

vCon Agent Session
draft-howe-vcon-agent-session-00

Abstract

This document defines an "agent_session" extension for Virtualized Conversations (vCon) that profiles the use of vCon parties, dialog, analysis, and attachments to carry the record of an autonomous AI agent's internal session - its prompts, tool invocations, tool results, reasoning, and file or artifact provenance - alongside the human-facing conversation that the agent participated in or acted upon.

The extension is a Compatible vCon extension. It introduces no new top-level fields and does not alter the semantics of existing ones. Instead, it specifies (a) how an autonomous agent is represented as a vCon party, (b) how agent message turns are placed in the dialog array, (c) how the internal agent trace (tool calls, results, reasoning, system events) is carried as a structured analysis entry whose body conforms to the Verifiable Agent Conversations (VAC) CDDL schema [I-D.draft-birkholz-verifiable-agent-conversations], and (d) how files and artifacts modified by the agent are carried as attachments.

By projecting agent-session data into the vCon model, implementations inherit vCon's party/identity model, the lawful basis framework [I-D.draft-howe-vcon-lawful-basis], the lifecycle and redaction machinery [I-D.draft-howe-vcon-lifecycle], and JWS-based signing, without re-specifying any of those concerns.

About This Document

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

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

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-agent-session>.

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 21 November 2026.

Copyright Notice

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

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

Table of Contents

1. Introduction	3
2. Conventions and Definitions	4
2.1. Core Terms	4
3. Extension Classification and Registration	4
4. Representing the Agent as a Party	5
5. Representing Agent Dialog Turns	6
6. Representing the Internal Agent Trace	7
6.1. Analysis Entry Shape	7
6.2. Granularity Variants	7
6.3. CBOR Encoding (Optional)	8
7. Representing File and Artifact Provenance	8
7.1. Purpose Registry	9

8. Signing, Transparency, and SCITT Integration	9
9. Lawful Basis and Consent	9
10. Lifecycle, Redaction, and Reasoning Encryption	10
11. Security Considerations	10
12. IANA Considerations	11
13. References	11
13.1. Normative References	11
13.2. Informative References	12
Acknowledgments	12
Author's Address	13

1. Introduction

Autonomous AI agents now read, write, and act on conversations involving human participants. Two categories of evidentiary record are typically required:

1. The conversation between people (and between people and the agent), captured today by vCon [I-D.draft-ietf-vcon-vcon-core] (see also [I-D.draft-ietf-vcon-overview] for background).
2. The agent's internal session - the prompts it received, the tools it invoked, the results it observed, the chains of reasoning it produced, and the files or external systems it modified - captured by Verifiable Agent Conversations (VAC) [I-D.draft-birkholz-verifiable-agent-conversations].

These two records have substantial overlap (parties, turns, artifacts, provenance, signing) and substantial complementarity (vCon has a rich party and consent model; VAC has a rich tool-call and reasoning model). Maintaining them as fully independent containers forces every implementer to invent linkage, duplicate identity, and re-solve consent.

This document defines a Compatible vCon extension (Section 2.5 of [I-D.draft-ietf-vcon-vcon-core]) that allows a single vCon to carry both records together: the human-facing conversation in the usual vCon shape, and the agent's internal session as structured analysis and attachment entries whose bodies conform to the VAC schema.

Regulatory drivers include the [EU-AI-ACT], the [NIST-AI-RMF], and sectoral regimes (HIPAA, PCI DSS, ISO 42001) that require auditable, non-repudiable records of automated decision-making.

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.

All timestamps in vCon documents conforming to this extension are formatted as Internet date and time strings per [RFC3339], matching the requirement in [I-D.draft-ietf-vcon-vcon-core].

2.1. Core Terms

***Agent*:** An autonomous software system, typically backed by a large language model, that produces messages, invokes tools, and/or modifies external state in response to instructions.

***Agent Session*:** A single bounded run of an agent, with a defined start and end, identified by a session identifier.

***Agent Party*:** A vCon parties[] entry of role "agent" that identifies an agent participating in the conversation.

***Agent Trace*:** The ordered, possibly tree-structured set of internal entries produced during an agent session - tool calls, tool results, reasoning, and system events. Distinct from the agent's outward-facing dialog turns.

***VAC Record*:** A verifiable-agent-record as defined in [I-D.draft-birkholz-verifiable-agent-conversations], or its projection into JSON for use as an analysis body.

***Compatible Extension*:** A vCon extension that introduces additional data without altering the meaning or structure of existing elements, as defined in [I-D.draft-ietf-vcon-vcon-core].

3. Extension Classification and Registration

The agent session extension is a ***Compatible Extension*** as defined in Section 2.5 of [I-D.draft-ietf-vcon-vcon-core]. It:

- * Introduces no new top-level fields.
- * Defines new values for the role field of a party, the purpose field of an attachment, and the type field of an analysis entry.

- * Can be safely ignored by implementations that do not support agent session processing; the vCon remains a well-formed conversation record.
- * Does not require listing in the critical parameter unless the application requires that consumers honor agent session semantics (e.g. for redaction or replay).

This document defines the "agent_session" extension token for registration in the vCon Extensions Names Registry:

- * ***Extension Name***: agent_session
- * ***Extension Description***: Representation of an autonomous AI agent's internal session (tool calls, reasoning, file provenance) within a vCon conversation record.
- * ***Change Controller***: IESG
- * ***Specification Document***: This document

vCon instances that include agent session data SHOULD include "agent_session" in the extensions array. They SHOULD include "agent_session" in critical if downstream consumers MUST process the agent trace (for example, when the only record of an authorizing tool call lives there).

4. Representing the Agent as a Party

Each distinct agent that participated in the conversation MUST be represented as a single entry in the vCon parties[] array, with role set to "agent". The party entry MAY include a meta object containing an agent_session member that captures the agent's identity:

```
{
  "name": "Claude Opus 4.6",
  "role": "agent",
  "validation": "system",
  "meta": {
    "agent_session": {
      "model_id": "claude-opus-4-6",
      "provider": "anthropic",
      "recording_agent": "claude-code/1.2.0",
      "environment": {
        "cwd": "/Users/example/project",
        "vcs_branch": "main",
        "vcs_commit": "abc123def456"
      }
    }
  }
}
```

Fields under parties[i].meta.agent_session:

- * **model_id** (string, REQUIRED): Vendor identifier for the model.
- * **provider** (string, REQUIRED): Organization providing the model (e.g. "anthropic", "openai", "google").
- * **recording_agent** (string, RECOMMENDED): The harness, IDE, or CLI that recorded the session (e.g. "claude-code/1.2.0").
- * **environment** (object, OPTIONAL): Runtime environment context. When the session edited a source repository, *vcs_branch* and *vcs_commit* SHOULD be set.

When multiple agents (e.g. orchestrator + sub-agent) participated, each MUST appear as a distinct party. Dialog and analysis entries reference them by index in the usual vCon way.

5. Representing Agent Dialog Turns

User prompts and assistant replies (the externally-visible side of the agent session - what a person would read in a transcript) are ordinary vCon dialog[] entries. The user is one party; the agent is another. No new dialog type is required.

When an agent reply is generated by a single agent in response to a single prompt, the dialog entry's parties field MUST reference the agent party. When the reply was produced by a multi-agent collaboration, parties MAY list all contributing agents.

6. Representing the Internal Agent Trace

The internal agent trace - tool calls, tool results, reasoning entries, and system events - is carried in the vCon analysis[] array. A single agent session SHOULD produce one analysis entry spanning all of its dialog turns, with the full trace embedded as a JSON-encoded VAC record in the body.

6.1. Analysis Entry Shape

```
{
  "type": "agent_trace",
  "dialog": [0, 1],
  "vendor": "anthropic",
  "product": "claude-opus-4-6",
  "schema": "https://datatracker.ietf.org/doc/...vac.../",
  "encoding": "json",
  "body": "{\\"version\\":\\"1.0\\",\\"session-trace\\":...}"
}
```

The analysis entry MUST set:

- * ***type***: "agent_trace".
- * ***dialog***: Array of dialog indices to which this trace applies.
- * ***vendor***: The model provider (mirrors parties[i].meta.agent_session.provider).
- * ***product***: The model identifier (mirrors model_id).
- * ***schema***: The URL of the VAC specification (or a specific version thereof), e.g. the canonical datatracker URL for [I-D.draft-birkholz-verifiable-agent-conversations].
- * ***encoding***: "json".
- * ***body***: A JSON-encoded verifiable-agent-record per the VAC schema. The CDDL-defined structure - session-trace.entries[] with message-entry, tool-call-entry, tool-result-entry, reasoning-entry, and event-entry variants - is preserved verbatim in the body. The parent-id / children tree relationships are retained.

6.2. Granularity Variants

Implementations MAY choose finer granularity:

- * ***Per-tool-call***: one analysis entry per tool invocation, each referencing the specific dialog turn it served. This enables fine-grained redaction (e.g. removing a single reasoning entry containing PII) using the lifecycle extension [I-D.draft-howe-vcon-lifecycle], but produces more analysis entries.
- * ***Per-branch***: one analysis entry per sub-agent branch in a multi-agent session.

The whole-session form is RECOMMENDED for archival. The per-tool-call form is RECOMMENDED when granular redaction or selective disclosure is anticipated.

6.3. CBOR Encoding (Optional)

When CBOR [RFC8949]-native representation is required (for example, for interoperability with COSE [RFC8152]-signed VAC records or SCITT transparency services), implementations MAY set encoding to "base64url" and place a base64url-encoded CBOR encoding of the VAC record in body, with schema qualified by ?encoding=cbor. Consumers MUST examine the schema URL to determine the encoding.

7. Representing File and Artifact Provenance

When an agent modifies files or produces artifacts (a source code edit, a generated document, a database row insertion), each such change SHOULD be represented as a vCon attachments[] entry with purpose set to "agent_file_change" (or a more specific purpose, see Section 7.1).

```
{
  "purpose": "agent_file_change",
  "party": 1,
  "dialog": 5,
  "encoding": "json",
  "body": {
    "path": "src/foo.py",
    "contributor": "agent",
    "line_range": [10, 25],
    "operation": "edit",
    "commit": "abc123",
    "content_hash": "sha512-..."
  }
}
```


The party index **MUST** identify the agent party that made the change. The dialog index **SHOULD** identify the dialog turn whose tool call effected the change (or, if the change is summary-level, the closing assistant turn).

For binary or large file content, the attachment **SHOULD** use vCon's external media pattern (url + content_hash) rather than inlining the body.

7.1. Purpose Registry

This document defines initial values for the purpose field of an attachment produced by an agent session:

- * agent_file_change - source file modified by the agent.
- * agent_artifact - non-file artifact generated by the agent (e.g. a database write, an API call payload, a generated document).
- * agent_environment - snapshot of relevant agent environment state (working directory listing, package manifest, etc.).

Additional purpose values **MAY** be registered through the usual vCon attachment purpose registry process.

8. Signing, Transparency, and SCITT Integration

The agent session extension inherits vCon's signing model: a vCon containing agent session data is signed as a whole, using JWS as defined in [I-D.draft-ietf-vcon-vcon-core].

When the agent trace was independently signed and submitted to a SCITT transparency service (as defined in [I-D.draft-birkholz-verifiable-agent-conversations] Section 9), the SCITT receipt and the original COSE_Sign1 envelope **MAY** be carried as additional attachments with purpose set to "scitt_receipt" and "agent_trace_cose_sign1" respectively. Consumers can then independently verify the agent trace against both the vCon JWS signature and the COSE/SCITT chain.

9. Lawful Basis and Consent

An agent session that processes personal data **MUST** be governed by a documented lawful basis [I-D.draft-howe-vcon-lawful-basis]. Implementations **SHOULD** include a lawful_basis attachment that:

- * Identifies the data subject(s) by party index.

- * Lists purpose_grants covering at minimum: agent_session_recording, agent_session_analysis, and (where applicable) agent_session_redistribution.
- * Has expiration set per applicable regulation.

When the agent session was authorized only under a lawful basis that prohibits redistribution, the vCon containing the agent session MUST NOT be transmitted to parties outside the scope of that grant. The vCon redacted mechanism and the lifecycle extension [I-D.draft-howe-vcon-lifecycle] provide the standard means for producing a redacted form for broader distribution.

10. Lifecycle, Redaction, and Reasoning Encryption

Internal reasoning entries (reasoning-entry in the VAC schema) frequently contain sensitive intermediate state. The lifecycle extension's redaction mechanism MAY be used to remove or replace specific reasoning entries while preserving the rest of the agent trace. The recommended approach is the per-tool-call analysis granularity described in Section "Analysis Entry Shape" so that each reasoning entry can be addressed individually.

Where reasoning entries must be retained but kept confidential from some recipients, implementations MAY encrypt the analysis body using JWE and place the encrypted form in body with encoding set to "jwe".

11. Security Considerations

Carrying an agent's internal trace within a vCon expands the vCon's privacy surface. In particular:

- * Tool call arguments and tool results frequently contain credentials, identifiers, and PII not present in the human-facing dialog. Implementations MUST scrub or redact these before any distribution outside the lawful basis grant.
- * Reasoning entries may reveal internal heuristics that are themselves sensitive (e.g. fraud-detection logic). Such entries SHOULD be encrypted or removed before distribution.
- * The agent's identity, as recorded in parties[i].meta.agent_session, is asserted by the recording party and is not by itself cryptographic evidence that a particular model produced the trace. Trace authenticity rests on the JWS signature over the vCon and, when present, on the COSE/SCITT receipt for the embedded VAC record.

12. IANA Considerations

This document registers the `agent_session` extension in the vCon Extensions Names Registry (see Section "Extension Classification and Registration").

This document registers the following values in the vCon analysis type registry:

- * `agent_trace`

This document registers the following values in the vCon attachment purpose registry:

- * `agent_file_change`

- * `agent_artifact`

- * `agent_environment`

- * `scitt_receipt`

- * `agent_trace_cose_sign1`

13. References

13.1. Normative References

[I-D.draft-birkholz-verifiable-agent-conversations]
Birkholz, H., "Verifiable Agent Conversations", 2026,
<<https://datatracker.ietf.org/doc/draft-birkholz-verifiable-agent-conversations/>>.

[I-D.draft-howe-vcon-lawful-basis]
McCarthy-Howe, T., "vCon Lawful Basis", 2026,
<<https://datatracker.ietf.org/doc/draft-howe-vcon-lawful-basis/>>.

[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>>.
- [RFC3339] Klyne, G., "Date and Time on the Internet: Timestamps", July 2002, <<https://www.rfc-editor.org/rfc/rfc3339.html>>.
- [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>>.

13.2. Informative References

- [EU-AI-ACT] European Union, "Regulation (EU) 2024/1689 (Artificial Intelligence Act)", 2024, <<https://artificialintelligenceact.eu/>>.
- [I-D.draft-howe-vcon-lifecycle] McCarthy-Howe, T., "vCon Lifecycle", 2026, <<https://datatracker.ietf.org/doc/draft-howe-vcon-lifecycle/>>.
- [I-D.draft-ietf-vcon-overview] McCarthy-Howe, T., "The vCon - Conversation Data Container - Overview", 2025, <<https://datatracker.ietf.org/doc/draft-ietf-vcon-overview/>>.
- [NIST-AI-RMF] National Institute of Standards and Technology, "AI Risk Management Framework 1.0", January 2023, <<https://www.nist.gov/itl/ai-risk-management-framework>>.
- [RFC8152] Schaad, J., "CBOR Object Signing and Encryption (COSE)", July 2017, <<https://www.rfc-editor.org/rfc/rfc8152.html>>.
- [RFC8949] Bormann, C., "Concise Binary Object Representation (CBOR)", December 2020, <<https://www.rfc-editor.org/rfc/rfc8949.html>>.

Acknowledgments

Thanks to the vCon working group and the authors of [I-D.draft-birkholz-verifiable-agent-conversations] for discussions that motivated this extension.

Author's Address

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