Cargomail — A Revised Email System

Revision 2 (2023-11-03)

Igor Zboran
izboran@gmail.com

***Abstract*—Electronic mail (email) is the most pervasive form of business information exchange. Email is used not only as an interpersonal communication tool but also as a "default choice" for sending files.**

This paper introduces Cargomail, a revised email system that allows users to exchange digital resources, e.g., messages, documents, books, images, videos, and audio. To succeed in this endeavor, we propose to add a *pull request* layer on top of the existing email infrastructure.

Additionally, we explain the importance of separating the identifier (*email address*) and the locator (*resource mailbox*) in the design of the Cargomail architecture. Our paper concludes by highlighting the significant advantages of this design.

I. Introduction

The main components of the email system were designed between the early 1970s and mid-1990s by many inventors. Over time, email has become the most commonly used Internet application. Nowadays, email is the only ubiquitous communication system on the Internet built in a decentralized fashion. Moreover, the email infrastructure forms the backbone of the users' global digital identity.

II. Current Situation

Today, outgoing email is typically transferred from the source system to the destination system as a single text-encoded file using the Simple Mail Transfer Protocol (SMTP). SMTP is an over-40-year-old push-based protocol, and even though SMTP has been updated, modified, and extended multiple times to increase security and efficiency, it still lags behind modern web-based protocols.

III. Problems and Issues

Despite the importance of email infrastructure, the whole ecosystem still relies on more than 40-year-old architecture and protocol design. There have been spam and attachment issues from the very beginning. While conceptually sound as a communication means, the email system is structurally obsolete and functionally deficient.

A. Functional and Security Flaws

Even though the major email service providers claim their email services to be safe, the fact remains that fundamental security and functional flaws are not fixed. There is still a dichotomy in attachment delivery; bulky files are not transferred as an attachment but are shared via links. A "file sharing" is unnatural for the email system, where each message with attachments is expected to maintain data integrity. Additionally, shared links can pose a consent phishing attack threat, where an attacker tricks users into granting malicious application access to sensitive resources.

B. Confidentiality and Privacy

Now, if we (as users) want to use a single *email address*, we have no choice but to use a single email service provider for all categories of communication. Information about every email we send or receive—"buying a car or a home, applying for a loan, taking out insurance, purchasing potato chips, requesting a government grant, getting turned down for credit, going to work, seeing a doctor" [1]—is funneled through the same service providers. Such concentration of information within a single provider raises privacy concerns.

C. Hyperlinks to External Files

Messages, documents, images, videos, and audio should be an integral part of the email. Keeping a time-consistent, recipient-owned copy of the sender's files is critical in some industries. Here is the list of issues with hyperlinks to external files in email messages:

- expired, unknown, blocked, phishing, or malicious hyperlink
- masked hyperlink target using a URL shortener
- target updated – not the version it is expected
- target changed – forgery
- target encrypted – need a password
- access control – requires signup or sign-in
- consent phishing attack

Hyperlinks typically lack transparency and auditability. Given these points, you are buying a "pig in a poke" with each hyperlink to the external file in the email message.

D. Content Repository

The current email system is missing a content repository—a "file system on steroids" with the capability to create, store, locate, and exchange any content.

IV. Proposed Solution

Given that the current email system is lagging behind modern communication and collaboration tools, we propose to build a pull request layer on top of the existing email infrastructure to address inefficiencies and vulnerabilities in the current email system.

A. Goals and Objectives

At the core of the proposed solution is an attempt to enhance the usability of email as an interpersonal communication tool and as a default choice for sending files while supporting data sovereignty and governance. Our strategic decision is to ensure the new architecture is interoperable with the existing email system. This approach allows the proposed solution to be deployed incrementally. Due to the complexity of the topic, covering all design decisions and implementation details in one document is virtually impossible. Therefore, we will provide a high-level overview of the solution rather than delving into the details.

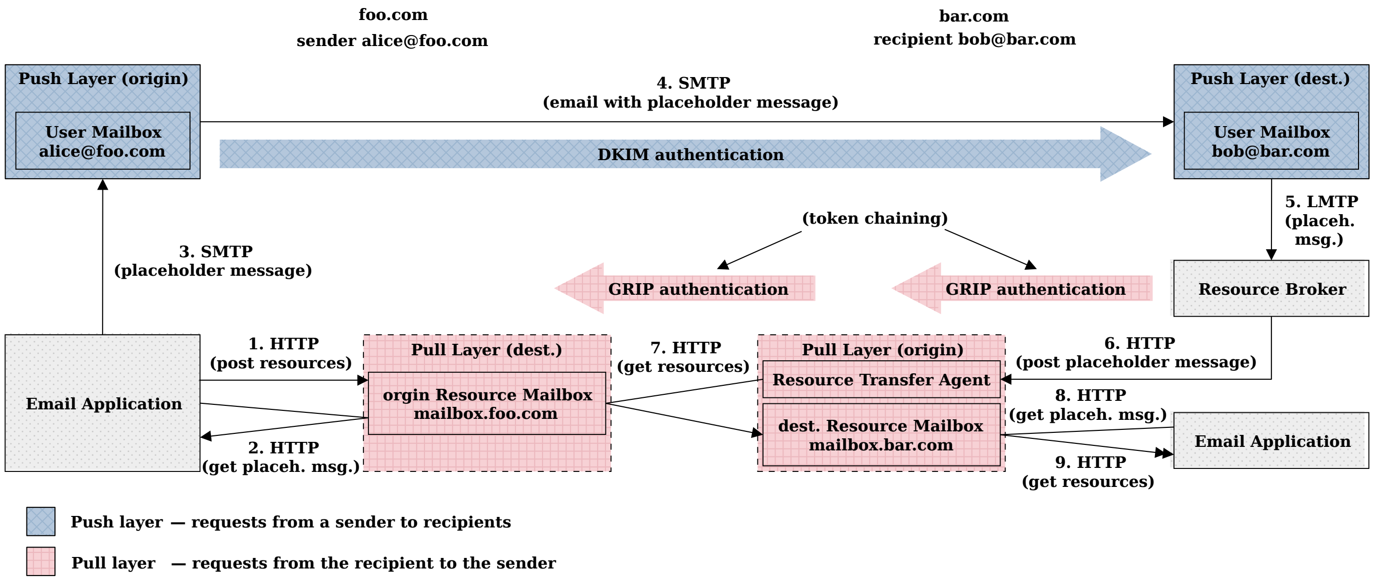

Fig. 1. Cargomail architecture with the *shared-mailbox-to-shared-mailbox* topology

B. Concept

According to RFC 5321 [2], an *email address* is a string of characters used to identify a person to whom mail will be sent or a location where mail will be deposited. In that sense, the terms mailbox and *email address* can be used interchangeably. However, Cargomail uses an architecture that separates the identifier and the locator of the address/ mailbox pair to introduce a new type of mailbox—a *resource mailbox*. This approach offers a different way of organizing data communication. In our design, each email includes a *placeholder message* created in the *resource mailbox* with references to external bodies kept in the same *resource mailbox* accessible via *content-addressed* URIs. The *resource mailboxes* are used for actual correspondence, while the default *user mailbox* of the current email system is used to send and receive the *placeholder message*, as shown in Figure 1. A newly designed service called *resource broker* coordinates email delivery.

The approach depicted in Figure 1 separates the identifier (*email address*) and locator (*resource mailbox* URL). By decoupling the locator and identifier, data can be exchanged between diverse *resource mailboxes* using one *email address* per user. Cargomail exchanges referenced resources between the *resource mailboxes* using the Global Reference Identity Protocol (GRIP) [3] authentication mechanism. This approach enables the sender and recipient to use multiple *resource mailboxes* while using their single *email addresses*.

In other words, this architecture uses a mechanism of push-then-pull requests over different routes to enable data exchange between mailboxes. The origin and destination of push requests differ from the origin and destination of pull requests. This mechanism has the potential to address privacy and attachment issues more effectively than push-pull systems with shared origin and destination. The added pull layer facilitates end-to-end encryption.

C. Key Points

• Each email consists of an envelope, a *placeholder message* created in the *resource mailbox*, and related external resources (message bodies) stored in the same *resource mailbox*. The *placeholder message* also acts as an access control list to external resources.

• The resources owned by the sender, stored in the origin *resource mailbox*, are temporarily shared with recipients. Following a successful sharing process, a *placeholder message* is sent to each recipient through the standard email system, usually as an email attachment. The *placeholder message* contains the origin *resource mailbox* URL, the cryptographic hash values of the referenced resources (Content-IDs), and the category of correspondence, e.g., personal, business, or healthcare (see Appendix A for a *placeholder message* example).

• After receiving the email with the *placeholder message*, the recipient's *resource broker* determines (according to the user's preferences and the category of correspondence) which destination *resource mailbox* will be used for communication. Once the destination *resource mailbox* is determined, the *resource broker* adds the header with the *destination resource* mailbox URL to the *placeholder message* and posts it to the relevant destination *resource mailbox* using the GRIP authentication mechanism.

• The *resource transfer agent* at the destination server gets the origin *resource mailbox* URL and the cryptographic hash values of the referenced resources in the *placeholder message*. Using the GRIP authentication mechanism, the agent tries to retrieve the external resources from the origin *resource mailbox*. After successful authentication, the data is retrieved and stored in the destination *resource mailbox*. Finally, the *email application* can download the relevant data from the destination *resource mailbox* and reconstruct the original message according to the *placeholder message* template.

D. GRIP Authentication

To enable controlled access to data within the pull request layer, we need an authentication/authorization mechanism designed in a decentralized manner.

Design decision: The initial idea was to use an authentication mechanism based on the OAuth 2.0 framework. However, this approach turned out to be inappropriate due to the fact that in the OAuth 2.0 framework, the roles of the authorization server, resource server, and client are co-located in the same security domain. Therefore, we developed a new identity propagation mechanism to address this issue. The main idea behind the GRIP authentication concept is that a cross-domain identity propagation system should work much like an email system. We were inspired by the fact that in a special case—as a de facto part of the globally accepted signup and password recovery mechanism—an email with a confirmation link works as a "token" in MIME format. Equally important for us was the finding that such a "token" is forwardable. We can say that these "tokens" are self-issued (the issuer is a Message Transfer Agent), DNS-bound (via the DKIM signing mechanism), chained (via the sender/recipient headers), and nested (forwarded as an attachment). With this in mind, we designed GRIP, a token-based zero-trust security protocol that authenticates service requests between untrusted hosts across the Internet.

Context conveyance: To meet the Cargomail architecture needs, we use self-issued DNS-bound tokens to propagate the user/broker context from the *resource broker* to the destination *resource mailbox* and the user/broker/agent context from the destination *resource mailbox* to the origin *resource mailbox*, all in different security domains. Proper chaining and nesting of tokens, as illustrated in Figure 2, is crucial to convey identity claims about the users, broker, and agent across the network.

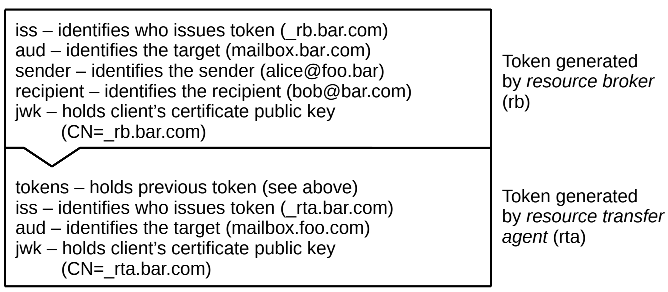

Fig. 2. Nested, chained, self-issued DNS-bound tokens

Actors authenticity: In some service-to-service communication scenarios, three identities are employed: user, client, and server identities. Fundamentally, mutual TLS (mTLS)/TLS certificates resolve client and server identities, while tokens resolve client and user identities. A DNS-bound token is a self-issued assertion in a JWT format signed by an mTLS private key that the first service uses to authenticate to the second service. The mTLS public key hash is published in the first service domain using the DNS TXT record, where the CN attribute of the mTLS public key certificate is used as a global client identifier in respect of the service it represents. Given that, to ensure that the proper context is conveyed and tokens cannot be replayed, the *resource broker* to the *resource transfer agent* and the *resource transfer agent* to the origin *resource mailbox* are authenticated via mTLS.

In summary: The zero-trust concept implemented in the GRIP authentication mechanism improves the usability and security of the proposed Cargomail system.

V. Advantages compared to the Current Email System

Cargomail has several decisive advantages over the current email system. These advantages encompass improved security and privacy, increased efficiency, and ease of use. Overall, Cargomail provides users with a more reliable and effective way of exchanging information.

A. Spam Protection

With Cargomail architecture, the *email address* and mailbox are separated. The actual email correspondence is exchanged between *resource mailboxes*, or "locators," while the *email address*, or "identifier," is only used to deliver a *placeholder message*. This architecture does not protect against unsolicited emails—anyone can send you an email—it allows for a more detailed assessment of the sender's reputation by individually evaluating their *email address* identifier and origin *resource mailbox* locator.

B. Content of Any Size or Quantity

Unlike the current email system, Cargomail enables users to transfer any content, regardless of its size or quantity, to ad hoc recipients while maintaining the data decentralized in their respective *resource mailboxes*. This capability encourages trustworthy communication across different domains or verticals (e.g., healthcare, insurance, finance).

C. Data Sovereignty

The *resource mailbox* is decoupled from the user's *email address*. This separation allows a user with a single *email address* to use multiple *resource mailboxes* side by side. With a single *email address* and numerous mailboxes, Cargomail can keep official, business, healthcare, and personal correspondence separately on designated *resource mailboxes*, ensuring compliance with relevant laws and regulations.

D. Data Governance and Auditability

Email messages and their resources are kept together in chronological tamper-resistant records. A content-addressed *resource mailbox* storage ensures the authenticity, integrity, and traceability of email messages and their resources. The cryptographic hash value guarantees the storage of only a single instance of a resource. Both data governance and auditability are crucial for organizations aiming to effectively manage their data assets, maintain data quality, comply with regulations, and ensure data security.

VI. Implementation Concerns

This chapter deals with the issues of implementing the proposed data exchange mechanism into the existing email infrastructure.

A. Transferring a Placeholder Message

The best way to send a placeholder message (typically in JSON format) to a recipient is to attach it to an email. There is a concern that some email providers may not accept JSON attachments.

B. Decentralized Push/Pull System from Scratch

Instead of relying on an SMTP-based email infrastructure, a newly designed web-based decentralized push/pull system may be worth considering. The system would use the GRIP authentication mechanism to send *placeholder messages* in the sender's context. Furthermore, the sender could request the recipients' public keys, which might be stored in their *user mailboxes*, and utilize them to encrypt the exchanged digital resources.

VII. Exchange Network for Digital Resources

The pull request layer creates an opportunity to build a global ecosystem around digital resources. The GRIP authentication mechanism (i) allows using a user-centric approach to manage access to individual resources, (ii) enables trust among all parties, and (iii) ensures interoperability among resource servers.

A. Cargomail Resource Mailbox System Topology

Cargomail has the capability to operate in several *resource mailbox* topology system scenarios. We will highlight three of these scenarios to demonstrate its architecture's versatility.

Two distinguished resource mailboxes: Each *resource mailbox* operates within its own security domain, which we refer to as the *distinguished-mailbox-to-distinguished-mailbox* topology. The email provider has no access to the exchanged data.

Two shared resource mailboxes: Each *resource mailbox* shares a security domain with its respective sender's or recipient's email provider, which we refer to as the *shared-mailbox-to-shared-mailbox* topology. This topology is suitable for industrial environments where the email system and *resource mailboxes* can function within the same security domain.

Single distinguished resource mailbox: A single *resource mailbox* operates within its own security domain. We called it the *single-distinguished-mailbox* topology. The email provider has no access to the exchanged data, and exchanged data never leaves the resource server.

B. Applications and Use Patterns

Cargomail overcomes the limitations of traditional email when working with digital resources. Its architecture offers new, not typical applications and use patterns. Notably, Pretty Good Privacy (PGP) can be used for signing, encrypting, and decrypting exchanged digital resources.

Digital signatures: Using public key cryptography to sign a *placeholder message* provides a reliable method to guarantee both the authenticity and integrity of all digital resources referenced by the message. This ensures that the recipient can have complete confidence in the origin and content of the linked resources.

End-to-End Encryption (E2EE): Encrypting digital resources can be tricky. The resources must be selectively encrypted using symmetric cryptography, where each symmetric encryption key must be separately encrypted using the recipient's corresponding public key. Encrypted keys are stored in the associated metadata. The digital resources are exchanged between resource mailboxes along with the associated metadata.

Publish-subscribe pattern: A publish-subscribe system can be implemented to keep track of any changes in resources. See Figure 3 for a visual representation of the basic idea.

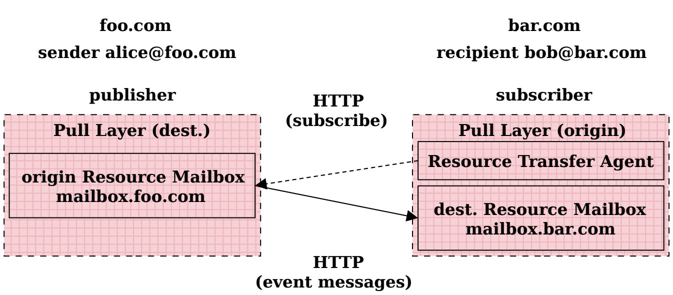

Fig. 3. Publish-subscribe pattern

This pattern helps to retrieve follow-up messages, new versions of attachments, and recurring messages, such as invoices, without sending a new email.

VIII. Use Cases

Numerous human activities in many industries can benefit from Cargomail technology. We describe some of them.

A. Manufacturing

Transferring digital resources is crucial in the manufacturing sector, where the exchange of design files, technical drawings, prototypes, and manufacturing data is typical.

- CAD/CAM Files: Manufacturing companies often need to exchange Computer-Aided Design (CAD) or Computer-Aided Manufacturing (CAM) files with designers, engineers, or suppliers.
- Prototyping and 3D Printing: In the prototyping and additive manufacturing processes, digital resources, such as 3D model files, are exchanged between design teams, engineers, and manufacturers.
- Supply Chain Collaboration: Manufacturing involves collaboration with suppliers, vendors, and contract manufacturers. Digital resource transfer allows the exchange of manufacturing specifications and technical drawings.
- Product Documentation and Manuals: Digital resources, such as user manuals, product catalogs, or assembly instructions, need to be shared with customers, distributors, or service technicians.

B. Healthcare

In the healthcare industry, digital resources are essential for sharing medical information, diagnostic images, and patient records.

- Medical Imaging: Healthcare providers often need to exchange large medical imaging files, such as X-rays, CT scans, or MRI scans, with specialists or physicians for consultations.
- Research Collaboration: Healthcare researchers and institutions collaborate on studies and clinical trials that involve large datasets, such as genomic sequencing data or medical research images.
- Electronic Health Record (EHR): Healthcare providers need to exchange complete electronic health records of patients with other providers or specialists involved in the patient's care.

C. Financial Sector

In the financial sector, it is crucial to securely and efficiently exchange sensitive financial data, reports, statements, and transaction records through digital means.

- Financial Statements and Reports: Financial institutions, accounting firms, and auditing agencies frequently exchange digital resources containing financial statements, annual reports, balance sheets, or income statements.
- Data Sharing for Risk Assessment: Financial institutions need to exchange large datasets for risk assessment and compliance purposes. This includes transferring data related to credit, market, or operational risks.
- Investor Communications: Asset management firms, investment banks, or private equity firms often need to exchange digital resources with their investors. This can include fund performance reports, investment prospectuses, or offering memorandums.
- Secure Client Data Exchange: Financial institutions frequently exchange sensitive client information, such as tax documents, loan applications, investment portfolios, or account statements.

D. Insurance

Transferring digital resources is essential in the insurance sector, where extensive documentation, policy files, claim records, and supporting evidence need to be exchanged securely and efficiently.

- Policy Documentation: Insurance companies often need to share large policy documents, including insurance contracts, terms and conditions, and coverage details, with policyholders or brokers.
- Claims Processing: Insurance claims often involve the exchange of large files, such as accident reports, medical records, repair estimates, or supporting evidence.
- Claims Investigations: Insurance investigations often require the exchange of large files, such as surveillance videos, forensic reports, or expert opinions.

E. Legal

In the legal sector, it is important to transfer digital resources such as documents, case files, evidence, and confidential information among attorneys, law firms, and legal professionals.

- Discovery and Litigation Support: During the discovery phase of a legal case, large volumes of documents, such as contracts, emails, financial records, or medical records, need to be exchanged between legal teams, clients, and opposing parties.
- Information Sharing with Co-Counsels and Experts: Attorneys often collaborate with co-counsels and subject matter experts who may be located in different offices or jurisdictions. Digital resource transfer enables the exchange of legal briefs, research materials, expert reports, or trial exhibits, allowing for seamless collaboration.
- Secure Document Sharing with Clients: Attorneys need to securely exchange sensitive documents with their clients, such as legal contracts, confidential agreements, or client records.
- Transcription Services: Attorneys often require transcription services for recorded statements, depositions, or hearings. Large audio or video files need to be securely transferred to transcription service providers for accurate and timely transcriptions.
- Document Archiving and Storage: Legal professionals often need to securely store and archive large volumes of case files, contracts, or other legal documents for future reference or compliance purposes.

IX. Models and Scenarios

Although Cargomail can be integrated into the email system of any email service provider, we slightly digress to introduce two visionary models of what a global email ecosystem might look like in the future.

A. Estonian Model

In this model, the government provides email services with *user mailboxes*. To avoid the risk of governmental surveillance, Cargomail allows citizens to use non-governmental *resource mailboxes*, e.g., from financial institutions or healthcare providers. Using non-governmental, sector-specific *resource mailboxes* increases the privacy of individual citizens, as the government cannot obtain detailed information about their activities.

B. Postal Model

According to UPU research [4], more than 93% of postal operators provide some form of digital postal service either directly or in partnership with other companies. Cargomail allows postal operators to expand and become public email service providers or innovate their existing email services and provide the *user mailbox* services with the ability to attach the *resource mailboxes* from the government as well as other institutions and organizations.

X. Related Work

While this proposal seems closely related to the Internet Mail 2000 [5] concept proposed by Daniel J. Bernstein, we differ in the new idea that the storage of email messages and their resources should be the responsibility of users and not of email providers.

XI. Conclusion

Cargomail can play an essential role in communication across various industries in the public and private sectors.

A. Overall Summary

Combining Cargomail with the current email system creates a hybrid architecture that meets the needs of a modern communication tool. Exchanging data between the appropriate mailboxes eliminates privacy concerns.

B. Digital Resource Infrastructure

Thanks to the ability to create, store, search, and exchange any content, Cargomail is the ideal platform for building a global *digital resource infrastructure* that helps exchange coherent and trustworthy information on the Internet. Cargomail can also serve as an extension of email infrastructure.

C. Future Work

Cargomail brings content-addressed storage and a new data exchange mechanism into the email ecosystem, predestining the proposed system to become more than a bare messaging tool. It would be fascinating to build a prototype of the proposed solution to serve as a proof of concept.

XII. Discussion

This proposal offers a fresh and uncharted outlook on the email system. The main question is whether it is worthwhile to integrate Cargomail into the current email system or to build up a new communication network based on the Cargomail architecture and GRIP mechanism.

References

[1] Jeffrey Rothfeder. 1992. Privacy for Sale: How Computerization Has Made Everyone's Private Life an Open Secret (pp. 22-23). Simon & Schuster Trade.

[2] J. Klensin, "Simple Mail Transfer Protocol", IETF RFC 5321, 2008, https://www.ietf.org/rfc/rfc5321.txt.

[3] I. Zboran, "Global Reference Identity Protocol (GRIP)," GitHub repository, March 2023, https://github.com/cargomail-org/grip.

[4] Universal Postal Union/Activities/Digital Services, accessed 27 March 2023, https://www.upu.int/en/Universal-Postal-Union/Activities/Digital-Services.

[5] Internet Mail 2000, accessed 23 April 2023, https://en.wikipedia.org/wiki/Internet_Mail_2000.

Appendix A.

Placeholder Message

```
{
  "headers": {
    "X-Origin-Resource-Mailbox-URL": "mailbox.foo.com",
    "X-Destination-Resource-Mailbox-URL": "mailbox.bar.com",
    "From": "Alice Sanders <alice@foo.com>",
    "Subject": "Meeting",
    "To": "Bob Sanders <bob@bar.com>",
    "Cc": "Carol <carol@bar.com>, Daniel <dan@bar.com>",
    "Date": "Tue Sep 19 20:52:05 CEST 2023",
    "Message-ID": "<b07d0cdf-c6f4-4f67-b24c-cc847a4c2df4@foo.com>",
    "X-Thread-ID": "<68fb9177-6853-466a-8f7d-c96fbb885f81@foo.com>",
    "X-Correspondence-Category": "primary",
    "Content-Type": "multipart/mixed"},
  "parts": [
    {
      "headers": {
        "Content-Type": "multipart/alternative"
      },
      "parts": [
        {
          "headers": {
            "Content-Disposition": "inline",
            "Content-ID": "<aSQnmlBT6RndpDnwTSStJUVhlh9XL9_y2QXX42NhKuI>",
            "Content-Type": [
              "message/external-body; access-type=\"x-content-addressed-uri\"; hash-algorithm=\"sha256\"; size=\"42\"",
              "text/plain; charset=UTF-8"
            ]
          }
        },
        {
          "headers": {
            "Content-Disposition": "inline",
            "Content-ID": "<Y_ION3g8WQuqGzhsDlVrhAgQ0D7AbXu9T-HSv3w--zY>",
            "Content-Type": [
              "message/external-body; access-type=\"x-content-addressed-uri\"; hash-algorithm=\"sha256\"; size=\"109\"",
              "text/html; charset=UTF-8"
            ]
          }
        }
      ]
    },
    {
      "headers": {
        "Content-Type": "multipart/mixed"
      },
      "parts": [
        {
          "headers": {
            "Content-Disposition": "attachment; filename=\"Cargomail.pdf\"",
            "Content-ID": "<6G6Mkapa3-Om7B6BVhPUBEsCLP6t6LAVP4bHxhQF5nc>",
            "Content-Type": [
              "message/external-body; access-type=\"x-content-addressed-uri\"; hash-algorithm=\"sha256\"; size=\"153403\"",
              "application/pdf"
            ]
          }
        }
      ]
    }
  ]
}
```