draft-ietf-sml-structured-email-03

IETF 122, Bangkok

Status

- Scope: Main specification for structured email → allow to describe content of email messages in a machine-readable format

- Example implementation: https://github.com/audriga/roundcube-structured-email

- Updates since IETF 121
  - New draft revision
  - Draft discussed in WG Interim on 2025-02-18
  - Conducted compatibility checks

Agenda

- 3.2 Vocabularies (github.com/hhappel/draft-happel-structured-email/issues/2)

- Structured data within email messages
  - 4.1 Structured data placement (github.com/hhappel/draft-happel-structured-email/issues/3)
  - 4.2.2 Referencing structured data in text/html (github.com/hhappel/draft-happel-structured-email/issues/5)

- Structured data across email messages
  - 5.3 Error replies (https://github.com/hhappel/draft-happel-structured-email/issues/8)
  - 5.4 Updates of structured data (https://github.com/hhappel/draft-happel-structured-email/issues/9)

3.2 Vocabularies

- Current text recommends Schema.org vocabulary if covering use case, allowing any valid JSON-LD otherwise

- Suggestion: create an IANA registry for RDF vocabularies used in structured email
  - Goal: help discover vocabularies and their documentation (both senders + receivers)
  - Register top-level vocabularies, not their individual concepts

- Template: Namespace / Website / Scope
  - https://schema.org / https://schema.org / General purpose concepts
  - https://sml.iana.org / https://sml.iana.org / Email-related concepts
  - https://rdf.who.org / https://who.org/resources/rdf / WHO medical vocabulary
  - …

- Open issue: Does IANA technically allow documentation in the style of https://schema.org?

4.1 Placement of structured data (I)

- Insight: messages might contain multiple “multipart/alternative” or “multipart/related” parts

- Implications
  - There might be multiple pieces of structured data in one email
  - Current wording is probably misleading in this case: “The email message SHOULD in this case also contain a text/plain and a text/html version of the content”
  - Suggestion: rewrite to sth like “the corresponding body part”?

Initial client tests of “multipart/alternative” rendering

Client	multipart/alternative (text/html/json-ld)	multipart/alternative (text/json-ld/html)*	multipart/related (text/html/json-ld)
Apple Mail	✅	✅	✅ 📎
Outlook (incl. OWA)	✅	✅	✅ 📎
Thunderbird	✅	✅	✅ 📎
Gmail	✅ 📎	✅ 📎	✅ 📎
OpenXChange 7.10	✅ 📎	✅ 📎	✅ 📎
Roundcube	✅ 📎	✅ 📎	✅
WEB.DE	✅ 📎	✅ 📎	✅ 📎
Yahoo	✅ 📎	✅	✅ 📎

📎 = will show as attachment

* formally incorrect order, just added for legacy comparison

4.1 Placement of structured data (II)

- Insights from testing
  - JSON-LD in “multipart/alternative” and “multipart/related” do not severely break major clients
  - Nag: JSON-LD sometimes shown as attachment
- Suggestion: encourage “content-disposition: inline” / ask clients to hide
  - Alternative: specific header? (vs. earlier draft on autoprocessing header?)
- Still open: automate testing

4.1 Placement of structured data (III) - Further open issues

- Allow/mention top-level message content-type “application/ld+json”?
  - For cases involving no user clients
  - Mainly relates to “full representation” case

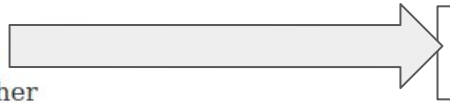
- Also allow signed/encrypted structured data? (e.g., “application/jose+json”) besides “application/ld+json”

- Consider “structured email signature”, which might likely be required for “full representation” in some jurisdictions (as part of core draft?)

4.2.2. Referencing structured data in text/html

and plump, tender shrimp, she pairs smoked paprika with lemon juice for a bright and earthy edge. She says it serves four but, let's be real, that's just a suggestion.

In the recipe notes for Kay Chun's sheet-pan chopped salad with chicken and a feta topping, you'll find a lively debate about whether to replace the zucchini with olives. Personally, I'd use both, but then I'm a known maximalist when it comes to sheet-pan meals. Kay's



```
{
 "@context": "http://www.schema
 "@type": "Recipe",
 "@id": "https://cooking.nyt.co
/sheet-pan-chopped-salad",
 "name": "Sheet-Pan Chopped Sal
    Chicken", (...)
}
```

- Forward reference (text/html → Structured data)
- Suggestion: Use HTML “data-id” property
  - <a **data-id=”https://cooking.nyt.com/sheet-pan-chopped-salad**” href=”...”>sheet-pan chopped salad with chicken
  - No direct reference to body part (content-id) → no problem? (MUA supposed to map by id)
- https://github.com/hhappel/draft-happel-structured-email/issues/5

5.3 Error replies

- Scenario: MUA is replying to a structured email *with a structured email* (e.g., “confirm flight”) → what to do in error cases on the recipient side?

- List of error cases
  - Syntax error in structured email
  - Semantic error (e.g., unknown property)
  - …

- Slightly related to DSN / System status codes (RFC 3463)
- Suggestion: keep on “SML layer”; create a proposal

- See also https://github.com/hhappel/draft-happel-structured-email/issues/8

5.4 Updates of structured data

- Scenario: structured data is is updated (e.g., flight departure)
- Current spec: leverage SUPERSEDES header (RFC4021); dismiss/update prior structured data if possible

- Open issues
  - Spec currently focuses on recipient case
    - Suggestion: add sender perspective
  - Deletion of structured data (e.g., flight booking cancelled)
    - Send empty structured data?

- See also https://github.com/hhappel/draft-happel-structured-email/issues/9

Next steps

- General
  - Automated client testing
  - Setting up WG GitHub repo
- Spec
  - Section 5 (Structured data across messages)
  - Section 6 (Header fields and flags)
- Further aspects
  - Provenance?
- Further co-authors?