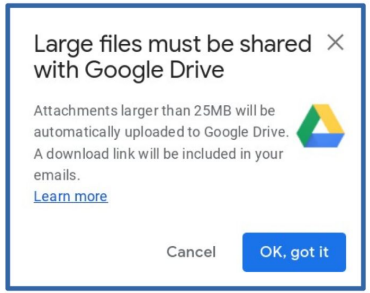
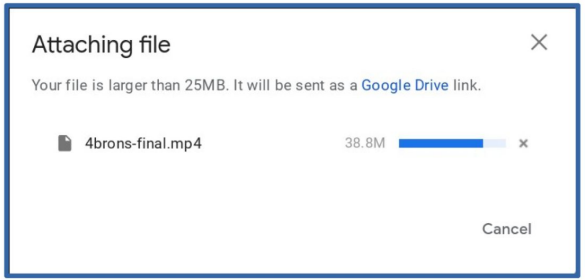
The Big Files Problem

SML@IETF127 Bron Gondwana

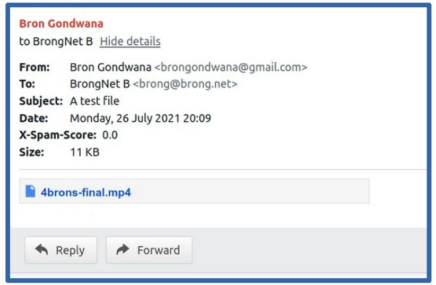
We spoke about this before

- https://datatracker.ietf.org/meeting/111/materials/slides-111-dispatch-gondwana-email-big-files-problem-00.pdf
- Today every client and service does its own thing
  - for example Gmail → Google Drive links









The problems with plain links

An Attached File	**A link**
Lasts as long as the message	Can stop working
Can’t be changed after sending	Can be edited later
Can be scanned by systems that handle the mesage	Can’t be scanned in transit reliably
Is available to anyone who has a copy of the message	May require a separate login

Design

- Have linked attachments be as similar as possible to “real” attachments
- A new header field, Content-External, on a html body part with a legacy link
  - Clients that don't support it show the link
  - Clients that do hide the link and show the attachment
- Not message/external-body: many clients show it with no link
  - and it has no digest or key
- Optional encryption with AEAD
- Lots of security considerations text

Example

```
Content-Type: text/html; charset=utf-8
Content-Disposition: inline
Content-External: video/mp4; name="site-survey.mp4"; size=734003200;
    url="https://files.example.com/la/Hq3Ld9YtW2xKq8vZp0aN5uFbE7cJr1sM";
    digest-sha-256="X48E9qOokqqrvdts8nOJRJN3OWDUoyWxBf7kbu9DBPE=";
    key="AEAD_AES_128_GCM:9Mzuq2AvtfQJxwuIzYB4Gg:Xq3mC8vT0bLw5RzH";
    expiration="Thu, 17 Dec 2026 09:15:00 +0000"

<div class="content-external">
<a
href="https://dl.example/pR7wQxV3mTz9KcLe2Hd8Fy?k=AEAD_AES_128_GCM:9Mzuq2
AvtfQJxwuIzYB4Gg:Xq3mC8vT0bLw5RzH">site-survey.mp4</a>
(700 MiB)
</div>
```

Who Does What?

- Sender / attachment server: stores the file (optionally encrypted) and gives a download link
- Receiving server: fetches on arrival with no cookies or credentials, checks the size and digest, decrypts, scans, and applies the same policy as for an attached file
- Message store: keeps the file as long as the message, with one copy per digest (ref-counted), and serves it to the user's clients like an attached file
- Client: treats it like an attached file

Potential for a JMAP extension to help manage these across all types (calendar attachments, contact resources)

Privacy

- The attachment server sees every fetch, no way around that
- If the client fetches when the message is displayed, it's a read receipt, like a remote image
- If the server fetches on arrival, it only shows that the message was delivered, and hides who the recipient is
- So the draft prefers: fetch on arrival, and clients SHOULD NOT fetch just because the message was displayed

There’s parallel work happening in MIMI, we’ll be similar

- https://www.ietf.org/archive/id/draft-ietf-mimi-content-10.html#name-external-content

```
ExternalPart = (
    cardinality: external,
    contentType: tstr,      ; An IANA media type {8}
    url: tstr,              ; A URL where the content can be fetched
    expires: uint .size 4,  ; expiration in seconds since UNIX epoch
    size: uint .size 8,     ; size of content in octets
    encAlg: uint .size 2,   ; An IANA AEAD Algorithm number, or zero
    key: bstr,              ; AEAD key
    nonce: bstr,            ; AEAD nonce
    aad: bstr,              ; AEAD additional authentiation data
    hashAlg: uint .size 1,  ; An IANA Named Information Hash Algorithm
    contentHash: bstr,      ; hash of the content at the target url
    description: tstr,      ; an optional text description
    filename: tstr          ; an optional suggested filename
)
```

Changes since Alexey’s Draft

- Explicitly always on separate MIME parts, not top-level headers
- one Content-External per part, so Content-ID works and each attachment is separate
- added KEY, STORED-DIGEST-, DESCRIPTION and date parameters
- SIZE and DIGEST-SHA-256 are required
- URLs must be https
- removed PERMISSION, because a linked attachment can't change

Open questions

- How does a client get the stored copy?
- JMAP: list linked attachments among an Email's attachments, each with a blobId
- IMAP: a new FETCH data item?
- Should these be separate documents?
- A standard way for a client to get proxied remote images?

Next Steps

Adopt here?

Send to Mailmaint?

Alternative designs? (top-level headers instead? Embedded microformat inside main HTML body?)