vCon
Internet-Draft
Intended status: Standards Track
Expires: 6 October 2026

T. McCarthy-Howe
VCONIC

April 2026

vCon Extension for SIP Signaling and STIR/SHAKEN Data
draft-howe-vcon-sip-signaling-00

Abstract

This document defines a vCon extension for capturing Session Initiation Protocol (SIP) signaling metadata, STIR/SHAKEN certificate data, and related telephony information within the vCon conversation data container. The extension uses the vCon Attachment Object to store SIP messages and certificates, and introduces optional parameters on Party and Dialog Objects to carry SIP-specific identifiers. This extension is classified as Compatible per the vCon extension framework, allowing implementations that do not recognize it to safely ignore the additional data.

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://vcon-dev.github.io/draft-howe-vcon-sip-signaling/draft-howe-vcon-sip-signaling-00.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-howe-vcon-sip-signaling/.

Discussion of this document takes place on the vCon 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/vcon-dev/draft-howe-vcon-sip-signaling.

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/.

Copyright Notice

Table of Contents

1. Introduction

The vCon conversation data container [I-D.draft-ietf-vcon-vcon-core] provides a standardized framework for exchanging conversational data across platforms and trust boundaries (see [I-D.draft-ietf-vcon-overview] for an overview of vCon use cases and architecture). The core vCon specification includes basic support for SIP-originated conversations through the Party Object's "sip" and "stir" parameters, and the Dialog Object's "session_id" parameter. However, many use cases require richer SIP signaling data to be preserved alongside the conversation.

Telephony platforms, regulatory compliance systems, fraud detection tools, and call center quality assurance systems all benefit from access to detailed SIP signaling metadata. The TRACED Act [TRACED] and similar legislation in various jurisdictions increasingly require retention of call authentication data, including STIR/SHAKEN attestation levels and certificate chains. Emergency services systems need SIP signaling to preserve location information, priority indicators, and routing metadata.

This document defines the "sip-signaling" vCon extension, which provides a structured approach to capturing SIP signaling data within the vCon container. The extension follows three design principles:

* SIP signaling messages are stored as Attachment Objects, using the existing attachment mechanism with well-defined purpose values.

* Conversation media (audio, video, text) continues to be stored in Dialog Objects per the core specification.

* SIP-specific identifiers that are useful for correlation across systems are added as optional parameters on Party and Dialog Objects.

The extension is classified as Compatible per the vCon extension framework defined in Section 2.5 of [I-D.draft-ietf-vcon-vcon-core]. Implementations that do not recognize this extension can safely ignore the additional parameters and attachment objects while continuing to process the core vCon data.

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.

This document uses the terminology defined in [I-D.draft-ietf-vcon-vcon-core] including Party Object, Dialog Object, Attachment Object, and Analysis Object.

The following additional terms are used:

SIP Dialog: A peer-to-peer SIP relationship between two user agents as defined in Section 12 of [RFC3261]. Not to be confused with the vCon Dialog Object, which represents a piece of captured conversation content.

Call-ID: A globally unique identifier for a SIP call as defined in Section 8.1.1.4 of [RFC3261].

PASSporT: Personal Assertion Token as defined in [RFC8225], used in STIR/SHAKEN caller ID authentication.

Attestation Level: The level of trust an originating service provider has in the calling party identity, classified as Full Attestation (A), Partial Attestation (B), or Gateway Attestation (C) per [RFC8588].

3. Extension Overview

3.1. Extension Name and Classification

This extension is identified by the token "sip-signaling" in the vCon extensions parameter (Section 4.1.3 of [I-D.draft-ietf-vcon-vcon-core]).

This extension is a Compatible extension as defined in Section 2.5 of [I-D.draft-ietf-vcon-vcon-core]. It introduces additional data without altering the meaning or structure of existing vCon elements. Implementations that do not recognize this extension can safely ignore it while maintaining valid processing of the vCon.

The "sip-signaling" token MUST NOT be listed in the vCon critical parameter (Section 4.1.4 of [I-D.draft-ietf-vcon-vcon-core]).

A vCon that uses any parameter or purpose value defined in this document SHOULD include "sip-signaling" in its extensions parameter.

3.2. Architecture

The extension distributes SIP-related data across the existing vCon object types as follows:

* Party Objects carry SIP endpoint identification data through new optional parameters (sip_contact, sip_user_agent, sip_display_name) that supplement the existing "sip" and "stir" parameters.

* Dialog Objects carry SIP dialog identification data through new optional parameters (sip_call_id, sip_from_tag, sip_to_tag, sip_cseq) that supplement the existing "session_id" parameter.

* Attachment Objects carry complete SIP messages, message traces, SDP bodies, header summaries, STIR certificates, and verification reports. Each attachment type is identified by a registered purpose value.

Conversation media (audio recordings, video, text transcripts) continue to be stored in Dialog Objects per the core specification. This extension does not define any new Dialog types.

3.3. Relationship to Existing vCon Parameters

The existing Party Object parameters "sip" (Section 4.2.2 of [I-D.draft-ietf-vcon-vcon-core]) and "stir" (Section 4.2.3 of [I-D.draft-ietf-vcon-vcon-core]) remain the primary mechanism for basic SIP identity information and caller authentication. The parameters defined in this extension supplement but do not replace them.

The existing Dialog Object parameter "session_id" (Section 4.3.10 of [I-D.draft-ietf-vcon-vcon-core]) provides the RFC 7989 session identifier for correlation. The sip_call_id parameter defined in this extension provides the SIP Call-ID header value, which serves a different correlation role and is more commonly available in SIP deployments than the RFC 7989 session identifier.

When both session_id and sip_call_id are available, both SHOULD be included to maximize interoperability.

4. Party Object Extension Parameters

The following parameters are defined as extensions to the Party Object (Section 4.2 of [I-D.draft-ietf-vcon-vcon-core]). All parameters are optional. Parameter names follow the snake_case convention required by Section 2.5 of [I-D.draft-ietf-vcon-vcon-core].

4.1. sip_contact

The SIP Contact header URI for the party, as received in or extracted from SIP signaling. The Contact header provides the direct reachability address for the user agent and may differ from the address-of-record in the "sip" parameter.

sip_contact: "String" (optional)

The value is the addr-spec portion of the Contact header field as defined in Section 20.10 of [RFC3261]. The URI scheme (e.g. "sip:" or "sips:") SHOULD be included.

This parameter captures the actual contact address used during the session, which is useful for troubleshooting registration and routing issues.

4.2. sip_user_agent

The User-Agent header value from the SIP signaling for this party.

sip_user_agent: "String" (optional)

The value is the complete User-Agent header field value as defined in Section 20.41 of [RFC3261].

This parameter is useful for identifying the SIP endpoint software and version, which aids in troubleshooting interoperability issues and identifying capabilities.

4.3. sip_display_name

The display name from the From or To header for this party, as carried in SIP signaling.

sip_display_name: "String" (optional)

The value is the display-name component of the name-addr form of the From or To header, as defined in Section 20.20 and Section 20.39 of [RFC3261].

This parameter preserves the display name as presented in SIP signaling, which may differ from the "name" parameter in the Party Object. The Party Object "name" parameter (Section 4.2.5 of [I-D.draft-ietf-vcon-vcon-core]) represents the known identity of the party, while sip_display_name preserves the value as claimed in the SIP headers.

5. Dialog Object Extension Parameters

The following parameters are defined as extensions to the Dialog Object (Section 4.3 of [I-D.draft-ietf-vcon-vcon-core]). All parameters are optional. These parameters provide SIP dialog identifiers that enable correlation between the vCon and SIP infrastructure logs.

5.1. sip_call_id

The SIP Call-ID header value for the dialog.

sip_call_id: "String" (optional)

The value MUST be the complete Call-ID header field value as defined in Section 8.1.1.4 and Section 20.8 of [RFC3261].

The Call-ID uniquely identifies a particular invitation or all registrations of a particular client. It is used by SIP user agents and proxies to match requests to existing dialogs. Including the Call-ID in the vCon enables direct correlation with SIP server logs, CDR systems, and network monitoring tools.

When a vCon contains multiple Dialog Objects from the same SIP dialog (e.g. separate recording and text dialog objects), each Dialog Object SHOULD carry the same sip_call_id value.

5.2. sip_from_tag

The tag parameter from the From header of the SIP dialog.

sip_from_tag: "String" (optional)

The value is the tag parameter from the From header field. Together with the Call-ID and To tag, this forms the SIP dialog identifier as defined in Section 12 of [RFC3261].

5.3. sip_to_tag

The tag parameter from the To header of the SIP dialog.

sip_to_tag: "String" (optional)

The value is the tag parameter from the To header field. This value is assigned by the UAS and is not present until a dialog-creating response is sent. If the dialog was not fully established (e.g. an incomplete dialog type), this parameter MAY be absent.

5.4. sip_cseq

The CSeq number from the initial dialog-creating request.

sip_cseq: "UnsignedInt" (optional)

The value is the sequence number from the CSeq header field of the INVITE or other dialog-creating request, as defined in Section 20.16 of [RFC3261].

This is primarily useful when correlating with packet captures or SIP traces where multiple transactions share the same Call-ID.

6. SIP Signaling Attachments

SIP signaling data is stored in vCon Attachment Objects (Section 4.4 of [I-D.draft-ietf-vcon-vcon-core]). This section defines the purpose values, media types, and formats for SIP signaling attachments.

All SIP signaling attachments SHOULD include the following Attachment Object parameters when available:

purpose: A registered purpose value from Section 6.1.1.

start: The timestamp when the SIP message was sent or received.

party: The index of the party that sent the SIP message, when applicable.

dialog: The index of the Dialog Object this signaling relates to.

mediatype: The media type of the attachment body.

Attachment Content (body/encoding or url/content_hash) follows the conventions defined in Section 4.4.7 of [I-D.draft-ietf-vcon-vcon-core].

6.1. Individual SIP Messages

Individual SIP request and response messages MAY be stored as separate Attachment Objects. This approach is suitable when specific messages are of interest, such as the initial INVITE and final response for basic call setup metadata.

6.1.1. Purpose Values for SIP Methods

The following purpose values are defined for individual SIP message attachments. Each value identifies the SIP method or response category contained in the attachment.

"sip-invite": A SIP INVITE request as defined in Section 13 of [RFC3261].

"sip-response": A SIP response message. The response status code is carried within the message body itself. Producers SHOULD include both provisional (1xx) and final (2xx-6xx) responses when they are relevant to the use case.

"sip-ack": A SIP ACK request as defined in Section 13.2.2.4 of [RFC3261].

"sip-bye": A SIP BYE request as defined in Section 15 of [RFC3261].

"sip-cancel": A SIP CANCEL request as defined in Section 9 of [RFC3261].

"sip-update": A SIP UPDATE request as defined in [RFC3311].

"sip-refer": A SIP REFER request as defined in [RFC3515].

6.1.2. Media Type

Individual SIP messages stored as attachments MUST use the media type "message/sip" as defined in Section 7.1 of [RFC3261].

The SIP message SHOULD be stored in its complete wire format including the start line, all headers, and body (if present). The message MUST be UTF-8 encoded when stored inline using the "none" encoding value, or Base64url encoded when using the "base64url" encoding value.

Example of a SIP INVITE stored as an attachment:

```
{
  "purpose": "sip-invite",
  "start": "2026-01-15T14:30:00.000+00:00",
  "party": 0,
  "dialog": 0,
  "mediatype": "message/sip",
  "encoding": "none",
  "body": "INVITE sip:bob@example.com SIP/2.0\r\n
    Via: SIP/2.0/UDP pc33.example.com;branch=z9hG4bK776\r\n
    Max-Forwards: 70\r\n
    To: Bob <sip:bob@example.com>\r\n
    From: Alice <sip:alice@example.com>;tag=1928301774\r\n
    Call-ID: a84b4c76e66710@pc33.example.com\r\n
    CSeq: 314159 INVITE\r\n
    Contact: <sip:alice@pc33.example.com>\r\n
    Content-Type: application/sdp\r\n
    Content-Length: 142\r\n
    \r\n
    v=0\r\n
    o=alice 53655765 2353687637 IN IP4 pc33.example.com\r\n
    s=-\r\n
    c=IN IP4 pc33.example.com\r\n
    t=0 0\r\n
    m=audio 3456 RTP/AVP 0 111\r\n
    a=rtpmap:0 PCMU/8000\r\n"
}
```

Note: The body value above has been formatted with line wrapping for readability. In a real vCon, the SIP message would be a single string with \r\n line endings.

6.2. SIP Message Trace

A complete SIP message exchange for a dialog MAY be stored as a single structured attachment. This approach is more efficient than individual message attachments when the full signaling exchange is needed.

The purpose value for a SIP message trace is "sip-message-trace".

6.2.1. Trace Format

The SIP message trace attachment uses the media type "application/json" with the following JSON structure:

```
{
  "version": "1.0",
  "call_id": "String",
  "messages": [
    {
      "timestamp": "Date",
      "direction": "sent" | "received",
      "party": UnsignedInt,
      "method": "String",
      "status_code": UnsignedInt,
      "status_text": "String",
      "headers": { ... },
      "body": "String"
    }
  ]
}
```

The fields of the trace object are:

version: The version of the trace format. This document defines version "1.0".

call_id: The SIP Call-ID for this trace. MUST match the sip_call_id parameter on the associated Dialog Object if present.

messages: An array of SIP message objects in chronological order.

The fields of each message object are:

timestamp: The date and time the message was sent or received, in

the format defined in Section 4.3.2 of [I-D.draft-ietf-vcon-vcon-core].

direction: Either "sent" or "received", from the perspective of the vCon producer.

party: The index into the vCon parties array for the party that sent this message. This parameter is optional when the sender cannot be determined.

method: The SIP method name (e.g. "INVITE", "BYE"). Present for requests. MUST NOT be present for responses.

status_code: The SIP response status code (e.g. 200, 486). Present for responses. MUST NOT be present for requests.

status_text: The SIP response reason phrase (e.g. "OK", "Busy Here"). Present for responses. MUST NOT be present for requests.

headers: A JSON object containing selected SIP headers as key-value pairs. Header names SHOULD use their canonical form. Multi-valued headers SHOULD use JSON arrays. This field is optional; producers MAY include all headers or a subset relevant to their use case.

body: The SIP message body as a string, if present. For SDP bodies, the complete SDP text is included. For binary bodies, Base64url encoding SHOULD be used with a separate "body_encoding" field set to "base64url".

Example trace attachment:

```
{
  "purpose": "sip-message-trace",
  "dialog": 0,
  "mediatype": "application/json",
  "encoding": "json",
  "body": {
    "version": "1.0",
    "call_id": "a84b4c76e66710@pc33.example.com",
    "messages": [
      {
        "timestamp": "2026-01-15T14:30:00.001+00:00",
        "direction": "sent",
        "party": 0,
        "method": "INVITE",
        "headers": {
          "From": "<sip:alice@example.com>;tag=1928301774",
```

```
            "To": "<sip:bob@example.com>",
            "CSeq": "314159 INVITE",
            "Contact": "<sip:alice@pc33.example.com>"
          }
        },
        {
          "timestamp": "2026-01-15T14:30:00.050+00:00",
          "direction": "received",
          "party": 1,
          "status_code": 180,
          "status_text": "Ringing",
          "headers": {
            "From": "<sip:alice@example.com>;tag=1928301774",
            "To": "<sip:bob@example.com>;tag=a6c85cf",
            "CSeq": "314159 INVITE"
          }
        },
        {
          "timestamp": "2026-01-15T14:30:05.200+00:00",
          "direction": "received",
          "party": 1,
          "status_code": 200,
          "status_text": "OK",
          "headers": {
            "From": "<sip:alice@example.com>;tag=1928301774",
            "To": "<sip:bob@example.com>;tag=a6c85cf",
            "CSeq": "314159 INVITE",
            "Contact": "<sip:bob@192.0.2.4>"
          }
        }
      ]
    }
  }
```

6.3. SDP Attachments

Session Description Protocol [RFC8866] bodies from SIP signaling MAY be stored as separate attachments when detailed media negotiation data is needed independently of the SIP messages that carried them.

The purpose value for SDP attachments is "sip-sdp".

The media type MUST be "application/sdp" as defined in [RFC8866].

Storing SDP separately is useful when media negotiation details are needed for quality analysis, codec identification, or network troubleshooting, without requiring storage of the complete SIP message exchange.

```
{
  "purpose": "sip-sdp",
  "start": "2026-01-15T14:30:00.000+00:00",
  "party": 0,
  "dialog": 0,
  "mediatype": "application/sdp",
  "encoding": "none",
  "body": "v=0\r\no=alice 53655765 2353687637 IN IP4 ...\r\n..."
}
```

6.4. SIP Header Summary

A summary of selected SIP headers MAY be stored as an attachment when full SIP messages are not needed but specific header values should be preserved.

The purpose value for header summary attachments is "sip-headers".

The media type MUST be "application/json".

The body is a JSON object where keys are SIP header names and values are the header values. Multi-valued headers use JSON arrays. The producer selects which headers to include based on their use case.

```
{
  "purpose": "sip-headers",
  "dialog": 0,
  "mediatype": "application/json",
  "encoding": "json",
  "body": {
    "Call-ID": "a84b4c76e66710@pc33.example.com",
    "From": "<sip:alice@example.com>;tag=1928301774",
    "To": "<sip:bob@example.com>;tag=a6c85cf",
    "P-Asserted-Identity": "<sip:+12125551234@example.com>",
    "History-Info": [
      "<sip:+12125551234@example.com>;index=1",
      "<sip:+12125559876@example.com>;index=1.1"
    ]
  }
}
```

7. STIR/SHAKEN Extended Data

The STIR (Secure Telephony Identity Revisited) framework and the SHAKEN (Signature-based Handling of Asserted information using toKENs) framework provide mechanisms for authenticating caller identity in SIP-based telephony. The core vCon specification provides the Party Object "stir" parameter for basic PASSporT storage. This section defines attachment types for extended STIR/SHAKEN data that does not fit in the compact "stir" parameter.

7.1. Relationship to Party stir Parameter

The Party Object "stir" parameter (Section 4.2.3 of [I-D.draft-ietf-vcon-vcon-core]) remains the primary location for the PASSporT token in JWS Compact Serialization form. This extension does not change the semantics of that parameter.

The attachments defined in this section carry supplementary data: certificate chains used to validate the PASSporT, verification results from the terminating side, and extended PASSporT data that does not fit the compact serialization.

Producers SHOULD always populate the Party Object "stir" parameter when a PASSporT is available, even if extended data is also stored as attachments. This ensures that implementations that do not support this extension can still access the basic authentication token.

7.2. STIR Certificate Attachments

The X.509 certificate or certificate chain used to sign the PASSporT MAY be stored as an attachment.

The purpose value is "stir-certificate".

The media type for a single certificate MUST be "application/pkix-cert" as defined in [RFC5280]. For a certificate chain, the media type MUST be "application/pem-certificate-chain" as defined in [RFC8555].

The party parameter SHOULD reference the Party Object whose "stir" parameter contains the PASSporT that this certificate validates.

```
{
  "purpose": "stir-certificate",
  "party": 0,
  "dialog": 0,
  "mediatype": "application/pem-certificate-chain",
  "encoding": "none",
  "body": "-----BEGIN CERTIFICATE-----\nMIIB...\n
    -----END CERTIFICATE-----\n
    -----BEGIN CERTIFICATE-----\nMIIC...\n
    -----END CERTIFICATE-----\n"
}
```

Implementations that retrieve certificates from the STIR certificate repository [SHAKEN-CERT] SHOULD store the full chain to enable offline validation.

7.3. STIR Verification Report Attachments

The results of PASSporT verification performed by the terminating service provider or verifying entity MAY be stored as an attachment.

The purpose value is "stir-verification-report".

The media type MUST be "application/json".

The verification report body is a JSON object with the following fields:

verifier: A string identifying the entity that performed verification.

timestamp: The date and time of verification in the format defined in Section 4.3.2 of [I-D.draft-ietf-vcon-vcon-core].

result: A string with one of the following values:

* "verified" - the PASSporT signature was successfully validated and the certificate chain is trusted.
* "failed" - the PASSporT signature validation failed.
* "no-signature" - no PASSporT was present in the call signaling.
* "stale" - the PASSporT iat (issued at) claim was outside the acceptable freshness window.
* "certificate-error" - the certificate could not be validated or the chain was incomplete.

attestation: The attestation level from the PASSporT, one of "A" (Full Attestation), "B" (Partial Attestation), or "C" (Gateway Attestation), as defined in [RFC8588].

reason: An optional free-form string providing additional detail about the verification result.

orig_tn: The originating telephone number from the PASSporT.

dest_tn: The destination telephone number(s) from the PASSporT.

```
{
  "purpose": "stir-verification-report",
  "party": 0,
  "dialog": 0,
  "mediatype": "application/json",
  "encoding": "json",
  "body": {
    "verifier": "example-telco-verification-service",
    "timestamp": "2026-01-15T14:30:00.100+00:00",
    "result": "verified",
    "attestation": "A",
    "orig_tn": "+12125551234",
    "dest_tn": ["+12125559876"]
  }
}
```

7.4. Extended PASSporT Attachments

When PASSporT extensions such as Rich Call Data [STIR-PASS-RCD] produce data that exceeds what is practical for the compact JWS serialization in the Party Object "stir" parameter, the full PASSporT MAY be stored as an attachment using the JWS JSON Serialization form.

The purpose value is "stir-passport-extended".

The media type MUST be "application/passport" as defined in [RFC8225].

The Party Object "stir" parameter SHOULD still contain the compact form when possible, with the extended attachment providing the complete data.

```
{
  "purpose": "stir-passport-extended",
  "party": 0,
  "dialog": 0,
  "mediatype": "application/passport",
  "encoding": "none",
  "body": "{ ... full JWS JSON Serialization ... }"
}
```

8. Implementation Guidance

This section provides non-normative guidance for implementers.

8.1. Minimal Producers

A minimal implementation producing vCons with SIP signaling data SHOULD include at minimum:

* The initial INVITE and the final response (2xx or error) as individual SIP message attachments with purpose values "sip-invite" and "sip-response".

* The sip_call_id parameter on the Dialog Object.

* The existing Party Object "stir" parameter with the PASSporT, when STIR/SHAKEN is deployed.

This minimal set provides sufficient data for basic call correlation, authentication verification, and troubleshooting.

8.2. Full Producers

A full implementation MAY additionally include:

* Complete SIP message traces using the "sip-message-trace" purpose.

* All Dialog Object extension parameters (sip_call_id, sip_from_tag, sip_to_tag, sip_cseq).

* All Party Object extension parameters (sip_contact, sip_user_agent, sip_display_name).

* STIR certificate chains using the "stir-certificate" purpose.

* Verification reports using the "stir-verification-report" purpose.

* Separate SDP attachments for media analysis.

* SIP header summaries for selected headers of interest.

Producers SHOULD select the level of detail based on the intended use case. Regulatory compliance and fraud investigation use cases typically require more comprehensive data than basic call center quality assurance.

8.3. Consumers

Implementations that consume vCons SHOULD:

* Detect the presence of this extension by checking for "sip-signaling" in the extensions parameter.

* Gracefully handle vCons that do not include this extension.

* Ignore attachment purpose values and Party/Dialog parameters that are not recognized.

* Not require the presence of any extension parameter; all parameters are optional.

Consumers SHOULD NOT reject a vCon solely because it contains unrecognized purpose values or parameters, in keeping with the Compatible extension classification.

8.4. Storage Considerations

SIP messages are relatively small (typically under 10KB), so inline storage using the "body" and "encoding" parameters is generally appropriate. SIP message traces for complex call flows with many transactions may be larger, and producers MAY use external references (url and content_hash) for traces exceeding a deployment-specific size threshold.

For inline storage, SIP messages in their wire format SHOULD use "none" encoding when the message is valid UTF-8, or "base64url" encoding otherwise. JSON-formatted attachments (traces, headers, verification reports) SHOULD use "json" encoding.

Producers that generate large volumes of vCons with SIP signaling data SHOULD consider compression of the vCon container. The GZIP format described in the Informative References of [I-D.draft-ietf-vcon-vcon-core] is suitable for this purpose.

9. Security Considerations

SIP signaling data contains information that may be sensitive from both security and privacy perspectives.

SIP messages may contain authentication credentials, particularly in Authorization and Proxy-Authorization headers. Producers MUST NOT include authentication credentials in SIP message attachments. Producers SHOULD strip or redact Authorization, Proxy-Authorization, and WWW-Authenticate headers before storing SIP messages in vCon attachments.

SIP signaling data may reveal network topology information through Via, Record-Route, and Path headers. Organizations SHOULD evaluate whether this information should be included or redacted based on their security policies.

The vCon signing mechanism (Section 5.2 of [I-D.draft-ietf-vcon-vcon-core]) SHOULD be used to ensure the integrity of SIP signaling data within the vCon. This is particularly important when SIP data is used for regulatory compliance or legal proceedings, where tamper evidence is required.

STIR certificates stored as attachments enable offline verification of caller identity. The integrity of these certificates SHOULD be protected using the vCon signing mechanism or the content_hash parameter for externally referenced certificates.

10. Privacy Considerations

SIP signaling frequently contains personally identifiable information (PII) including telephone numbers, SIP URIs, display names, IP addresses, and geolocation data. Implementors should consult [I-D.draft-ietf-vcon-privacy-primer] for general guidance on privacy best practices for vCon developers.

The vCon redaction mechanism (Section 4.1.8 of [I-D.draft-ietf-vcon-vcon-core]) SHOULD be used to create redacted versions of vCons when SIP signaling data must be shared with parties that should not have access to PII.

The vCon encrypted form (Section 5.3 of [I-D.draft-ietf-vcon-vcon-core]) SHOULD be used to protect SIP signaling data in transit and at rest.

Producers SHOULD apply data minimization principles by including only the SIP signaling data needed for the intended purpose. Diagnostic use cases may require different data than compliance use cases. The

attachment-based architecture of this extension supports selective inclusion by allowing producers to choose which attachment types to create.

SIP headers such as P-Asserted-Identity, Remote-Party-ID, and Geolocation contain particularly sensitive data. Producers SHOULD carefully evaluate whether these headers should be included in header summaries or SIP message attachments.

11. References

11.1. Normative References

[I-D.draft-ietf-vcon-vcon-core]
Petrie, D. G., "The JSON format for vCon - Conversation Data Container", Work in Progress, Internet-Draft, draft-ietf-vcon-vcon-core-02, January 2026, <https://datatracker.ietf.org/doc/draft-ietf-vcon-vcon-core/>.

[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>.

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

[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>.

[RFC8225] Wendt, C. and J. Peterson, "PASSporT: Personal Assertion Token", RFC 8225, DOI 10.17487/RFC8225, February 2018, <https://www.rfc-editor.org/rfc/rfc8225>.

[RFC8588] Wendt, C. and M. Barnes, "Personal Assertion Token (PaSSporT) Extension for Signature-based Handling of Asserted information using toKENs (SHAKEN)", RFC 8588, DOI 10.17487/RFC8588, May 2019, <https://www.rfc-editor.org/rfc/rfc8588>.

[RFC8866] Begen, A., Kyzivat, P., Perkins, C., and M. Handley, "SDP: Session Description Protocol", RFC 8866, DOI 10.17487/RFC8866, January 2021, <https://www.rfc-editor.org/rfc/rfc8866>.

11.2. Informative References

[I-D.draft-ietf-vcon-overview]
McCarthy-Howe, T., "The vCon - Conversation Data Container - Overview", Work in Progress, Internet-Draft, draft-ietf-vcon-overview-01, March 2026, <https://datatracker.ietf.org/doc/draft-ietf-vcon-overview/>.

[I-D.draft-ietf-vcon-privacy-primer]
James, D. and T. McCarthy-Howe, "Privacy Primer for vCon Developers", Work in Progress, Internet-Draft, draft-ietf-vcon-privacy-primer-00, July 2025, <https://datatracker.ietf.org/doc/draft-ietf-vcon-privacy-primer/>.

[RFC3311] Rosenberg, J., "The Session Initiation Protocol (SIP) UPDATE Method", RFC 3311, DOI 10.17487/RFC3311, October 2002, <https://www.rfc-editor.org/rfc/rfc3311>.

[RFC3515] Sparks, R., "The Session Initiation Protocol (SIP) Refer Method", RFC 3515, DOI 10.17487/RFC3515, April 2003, <https://www.rfc-editor.org/rfc/rfc3515>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/rfc/rfc5280>.

[RFC7989] 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>.

[RFC8555] Barnes, R., Hoffman-Andrews, J., McCarney, D., and J. Kasten, "Automatic Certificate Management Environment (ACME)", RFC 8555, DOI 10.17487/RFC8555, March 2019, <https://www.rfc-editor.org/rfc/rfc8555>.

[SHAKEN-CERT]
Barnes, M., Wendt, C., and J. Peterson, "SHAKEN: Secure Handling of Asserted information using toKENs", n.d..

[STIR-PASS-RCD]
Wendt, C. and J. Peterson, "PASSporT Extension for Rich Call Data", Work in Progress, Internet-Draft, draft-ietf-stir-passport-rcd-26, June 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-stir-passport-rcd-26>.

[TRACED] United States Congress, "Pallone-Thune Telephone Robocall Abuse Criminal Enforcement and Deterrence Act (TRACED Act)", Public Law 116-105, December 2019.

Appendix A. IANA Considerations

A.1. vCon Extension Names Registry

This document registers the following entry in the vCon Extension Names Registry defined in Section 6.4 of [I-D.draft-ietf-vcon-vcon-core].

Extension Name: sip-signaling

Extension Description: SIP signaling metadata, STIR/SHAKEN certificates, and related telephony data captured as vCon attachments and extended Party/Dialog Object parameters.

Change Controller: IESG

Specification Document(s): RFC XXXX (this document)

A.2. Party Object Parameter Names Registry

This document registers the following entries in the Party Object Parameter Names Registry defined in Section 6.3.3 of [I-D.draft-ietf-vcon-vcon-core].

Parameter Name	Parameter Description	Change Controller	Specification Document(s)
sip_contact	SIP Contact header URI	IESG	Section 4.1, RFC XXXX
sip_user_agent	SIP User-Agent header value	IESG	Section 4.2, RFC XXXX
sip_display_name	SIP display name from headers	IESG	Section 4.3, RFC XXXX

Table 1

A.3. Dialog Object Parameter Names Registry

This document registers the following entries in the Dialog Object Parameter Names Registry defined in Section 6.3.4 of [I-D.draft-ietf-vcon-vcon-core].

Parameter Name	Parameter Description	Change Controller	Specification Document(s)
sip_call_id	SIP Call-ID header value	IESG	Section 5.1, RFC XXXX
sip_from_tag	SIP From header tag parameter	IESG	Section 5.2, RFC XXXX
sip_to_tag	SIP To header tag parameter	IESG	Section 5.3, RFC XXXX
sip_cseq	SIP CSeq number from dialog-creating request	IESG	Section 5.4, RFC XXXX

Table 2

A.4. Attachment Purpose Values

This document does not define a formal registry for Attachment Object purpose values, as the purpose parameter in [I-D.draft-ietf-vcon-vcon-core] is a free-form string (Section 4.4.1 of [I-D.draft-ietf-vcon-vcon-core]). However, the following purpose values are defined by this document and their semantics MUST be preserved by implementations claiming conformance with this extension.

Purpose Value	Description
sip-invite	SIP INVITE request
sip-response	SIP response message
sip-ack	SIP ACK request
sip-bye	SIP BYE request
sip-cancel	SIP CANCEL request
sip-update	SIP UPDATE request
sip-refer	SIP REFER request
sip-message-trace	Structured SIP message exchange
sip-sdp	Session Description Protocol body
sip-headers	Selected SIP header summary
stir-certificate	STIR/SHAKEN certificate or chain
stir-verification-report	PASSporT verification results
stir-passport-extended	Extended PASSporT (JSON Serialization)

Table 3

If a future version of [I-D.draft-ietf-vcon-vcon-core] establishes a formal registry for attachment purpose values, the values defined in this document SHOULD be registered in that registry.

Appendix B. Example vCons

B.1. Minimal SIP Call

This example shows a minimal vCon for a two-party SIP call with basic signaling metadata. It includes the INVITE and 200 OK as attachments, the sip_call_id on the dialog, and a PASSporT on the originating party.


```
{
  "vcon": "0.0.2",
  "extensions": ["sip-signaling"],
  "parties": [
    {
      "tel": "+12125551234",
      "sip": "alice@example.com",
      "name": "Alice",
      "stir": "eyJhbGciOiJFUzI1NiIsInBwdCI6InNoYWtlbiIsIn..."
    },
    {
      "tel": "+12125559876",
      "sip": "bob@biloxi.example.com",
      "name": "Bob"
    }
  ],
  "dialog": [
    {
      "type": "recording",
      "start": "2026-01-15T14:30:05.200+00:00",
      "duration": 312.5,
      "parties": [0, 1],
      "mediatype": "audio/x-wav",
      "encoding": "base64url",
      "body": "UklGRi...",
      "sip_call_id": "a84b4c76e66710@pc33.example.com"
    }
  ],
  "analysis": [],
  "attachments": [
    {
      "purpose": "sip-invite",
      "start": "2026-01-15T14:30:00.000+00:00",
      "party": 0,
      "dialog": 0,
      "mediatype": "message/sip",
      "encoding": "base64url",
      "body": "SU5WSVRFIHNpcDpib2JA..."
    },
```

```
      {
        "purpose": "sip-response",
        "start": "2026-01-15T14:30:05.200+00:00",
        "party": 1,
        "dialog": 0,
        "mediatype": "message/sip",
        "encoding": "base64url",
        "body": "U0lQLzIuMCAyMDAgT0sN..."
      }
    ]
  }
```

B.2. Full SIP Trace with STIR/SHAKEN

This example shows a more comprehensive vCon with a full SIP message trace, STIR/SHAKEN certificate chain, verification report, and all extension parameters.

```
{
  "vcon": "0.0.2",
  "extensions": ["sip-signaling"],
  "parties": [
    {
      "tel": "+12125551234",
      "sip": "alice@example.com",
      "name": "Alice Smith",
      "stir": "eyJhbGciOiJFUzI1NiIsInBwdCI6InNoYWtlbiIsIn...",
      "sip_contact": "sip:alice@198.51.100.5:5060",
      "sip_user_agent": "ExamplePhone/2.1.0",
      "sip_display_name": "Alice Smith"
    },
    {
      "tel": "+12125559876",
      "sip": "bob@biloxi.example.com",
      "name": "Bob Jones",
      "sip_contact": "sip:bob@203.0.113.10:5060",
      "sip_user_agent": "BiloProvider-UA/3.4",
      "sip_display_name": "Bob Jones"
    }
  ],
  "dialog": [
    {
      "type": "recording",
      "start": "2026-01-15T14:30:05.200+00:00",
      "duration": 312.5,
      "parties": [0, 1],
      "mediatype": "audio/x-wav",
      "encoding": "base64url",
```

```
        "body": "UklGRi...",
        "sip_call_id": "a84b4c76e66710@pc33.example.com",
        "sip_from_tag": "1928301774",
        "sip_to_tag": "a6c85cf",
        "sip_cseq": 314159,
        "session_id": {
          "local": "aeffa location-ca11-ab01-0123456789ab",
          "remote": "bef1a location-ca11-ab01-0123456789cd"
        }
      }
    ],
    "analysis": [],
    "attachments": [
      {
        "purpose": "sip-message-trace",
        "dialog": 0,
        "mediatype": "application/json",
        "encoding": "json",
        "body": {
          "version": "1.0",
          "call_id": "a84b4c76e66710@pc33.example.com",
          "messages": [
            {
              "timestamp": "2026-01-15T14:30:00.001+00:00",
              "direction": "sent",
              "party": 0,
              "method": "INVITE",
              "headers": {
                "From": "\"Alice Smith\" <sip:alice@example.com>;tag=1928301774",
                "To": "<sip:bob@biloxi.example.com>",
                "CSeq": "314159 INVITE",
                "Contact": "<sip:alice@198.51.100.5:5060>",
                "Identity": "eyJhbGciOiJFUzI1NiIsInBwdCI6InNoYWtlbiIsIn..."
              }
            },
            {
              "timestamp": "2026-01-15T14:30:00.050+00:00",
              "direction": "received",
              "party": 1,
              "status_code": 100,
              "status_text": "Trying"
            },
            {
              "timestamp": "2026-01-15T14:30:01.200+00:00",
              "direction": "received",
              "party": 1,
              "status_code": 180,
              "status_text": "Ringing",
```

```
              "headers": {
                "To": "<sip:bob@biloxi.example.com>;tag=a6c85cf"
              }
            },
            {
              "timestamp": "2026-01-15T14:30:05.200+00:00",
              "direction": "received",
              "party": 1,
              "status_code": 200,
              "status_text": "OK",
              "headers": {
                "To": "<sip:bob@biloxi.example.com>;tag=a6c85cf",
                "Contact": "<sip:bob@203.0.113.10:5060>"
              }
            },
            {
              "timestamp": "2026-01-15T14:30:05.250+00:00",
              "direction": "sent",
              "party": 0,
              "method": "ACK"
            },
            {
              "timestamp": "2026-01-15T14:35:17.700+00:00",
              "direction": "sent",
              "party": 0,
              "method": "BYE"
            },
            {
              "timestamp": "2026-01-15T14:35:17.750+00:00",
              "direction": "received",
              "party": 1,
              "status_code": 200,
              "status_text": "OK"
            }
          ]
        }
      },
      {
        "purpose": "stir-certificate",
        "party": 0,
        "dialog": 0,
        "mediatype": "application/pem-certificate-chain",
        "encoding": "none",
        "body": "-----BEGIN CERTIFICATE-----\nMIIBxTCCAWugAwI...\n-----END CERTIFICATE--
---\n-----BEGIN CERTIFICATE-----\nMIICIjCCAcigAwI...\n-----END CERTIFICATE-----\n"
      },
      {
        "purpose": "stir-verification-report",
        "party": 0,
```

```
        "dialog": 0,
        "mediatype": "application/json",
        "encoding": "json",
        "body": {
          "verifier": "biloxi-telco-verification-service",
          "timestamp": "2026-01-15T14:30:00.100+00:00",
          "result": "verified",
          "attestation": "A",
          "orig_tn": "+12125551234",
          "dest_tn": ["+12125559876"]
        }
      },
      {
        "purpose": "sip-sdp",
        "start": "2026-01-15T14:30:00.001+00:00",
        "party": 0,
        "dialog": 0,
        "mediatype": "application/sdp",
        "encoding": "none",
        "body": "v=0\r\no=alice 53655765 2353687637 IN IP4 198.51.100.5\r\ns=-\r\nc=IN I
P4 198.51.100.5\r\nt=0 0\r\nm=audio 49170 RTP/AVP 0 8 97\r\na=rtpmap:0 PCMU/8000\r\na=rtp
map:8 PCMA/8000\r\na=rtpmap:97 opus/48000/2\r\n"
      }
    ]
  }
```


Appendix C. JSON Schema Extension

The following JSON Schema fragment extends the vCon JSON Schema defined in Appendix B of [I-D.draft-ietf-vcon-vcon-core] with the parameters defined in this document.

Party Object extension (merge with existing Party Object schema):

```
{
  "properties": {
    "sip_contact": {
      "type": "string",
      "description": "SIP Contact header URI"
    },
    "sip_user_agent": {
      "type": "string",
      "description": "SIP User-Agent header value"
    },
    "sip_display_name": {
      "type": "string",
      "description": "Display name from SIP From/To header"
    }
  }
}
```

Dialog Object extension (merge with existing Dialog Object schema):

```
{
  "properties": {
    "sip_call_id": {
      "type": "string",
      "description": "SIP Call-ID header value"
    },
    "sip_from_tag": {
      "type": "string",
      "description": "SIP From header tag parameter"
    },
    "sip_to_tag": {
      "type": "string",
      "description": "SIP To header tag parameter"
    },
    "sip_cseq": {
      "type": "integer",
      "minimum": 0,
      "description": "CSeq number from dialog-creating request"
    }
  }
}
```

SIP Message Trace schema (for sip-message-trace attachment body):

```
{
  "type": "object",
  "required": ["version", "call_id", "messages"],
  "properties": {
    "version": {
      "type": "string",
      "enum": ["1.0"]
    },
    "call_id": {
      "type": "string"
    },
    "messages": {
      "type": "array",
      "items": {
        "type": "object",
        "required": ["timestamp", "direction"],
        "properties": {
          "timestamp": {
            "type": "string",
            "format": "date-time"
          },
          "direction": {
```

```
              "type": "string",
              "enum": ["sent", "received"]
            },
            "party": {
              "type": "integer",
              "minimum": 0
            },
            "method": {
              "type": "string"
            },
            "status_code": {
              "type": "integer",
              "minimum": 100,
              "maximum": 699
            },
            "status_text": {
              "type": "string"
            },
            "headers": {
              "type": "object"
            },
            "body": {
              "type": "string"
            },
            "body_encoding": {
              "type": "string",
              "enum": ["base64url"]
            }
          }
        }
      }
    }
  }
```

STIR Verification Report schema (for stir-verification-report attachment body):

```
{
  "type": "object",
  "required": ["verifier", "timestamp", "result"],
  "properties": {
    "verifier": {
      "type": "string"
    },
    "timestamp": {
      "type": "string",
      "format": "date-time"
    },
    "result": {
      "type": "string",
      "enum": [
        "verified",
        "failed",
        "no-signature",
        "stale",
        "certificate-error"
      ]
    },
    "attestation": {
      "type": "string",
      "enum": ["A", "B", "C"]
    },
    "reason": {
      "type": "string"
    },
    "orig_tn": {
      "type": "string"
    },
    "dest_tn": {
      "oneOf": [
        { "type": "string" },
        {
          "type": "array",
          "items": { "type": "string" }
        }
      ]
    }
  }
}
```

Author’s Address

Thomas McCarthy-Howe
VCONIC
Email: ghostofbasho@gmail.com