWB\_qSJLjfuU6RzY0Ke2rksW\_4aaZDgqsTyu  
r9CXBjhpZmols\_0gaia-JhuZHsmiCHvEmdmat60PeZhKq7LyeNZ9Az32vrdKq

QrtM2R\_onDrOCwtMAmbxLTmrzdznUO-H-pdiGQwgFdDmQwO7Si4UHX\_i6ASXZ  
agk3YodlxuXJev1INnCENSgLabVkoTt1\_RTSAaesOrHPOZQ-t0ikwv41EFVv  
9vkhMBabg6VC7eBepRxLeZ7I-dBXUtGgK-quXYFgKxqt9usErcOK6euN1TyCv  
EHFYWxjhK2IWFbixO7SRsNC2wu\_oOTYpbxKkJy9GpCt8Gjc2erprYHdlwyu8O  
0h8YwWbfbJqhDNGorUU6O7seTfhZYFdn7fs-7QpCXv\_iMWH96f6ft7Yq38UqU  
Tun6FiM30mx9K6hpWJM493GcIVWGEH55g37k7Xv1f9EcSRViC-dOo9I\_Bmit0  
XkFwDpc4ZUOFV78RpvaLkUUjCzjcxim4Z3YHZwiftOnhxBDxVW9fgC-Z5molN  
UNu19WTqsSjRH8pHgmyAOp-6pTsar8ZM1LVFZTRtjVhvVSkjfmvU4FZ1yF4gY  
GuaDD0sJhXsd2cHBwlkUIvEEh7S9bd3rrEoICcRcnUWxDCeKVQA7UH4OHZP-T  
ohRAcRnxvw0Wlxh1BOWD-h8yfu\_ZeS9aXKXr6qNGBMXUKi9OsPq9tzDA7PsD1  
BKdCWvJZ0\_9oNNdcJafk1GkJctS-kqyzT-lP0lRVRxaEam2n8\_6a7RHDx9Zfu  
G29Y00EjBUFK9uAWSpE5-apAv\_kjQUliGvaqVNmdk\_TN8xlziYUZvrP0aX61a  
-14bZm-XOSk7DpRn3Jaiqzwx\_tAbDs-RF686wsQDlxmsODfH5PT8vaZLabP92  
Nu8BBrf8BSdbgplbAbahA2KMWka340oCE-CH5WKQIwuVNzW6JZojZVAFzI1OV  
E\_pf3FiXXIQpeM5\_YFzaP8QY8rr\_6W9uvc16bhlJqRMVFeP9C6VA5ikjUCbFE  
FXUwQA01IX4ZP7zd3lFA4BisBLlvixYeB\_4cG1Zlfb6tA6jsBsKfJoSK4enzm  
utntVm6rthxbgc5ONcU8DrZXKplCwyiXB6LQOnPgTgs55c1ZPsDlhseXs9WoZ  
Qo2tbrF-Bnzx9WMY\_lyn4pd118VmJzt2jUhVduGHVJoYqUUBYrmXjst4QYYTV  
Xv4azmWUIXx-jGYnnI4En76o7ZVsZrZFhJzbL\_2Zb4mo2hzV2ppZl3CVdGaAl  
XcSUNez7im9CTuRZT3jk4ZYFcy6PE5E5RvS\_kVCk7U94PfKfU0Fh0GCzS9rOh  
q6KTNOctGz1B7DP49AFNb\_NS0spVnnaY\_jiCtzh0P5Lt\_btpLhw\_jnL6fn4nU  
179MGGfNdrdfHQu3\_Y1\_Rm4wI5UxzTwlFqw8HsDGA2Yt9gYoQiDr\_Abd4GePb  
jZzoCbyLh\_SmpOcHxfF7XFG6S6D\_fs6G8mc6cGbTW-LtIO3lu7TUEv7YX6F7y  
uF5VjTzMSbGWARQLAYNvCiG0A-dK5jx\_DB1xLOUKRPLVFE4W8puNiknd5Y6gP  
wkbeeTF2U\_sOk1x2\_Y39sfp8m2UX9xFRpd6G25fFbuHy9LpXZMptqW5uom344  
M1h5\_WbvjU3GsJkgj-La3N73ybnZeFcupHMzYIqOqsjO7NaH4JkNaZ2Vp9gPf  
4wFHRsZCFh3y2I9zw2a80IWua5yLNZsVPzOlQ\_Pxn1OPMYXy05tdsDLHKvhin  
kNWozt1JTOSEsoyV3G1KNSJVRIsFpea8PhpH0G3t-YZ3AO0jG82V6USv8O8eq  
2Xwa-WPazZlMsrlAiJsyhtzPQjsqqeRnl4pleMFZOrXhCIjUDmnSnFpjwioWA  
KKcW744c9P7QlwWqAaTrYqB--yGWMYLYzQR7eOsOQ4zYcEbvpOltqCwLrk159  
\_BPNlyqGDiMiW9qfBuTE1-jw1WlN2Y20BjsHvUyrcgXiFJ6Wh2UM\_EKovoJse  
HRucWWSR8rxxlSSd5GI55vpel43-yz2hKBSfwc12daEEfke8\_ZpmgnvqIUi-r  
K3zivoWEFm8-u5jflrGluia3UkjUAFPEkaJKtEf-mu22Ya4nG6H6dS6W32mj7  
U--pg7fWn6I88KSci6GbWuLkhEA00joYMq-w6KUarGwH6MZdT9EN6XUCZDxkA  
yx6SxLL1t1LPZeL9ZWeBU0so6AYpvKZH\_E68eCUHexmuaheMMxTbaArTdTr  
vBUYdIMuF3uQcOopxmbc5d9xCtY8bszj\_yeV8X-Ai3qlj7YOkA8gsCTAQBgG  
Z-KMrNawXg6oes5G4H1wY1APlgkLf5rY1PtAt4LTFvvG39\_WQ\_ewbWdOZJKux  
LHwxwuiB8tuxXw--\_6DxHXZBxsVqyJRIu3RPG0CLzo90fHf6NaWVY72yZktpU  
41Eo4nkgjyTELzJ0dwxlXVdf94Cf\_FwILkAFMKTLkQxzn3j\_Wl6zAFLAbyPp5  
Qxv\_QlP0vfAjlJVMaUAGcoTyHKy1Et8HteckZVZGgy6rePipI3seYPGTlvtMP  
PZBgwpJuUQFNoirG0d7q4CJsUiDfuOdSlpdn-1o3Ae0bWnwkelj9PhNOBk2xe  
s-lb9kmN1FEV7o8hIqOaJwfTRrkYsQFzz-p6XFiYyvoq8T9DAXvHttWV-3JDv  
zsmGsK8qUm3KHykI9ztYcQlX3MFfB8FCA1QMR0NWzytAsQc3SGJyNqefknOr  
XZR7frTP2AQpxqR6mr-HhaJZj3A74eY7Jja3me0RkQn4\_WmeQBbCwU0ABaFk  
PR9JYL6nZqM8OWPS6lK9iMnlKkJMPHqJvLlj\_yD0MxzawyH-uFJzgfDpALXQ  
tFvqdovhLqDQ2UwjgFAjQNAyhknk8j7w\_06F1lWFBSc38OCx2CFiw5D1OuJXB  
QyWdFNU7xrtJZ5-IY4Qz98yDloqKY40P0Q3TTGO3Qu9BkhluRNrTwqLQ11Efy  
cWEFi0vyXDUMqult97mjQ3dfVNNsVAZ\_FiSC23GKqH3BXO6bAiHdWLBcc8DwY

EYkAAJL9NHZmMrGk6Jo2BDBh9aSzSOzI7qHGNw-a3WPRQLWwNy9M-BwSiLam4  
pifeWgozWKECG20tmLG0Rp\_DYRrkYUQoZ-b\_AHf1qRWdHMEusUxZmE2\_9yWMv  
ttaIkuObkvwmCFfoPuAFEn2edJU0OMhNtqL6145VftAKd6Sa\_0WdYpNsAq6D4  
34atkPogDmn6JVq8woL4Ob3x3Hm7mG37nOG6ItR8vTH1A9Gg\_5ioLXm3TaoxS  
c7pXVQxe387hIRMjXdJY6\_klCPqmhKtUy32luLW0xVO7-RSJ5ylG9FeWMFa77  
tS5I2GXzRCUm6HR1hLtkA25qJG8WB7C3\_nFAumkgAFju7EkaBlMWysho\_RRDb  
1YVaIQpPpBUFAn2Xkra14RVPOclr0m1EAWGIH6ZG4kp\_vnoBmZMNvE3-chxz  
whX9n3AtOgXgQ649RrUPIhzAD3PsT3OUIbjxTR1hc2\_ag5mZZVw7tUdiPcY44  
uw8bGQnqhb9XPn52SYcpA7ykiEuvEzVUQivmtmkQRKKH2SBAGc\_trt4eRYa1M  
F1QEZisYNKw2V-Kjwv10EilbHMrn3v-r4PnaseeamzzHiBwxJF9DeBhp\_geGE  
avVZ9iTkAaM4HWDPrmiNs07LK0SPQ3SKU4yco7eD9bZzMAyUiHQCbjbqmWtZ  
2bxvH5CipMBuB7aElBvQ0zZdgZ-pOox8jvYFWfw5RIQ7YMzkhC3auS16\_RFEi  
TcaJJz0\_ZTsvCc5orMTBgg6pD3RvhjNMO-fJ8uDinSuGs2bPJcwc5G8IN4Z5o  
K4uErvIKY2z2ZeQVkwL4QvSY6AB2WZTYSa9bIHO4PHKbWLTs5-D-xH70X\_XSE  
1\_poLgBUpvCwiMx\_2ac\_HPXP3kNtI66949rU8Nk9\_-s1M1xE7B3\_Xi90kKCp  
hTwcfoxt-\_t5fDIZBzTX-BH\_JiMeOMHgmVzMpDV3UshvPACNs1lr-Mjgrzp0o  
F5vMSu\_goAsjB9NElnSuN3X3RHKpm5859vUZCQnVeM8r0bDHWOAEgyeXJ0njy  
d9Rtl6F-2ogZ5mspDCR6Gby-Q0UDCY0LukF09g2T8KD5T1BbKgnplvZDCLd-C  
M0y3PzMMWBi2mUewtRxYQB5H\_47WLq3s\_UZxxyVL1IECBo6mx56FEH1LiEhnX  
5ioACHAAULrmICnqkcaDcSY1TwfSt9UEqZ-\_D8o3S7DropN6oSjQiUYigrpo  
QtRZlkEHRzY66MC4fjhlEnSgHECW12x10Rg\_a6PjVnt4Fs\_tSY4dDcJXhLp\_M  
dVs9ocXiLkaWmd6Ac7tYregLha0Zh9MMLCKFTQfeUv6\_jXjlyqBbMEcatXhtZ  
PJIFe64IrUKnHvBql7A6CToylP8zcrh6LMjvBHTLD7NcWZSbcSoFLCaqhcxum  
cLqPgRySAfhle4-ipve7rFYTXaNRyMbWftw1Lb22h4DNmMVuJRZsdMvENNQGt  
lpbx\_XCErwVabJ6fo7vs8K5MtVYymYs8qkVSSp6HrOdN9gd7tf9Ls4CZ11Ozf  
xjPht4UKY9WRkNYzdGoBYVeMF\_Yc3jQQQEiRVfPJ-JHwi4S0JmIx2VTHSnnC5  
KgxiVkjOwj5nxzpf9sltdJdJeXSegMWS1q0sgwpoudpPQC\_yKVbULnpVuiH20  
pgAD28M2UzUoVJ-\_UQ4TYC1S15dFQ4j5EYBJPhLmHfOr8m9lXiwpXBlq8dpS5  
K0lDc7a9nHBrnwpAH0z7pV5RIKZIKX3sWyASRr487Rp8Vc6LhAofX2sz6mIzb  
alpZTDxdmYtYbDyeZdmvFYpaB0F1lHzcXVfGd5wCmGh4jpqLglxXUnb-KQOS  
WRN0MBCP0AURSj8OSqMfUmkobE6spy28Tbde6bkJ8J4-vOYlC9mXYKzYml38L  
YmXdqIETdcnXTchuhNDwcHyiBsCVjAyWNR6SUvM4k4L\_i0jpn3B0z9rZU380Q  
qUUTY-U3DVvlbPprM7JE1\_1KvSmG\_p0OZrt0ICe-VFRf4brn8W2NugQr83DHS  
qzpTJzIo5UbCAGP-VIM0vOnyKs3QrZdS4K-luHwcAdmPrSWJ\_W7jv16ShK\_KO  
A15UkkLN1k-WfzL97fzGFloMwErSgluFB2AkUL2mJUxNcFZrnuH32M-W-qtob  
LRaxT\_bkELx3B8qe5GJHOYVVOFc-jJgkDR\_sQF0mh7qC8XQI2ZWCRHPFxCuAM  
SJF56-LZELrZLbqbVzpjV2VLkHxzZIUkLdFDXGBgHkactGP\_O-f3zxHS\_2HZ1  
nK4sQiVos-170aH\_TnmSicaG0K2VOavtfigZozYfxOuDqSMQKK05PqBxVtqdU  
kdCdJnQSzXRsI2bmhX6IatQsRzU6OqRopYk-ezd4Gy-rBlGBPztpiCo3q\_WmM  
DpsT1FqlHQ1kGscTzcQAgRoSRkG0VOjM3KCGe0WLv86QGkmur7-2oDA56S7zb  
a7JlZRONqW9aEqtZrZ34i-SZY0RX6NjHpDVi0V0naCqs3OhrqIBO54i1VNiI  
vwoleO-YEj6GShMoEJfayZTSFBM6F5a0arce37TVy5oNr87nBZI9b9DuA612V  
\_CYpJE-PPpm\_-wJPSmch6f2\_9B7FGqVU1A9YsAHlq-flmdlpL5xFAIZvaemyc  
3Iq3PCXQrNJdF8m78h6O4e3fg\_4H4JaK8\_pnBU4oAzaSfmiofdHnSpVaQulL  
cTjCQmalO8dpBsf0Mb7BuwueelosuLVfdQkSfdLlscRi7QZXU5b5xz7uEZTnD  
mHM97RkUhn2tIEl9DDDOgybXKwXN5mvph2XuX8JPXsdbueFrKmmj\_WDliF8dl  
gtSI\_O3avgo06TyvzP48fb-cWZRsgET6YWBReIwBp-ZwmaqOhcKgQM9uie7BE  
fc6xWJVOya\_DmbkdbmZCFscGsxCai9RA0lgo5g\_dVk0rrblDk8bUJ4qn6V1Vc

ACReYg-knSf941bw-KwfgvswZHHN2vZkRsueiQxqJ4dESj1lwYSTZHit63DO  
Zfu-4RLUXbw27R4y57dbQc7TP42z\_s-RvaddiyBl-mTePMorZOVy8p3XAdp8l  
YN1OFkiof-iW0dCODHwuhUPAV24zjVv7b6zFUoi0DqyYrE7smWaTMcz9c74We  
aZxFtXSyqo04khEsfPXIKTdUNZ6Hv52ViVlUGNlmrDjEcFs5jQIXXf2YEPSFu  
lfzZYZ-teFpCiWhts5xfJdSdQ-Cy-95xBnlrBgkMffNhVvzZ6iB-Cg7Lir7JA  
NAsGo4w\_oJOf4FPBaM6rCt3vCOJ28IsJ2v5EoDD4GgDGHEeAGIkZHtVxz\_r\_T  
PlsvKFu2awqVB4vG41\_y3OzmgslPSBrN7ztcs1fxIJ4zsSV4pQUDwKCzjOS-Y  
oYyH4LjNm4yA-X6BraWejYsWP4sPTZDKKT3IaK4XpDFJP649NkdkVctFIN2Ys  
zcinC14PDen9qCQPxrNL8p\_CIMGLjPfstT3Gs-DJrCTlJ8iDG5QQ7LkCT4ul-S  
kvCdc4FtJtd53BlxPRB\_s4uFdgZIfCn2lY5qnoRxx0j8fH90rrZqubNhBon6s  
I8wyaZBZmaA857xUtybs6\_tlbfgU94m6jeJDC\_07evqJ9fsD0ONx-gMtKo4tj  
2kcYW7tKRfdImGPQ6TiQ7lR0iorpj-aljWSGgsYeNSWakzwsKRI7ovrak99AE  
gi6hRZFqdCjyOW0nFjrtBs\_K7Gb00Tnl1KPOWgMmBBI001ojyxkVtdxT2l6jq  
X3ewze6xfNLSXfSVGBMKS9ZHHTbIG5qOmVOz2aVB48rc\_-6JrxUVVT7h6ps-X  
uP-zKpZJRumMXUYtRBkJOKsi\_nnIhiU2Smc8DCqCoJdUhoU-elr09g-VN40yE  
vAcTpaLjBzE9jXRqTTp1vl-q6clkuI2ieakrPIdoAG6nTLDGD7zS87hlpfZGs  
xZDieBwHlNAu92NmpE69NGO9xmJtUPIyXwh4OiIpZLkWAXGspBV7qroERL0QM  
uXrHzpcmvRostLfCd9o3l1Oknflwd4wLEi7vQRHklu08MY9ln\_WmzTnu9LBjt  
xyHOJRfaewMhlrSvh0Ac3Z6UU86nsWCohH8INIFCgi\_Qldnor4YkCdsREWic1  
MD4He0fqVcmPTz58gene99dNSodYlkjCtIjAetAnZDGC8RNz-X3S82\_I3eYzY  
dt000Uu3vFAMTilHJlfiwvfoCBMQd\_kPMzB2AzUW0HE5Y6j6sXoPhiQyEoglu  
w16qcDPkEQonn9AnV3qmE00nMqC2eNcZ2RWV1E0lPQApnFG8CTNIj6lVJx0Lb  
l8TNpE0GRvlpjhM5\_qDZ4-aiMiq6hl\_nHEK7jtsTSbNWu6Mr4GjAJ-TqZPqh\_  
IDHsHwPkrLKZy0A6in93PPDYCE\_kaHhZfSWvZpAOX6H\_c968QWtr\_SWHgWvx0  
TdLvbwzgyCZ53kpcyQV\_1CVHzKWn2PnvG4beqrfVlp4tjBiYDndz1F198PA\_b  
ldkX4SuxU8JuvA0ODCAISJEZO7AmLnDrjzipm3sEKAarpyEJnjMv-uuDv761f  
SKFXHaTpXumZSVgVpM\_osolF9\_M93kJEAQjLvWF2XAE6QT2oHOTw\_B-ta9XZM  
7tKEelYz-l\_ByWo\_DRXX5FC6HAX6wRMyjzTYh16DfEyHKJBwDEg5j0eNrVSgY  
Ux3PIYSRbh\_wGnSGSjFcr3aOEbiiYpKNlIwGK8Wk4oyP2rllq9ewar39F03j  
MUOnG2ThEuJhJtrHlZVDY9MI9NifaA5RnLd-63JAetGQK\_aSkM\_K6-Asz\_H5S  
w\_JOevadNY29BmiVaccELs4SRdKWbnSbR-cjQhz29Z15BlrqiTjCXvnusHa4r  
W-1J4B5hzoRB45mxMfjaChsrYK4fb-rDKkN49J2kwcbyDBvn0BSQYY33ch0nB  
LU4pT-qPic9r53DapHtRdD0--FS50Bu-h0F0UucQbVhAkUzeryPUkLEJgfpV  
gVLRyAIOMzBlnsAIocEaG90TnGD8kGGisLS3jhwqWHO2iJKE4ey\_pUeSc\_Taw  
DaPKPNeEtBDFJQoL705JmJ\_Sd6yDdyzTdSo5pt2ECSe1963g6GUjAlZ9ujL4l  
NQ\_D2\_d9j7JffrtoWhxoSh6\_RnckmhUVHorDS5VL0-9CT3HoULVBkxJhdiTvZ  
JY12UyhLZ6i6Sw5\_Tm0914J7yb2kkZ4UjSltdVzn9Ij2Lx5ATPTArfZVNLuhb  
D5VRta3qqYdlZDmu8psWjvkPzxByH1FyRO3e4GYdiN3dAg\_l2IU-GN4yS2Mj-  
4ccEZ6IF09OceQlpT\_4vwf0AffUQ\_2n1xVe1\_e6qKQ2QZjU\_D\_MithdQHmVYX  
P7dly0yIIRCGrjadB87LM9hjKgN6WKbrd2J9XRE31SLvowa5Juw0Nhq1GJALa  
aZh-\_PTuuaafaGuMLIXpdKTVY8uOE6uUp3J2ofMRu-rM67Gg-p8VZnhFkg-AQz  
TsMqmJls0N8IOtw0l1fSNzl4BkJcYm4bJM0JgHko7cPqMxV-0124o4LICKrAE  
\_KwbmsnsV7rt1EV0Z4vg999DBTJkzk40Ad-FdC0c7nOio4kOy3tg2wGweYXKD  
dCLpAst9hAcV5ZYTgBpNnVtPYxOK5RgZmYFvrrHULuwclh0Lmn0HS9SyZjpSl  
wB\_ly6DCTyIkFMHs0YRXQjO3dTSEyYjyGKSyu8s08mJbgo0cqXvt-qrwnHAuA  
YzXroSY5UC-ROI5giAHkLFAESj5XzWWpCkPCzV-8MpaaDevFrtDKLd3StNT-Y  
udjuivupPf9JSTplkUGBify\_GdTniZchElGbU8sBuJQmTXVoW2h1\_Tu3j2VZm  
IjwxWfwcrKcmjF-2TFho-Uz5LD0r97GrSXI\_Zlp\_EzDN4b8bqQ-qMyH3OlQvH

vjkxZ3Qldy8ErmGBNclmekgbs24e75UfM0m4cd7-Qh4ro04yRGmfASY2SsLyQ  
BsBC0Et1LiyLnRemkhRCKIUo62wqRZJEIS1LoKQrrSgMAN6yibeJAJX2nyTwF  
RyeIZVM19oRwDI6etQJQkPRpBfCmgwEyKVKo6loV09Fd\_kRVqr9cw5mgeCVZb  
CRo\_gSFwmptdM-sABWowMh7hRCC8Nm2\_WnMtIRfq7oKo2feARrA9tHAOpCgdk  
D1ZxgxrrzbxAYGw03CUonAl6d7ktZ9MUQDotd1TElc\_bfamw1ASYKCU7WVmvR  
FBMiv8KuQKDPFAEKNGS2s9Vos3ZTa3EVbSXXMCfeOg4MXZeoJHoSzZ3WEtDf8  
Alm1lpSVMq7d5035PrX4g60e6tHVwbYaYxyXeGCGoTQouktaKndvi6vwhJxg\_  
86-Q8thPX7HLJDxmlSEKoYYBRq8OVny8\_ui-sK5hMcWnSzZg9Vl75iJwdg7OC  
iu6L5chD3CRk69b4OeDkU73v2F3\_OAF\_bgTJO8nJdZOJENq5EPgV-UWVQnMY1  
3o9m3PLw48Q5gUC6twe4C8OgmqbH2yfPKV-N6Ka9GvBDcv01TgST3CO0LWCmh  
eCeygxVRlGXu\_eYSdHYu8aT9iSpgyMBZUP71st30ArmOslfuam\_MMbts5eD5  
OaQA\_a6mSfTa3d9nOK7rqY8T6-vuxlo-Hggc2r8BjoE4lyNbWzUXeD-Jfwr8  
PdUMvbEgF2bErs8oDzuV5EXhMT-C7Mwm-gx8qdVuwxDWDphHy3Cd6HibH\_z2q  
HMBza6Zw5jSw5Mr\_INYrUGw-0eI-17pAv52U\_LqDwSwAkZBUnfo2GCYGYJmK6  
VoDhhZQLsZG0EtltH3Orfw38yXBD2S9-6CmeMQeJV4l6g9vROS08ggqXhWsQk  
gEePeGYem6cegTEJ4r9yopMxMy7KvvhB4vN6pp2FoC03\_\_NqjW4u-\_DwNfuYZ  
8BCLMt6NG6oC3lokXeKx7WBUPI5abUbVDXmWHZLRqjtqv-esY1uPalBYQka0x  
CSUVNv7YyP258HLPfwl2-dudUkxsNOyW6NGskSmux56S4D0QTTn56PMVC3Th6  
vS1Vw-t5GYp7QRtXCEpgQ\_Deyyz-M9UHOJREJnNxVGIqrScSZew3Zklts6K-a  
hAimGKCwqKEl-zf2xp6pdjxdrXdl56GVB646XBhC\_PwWAmDPWB0aglvntfkfn  
-VM9FhAiIxcIEAS9rs2lXO3ujlRXMbWyl4xfgfVd4udoUZb-eekZWphYtcuGw  
8\_KzUdJomtDNqie4R-vc8UOcGARNg4CdsPuHhyTtxPNnbNTf-IkAlDH3oZt6x  
hAv8QxKJii2g8brDVgN0vi7HUXKbbATTBlgg\_WZBZ-iRC7ckaC-0NCiP0cRkJ  
Yz7Slk4k7ICKEGvmQRGic4A5ZAo2qj4Z6E89NlmsOPRHVkuTup9SFxF0Cfwe  
9YrKPZo37q6\_XyJAg2ZXmSIRwTKgJIwtXwa7ygXmXiBmkNO9cE-X-RZA6sVkg  
l2PYHZEi\_VPa6r4t6eTs4TJegDFibnCl44wbr-bcFKJ8uCRaUVnfOvfEDfmPR  
6SIYw2auUrmJ4JrTeo-t79NgXPJwDVCN6p2enP7CjsIeBCa23DmYGfxcS52WG  
dEKwebIgg-Cx0eGF2KAW64HGkouWSIy3j3CrBX4nshzAk-M-zosrYOV7hHiZX  
jbSfZiLe8rPMQOjgl1o9sSy2R2WLwiHInIYKKZ4sWSDL-aOjeuq-WRApLlpGk  
lBrg3YQf67otXP1lKIC9Ev9Pifm3E68oSbOIryiX2EwrVuuET03Z5WfNw8Yt\_  
nPKyW\_iBKNQ3dHPCGn-\_M0DlXojPQhNmRQrof5wVekZhexZIA-LrHlyOi5O4  
6L8ItBV9gLXOMl34zoWdJFeAT3SUPY4h6J7-\_By-z4JmMeWk9ht0TS5loEOBq  
Qx3TFFhN90oPW8s9MjpaqTF7c8wmPeElqicdo7RyzZph5DhxcakxOFZSLijmQ  
2aAXOkYL7\_cORdacL-zgSgyS9GYWL2BH7ReTRL7vz4NGBs6NGs4Fh8AY37hmN  
tZ0rN\_tTW7EPdGa-tc9PVPB9bLnAK2I6ehOPQvIpyGJhSP8HS2ErE68bV6UNo  
2LbZhFJlgMKbCWWhRoA2USLv9EhgHXsBc2MkJpk4qTCSacwerWI514BzARBjN  
I2v2tjuBdQubLnje2\_MvTYVWm2Nj4jysdr0UJynJhmJPLAXP8Gd5YKE2-uziZ  
ZylNqSk5WVrLc2YpeOJU7L-34s3Lno5hEfDhoLN-3QDu--BVyrr7iV7X6ybi  
Mwb015Ja37H3U\_Qw7NvrXuyIcSidGQNXbTjk8es1EQ2lfkqkeBtSGhnnLhiaF  
ozMutaB372y\_bdwoakpMnSrYRRRER3-pUhT36Bke5lwCHv9BcWOGpIYrOGjsa  
liIVnC9der8f6Lp9uzHLTpBZSV0ZkVQQ4\_4\_c2n\_Hm3nnyUcFsH0zgOJAmcZj  
0Z4807z\_HZNwzRCxq09VrHaGDH5eXZW1hL4-Z8g\_GHhdiCmlSIkYi3kovyssk  
hm2rlzG8oNDvLx2PMF0jYPICIW7M18T84GzwQlON08xP-hawtGFRI9P2G4uf2  
FcVlK-4vbTV\_v936ePSEnSWoPDgaOHX-q5Y-g8HfzISB7lmkZxoqvn09JqIZR  
o2oQlcoEXqI0XpTzHdg-rooChzHvV7XBkEcH\_ytClFirJpNLv146kWTiiejSS  
Xm3Abgus9lRmW6N9Lc3uvf5yf15Gwluyly\_ad2LZGdUkDcmgFD6HwNGrEC14K  
VsWMivcuw0W-sQKjHNfgDAHqMXYTByx8TbWIRgl-dhfzMZlSd41acK4C2O5U0  
4pSAMBzowPKAQFMKMEK4DFgwy0CrdWGfzM1kTZP76TMgDzvnDsGycujamRfL

```

fkuB87af5IPUvmgShIpC6F-9-mXpGB3yI99MpeJ_p2QAcyfxneXBeEcfEMmsp
j98ejs7dJsvXCjfpCQYkQ7iyxMH6U_0iEnOkw9G6eRBD00TZWrcsawgUj1urt
mn8F_D9qMBgKTfiCiQr-5R2qfKrvO2UJisWKC0KhIddR4Eue2W5PFhU-wP-na
bo3MjOEXvLsewm4Dr36wembHAnPRL6h_3oDu7Sr7dFVnzQaf6LLhbuId2WXX2
X6lcwxQ2UjLjzpUoQbnbnzZ3S-IS8LhyBskQNeIq45p40AlzohHPm_S00_1Ru
UyNZjl1v1fg0S5evXqw17jFljX57m5VksWy6uHTGDme60-SzXRZ4MqTpUDgcJo
M4snw6IcWSxIPJmmleVnKzKLazVeSEEv_56miRrzX6ap65vd01fyPtaeHh9rE
iDzngsEaVlJia5Gi32X6GuDkd18_82u-VGoI8ncnN5QOODAISwL4_Dk-ovmkn
QTzMBfV_AxDLBE3eonkUG6pAGJ7IJAAbMP_2lq1TFvqOt3OChL8t5T1JRsDdNs
wM1JmvL5ECTbUTKn7mlCP-6N9pBfuMJuxXAZGLqlyXsg_dyDYe58rrMOMlftS
2RLlvGLSE3XLKPH2gujj8_7Yu070aPm_APB9F34BSkl7lRrjnYj02wsbiHoZR
0PGyJBLt5skpanyiuyFqXIbDbqiWHQbBr_UryyGTyIRnfocJdVJ4f47cgQlPj
MzFwxwBBBkXym7b1WnIQGbZ_p84OolpGgqBWCoYt46XrM9Tz1Btiafu4RR4ii
afPa_MrQaFwHBapPFadJNQAGghBp-1jM_xQ7X5XC9QRJdFQtUUhagIhTAY_p0
nThos4tSJzZolqQiQ7kcXMaXqO8p4q4lh9ePbVn1bSek2EnYk5jFT3x9_cRPl
-b7eavQ10QKMeElegiZAWwOMsmU1eGJ1_VPd1qNtxmDMXnYBeVmK0FSDXCf1E
To9d3cC6ahenVBxHwvD3PQkP8BB9bqhGYBU6DgPh9ilS4nINZHou9PwseHLyp
UtnUv2RzCHZ2kG2HK9s-0AdGp9d_CZU3wZeFKcNn8gaEWM-jYh7CtfCoCG1VP
l-WE8ld3u5xWAnq5TcvN3toguZJ9juZ4wz65D-OM_Jsg-I_asoJqURI-lSYYM
Ql1Yhbr0gLLn9L-k6QFoLWNq43Zxgk0gHh9mYMLT08CpdMKybk-SRhPD94cRK
RBrjGEZoafRmVOBn-_ze7EDn-Btg7jobsLxkz2OEWz_xf672_zqMm40Xv7Dyo
19V8UWLI-GZ9hzhc2OkBKtaAmDFpdtzLkySW8hPe7lO3qhSfDqEmIj6-4Vcg3
RlJ0AmlS8UpKNEBIGRD6ghn6qids6slytd4t6F4xy7mybTMeMI0gzLDEwQcAK
-NeNGhHBjopMjdHZsbRp8qN80IR_om4J8_kTzfncFCgORCiXmVuJvnpM9BWZj
Pk1fgX5TpLziJx68xrHKHYTpVdNpXINESy00JkBCoAc7OGf25tR-9R3v5py_F
niWVi_B_68i0t8WxQSQgZN4MPuOzKnnNaev36pLxAOKK6pdwVMmP9tnyuIqtC
7ITAqlbI4VQFocGJyo46bn-sB7mE_m2Jt199g7p_Izwh2awrsk7lNsUzuoWVu
yR48j7ef_7wEvoqKkcYJvuKsw4BkwvOZubMBuCP1oJhbsxy0CG_Ojy57J_FPj
uWrw-XowWN8pn58rjjeATOAdtYrfo5aEqD8a47noQZ-tSRuumKiOlKUN5b3BL
AEGth504BvfcofNxfiXtHYv4V6OSRKIKjDRVO59vhUMuE7Vj5RPJbYFIM6GoB
f6X_t9-kQyamWk3bHDT0ivUi7Bcax1hzI3iV-fYfxK7AzX7gU1-VWBGy-xFnL
G_r8oq0g45k7OFU42qmpWPU8X0ETHqQyFFLP7TgrluihVTSIO49kH1zBixe57
Qszpl_rjXZVXSC3ltkq3ORloy35AtACpUK917bWqZ7eOchJnthdRJB_gEDnkb
bBXcW5F6RPvrmb79w8-RvSUi_DfrThR3bqvM98HVjwlJgeNDu0Lpc5tPcDhP
5J_OPct-Mre6-AL0Z8Kp5PWbS2aIskLneZHK_x85Kv",
"tag": "yUtoAEWMDlRDiYclUSHGMRYC9wddGWhYbQbkDRCxjls",
"recipients": [
{
  "header": {
    "enc": "A256CBC-HS512"
  },
  "encrypted_key": "XyeGomw_HGQj0bqeWDEdYnFug_m8kFPO-lRUtAn4K
l-T6dbg95JUPDItGG2IL7j4lfoE1HhCzI-K26vvAy6Gft7EcajZ-MaczG
XMRwYgGWM1F2gpvyVEhPtBHDrx8I_UfHFwOed45-FFKC2NzdIu6knhzDM
PGwSiUoVJWNboTcdAtXHpvGPaY4sn6zrKKiJJurGaiFdPCBo9Zc8zZVh8
YVGzBNF1Q69DjH_fYPnI_XmeA2whuh3OrBYtYECA-ezVCBYfngzIqaNyg
0jYcKeRKhqkMWo7E6nYHLz2-7-b8ran__migQBTzJvg6VctzfHzzOGWoa
ciolkRhx2VZxyq2g"
}

```

```

    }
  ],
  "unprotected": {
    "uuid": "0195544a-b9b1-8ee4-b9a2-279e0d16bc46",
    "cty": "application/vcon+json",
    "enc": "A256CBC-HS512"
  }
}

```

#### A.9. Redacted vCon

This is an example redaction of the vCon in the example Two Party Call vCon With Externally Referenced Recording (Appendix A.5). 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](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": 4.9800001,
        "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": 15.1950003,
        "text": "I'd like to add Sue 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": "2022-06-21T17:53:49.015000+00:00",
        "duration": 6.239874999999998,
        "text": "{{US_STATE}}. I've added two to your service.
          Is there anything else I can help you with? No. T
          hank you."
      },
      {
        "parties": 0,
        "start": "2022-06-21T17:53:55.254875+00:00",
        "duration": 2.8200000000000003,
        "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": "0195544a-b9b1-8ee4-b9a2-279e0d16bc46",
  "type": "PII Redaction"
}
```

```
}
```

#### A.10. Appended Signed vCon

TODO: appended vCon example

#### A.11. vCon Group

TODO: group vCon example

#### Acknowledgments

- \* Thank you to Thomas McCarthy-Howe for inventing the concept of a vCon and the many discussions that we had while this concept was developed into reality.
- \* Thank you to Jonathan Rosenberg and Andrew Siciliano for their input to the vCon container requirements in the form of I-D: draft-rosenberg-vcon-cc-usecases.
- \* Thank you to Rohan Mahy for his help in exploring the CDDL schema and CBOR format for vCon.
- \* The examples in this document were generated using the command line interface (CLI) from the py-vcon [PY-VCON] python open source project.
- \* Thank you to Steve Lasker for formatting and spelling edits.

#### Author's Address

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