Workgroup: Network Working Group
Internet-Draft:
draft-happel-sml-problem-statement-00
Published: 27 January 2023
Intended Status: Informational
Expires: 31 July 2023
Authors: H.-J. Happel C. Junghans
audriga GmbH

Structured Email: Problem Statement and Areas of Work

Abstract

This document discusses benefits of complementing existing email standards by means that allow to replace or extend text-based email messages with message parts that describe content (full or in parts) in a machine-readable way. This would enable rich and structured interaction for its recipients - may it be human users or agents on their behalf.

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 31 July 2023.

Table of Contents

1. Introduction
2. Problem statement
  2.1. Email content is not machine-readable
  2.2. Using email as a "personal API"
  2.3. Slow adoption of existing solutions
3. Areas of work
  3.1. Core areas of work
    3.1.1. Enable email clients to understand email content
    3.1.2. Enable users to compose structured email
  3.2. Cross-cutting areas of work
    3.2.1. Internet Message Format extensions
    3.2.2. Trust
    3.2.3. Interaction with HTTP backend APIs
  3.3. Optional areas of work
    3.3.1. Email search and filtering
    3.3.2. Interaction with other data types and external APIs
    3.3.3. Email storage formats
    3.3.4. Common practices in email processing
  3.4. Out of scope areas of work
    3.4.1. Data extraction
    3.4.2. UI design
4. Privacy considerations
5. Security considerations
6. IANA Considerations
7. Informative References
Authors' Addresses

1. Introduction

Email is a successful and widespread technology which is likely to remain important, even if new vendors and technologies continuously challenge its role.

Email, to some extent, can be considered a victim of its own success. Interrelations of various components have created a relatively stable software ecosystem, which makes it difficult to introduce larger improvements. While many RFCs have updated and extended initial email specifications, the major inner workings of email remain widely unchanged since their inception.

This documents lines out certain issues with email content that might be addressed by standardization work. The proposed approach aims to enable novel ways of how users and automated programs can interact via email while retaining downwards compatibility with

existing email standards. For additional background, see also [I-D.happel-structured-dynamic-email-00].

2. Problem statement

2.1. Email content is not machine-readable

A large share of today's emails is of transactional nature, i.e., sent by automated agents or processes to human users. While those emails often have relatively clear semantics (such as *Invoice* or *Reservation*), the medium of those emails is still human-readable text.

In order to help users with managing and processing their emails more efficiently, a machine-readable representation of email content can be an important building block.

2.2. Using email as a "personal API"

Beyond improving the handling of existing email interactions, it is worthwhile to note that email can be considered a prime technology in the recent discussion about data sovereignty, which aims to give back control to users over their personal data.

Email is inherently open and decentralized. It is probably the only widely-used and standardized technology which seamlessly connects the local data space of users (i.e., emails on their PCs, mobile devices, or private hosted mailboxes) with the public internet.

It might even be considered a ubiquitous "personal API" of internet users, which to date is mostly based on text-based instructions ("John, could you please send me this presentation?"). Along these lines, structured email might serve as an enabling technology for granting users additional sovereignty in date storage and exchange when used to interact with internet services.

2.3. Slow adoption of existing solutions

In a mature technological space such as email, any novel approach such as structured email needs to address the issue of user adoption.

Existing attempts to structure email interaction can be distinguished in *standards-based* and *vendor-driven* approaches. Several RFCs address very particular problems such as meeting workflows [RFC2447], message delivery notifications [RFC8098], mailing list subscriptions [RFC8058], message metadata [RFC6477] or message content interpretation [RFC9078]. Creating independent RFCs for each possible type of structured email interaction might not be a feasible approach for both standards makers and client developers.

With respect to vendors, Google ([EMarkup], [AMPemail]) and Microsoft ([AM]) have made attempts to structure email interactions. Even though some other vendors support, e.g., [EMarkup], a closer look shows that those implementations are mostly incompatible among each other ([SmartInbox], [YahooPS], [ZohoQAES]). Besides a lack of standardization (and according tool support), widespread adoption also suffers from various sender restrictions, including manual approval processes.

3. Areas of work

This section tries to identify and structure areas of work to address the aforementioned topics. We distinguish some core work, additional, and optional topics.

3.1. Core areas of work

3.1.1. Enable email clients to understand email content

As a primary building block for structured mail, there needs to be a specification on how to represent email content or metadata about email content in a machine-readable form.

Such an approach needs to be downwards compatible with existing practices and clients, probably in a similar way the "multipart/alternative" MIME part type is used for HTML email ("text/html"). [AMPemail] for instance, is leveraging this approach by adding an additional MIME part in conjunction with "text/plain" and "text/html".

Regarding content semantics, [EMarkup] is using a subset of [SchemaOrg] types in a JSON-LD [W3C.REC-json-ld11-20200716] representation. Ideally an open approach to structured email would allow for decentralized extensions of content semantics.

3.1.2. Enable users to compose structured email

To sustain the decentralized and interactive nature of email, it is insufficient to just allow professional email senders to send structured emails to users. Instead, users need to be enabled to answer and also to initiate structured email exchanges.

[AMPemail] (initiated by Google) and Microsoft Actionable Messages [AM] are approaches that allow users to reply with a structurd answer based on input forms. However, the required input is not machine-understandable, making it difficult to assist users in data entry. Both approaches allow responses to be sent to a dedicated HTTP API only, but not as a regular email response.

Both approaches also do not allow users to initiate a structured email exchange. For this purpose, a discovery mechanism would be required, which allows users resp. their email clients to determine which types of structured email content an intented email recipient is willing to accept besides standard email.

3.2. Cross-cutting areas of work

3.2.1. Internet Message Format extensions

While structured email should be designed to maintain downwards-compatibility with the existing email technology stack, there might be some smaller helpful extensions:

* Additional email header information might inform clients about the presence of structured email content.
* It might be helpful to distinguish structured email markup which represents the complete message content (in the sense of "multipart/alternative") versus markup which provides partial annotations or metadata to regular email content. For the latter case, a standardized option to help clients identify such MIME parts (e.g., based on a MIME part header) might be useful.
* A distinct characteristic of email technology is the multitude of processing options on both server-side or on client-side with multiple possible email clients operating on one account. This challenge is, e.g., also present in the email filtering design space [RFC5228]/[RFC5804]. Accordingly, there might be use cases in which users may want to direct structured email processing to certain clients (e.g., travel information to the mobile email app) in which additional header information might help guide processing.

3.2.2. Trust

(Semi-)automated processing of structured email requires increased scrutiny with respect to security issues. Accordingly, current vendor-specific approaches ([EMarkup]/[AMPemail]/[AM]) require DKIM [RFC6376] and/or SPF [RFC7208] set up in addition to a manual sender registration process.

While the latter is already a clear inhibitor of adoption, this process will not work if users shall be enabled to send structured mail by themselves - a scenario without a central gatekeeper. Accordingly, suitable measures of trust need to be in place. Candidate approaches might be a consolidation of existing email trust indicators, specific concepts of "trusted senders", or drawing inspiration from sources such as the ACME protocol [RFC8555].

3.2.3. Interaction with HTTP backend APIs

Structured email may benefit from certain server-side (HTTP) APIs, serving different purposes. An API for the discovery if an intended receiver supports structured email has already been discussed (see Section 3.1.2).

Additional APIs could be useful to allow synchronous responses of structured data in certain uses cases in which an asynchronous message response cycle would be infeasible. All current vendor-specific approaches ([EMarkup]/[AMPemail]/[AM]) allow for some sort of direct data transmission against an HTTP backend API.

3.3. Optional areas of work

3.3.1. Email search and filtering

Machine-understandable interpretation of email content could also be used to improve some existing practices based on text/keyword-based processing of email content. In particular, the IMAP SEARCH command (Section 6.4.4 of [RFC9051]) and email filtering [RFC5228] might benefit from corresponding extensions. However, those might also be subject to work of other IETF working groups.

3.3.2. Interaction with other data types and external APIs

Email clients have grown over years to support various types of user data beyond email. This typically involves contacts/address books, calendars, tasks, notes, or files.

Besides the case of iMIP [RFC2447], interrelation between email and other data types is vendor-specific, if present at all. Structured email could help to enable easier interaction within email clients, with other applications on the same machine and with services on the internet (see also Section 2.2).

Ongoing work in the JMAP working group [RFC8620] could overlap with some of these aspects. The W3C Solid initiative [Solid] is another loosely related

3.3.3. Email storage formats

Several vendors use machine learning or data extraction approaches (see also Section 3.4.1) to derive information about email content which is similar to or complementary to structured email content.

For interoperability and data portability, it would be helpful to standardize storage for such metadata. This might also help with current portability issues stemming from user-defined metadata such as IMAP flags (Section 2.3.2 of [RFC9051]).

3.3.4. Common practices in email processing

While structured email aims to provide a user-driven approach to make email content machine-understandable, there may be a number of more technical issues related to email-processing which might be addressed en route.

Besides consideration of already existing RFCs (see Section 2.3), this may involve currently unstandardized issues such as email signatures, vacation notices or "noreply" addresses. Those are probably more specific application areas of structured email, since they can impact multiple layers of email processing and corresponding standards.

3.4. Out of scope areas of work

This section lists aspects which are relevant to structured mail but which might not be addressed by formal standardization work in the scope of this proposal.

3.4.1. Data extraction

Several service providers or tools (e.g., [KDEItinerary]) apply data extraction techniques to derive structured data from textual email content. Data extraction is therefore an interesting approach to help bootstrapping the adoption of structured email, until it has become common practice among email senders.

While cooperation on shared data extractors might be a useful activity for parties interested in structured email, it is probably not a subject for standardization.

3.4.2. UI design

Prescription of how user interfaces should display structured email or organize interaction with it are - besides illustration - typically out of scope of RFCs. As in the case of data extraction, there might however be a shared interest among vendors to collaborate on certain best practices.

4. Privacy considerations

Since email content often contains personal data, it is subject to privacy considerations. From a high level perspective, privacy issues in structured email should not significantly differ to privacy issues in existing email standards.

On a more fine granuar level, structured email might both raise certain novel privacy issues (i.e., since structured data is more easy to process and share), but it could also improve certain

privacy concerns. For instance, it could make certain data extraction practives (see Section 3.4.1) obsolete, which currently cannot distingish senstive from non-sentitive parts of an email.

5. Security considerations

As a problem statement document, no particular security considerations apply as such. Similar to privacy considerations, security issues might also widely overlap with those of existing email standards.

See Section 3.2.2 for the discussion of some general security aspects with respect to structured email.

6. IANA Considerations

This document has no IANA actions at this time.

7. Informative References

[AM] Microsoft Inc., "Actionable Messages", <https://learn.microsoft.com/en-us/outlook/actionable-messages/>.

[AMPemail] OpenJS Foundation, "AMP email", <https://amp.dev/about/email/>.

[EMarkup] Google Inc., "Email Markup", <https://developers.google.com/gmail/markup>.

[KDEItinerary] KDE e.V., "KDE Itinerary", <https://apps.kde.org/itinerary/>.

[RFC2447] Dawson, F., Mansour, S., and S. Silverberg, "iCalendar Message-Based Interoperability Protocol (iMIP)", RFC 2447, DOI 10.17487/RFC2447, November 1998, <https://www.rfc-editor.org/info/rfc2447>.

[RFC5228] Guenther, P., Ed. and T. Showalter, Ed., "Sieve: An Email Filtering Language", RFC 5228, DOI 10.17487/RFC5228, January 2008, <https://www.rfc-editor.org/info/rfc5228>.

[RFC5804] Melnikov, A., Ed. and T. Martin, "A Protocol for Remotely Managing Sieve Scripts", RFC 5804, DOI 10.17487/RFC5804, July 2010, <https://www.rfc-editor.org/info/rfc5804>.

[RFC6376] Crocker, D., Ed., Hansen, T., Ed., and M. Kucherawy, Ed., "DomainKeys Identified Mail (DKIM) Signatures", STD 76, RFC 6376, DOI 10.17487/RFC6376, September 2011, <https://www.rfc-editor.org/info/rfc6376>.

[RFC6477] Melnikov, A. and G. Lunt, "Registration of Military Message Handling System (MMHS) Header Fields for Use in Internet Mail", RFC 6477, DOI 10.17487/RFC6477, January 2012, <https://www.rfc-editor.org/info/rfc6477>.

[RFC7208] Kitterman, S., "Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1", RFC 7208, DOI 10.17487/RFC7208, April 2014, <https://www.rfc-editor.org/info/rfc7208>.

[RFC8058] Levine, J. and T. Herkula, "Signaling One-Click Functionality for List Email Headers", RFC 8058, DOI 10.17487/RFC8058, January 2017, <https://www.rfc-editor.org/info/rfc8058>.

[RFC8098] Hansen, T., Ed. and A. Melnikov, Ed., "Message Disposition Notification", STD 85, RFC 8098, DOI 10.17487/RFC8098, February 2017, <https://www.rfc-editor.org/info/rfc8098>.

[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/info/rfc8555>.

[RFC8620] Jenkins, N. and C. Newman, "The JSON Meta Application Protocol (JMAP)", RFC 8620, DOI 10.17487/RFC8620, July 2019, <https://www.rfc-editor.org/info/rfc8620>.

[RFC9051] Melnikov, A., Ed. and B. Leiba, Ed., "Internet Message Access Protocol (IMAP) - Version 4rev2", RFC 9051, DOI 10.17487/RFC9051, August 2021, <https://www.rfc-editor.org/info/rfc9051>.

[RFC9078] Crocker, D., Signes, R., and N. Freed, "Reaction: Indicating Summary Reaction to a Message", RFC 9078, DOI

10.17487/RFC9078, August 2021, <https://www.rfc-editor.org/info/rfc9078>.

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

[SmartInbox] 1&1 Mail & Media GmbH, "WEB.DE Smart Mailbox", <https://postmaster.web.de/en/smartinbox>.

[Solid] W3C Solid Community Group, "Solid Protocol", <https://solidproject.org/TR/protocol>.

[W3C.REC-json-ld11-20200716] Longley, D., Ed., Kellogg, G., Ed., and P. Champin, Ed., "JSON-LD 1.1", W3C REC REC-json-ld11-20200716, W3C REC-json-ld11-20200716, 16 July 2020, <https://www.w3.org/TR/2020/REC-json-ld11-20200716/>.

[YahooPS] Yahoo Inc., "Promotions & Schema", <https://senders.yahooinc.com/promotions-and-schema/>.

[ZohoQAES] Zoho Corporation Pvt. Ltd., "Quick Actions and Email Snippets", <https://www.zoho.com/mail/help/quick-actions.html>.

Authors' Addresses

Hans-Joerg Happel
audriga GmbH

Email: happel@audriga.com
URI: https://www.audriga.com

Conny Junghans

Email: conny.junghans@1und1.de