Episode Book

EPISODE #6372

Where Email Attachments Actually Live and Go

PUBLISHED	30 October 2026 at 16:27 UTC
DURATION	23:12
LISTEN	myweirdprompts.com/episode/email-attachment-storage-mime-45008/
HOSTS	Corn, Herman, Larry, Hilbert
SCRIPT MODEL	openai/gpt-6.1-sol
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.26
SOURCES	19 cited by the research (listed at the end)
BOOK GENERATED	31 October 2026 at 05:58 UTC

SYNOPSIS

Email attachments seem like simple files handed between computers, but MIME transforms them into text packed directly inside the message body. Storing all that data poses an enormous engineering problem: attachments accounted for 85 percent of storage at Mail.Ru before deduplication. Mail systems dissect complete messages on disk, balance reference counters, hash encoded payloads, and handle transport overhead, revealing what really happens to old files stored across remote data centers.

TRANSCRIPT



Corn

A document you emailed in two thousand fourteen can still open today, even if the computer you sent it from has become landfill. Daniel wants to know where those bytes actually went. Whose machines are keeping them alive, how well are they guarded, and can you ever make an attachment stop being retrievable?



Herman

The unsettling part is that opening it today probably has nothing to do with your original computer. Your mail provider and the recipient's provider have their own copies. The little paperclip makes the document look like an object you handed over. Underneath, you distributed its contents.



Corn

My laptop died. Its paperwork remains employed.



Herman

There can also be intermediate mail servers, local downloads, archives, and backups. Those aren't all necessarily permanent, but they are separate places with separate retention decisions.



Corn

What catches me is the ordinary experience. I search an ancient subject line, click the paperclip, and the file appears. No visible hosting contract, no renewal. Somebody has continued paying for disks and electricity on behalf of a spreadsheet I regretted immediately.



Herman

Your mailbox account supplies that continuing relationship. The first technical detail is that the attachment travels inside the message body. There isn't a second little parcel following the email.



Corn

Then the paperclip is already lying about the packaging.



Herman

MIME gives the message multiple parts. One might contain plain text, another HTML, another your PDF. Each part has information describing its content and encoding. Binary attachments are commonly converted into base sixty-four text, using characters that older mail infrastructure can carry reliably.



Corn

The PDF becomes text, crosses the network, and gets decoded back into the PDF at the other end. No instruction to fetch it from my hard drive later.



Herman

Right … no, more specifically, no dependency on your hard drive after successful submission. The receiving system has the data needed to reconstruct it. Modern transport has extensions beyond the old seven-bit restrictions, but base sixty-four remains extremely common.



Corn

Email has spent decades accommodating equipment nobody wants to meet.



Herman

The history explains that. Before polished attachment interfaces, people used uuencode and pasted the resulting text into messages. Early PC mail software such as cc:Mail put an attachment interface over that process. MIME made richer message structures interoperable.



Herman

Nathaniel Borenstein and Ned Freed developed MIME, with the original standard appearing in nineteen ninety-two. RFC twenty forty-five, from nineteen ninety-six, is a later foundational specification. Borenstein is associated with the first MIME attachment, rather than the first occasion anyone transmitted a file through email.



Corn

Useful distinction. Otherwise somebody with a very old terminal gets furious.



Herman

Encoding also costs space. Base sixty-four itself adds about a third, and line wrapping takes the usual overhead to roughly thirty-seven percent. That overhead counts when a server limits the whole encoded message.



Corn

Twenty megabytes can become more than twenty-five before you've written, please find attached.



Herman

Depending on how the service measures its limit, yes. A ten-megabyte whole-message allowance leaves roughly seven megabytes for the actual file. The filename displayed by the mail app doesn't show you all those transport bytes.



Corn

I object to a file gaining weight in transit.



Herman

Would you rather the interface showed the encoded size?



Corn

Both sizes. Tell me what I own and what the post office charges for. Hiding the second number just makes the eventual rejection look arbitrary.



Herman

That distinction matters again when we get inside the storage system.



Corn

The obvious route is my provider's outgoing server to your provider's incoming server. I assume my Sent folder is one copy and your Inbox is another. Where do the intermediaries fit?



Herman

Mail transfer agents accept messages and forward them onward. A corporate filtering gateway might sit before the destination mailbox, for instance. Once a server accepts responsibility for a message, it generally needs a queued copy while arranging delivery.



Herman

That does not mean every relay keeps a permanent archive. A relay can remove its queued copy after the next system accepts it. Temporary custody and long-term retention are different.



Corn

Otherwise we'd accidentally claim every birthday invitation lives forever at every junction.



Herman

Precisely ... actually, let me narrow that too. Some routes have very few junctions. Others include filtering, journaling, or organizational routing. You can't infer the number of retained copies from the number of addresses in the message.



Corn

If the attachment is text inside that message, and each accepting system needs the whole thing, who exactly ends up holding all of it?

L

Larry

Folks, let me tell you about MimeMinder, the only attachment protection plan that understands that your file is not a file, it is a base sixty four encoded block inside a multipart message, and that is exactly why our satisfied customer, a lovely lady named Deborah, no wait, Denise, from a town I will not name, signed up and never looked back. Denise sent one email with one photo, and because every server that accepted it stored its own full copy, she now receives a personal letter from each and every one of those servers every single morning, and MimeMinder prunes them for her, one by one, with a little pair of scissors I assume. She told me, Larry, I have not thought about my attachments in years, and I said Denise, that is the whole point.



Herman

Accepting a queued message doesn't establish an ongoing correspondence with its sender. Delivery status notifications concern particular delivery outcomes, and successful relays normally clear their queues. They do not send daily letters.



Corn

Let the scissors go, Herman.



Herman

The long-term holders are usually the mailbox providers, plus whatever archives and clients are involved. Physically, the data lives on storage devices in data centers. The mail application finds it through indexes and storage identifiers. An independent provider on the other side has its own infrastructure.



Corn

Unless both addresses belong to the same provider. Then I can imagine one underlying object with two mailbox entries pointing at it. Moving a mountain of identical company brochures onto separate disks would be expensive.



Herman

Providers have a powerful reason to avoid that duplication. Mail.Ru described attachments taking eighty-five percent of its mailbox storage, with message bodies and indexes occupying the rest.



Corn

Most of email is the thing you were meant to read instead.



Herman

Their deduplication project reduced storage from fifty petabytes to thirty-two. They identified identical stored files using hashes and maintained reference counts. A mailbox entry could point to an existing attachment object instead of requiring another complete payload.



Corn

Eighteen petabytes recovered by noticing everybody got the brochure.



Herman

Their examples included repeated price lists and newsletters. It is a substantial saving, but there is engineering behind it. The lookup index needs memory, replication needs capacity, and the counters must remain correct as messages arrive and disappear.



Corn

Herman, you've reached the part where a saved disk becomes a new bookkeeping department.



Herman

I accept the accusation. Incorrect bookkeeping here can lose someone's document, so I find it rather more compelling than your system of leaving every downloaded PDF on the desktop.



Corn

My reference counter is visual.



Herman

Wild Duck, an open-source mail server, makes the design especially clear. It hashes attachment content, keeps a reference count, and can place the payload in an object store or MongoDB GridFS. Message records carry the references needed to put the original message back together.



Corn

Wait. We just said the attachment is inside the message. Now we've taken it out again.



Herman

On disk, yes. The message format and the physical storage layout are separate decisions. A server can receive a complete MIME message, split it into internal components, and reconstruct the complete message when a client requests it.



Corn

The protocol promises the meal. It doesn't prescribe the refrigerator.



Herman

Google has patents describing that separation, including partial message structures that omit attachment contents and routers that reconstruct complete messages for delivery. A patent doesn't establish exactly how today's Gmail storage works, but it demonstrates the architecture.



Corn

Does a shared object mean someone who guesses its hash can retrieve my attachment?



Herman

No. A content hash identifies bytes within the storage design. Authorization still has to determine whether your account may access the message referencing those bytes. Knowing an identifier and having permission are separate.



Corn

I'd be more worried about a counter error than the sharing itself. If three messages point to one object, removing one reference mustn't remove the payload.



Herman

Wild Duck waits until the reference count reaches zero, then allows a twenty-four-hour grace period before garbage collection removes an unreferenced attachment. That gives the system room to handle operations without immediately destroying newly unreferenced data.



Herman

Here's a complication, Corn. Would you deduplicate the decoded PDF, or the exact encoded attachment text?



Corn

The PDF. Surely the contents are what we're paying to keep. Different line breaks in the envelope shouldn't buy another disk allocation.



Herman

Wild Duck chooses the encoded bytes to preserve message fidelity and signature validity, including DKIM and GPG schemes. Two encodings can decode to the same file but differ in the representation needed to reproduce a signed message.



Corn

I concede the signature. I retain the complaint.



Herman

Different wrapping of the attachment's encoded text can prevent deduplication. Merely changing the surrounding email doesn't necessarily do that. The relevant bytes have to differ.



Corn

Then neither of us can look at the paperclip and count physical copies. It might represent one of several independent objects, or one reference among thousands. The provider knows the layout. We know the document opens.



Herman

Security adds another distinction. Preventing a dangerous attachment from arriving, protecting stored data, and letting a sender revoke access are three different jobs.



Corn

Daniel asked how well the storage is guarded. A malware scanner doesn't answer who can read a harmless tax return.



Herman

Stored mail can be protected through encryption at rest, access controls, operational monitoring, and account security. Connections can be protected in transit too. Those controls do not ordinarily make conventional email end-to-end encrypted. The provider operates the service that reconstructs and serves the attachment.



Corn

Encryption against a stolen disk doesn't necessarily protect me from the organization holding the keys.



Herman

Correct. Nor does it protect against every compromised account or authorized export. Exact arrangements vary by provider and organization. We shouldn't assign a universal security grade to a whole category of mail storage.



Corn

There is still a useful boundary. My recipient's provider is now part of the security arrangement, whether I chose it or not. Their downloads and backups add more arrangements I can't inspect.



Herman

That is the loss of custody people often miss. You can secure your account extremely well and still deliver the file into an account governed by somebody else's administrator.



Herman

Gmail's attachment filtering tackles a different risk. It blocks executable programs, scripts, installers, and numerous other formats. It can also block prohibited content inside archives and documents containing malicious macros.



Corn

The zip file doesn't count as a disguise.



Herman

Password-protected archives can also be blocked under its rules, including when archived contents prevent inspection. Scanning and format restrictions reduce risk, but a file passing inspection is not proof that it is harmless. New exploits and deceptive documents remain possible.



Corn

Your clinical voice has appeared.



Herman

You wouldn't want the DJ voice explaining a malicious macro.



Corn

It would still have a drop.



Herman

Google's advice when a legitimate file is blocked is revealing. Upload it to Drive and send a Drive attachment. That moves the delivery into a separate sharing service, with its own access rules.



Corn

The email app quietly changes products while keeping the paperclip.



Herman

What would you expect a delete-attachment button to do, Corn? Just save mailbox space, or withdraw what you sent?



Corn

In my Inbox, save space while preserving the correspondence. In Sent, I might hope it also withdraws the document, but I wouldn't trust that hope. The interface would need to distinguish those operations very loudly.



Herman

Gmail doesn't provide a native way to remove just the attachment from an existing message. Some mail clients offer a detach or delete-attachment operation, but the common IMAP approach creates a replacement message without that part and removes the original.



Corn

A small editorial revision with a surprisingly large technical footprint.



Herman

It can preserve the readable correspondence while changing the underlying message. That distinction can matter for signatures and archival requirements. Third-party products such as Unattach and Clean Email address the mailbox-cleanup niche, but they don't acquire authority over somebody else's Inbox.



Corn

Even the modest version is more involved than it looks. I save the file elsewhere, replace the email, remove the old message, then wait for the provider's retention and storage processes. The word delete covers several events.



Herman

Removing something from the visible mailbox, removing references from active storage, and purging retained copies need not happen together. Backups may follow separate schedules. Deduplication also means the underlying payload can remain because other messages still reference it.



Corn

None of those events reaches across to my recipient.



Herman

A normal delivered attachment supplies no sender-controlled revocation channel. Some managed mail environments have recall features under specific conditions, but recall is not a general power to erase delivered attachments across providers, downloaded files, and forwarded messages.



Corn

The recipient doesn't have to keep cooperating with the sender's software. They already have the bytes.



Herman

Mm.



Herman

Links change that relationship because the message contains a reference to a remotely hosted file. The sharing service can check permission on each later request. Remove access there, and future requests can fail.



Corn

I'd still distinguish stopping the next request from retrieving yesterday's download. The first is credible. The second requires a remarkably obedient recipient.



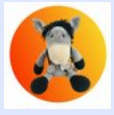
Herman

Drive supports restricted sharing and removing access for particular people. Eligible work and school accounts can set per-person expiration, with a date within one year. That expiration option isn't the same feature on a personal Gmail account.



Corn

A document can have a retirement date, subject to account class.



Herman

Viewers and commenters can also have downloading, copying, and printing disabled through sharing settings. Those controls affect the supported interface. They don't make readable information impossible to reproduce.



Corn

Screenshots remain aggressively interoperable.



Herman

Ownership matters too. Moving your own Drive file to trash doesn't immediately stop collaborators viewing, commenting, or making copies. Google's sharing guidance says that can continue until permanent deletion. If you don't own the file, you cannot delete the owner's original.



Corn

I would have expected trash to end sharing. That's an unfortunate place for an access decision to hide.



Herman

The service has separate concepts for ownership, trash, and permission. Visually disposing of an item doesn't necessarily alter all three.



Corn

An expiring link has another cost that attachments avoid. If I need the document years later, I now depend on the owner's account, the sharing service, and an unchanged permission decision. An ordinary attachment may survive the relationship that produced it.



Herman

You're defending the old durable copy.



Corn

You asked where I'd draw the line. A quotation I'm evaluating can expire. A document I'm required to retain should not vanish because someone reorganized their cloud drive.



Herman

That is a genuine benefit of distributing a complete message. Later retrieval doesn't require asking the sender again. It is also why a sender can't simply change their mind about everyone else's retained version.



Corn

What about an attachment identifier in the Gmail API? It sounds like something you might invalidate.



Herman

It's a handle used to retrieve attachment data in the context of a message and authorized account. It isn't a sender-issued lease on all copies. Descriptions differ on its stability, but neither a durable identifier nor a short-lived pointer creates remote revocation.



Corn

A pointer expiring doesn't imply the thing it pointed to disappeared.



Herman

Nor does a fresh pointer mean a fresh copy. There are several layers between the identifier an application sees and the storage object a provider maintains.



Hilbert

Yes, that does happen. I spent a Sunday comparing two copies of a stock list because I needed to show I hadn't changed the document. This was in the early nineties, when I was counting stored mail files for a distributor. There was rather more repetition than we've made it sound. My own test messages were enough to annoy me.



Corn

A whole Sunday devoted to a stock list sounds like a dispute with no refreshments.



Hilbert

There was toast. I sent the file to myself through two mail programs. Same document when decoded. Different encoded text in the saved messages. Both occupied space. Herman, you were careful about the wrapping earlier. I appreciated that. People say identical when they mean it looks identical after the software has finished with it.



Herman

Comparing the recovered files would settle whether the document changed, but it wouldn't make the stored representations match.



Hilbert

I printed both encoded versions. I crossed out matching lines in green pencil until I found the differences. I wouldn't use red because nothing in the document was wrong. The printer took most of the afternoon. I knew Colin Mercer could clear a paper jam in seventeen seconds. He'd timed himself. That was an irritating thing to remember while sitting there alone.



Corn

Green pencil was kinder to the document than to your Sunday.



Hilbert

I still use the printer. It's beside the bookcase in my flat, plugged into a print server. It prints mail attachments perfectly well. The word deduplication still makes me check what somebody counted. Those two saved messages were six pages each.



Herman

Twelve pages of transport representation for a document that hadn't changed. Reconstructing the original bytes and retaining the exact message are different requirements.



Corn

The paper cost was visible, at least. In a mailbox the extra representation disappears behind the same filename and icon. That's part of why this feels mysterious. The interface suppresses nearly every distinction we've spent the episode making.



Herman

One adjacent detail we haven't used is Drive's sharing scale. A single file can be shared with up to six hundred individual email addresses, while simultaneous editing is limited to a hundred open tabs or devices.



Corn

Even a link has a crowd-control policy.



Herman

Those are sharing-service limits, separate from the mail transport limits. Once again, the same attachment-looking interface can conceal quite different machinery.



Corn

I'd like future mail interfaces to make that choice visible before sending. Am I distributing a durable copy, or inviting someone to retrieve a document whose access may end? Twelve printed pages or a cloud object, the distinction shouldn't require investigating the packaging.



Herman

Thanks to Hilbert Flumingtop for producing the show and keeping our side of the machinery running.



Corn

If this was your kind of episode, go back for episode thirty-five, The Privacy Gap; episode sixteen, On Deepfakes, SynthID, And AI Watermarking; and episode forty eighty-five, The Clinton Email Server. This has been My Weird Prompts, the human-AI collaboration podcast. No question too obscure, no rabbit hole too deep.



Herman

If you enjoy the show, leave us a review wherever you listen. It helps other curious people find us.



Corn

See you tomorrow.

SOURCES

What the research for this episode read before the script was written. Primary sources first.

1. **High Scalability, Mail.Ru engineering case study [primary]**
   https://highscalability.com/efficient-storage-how-we-went-down-from-50-pb-to-32-pb/

2. **Wild Duck IMAP server docs [primary]**
   https://github.com/nodemailer/wildduck/blob/master/docs/in-depth/attachment-deduplication.md

3. **Google Gmail Help, blocked file types, accessed 2026-10-30 [primary]**
   https://support.google.com/mail/answer/6590?hl=en

4. **Google Drive Help, sharing and expiration, accessed 2026-10-30 [primary]**
   https://support.google.com/drive/answer/2494822?hl=en

5. **Microsoft Learn, MAPI attachments [primary]**
   https://learn.microsoft.com/en-us/office/client-developer/outlook/mapi/attached-files-and-messages

6. **Wikipedia, last edited 2026-08-03 [primary for technical detail]**
   https://en.wikipedia.org/wiki/Email_attachment

7. **Wikipedia, last edited 2026-10-12**
   https://en.wikipedia.org/wiki/MIME

8. **Wikipedia, last edited 2026-08-30**
   https://en.wikipedia.org/wiki/Nathaniel_Borenstein

9. **Wikipedia, last edited 2026-02-24**
   https://en.wikipedia.org/wiki/Message_transfer_agent

10. **Wikipedia, last edited 2026-10-18**
    https://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol

11. **Google Patents, attachment de-duplication**
    https://patents.google.com/patent/WO2014028261A1/en

12. **Google Patents, attachment handling during email delivery**
    https://patents.google.com/patent/US20140122620A1/en

13. **Gmail Community, Delete attached photos but keep the email?, Nov 3 4, 2024**
https://support.google.com/mail/thread/305500322/

14. **Gmail Community, how can I detach attachments in gmail**
https://support.google.com/mail/thread/33445561/

15. **Stack Exchange, Can one delete a Gmail attachment but keep the email?**
https://webapps.stackexchange.com/questions/95427/

16. **Clean Email blog**
https://clean.email/blog/email-management/how-to-delete-attachments-in-gmail

17. **Unattach**
https://unattach.com/blog/how-to-delete-gmail-attachments

18. **Stack Overflow, Gmail attachmentId stability**
https://stackoverflow.com/questions/28104157/

19. **Nylas guide**
https://cli.nylas.com/guides/gmail-attachment-id-stability