

SML
Internet-Draft
Intended status: Standards Track
Expires: 23 April 2026

H.-J. Happel
audriga GmbH
20 October 2025

Structured Email
draft-ietf-sml-structured-email-05

Abstract

This document specifies how a machine-readable version of the content of email messages can be added to those messages.

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 23 April 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	3
2. Conventions Used in This Document	3
3. Representing structured data	3
3.1. Knowledge representation language	4
3.2. Vocabularies	4
4. Structured data in email messages	5
4.1. Designation	5
4.2. Placement	5
4.2.1. Full representation	6
4.2.2. Partial representation	7
4.2.3. Non-representation	7
4.3. Identifiers	7
4.3.1. Using identifiers in structured data	7
4.3.2. Using structured data identifiers in text/html	8
5. Structured data across email messages	8
5.1. Forwarding	9
5.2. Replies	9
5.3. Updates	9
6. Message flags	10
7. Examples	10
7.1. Full representation	10
7.2. Full representation (machine-readable only)	11
7.3. Full representation with inline image	11
7.4. Partial representation	11
7.5. Non-representation	11
8. Appendix (Structured Email Signature)	11
9. Security and trust	12
10. Implementation status	12
10.1. Structured Email for Nextcloud Mail	13
10.2. Structured Email plugin for Roundcube Webmail	13
10.3. Yatagarasu Mail	13
11. Security considerations	13
12. Privacy considerations	14
13. IANA Considerations	14
13.1. Creation of the SML registry group	14
13.1.1. Creation of the SML Email Vocabulary Registry	14
13.1.2. Creation of the SML Vocabularies Registry	14
13.1.3. Initial Entries for the SML Vocabularies Registry	14
13.2. Registration of the Content-Purpose MIME header field	15
13.3. Registration of the \$hasStructuredData keyword	15
13.4. Registration of the \$MRM keyword	16
14. Informative References	17
Author's Address	19

1. Introduction

Information on websites and in email messages mostly addresses human readers. However, various attempts have been made to make such information (U+2013) fully or in part (U+2013) machine-readable, so that tools can assist users in dealing with that information more efficiently.

One widespread approach is the usage of [SchemaOrg] vocabulary, which can be embedded in the HTML markup of websites. It is then crawled by web search engines and used to improve the quality of search result snippets (e.g., by showing ratings, opening hours, or contact information).

Similarly, a number of web shops, hotels, and airlines include Schema.org vocabulary in order receipt email messages sent to customers. This information is extracted and used by some ISPs and open source tools ([SchemaOrgEmail]). However, these implementations differ in many details.

The goal of this specification is to provide a clear and comprehensive specification for this practice and to provide ground for potential future extensions.

2. Conventions Used in This Document

The terms "message" and "email message" refer to "electronic mail messages" or "emails" as specified in [RFC5322]. The term "Message User Agent" (MUA) denotes an email client application as per [RFC5598].

The terms "machine-readable data" and "structured data" are used in contrast to "human-readable" messages and denote information expressed "in a structured format (..) which can be consumed by another program using consistent processing logic" [MachineReadable].

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.

3. Representing structured data

In order to exchange structured data, one needs to choose a formal language and a serialization format. Based on this, vocabularies can be used to establish a shared understanding of structured data among different parties, such as email senders and receivers.

3.1. Knowledge representation language

The Resource Description Framework ([RDF]) is a formal language for knowledge representation standardized by the W3C. It is underlying [SchemaOrg] and thus already used for annotating websites and emails. Among the various serializations for RDF, JSON-LD ([JSONLD]) has become the most commonly used serialization used on websites ([WDCStats]).

Hence, structured data in email messages SHOULD be expressed in the JSON-LD serialization of RDF.

For discussion, see also:

<https://github.com/hhappel/draft-happel-structured-email/issues/1>

3.2. Vocabularies

Using RDF/JSON-LD, users are free to express any kind of information in structured data. For reuse and reference however, it is common to agree upon core concepts/entities and properties for a certain domain. Those are typically collected and shared in so-called vocabularies.

[SchemaOrg] is a widespread vocabulary, which was designed for annotating content on websites. A small subset of its concepts is already used by email senders and processed by email providers.

Users that want to add structured data into email message SHOULD use concepts from [SchemaOrg], if they fit their use case. They MAY however use any valid JSON-LD.

There might also be certain vocabularies for email-specific use cases (such as [I-D.ietf-sml-structured-vacation-notices-01]), that will be specifically endorsed by the IETF or by respective RFCs.

MUAs may choose freely if and how to use structured data extracted from messages. If they do not explicitly support a certain vocabulary, MUAs may also rely on extensions or passing data to outside applications, similar to the case of "email attachments" (i.e., MIME body parts with content-disposition type attachment [RFC2183]).

For discussion, see also:

<https://github.com/hhappel/draft-happel-structured-email/issues/2>

4. Structured data in email messages

This section defines aspects of adding structured data to a MIME message and its interrelation with other body parts.

As a basic distinction, we define two types of messages:

- * Machine-readable messages (MRM), for which their content is entirely machine-readable
- * Partly machine-readable messages (PMRM), which contain machine-readable content describing some of the human-readable content

4.1. Designation

This document targets structured data describing the content of an email message itself. Since users may add other arbitrary structured data (e.g., as MIME body parts of type `application/ld+json`) to an email message, we need to define which kinds of structured data are supposed to be representative of the email message content.

For this reason, senders **MUST** set a header field `Content-Purpose` to the value `Machine-readable` on an `application/ld+json` body part, which is meant to provide a machine-readable description of the message content. A MUA **SHOULD NOT** show such body parts as a file attachment in the list of email attachments.

The `Content-Purpose: Machine-readable` header field **MAY** also be set for body parts with other media types than `application/ld+json`. A system **SHOULD** treat such body parts as if their media type would be `application/ld+json` according to this specification, if they can extract JSON-LD data (e.g.: `application/jose`; [RFC7515]).

4.2. Placement

When aiming to describe human-readable content in a machine-readable way, there may exist three general relations between both types of content in which the machine-readable version of the content may be:

- * fully representative of the human-readable content
- * describing only parts of the human-readable content
- * describing none of the human-readable content

From the perspective of the machine-readable content, we call those cases "Full representation", "Partial representation" and "Non-representation". Those distinctions matter for MUAs, as they can make choices for the autoprocessing or presentation of messages and their body parts.

For discussion, see also:

<https://github.com/hhappel/draft-happel-structured-email/issues/3>

4.2.1. Full representation

Full representation denotes the case in which structured data describes the content of a certain body part, in the sense of providing an "_alternative_" version of the same information" as in the informal definition of multipart-alternative in [RFC2046].

If a message is sent to a definitely non-human recipient (e.g., an API), application/ld+json SHOULD be used as Content-Type in the message header.

If a message is sent to a human recipient, a sender MUST use a multipart/alternative for each body part that is _fully_ described by structured data. In this case, the multipart/alternative should contain a text/plain and a text/html version of the content for backwards compatibility, plus the application/ld+json body part containing the structured data representation.

If the MIME tree of a message consists entirely of body parts which have a "full representation" machine-readable alternative of their content, this is called a "machine-readable message" (MRM). A MRM may be suitable for special forms of automated processing, since it contains no information for which interpretation of a human reader is required.

While it may be tricky to derive if a message is a MRM in case of complex MIME trees, the majority of messages in practice will contain just one multipart/alternative body part, for which such conclusion is easy to derive.

If a MUA is able to process the vocabulary of a MRM or is able to process the structured data otherwise, it SHOULD prefer the application/ld+json representation, unless instructed otherwise by the user.

In case of more complex MIME structures, it is up to the discretion of the MUA how to process or render the message.

Some countries require senders to include legal disclaimers in email messages. In the case of "full representation", a sender MAY include a "structured email signature" as shown in the Appendix either in the "full representation" structure data or in an additional "non-representation" body part.

4.2.2. Partial representation

If structured data is intended to describe only a `_subset_` of a certain human-readable body part, it **MUST** be added as a multipart/related entity with the content type `application/ld+json`.

This multipart/related entity **MUST** also contain the human textual content of the body part (e.g., `text/plain` and `text/html`). Also, any MIME body part referenced from the structured data in the `application/ld+json` body part, **MUST** be enclosed in this multipart/related entity.

MUAs **SHOULD** render such messages as if no `application/ld+json` would be included. MUAs **MAY** process the `application/ld+json` data for providing an enhanced user experience of their resp. the user's choice.

4.2.3. Non-representation

In the case of non-representation, there is no overlap between structured data and the human readable content.

This may be useful for special scenarios, such as embedding "preemptive" structured vacation notices as described in [I-D.happel-sml-structured-vacation-notices-00] into email messages.

As in the case of partial representation, MUAs receiving such messages may take according action based on the structured data extracted.

4.3. Identifiers

There are existing use cases for cross-referencing between different parts of a MIME message, for which [RFC2392] defines the `cid:` and `mid:` URI schemes.

In a similar fashion, cross-referencing might occur between structured data and other message parts.

4.3.1. Using identifiers in structured data

Most nodes and properties in JSON-LD are identified using IRIs [RFC3987]. Since any [RFC2392] (`cid:/mid:`) reference forms a valid IRI, those references can be directly used in JSON-LD.

There are two main cases for which `cid:-`identifiers **SHOULD** be used in structured data.

First, if structured data references binary content such as images or other files, which already exist as MIME body parts within the same message.

Second, if a `cid:` value is used in a JSON-LD `@id` property, the corresponding JSON-LD node can be considered to describe the MIME body part identified by that `cid:`. This MAY be used to denote that certain structured data is explicitly describing that MIME body part. This MUST NOT be used for the main `text/plain` or `text/html` body parts, though.

For discussion, see also:

<https://github.com/hhappel/draft-happel-structured-email/issues/4>

4.3.2. Using structured data identifiers in `text/html`

In the case of "partial representation", a MUA will still primarily display the human readable part of a message (e.g., `text/plain` or `text/html`).

It might however be helpful if the MUA is able to determine which parts of human readable text refer to certain structured data - e.g., to offer actions based on structured data directly in the context of the corresponding human-readable content.

For this purpose, the sender may add a HTML `"data-id"` property (`[HTMLData]`) to any HTML entity in the `text/html` body, which references the `@id` property of a JSON-LD node in the structured data.

Besides referencing the corresponding JSON-LD node, a sender might also want to denote if the underlying data is "extensively" described or just mentioned in the human readable representation. For example the New York Times cooking newsletter typically `_features_` few recipes, while mentioning a larger number of recipes, also referencing their web URL.

For providing an adequate user experience, the MUA should be able to understand which recipes are featured in an email and which are just mentioned.

For discussion, see also:

<https://github.com/hhappel/draft-happel-structured-email/issues/5>

5. Structured data across email messages

This sections deals with aspects that go beyond the scope of an individual MIME message.

5.1. Forwarding

Forwarding messages including structured data needs to be considered from a privacy perspective, particularly in cases of "non-representation", when the user has no way to determine structured data from the human readable part of the message.

A MUA MUST strip non-representative structured data when a user is forwarding messages to somebody else in her MUA. Note that this does not apply to automated forwarding of messages.

Beyond that, privacy issues also apply to forwarding regular email messages. Improvements of the status quo might hence be considered beyond the specific context of structured email.

For discussion, see also:

<https://github.com/hhappel/draft-happel-structured-email/issues/6>

5.2. Replies

Users or agents might want to reply to structured emails with a structured email.

Details for this will be specified in a separate future draft.

5.3. Updates

In human-readable messages, human language can be used to update or recall information that was conveyed in prior messages. Accordingly, there needs to be a machine-readable mechanism that allows to express the update or recall of structured data.

To update or recall structured data, senders MUST set the SUPERSEDES header field ([RFC4021]) of the "update" message with the message id of the original email message. An "update" message with empty structured data can be used to signal a full recall of previously send structured data.

Every "update" message MUST have its own unique message id.

The processing of an "update" message by the receiving MUA is up to its own discretion, as meaningful action may depend an multiple factors.

MUAs MAY consider:

- * An update might be triggered by a previous action of the user

- * Adding the original message id as an identifier property to the structured data to preserve its origin

For discussion, see also:

<https://github.com/hhappel/draft-happel-structured-email/issues/9>

6. Message flags

In some use cases, MUAs might benefit from information about message details without having to evaluate the full message body.

For example, the `$hasAttachment` IMAP flag (`[HasAttachment]`) was proposed to signal the existence of MIME attachments in a message which otherwise would need to be redetermined based on complex MIME parsing.

In case of structured data, the receiving MTA (or any later MUA) MAY determine two facts when parsing the MIME tree of a message:

- * If the message contains any structured data according to this document
- * If so, if the message is a fully machine-readable message (MRM)

If found, the following IMAP flags MAY be set for a message:

- * `$hasStructuredData` (in the first case)
- * `$MRM` (in the second case)

The first case may be helpful for filtering or client-side preloading of message content. The second case may be particularly helpful for automated processing without user interaction.

For discussion, see also:

<https://github.com/hhappel/draft-happel-structured-email/issues/10>

7. Examples

The following section shows some example MIME hierarchies of email messages containing structured data.

7.1. Full representation

```
multipart/alternative/  
├─ text/plain  
├─ text/html  
└─ application/ld+json
```

7.2. Full representation (machine-readable only)

```
application/ld+json
```

7.3. Full representation with inline image

```
multipart/alternative/  
├── text/plain  
└── multipart/related/  
    ├── multipart/alternative/  
    │   ├── text/html  
    │   └── application/ld+json  
    └── image/png
```

7.4. Partial representation

```
multipart/related/  
├── multipart/alternative/  
│   ├── text/plain  
│   └── text/html  
└── application/ld+json
```

7.5. Non-representation

```
multipart/mixed/  
├── multipart/alternative/  
│   ├── text/plain  
│   └── text/html  
└── application/ld+json
```

8. Appendix (Structured Email Signature)

The following snippet of structured data uses the Schema.org publisher property of an EmailMessage.

```

{
  "@context": "https://schema.org/",
  "@type": "EmailMessage",
  "publisher": {
    "@type": "Organization",
    "legalName": "MUSEO NACIONAL DEL PRADO DIFUSIN, S.A.U., S.M.E.",
    "legalAddress": {
      "@type": "PostalAddress",
      "addressLocality": "Madrid, Spain",
      "postalCode": "28014",
      "streetAddress": "Casado del Alisal, 10, bajo B"
    },
    "legalRepresentative" : {
      "@context": "https://schema.org",
      "@type": "Person",
      "name": "Jane Doe",
    },
    "identifier": {
      "@type": "PropertyValue",
      "name": "Registration data in the Company Register",
      "value": "Volume 23578, Entry 1, Section 8, Sheet M-423094, 74 Folio"
    },
    "vatID": "A84888056"
  }
}

```

9. Security and trust

Email user agents that want to support structured email should follow guidance to ensure trust and security standards. These will be elaborated in a separate specification (see [I-D.draft-happel-structured-email-trust-04]).

10. Implementation status

< RFC Editor: before publication please remove this section and the reference to [RFC7942] >

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [RFC7942], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

10.1. Structured Email for Nextcloud Mail

Nextcloud Mail is an open source Webmail app which includes a subset of structured email support since 2020 [NC-Itinerary].

In recent months, code has evolved to support structured email in a more general fashion [NC-SML].

10.2. Structured Email plugin for Roundcube Webmail

An open source plugin for the Roundcube Webmail software is developed to serve as an example implementation for this specification ([RC-SML]).

Beyond that, some ISPs and open source tools provide implementation partly compliant with this specification ([SchemaOrgEmail]).

10.3. Yatagarasu Mail

[Yatagarasu] Mail is a fork of the open source Thunderbird Mobile client for Android. It adds structured email support for both reading and sending messages.

11. Security considerations

See section "security and trust".

12. Privacy considerations

See section "security and trust".

13. IANA Considerations

13.1. Creation of the SML registry group

IANA will create a new registry group called "Structured Email (SML)". This group includes the "SML Email Vocabulary Registry" and "SML Vocabularies" registries described below.

13.1.1. Creation of the SML Email Vocabulary Registry

IANA will create the following registry:

Registry Name: "SML IETF Email Vocabulary"

Registration Procedure: TBD

13.1.2. Creation of the SML Vocabularies Registry

IANA will create the following registry:

Registry Name: "SML Vocabularies"

Registration Procedure: TBD

(TABLE OF REGISTRATIONS)

The registry entries contain the following fields: - Vocabulary name: a human readable name for reference - Namespace: the URI namespace prefix for the vocabulary - Documentation page: a web page with further documentation about the vocabulary - Scope: the domain described by that vocabulary - Reference: (IANA) - Notes: any additional notes

13.1.3. Initial Entries for the SML Vocabularies Registry

The registry initially contains these entries:

Vocabulary name	Namespace	Documentation page	Scope	Reference	Notes
Schema.org	https://schema.org/ (https://schema.org/)	https://schema.org/ (https://schema.org/)	General purpose concepts		
SML Email Vocabulary	TBD	TBD	Email-specific concepts defined in IETF drafts	(this draft)	

Table 1

13.2. Registration of the Content-Purpose MIME header field

The following MIME header field is registered in the Message Headers registry, as established in [RFC3864]/BCP 90.

Header Field: Content-Purpose

Description: If the MIME entity is intended as machine-readable information for the MUA.

Applicable protocol: MIME [RFC2045]

Status: standards-track

Author/change controller: IETF

Specification document(s): This document

Related information:

13.3. Registration of the \$hasStructuredData keyword

The following IMAP/JMAP keyword is registered in the IMAP and JMAP Keywords registry, as established in [RFC5788].

IMAP/JMAP keyword name: \$hasStructuredData

Purpose Indicate to the client that a message contains structured data according to this spec.

Private or Shared on a server: SHARED

Is it an advisory keyword or may it cause an automatic action: This keyword can cause an automatic action.

When/by whom the keyword is set/cleared: This keyword is set by the server on delivery.

Related keywords: MRM;hasAttachments is peripherally related

Related IMAP capabilities: None

Security considerations: None

Published specification: This document

Intended usage: COMMON

Scope: BOTH

Owner/Change controller: IESG

13.4. Registration of the \$MRM keyword

The following IMAP/JMAP keyword is registered in the IMAP and JMAP Keywords registry, as established in [RFC5788].

IMAP/JMAP keyword name: \$MRM

Purpose Indicate to the client that a message is fully machine-readable according to this specification.

Private or Shared on a server: SHARED

Is it an advisory keyword or may it cause an automatic action: This keyword can cause an automatic action.

When/by whom the keyword is set/cleared: This keyword is set by the server on delivery.

Related keywords: any message with MRM should have also set thehasStructuredData keyword

Related IMAP capabilities: None

Security considerations: None

Published specification: This document

Intended usage: COMMON

Scope: BOTH

Owner/Change controller: IESG

14. Informative References

[HTMLData] WHATWG, "HTML Living Standard: Embedding custom non-visible data with the data-* attributes",
<https://html.spec.whatwg.org/multipage/dom.html#attr-data-*>.

[HasAttachment]
IETF imapext WG mailing list, "Registering \$hasAttachment & \$hasNoAttachment",
<<https://mailarchive.ietf.org/arch/msg/imapext/MVE5eNHOaNIVGUvN1RKtBL8b278/>>.

[JSONLD] W3C JSON-LD Working Group, "JSON-LD 1.1",
<<https://www.w3.org/TR/json-ld/>>.

[MachineReadable]
NIST, "NIST IR 7511 Rev. 4",
<https://csrc.nist.gov/glossary/term/Machine_Readable>.

[NC-Itinerary]
Nextcloud GmbH, "Nextcloud Mail: Itinerary support",
<<https://github.com/nextcloud/mail/pull/2214>>.

[NC-SML] audriga GmbH, "Nextcloud Mail: SML support",
<<https://github.com/audriga/nextcloud-mail/tree/enh/sml-markup-rendering>>.

[RC-SML] audriga GmbH, "Structured Email plugin for Roundcube Webmail",
<<https://github.com/audriga/roundcube-structured-email/>>.

[RDF] W3C RDF Working Group), "RDF 1.1 Concepts and Abstract Syntax", <<https://www.w3.org/TR/rdf11-concepts/>>.

[RFC2045] 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/info/rfc2045>>.

- [RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, DOI 10.17487/RFC2046, November 1996, <<https://www.rfc-editor.org/info/rfc2046>>.
- [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/info/rfc2119>>.
- [RFC2183] Troost, R., Dorner, S., and K. Moore, Ed., "Communicating Presentation Information in Internet Messages: The Content-Disposition Header Field", RFC 2183, DOI 10.17487/RFC2183, August 1997, <<https://www.rfc-editor.org/info/rfc2183>>.
- [RFC2392] Levinson, E., "Content-ID and Message-ID Uniform Resource Locators", RFC 2392, DOI 10.17487/RFC2392, August 1998, <<https://www.rfc-editor.org/info/rfc2392>>.
- [RFC3864] Klyne, G., Nottingham, M., and J. Mogul, "Registration Procedures for Message Header Fields", BCP 90, RFC 3864, DOI 10.17487/RFC3864, September 2004, <<https://www.rfc-editor.org/info/rfc3864>>.
- [RFC3987] Duerst, M. and M. Suignard, "Internationalized Resource Identifiers (IRIs)", RFC 3987, DOI 10.17487/RFC3987, January 2005, <<https://www.rfc-editor.org/info/rfc3987>>.
- [RFC4021] Klyne, G. and J. Palme, "Registration of Mail and MIME Header Fields", RFC 4021, DOI 10.17487/RFC4021, March 2005, <<https://www.rfc-editor.org/info/rfc4021>>.
- [RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, DOI 10.17487/RFC5322, October 2008, <<https://www.rfc-editor.org/info/rfc5322>>.
- [RFC5598] Crocker, D., "Internet Mail Architecture", RFC 5598, DOI 10.17487/RFC5598, July 2009, <<https://www.rfc-editor.org/info/rfc5598>>.
- [RFC5788] Melnikov, A. and D. Cridland, "IMAP4 Keyword Registry", RFC 5788, DOI 10.17487/RFC5788, March 2010, <<https://www.rfc-editor.org/info/rfc5788>>.
- [RFC7515] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <<https://www.rfc-editor.org/info/rfc7515>>.

[RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <<https://www.rfc-editor.org/info/rfc7942>>.

[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/info/rfc8174>>.

[SchemaOrg] W3C Schema.org Community Group, "Schema.org", <<https://schema.org/>>.

[SchemaOrgEmail] Structured Email, "Schema.org for email", <https://structured.email/content/related_work/frameworks/schema_org_for_email.html>.

[WDCStats] Web Data Commons Project, "Web Data Commons - Microdata, RDFa, JSON-LD, and Microformat Data Sets", <<http://webdatacommons.org/structureddata/#toc3>>.

[Yatagarasu] audriga GmbH, "Yatagarasu Mail", <<https://github.com/audriga/thunderbird-android>>.

Author's Address

Hans-Joerg Happel
audriga GmbH
Email: happel@audriga.com
URI: <https://www.audriga.com>