Optimizing file routing in EDI using the new naming principle

Olha Yanholenko[1,†], Nataliia Khatsko[1,2,*,†], Andrii Kopp[1,†], Kyrylo Khatsko[1,†] and Yevhenii Shebanov[1,†]

[1] *National Technical University «Kharkiv Polytechnic Institute», Kyrpychova str. 2, Kharkiv, 61002, Ukraine*

[2] *National University of Radio Electronics, Kharkiv, Nauky Ave. 14, 61166, Ukraine*

Abstract

The use of electronic data interchange is becoming a de facto standard of organizational interaction. The number of documents transferred between counterparties in the network is constantly increasing. Large volumes of files circulating between counterparties, in the event of an emergency that causes a stop in processing in document exchange systems, provoke the accumulation of a message processing queue. Different types of messages affect business processes of enterprises in different ways, and prompt delivery of important messages should be a priority for data analysis and processing. Unfortunately, to distribute messages depending on the recipient and message type, it is necessary to analyze the file contents. The purpose of this study is to propose a method for increasing the speed of message distribution between data exchange participants based on the analysis of document data transferred in the network. In order to comply with the service level agreement (SLA), as well as to reduce the impact of accidents that occur during document exchange, the article presents a principle of file naming in electronic data interchange, which allows you to prioritize message processing without reading the file contents, get improved routing and change the priority of document processing depending on the message type, its sender and recipient, as well as its format.

Keywords

file routing, file naming algorithm, electronic data interchange, identification, information system, processing priority, network data analysis

1. Introduction

Electronic document management systems have been in use for quite a long time, but they have not lost their relevance, even with the advent of new technologies such as Applicability Statement 2. This can be seen from data forecasting the growth of the electronic data interchange (EDI) software market to USD 5.30 billion by 2032; in 2025, this market is estimated at USD 2.31 billion [1].

The first attempts at EDI date back to the 1980s, which marked the beginning of enterprise computerization. Over time, technology has evolved, and message transmission has become increasingly automated. Whereas previous messages were sent from supplier to client or in a similar manner, now the exchange occurs between systems. These may still be suppliers and consumers, enterprises, legal entities, or government agencies; in general, electronic document flow does not stop. What remains constant in this process is the ever-increasing volume of files transferred and the wide variety of file formats in which data is transmitted.

MoMLeT-2025: 7th International Workshop on Modern Machine Learning Technologies, June, 14, 2025, Lviv-Shatsk, Ukraine

[*] Corresponding author.

[†] These authors contributed equally.

olha.yanholenko@ khpi.edu.ua (O. Yanholenko); n.khatzko@gmail.com (N. Khatsko); kopp93@gmail.com (A. Kopp); kyrylo.khatsko@khpi.edu.ua (K. Khatsko); qwertyw1999@gmail.com (Y. Shebanov)

0000-0001-7755-1255 (O. Yanholenko); 0000-0002-2543-0280 (N. Khatsko); 0000-0002-3189-5623 (A. Kopp); 0000-0003-3315-1553 (K. Khatsko); 0009-0006-9032-8764 (Y. Shebanov)

There may be an opinion that modern powerful computers can easily cope with the task of determining file recipients, routing, and sending such significant volumes of files. However, this is not always the case. For example, in retail chains, problems with the transfer of urgent orders often occur, especially during the cold start of the system or after a shutdown due to a failure. In such situations, the system must work very quickly and distribute files to recipients. But to route orders correctly, you need to open the file and read the recipient information inside it. It can be argued that this operation can be entrusted to additional resources, but this also requires some time, which may be too long compared to what is permissible in a particular situation. One area of change in EDI to improve productivity is the unification of procedures and methods.

There are internationally accepted messaging standards such as X12, EDIFACT, ODETTE, VDA, XML, EANCOM, IDOC, and TRADACOMS [2 – 6]. However, providers of electronic document flow have thousands or even hundreds of thousands of counterparties with different business processes and must account for all their nuances in their workflows, as well as the interaction of each with their counterparty. In general, the exchange of different types of messages between counterparties is shown in Figure 1.

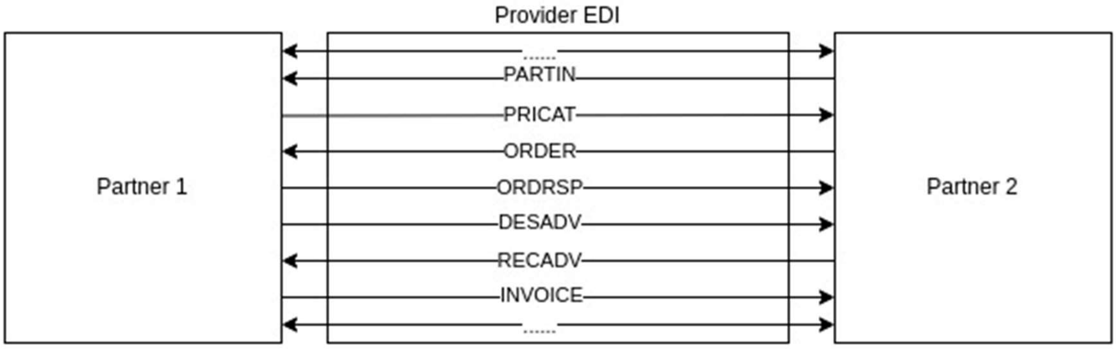


Figure 1: Exchange of messages between counterparties using a provider

Thus, the large number and variety of formats used in EDI document exchange complicate the timely delivery of messages to recipients and ultimately cause system failures in some cases. Therefore, further research into methods for improving data processing speed in EDI systems remains a relevant and important task.

2. Analysis of recent research and publications

It is entirely natural to strive to make such processes simpler and more understandable, so research continues on the use of EDI, for example, to reduce order variations and administrative costs for management [7]. One of the challenges in commerce is the so-called bullwhip effect, which refers to the amplification of orders along the supply chain, meaning that small variances in customer demand lead to increasing oscillations further upstream. Recent studies [8 – 11] propose the use of artificial intelligence to address this issue. However, EDI systems themselves have also proven to be highly effective in mitigating this problem [12].

Today, EDI remains a widely used technology and forms the foundation of many inter-organizational information systems. In various industrial sectors, alongside EDI, blockchain technology is employed to enhance transparency, security, and operational efficiency. In the study [13, 14], a comparison of these two technologies was conducted based on existing literature, leading to the conclusion that replacing EDI with blockchain technology is not advisable. Blockchain can increase trust, security, and reduce errors associated with EDI; however, it can only complement rather than fully replace EDI, despite the fact that blockchain enables simplified data formats and automation. The synergy of EDI and blockchain technologies allows us to maximize the benefits these technologies offer, yet the challenge of dealing with diverse data formats remains.

The integration of EDI with Industry 4.0 technologies, such as the Internet of Things (IoT), is being implemented on modern, high-performance platforms like IBM [15], and there are already practical examples of its use [16, 17].

The issue of receiving a large number of files of various types and formats arises as a consequence of increased demand for products or the failure of system components. In [18], an attempt was made to address the problem of system overload by classifying messages through the assignment of attributes and converting one message format into another. For example, the authors worked with the EDIFACT format and transformed it into XML. However, their approach is limited to simulating message flows with the ability to redistribute messages across infrastructure nodes. This solution cannot be applied in real-world scenarios, yet the study once again highlights the complexity and importance of the existing problem.

A sudden surge of messages in different formats often leads to failures in EDI systems, resulting in serious issues. Rapid recovery of the system after a failure is highly desirable. In [19], the authors suggest treating an error as a network intrusion and applying class balancing, which may yield results but slows down file transmission within the system. As the number of documents that must be processed in the shortest possible time increases, challenges arise regarding processing speed, the speed of restoring system nodes after a failure, and the need to establish and resolve priorities for processing and routing critical messages.

The question of file naming has been around for quite some time and is already well-established in IT. We can look at the following examples. File naming is regulated within operating systems, which impose limitations on file name length, prohibited characters, and rules for constructing paths. In programming, names are governed by language standards and corporate conventions. Databases often require adding prefixes, unique application identifiers, and specific delimiter characters to table names. In web development and DevOps, standards exist to ensure cross-platform compatibility, readability, and automated file processing. Finally, in the realm of integration with external systems and file exchange between systems (for example, via FTP, SFTP, cloud storage), corporate or industry-specific file naming standards are frequently introduced to automate processing, versioning, and status tracking. As shown above, standards in EDI systems also exist, but they are related to file structure and industry standards.

The current level of integration between exchange systems and the existence of businesses that simultaneously operate in various industrial sectors has significantly complicated the situation and substantially impacts the efficiency of file routing. Below, we will illustrate the problems in this area. The aim of this research is to propose a file naming method for exchange within a system that increases the routing speed of large volumes of files in provider systems.

3. Problem description

The principles of routing files over communication channels influence the quality and speed of exchange, the presence of single points of failure and bottlenecks in the exchange process, and the ability to collect metrics. In general terms, the message processing procedure on the side of an electronic document management provider or an intermediary message flow distribution system can be seen in Figure 2. It shows that the system takes files for processing from one storage location, the incoming messages (inbox/IN), processes them, and then transfers them to a storage location for further transmission of the message to the partner.

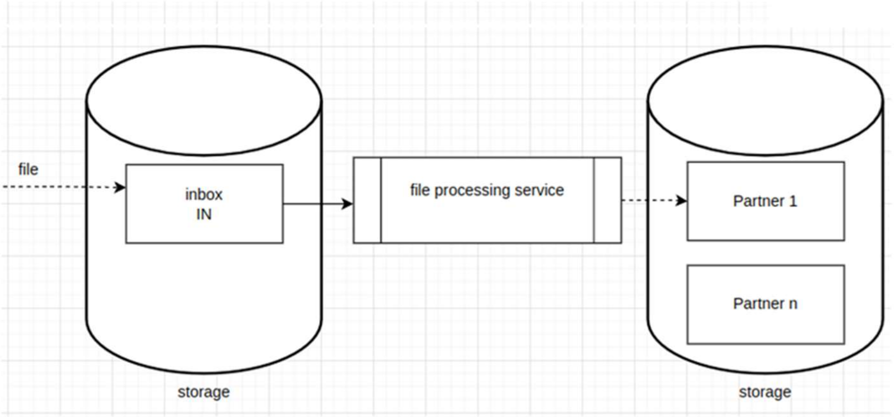


Figure 2: General view of the message processing procedure

In modern EDI systems, file analyzers are necessary to examine the content of files for routing purposes. The large number of file formats being transmitted makes the work of analysts a complex task. Figure 3 schematically illustrates the mechanisms of an Electronic Data Interchange provider that supports the processing of both documents in the system's native format (for example, XML) and other formats.

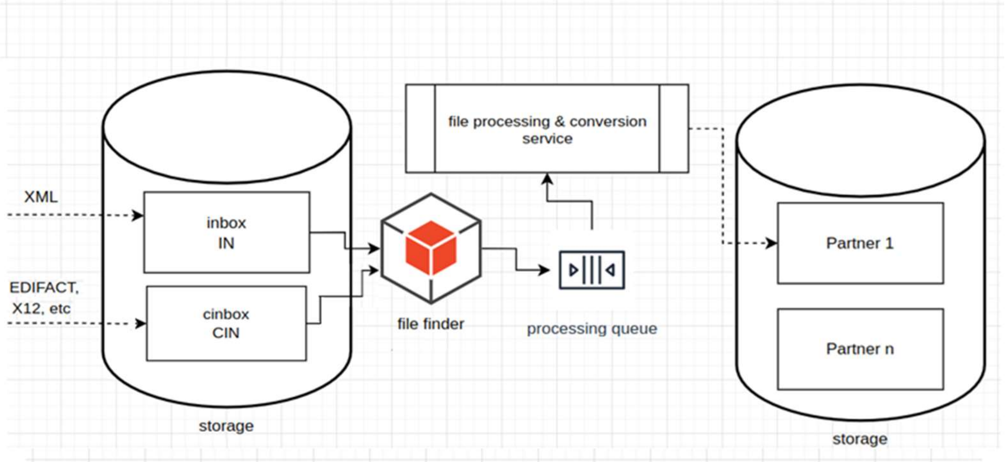


Figure 3: Handling multiple message formats

The electronic document management provider's system receives the file and saves it in its storage. If the file format is the main one for the system, then the message is saved in the “inbox/IN” storage. Figure 3 shows the additional directory “cinbox/CIN”, which is used only for messages that are not specific to the main message processing system.

The "file finder" component looks for files in storage and sends them to a message queue for processing by the processing service, which analyzes the file structure. However, each file with a non-standard format has an additional processing function that helps the processing service understand that this file needs to be handled with a preliminary conversion using mapping transformations. This pre-processing is performed by the processing service itself; that is, it first converts the file from a format atypical for the system into a typical format that is already analyzed by the system for further transmission.

The need to handle different message formats has led to the widespread use of Extensible Stylesheet Language Transformations and similar procedures for transforming the primary message into the resulting message format. Such conversion systems are integrated into the file processing service and use a separate conversion map created for each message type and format. Over time, systems developed as part of the transition to microservice architectures [20], as a result, the conversion procedure was separated into a separate service. Figure 4 demonstrates how the message conversion procedure became an independent service. Given the diverse formats and the need to enhance system resilience, the service received its separate queue for processing, and a separate message finder, which formed a queue based only on files of a message format not typical for the system, taking them from the cinbox/CIN directories.

After conversion, the file enters the inbox/IN directory, from where it enters the internal format file processing service, from the queue formed by a file finder that checks only the inbox/IN directory. Thus, converting to a separate service made it possible to distribute the responsibility of services, the developers of these services, for processing messages that the system should process. In the system architecture, it is possible to scale individual elements by increasing the number of processors using SAAS services and saving additional resources when not involved.

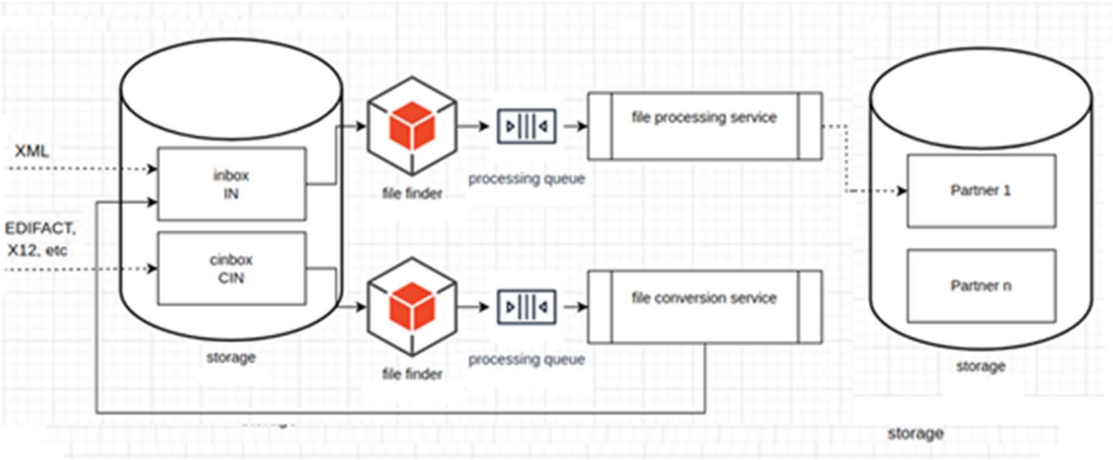


Figure 4: Handling multiple message formats with a separate conversion service

However, any system is susceptible to failures. Let's consider the previous diagram; in the event of a queue mechanism failure (message queuing failure), Figure 5 demonstrates that the number of unprocessed messages in the cinbox/CIN directory increases. Similarly, the connection for processing messages in the system's typical format could also fail. Given that the average message processing time is 0.04 seconds, which on average corresponds to the message processing rate [16, 21], a backlog of messages forms, and the system may require a significant amount of time to process them.

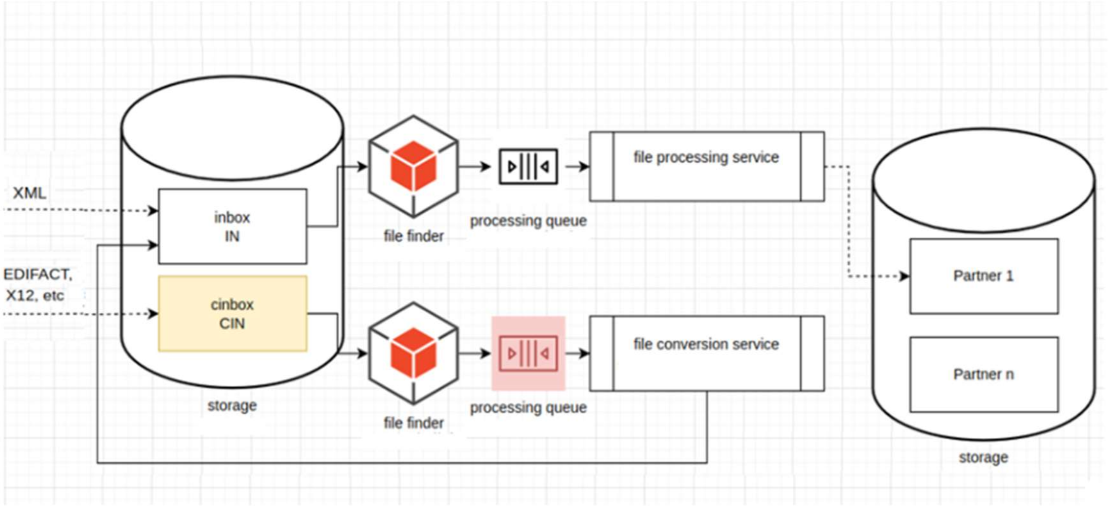


Figure 5: Failure of the conversion queue mechanism by a separate conversion service

It's also important to consider that resolving an outage takes time. Depending on the response speed and the failure's cause, the recovery process can take seconds, minutes, or even hours or days. The provider is very likely to fail to meet the Service Level Agreement (SLA). Depending on the business process and the terms of the counterparty, the provider risks incurring penalties. To ensure Service Level Agreement (SLA), a mechanism is required to process messages in priority mode depending on their type, format, recipient, and sender. However, the modern systems we've examined operate in FIFO mode. Determining the priority of a particular message can only be done indirectly through the message type (if correctly specified) in the file name and the sender's directory. This approach results in SLA violations in case of message processing failures.

The idea of the research is to add useful information to the file name, which allows to speed up the routing, as well as change the priority of message processing without the need to read the contents of the file. The application of a new principle of file naming helps to increase the amount of information that can be obtained in the intellectual analysis of data of the work of the service of the service transmission. And this, in turn, allows to get more metrics for a correct assessment of the effectiveness of routing.

The new naming principle is aimed at improving the routing of messages, in distribution systems, which should have great bandwidth. Such systems are usually the workflow providers and retail networks.

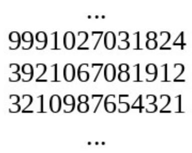
4. Solution of the problem

To achieve the stated goal:

1. The principles of indication of information about the recipients and senders, and the type of notice, are analyzed.
2. A new format EDI message name is proposed.

4.1. Analysis of current principles for defining the sender and receiver

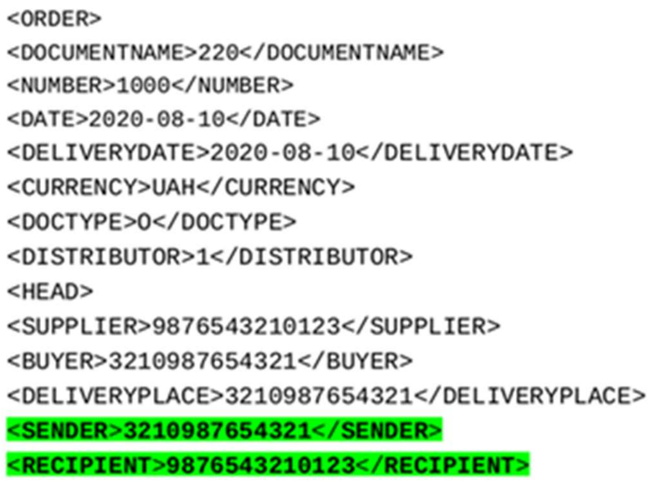
The Global Location Number (GLN) [22] is an identifier for both the recipient and the sender. Examples of GLNs are shown in Figure 6.



...
9991027031824
3921067081912
3210987654321
...

Figure 6: Examples of GLN identifiers

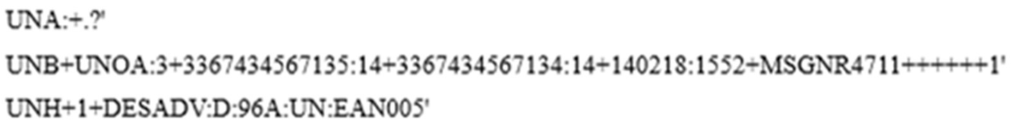
Currently, information about the recipient and sender of the message and the type of message is in the file's body. Here is an example of the contents of a file for an order with recipient and sender identifiers in XML format in Figure 7.



```
<ORDER>
<DOCUMENTNAME>220</DOCUMENTNAME>
<NUMBER>1000</NUMBER>
<DATE>2020-08-10</DATE>
<DELIVERYDATE>2020-08-10</DELIVERYDATE>
<CURRENCY>UAH</CURRENCY>
<DOCTYPE>O</DOCTYPE>
<DISTRIBUTOR>1</DISTRIBUTOR>
<HEAD>
<SUPPLIER>9876543210123</SUPPLIER>
<BUYER>3210987654321</BUYER>
<DELIVERYPLACE>3210987654321</DELIVERYPLACE>
<SENDER>3210987654321</SENDER>
<RECIPIENT>9876543210123</RECIPIENT>
```

Figure 7: Recipient and sender identifiers in ORDER XML format

The file contents in Figure 7 provide another example of recipient and sender IDs in EDIFACT format.



```
UNA:+.?'
UNB+UNOA:3+3367434567135:14+3367434567134:14+140218:1552+MSGNR4711++++++1'
UNH+1+DESADV:D:96A:UN:EAN005'
```

Figure 8: Recipient and sender identifiers in EDIFACT format

Essentially, to route a file, it's necessary to read it and locate the sender and recipient information within the document's body. Therefore, to determine the route a document should take, the following steps are required:

- Take the document for processing. There's usually a message queue from which documents are retrieved for processing.
- Determine the file format. In the case of EDI, common formats include X12, EDIFACT, ODETTE, VDA, XML, EANCOM, IDOC, and TRADACOMS, although files in other formats may also be received. Each of these formats has its unique structure and specifications.
- Identify the sender and receiver. Depending on the format of the file being transferred, these identifiers may be located in different tags, segments, or other elements within the file.

So, to identify the parties involved in the exchange, it is necessary to analyze the file's contents, which requires opening and reading the file. All of this takes time. The time needed to process different formats affects the speed of analysis and identification.

Based on the above, we can identify the following factors of influence on the speed of file distribution from the sender to the recipient: type of message, the need to read the file, the need to analyze the file, file format, file structure, and file volume.

4.2. Principles of file names

Currently, the filenames being used contain information about the sender and recipient identifiers within the message body, and they appear as follows:

- ORDER_e35v8b72bv92b8ceos.edifact
- ORDER_1147.edifact

- INVOICE_098654567890986457.xml
- INVOICE_20221110122422_9875467.xml
- ADA98435-2156FR.txt

We observe that the filename contains only the message type (ORDER, INVOICE), and the sequence of letters and numbers is unclear. This sequence is generated randomly to prevent duplication of previously created filenames. The fourth example is slightly different, with its name including the type, file creation time, and document identifier within the system where it was created. However, this information is unhelpful for file processing and does not improve routing. The fifth example provides no information whatsoever about the file's contents. The filename in the fifth example is formed by the sender according to some internal rules that are unlikely to be known by the provider. Thus, we see that there are no clear rules within the EDI systems that allow for the use of information from the filename for routing purposes.

It is proposed to include additional information in the filename that is found within the message body, specifically the sender and recipient identifiers, in addition to the file type, file identifier, and file format.

Counterparties may need to prioritize processing specific messages amidst the plethora of documents traversing electronic document management systems. Hence, the proposed solution is in incorporating an additional message "importance" indicator in the file name – a monogram. This attribute is typically unspecified by default. Utilize English alphabet letters as symbols, where "a" denotes the highest priority, followed by "b," and so forth in alphabetical order. This hierarchical marking system is intuitively logical and easily comprehensible, facilitating seamless interpretation by involved parties. Consequently, stakeholders in the exchange process and the facilitating providers gain the capacity to process vital messages labeled with this attribute sequentially, streamlining the handling of queued messages.

In this case, the name’s format will be as follows:

$$MesType\_priSign\_recGLN\_senGLN\_UUID.fmt, \tag{1}$$

where MesType – type of message,
priSign, – message priority symbol (monogram), optional,
recGLN – recipient GLN,
senGLN – sender GLN,
UUID - Universally Unique IDentifier,
fmt – format (specified as a file extension).

Let's illustrate the new name format with the following (Figure 9):

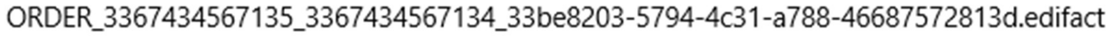

ORDER_3367434567135_3367434567134_33be8203-5794-4c31-a788-46687572813d.edifact

Figure 9: The example of a file name in the format we propose to enhance information transmission efficiency

In this example, the values with the payload are separated by the lower underlining symbol. The name "ORDER" indicates the message type, "3367434567135" – GLN recipient, "3367434567134"– GLN sender, and "33be8203-5794-4c31-a788-46687572813d" – message ID. Version 4 of UUID provides a unique name for each file among many files created and can be used as a message identifier on the sender's system.

Other useful information may also be added to the file name, depending on the type of message and the agreement of the counterparties, for example:

- File creation date.

- Date of product delivery.
- Date of expected payment on the account.
- Number of items with information about the product in the document's body.
- Number of retail outlets.
- Other.

Generally, it's recommended to include information in the filename that counterparties believe could expedite processing or allow for changes in the order of processing or message routing. At the same time, the filename length should not exceed 255 characters, unless this contradicts the counterparties' systems.

Thus, the file name, using UTF-8 encoding, may contain: Latin alphabet characters, both upper and lower case, numbers from 0 to 9, lower underlining characters (_); dash (-); dot (.) And in general, without specifying the type of message and other parts that will depend on the format of the document and the business process, it can be described by a regular expression:

$$(\backslash w|-|\_)\{1,\}\_((\backslash d)\{13\}\_)\{2\}(\backslash w)\{8\}-((\backslash w)\{4\}-)\{3\}(\backslash w\ )\{12\}.(\backslash w)\{1,\}. \tag{2}$$

It is not recommended to change the file name during its subsequent transfer, so downstream participants in the file transfer and processing chain can also utilize the additional useful information contained within the file name.

It is important to note that the file content typically contains commercially sensitive information, and based on the suggested file name, it is possible to identify the counterparties with whom a particular company has agreements. Therefore, it is recommended that all this information be transmitted through secure information systems that are protected against leakage. Similarly, when storing information, it is necessary to employ mechanisms to prevent information leakage. If a subsequent recipient in the transfer chain is not trustworthy, it is recommended that both the file content and its name be encrypted.

Using the file-naming principle mentioned above, avoiding the need to parse the message is possible. This means routing and setting processing priority can be done without reading the content of the file. As a result, an element can be added to the scheme shown in Figure 5 that can perform processing queue formation, considering the message processing priority. Figure 10 presents this new distribution element capable of reordering the message queue.

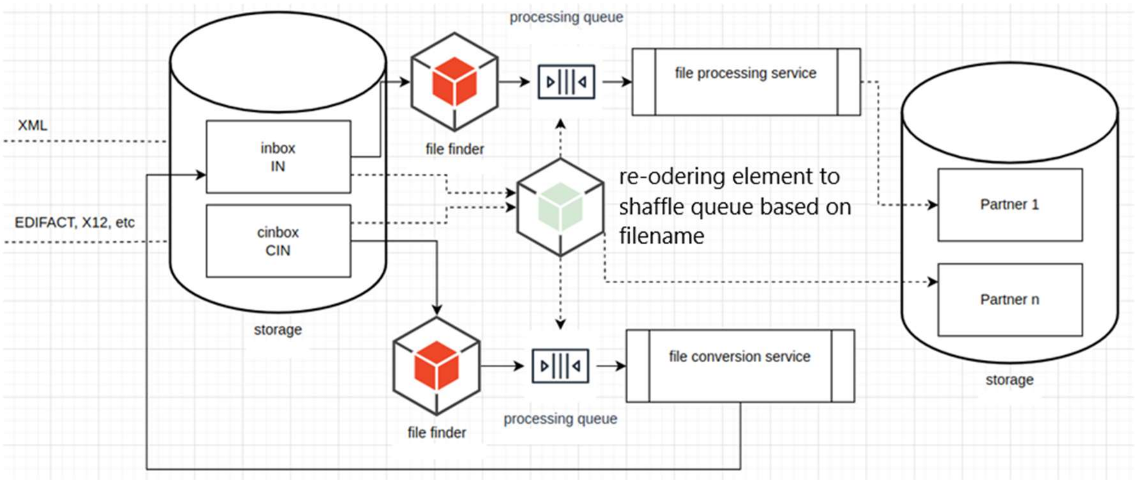


Figure 10: Schematic of handling messages of several formats with a distribution element based on filenames

5. Experimental results

For conducting experiments, a model of an EDI system, analogous to industrial systems, was developed. During the preparation for the experiment, a database of messages typical of the trade sector was created. The most frequently used formats in this sector are ORDER, which corresponds to the GS1 XML standard version 3.6; APERAK, which corresponds to the GS1 XML standard version 3.4.1; and INVOICE, which corresponds to the GS1 XML standard version 3.6. The message database contains 1000 messages of each format. The file sizes are ORDER – 4214 bytes; APERAK – 1046 bytes; INVOICE – 7563.
Ten experiments were conducted in each of two modes:

1. Distribution (file routing) based on information from the file name.
2. Distribution (file routing) based on information extracted from the file content.

For each of the 1000 iterations in every experiment, the transmission time of each specific message was recorded, from the element that created the message to the element that received it in its role as a client. Among the obtained data, the average, minimum, and maximum message processing times were determined for each dataset. The experimental results are presented in Table 1 and illustrated in Figure 11.

Table 1
The results of the experiment

Format	Format Method of Distribution	Average Processing Time, ns	Increase in Distribution Speed, %
APERAK	By the contents of the file	2217	
	By file name	1899	16,69
INVOICE	By the contents of the file	2001	
	By file name	1493	34,01
ORDER	By the contents of the file	2460	
	By file name	2008	22,51

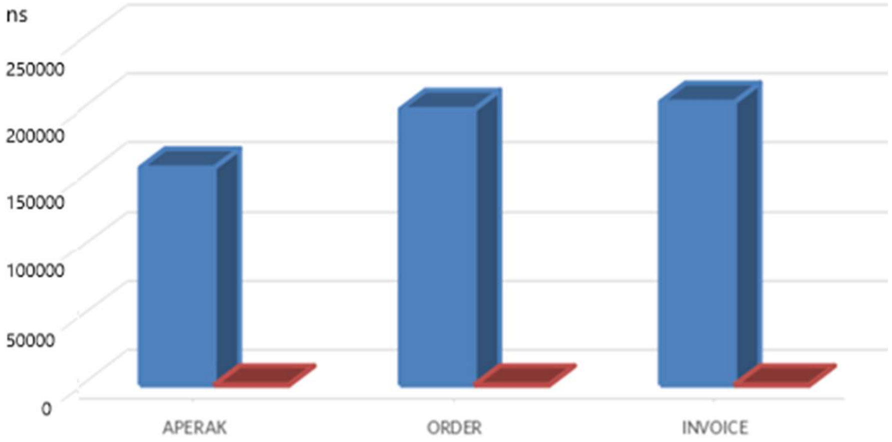


Figure 11: Comparison of the recipient's GLN receipt time by data source. Blue - from message body, Red - from file name

The experiment results demonstrate an increase in the speed of message processing due to changing the principle of file naming. For APERAK format files with a size of 1046 bytes, the improvement was 16.69%; for ORDER with a size of 4214 bytes, the improvement was 22.51%; for INVOICE with a size of 7563 bytes, the improvement was 34.01%. It can be noted that the larger the size of the message and the more complex the procedure for extracting identifiers from the body of the file, the more effectively the proposed principle of file naming demonstrates itself, also the size of the message still affects the speed of processing due to the need to transfer files through the network and other elements of the information system.

6. Discussion

6.1. Information security and anonymity

It should be noted that the file content typically contains information constituting a commercial secret, and based on the proposed file name, it's possible to identify the counterparties with whom a specific company has contracts. Therefore, a security concern arises. It is advisable to transmit all such information through securely protected information systems that prevent leaks. However, it's important to recognize that the nature of a GLN identifier inherently does not guarantee anonymity.

GLN addresses found in files and file names are publicly available information accessible through the GS1 service system. If anonymity is required, parties should resort to alternative identifiers. Data transmission within corporate networks generally occurs by default in secure systems, and data storage, including messages, also occurs in protected databases. In cases of message transmission or storage via compromised communication channels, changing the file naming format does not fundamentally increase risks, as intercepted messages can still be read. The information contained within the message body far outweighs the business value of any data that could be illicitly extracted from the file name. The research proposal helps expand potential ways to protect information within the message body; for example, encrypting the message body allows EDI providers to perform routing based on information in the message name without accessing the message body. In existing systems, routing requires extracting information about the exchanging parties from the message body, involving all intermediate exchange links, such as multiple EDI providers, who may gain access to and potentially use sensitive information from the message body.

6.2. Expose of information about existing systems

According to this work proposal file name formation should occur within the system responsible for the initial file creation to achieve maximum efficiency of the proposed solution. At the time of file formation, the system generating it already possesses all the necessary information to craft the proposed file name (excluding priority, as this parameter may not be typical for existing systems). Additionally, the file name can be adjusted to the proposed one at any subsequent stage of message transfer, such as by an electronic document management provider. Systems that do not support deriving useful information from a file name will continue to process messages as before the file naming format change. Systems that integrate filename processing logic with the proposed changes may benefit from the advantages outlined in this article.

7. Conclusion

This work presents a network data analysis of electronic message exchange within a document circulation system. By examining the patterns and characteristics of this communication network, we identified key opportunities for optimization. Specifically, our analysis led to the proposal of a novel file naming convention designed to embed critical metadata directly within the filename. The implications of this network-aware file naming scheme are significant for data flow and processing. We posit that it can lead to: reduced network latency in file routing; elimination of the need for content-based parsing at initial routing stages; the ability to implement priority queuing based on sender/recipient relationships observed within the network, without requiring deep packet inspection; and a simplified network path selection logic leveraging inherent network structure information encoded in the filename. To validate the efficacy of the proposed approach, we developed a simulation system for message transmission. This allowed for an experimental comparison between traditional routing methods and our proposed approach. The results of experiments demonstrate a substantial improvement in file routing performance across the simulated system.

Declaration on Generative AI

During the preparation of this work, the authors used Grammarly to check Grammar and spelling.

References

[1] Fortune Business Insights, The global Electronic Data Interchange (EDI) software market size, 2025. URL: https://www.fortunebusinessinsights.com/electronic-data-interchange-edi-software-market-103690.

[2] Structure of an EDIFACT Interchange, 2020. URL: https://www.odette.org/uploads/resources/document/OS11_Structure_of_an_EDIFACT_Interchange_v1.0_2021-05-07-125201.pdf.

[3] XML specification for electronic data interchange, 2023. URL: https://www.gs1.org/standards/gs1-xml/3-0.

[4] ORDER-XML Specification, 2024. URL: https://wiki.edin.ua/en/latest/Distribution/EDIN_2_0/XML/XML_structure.html#order-document-order.

[5] EDIFACT, EANCOM Syntax 4, 2020. URL: https://gs1.se/wp-content/uploads/sites/2/2020/08/edifact-syntax-eancom.pdf.

[6] X12 EDI Specification, 2020. URL: https://www.stedi.com/edi/x12.

[7] J. Bahija, The prevention and reduction of the bullwhip effect by electronic data interchange and collaborative forecasting, European Scientific Journal 17 (2021). doi:10.19044/esj.2021.v17n23p163.

[8] E. Weisz, D.M. Herold, S. Kummer, Revisiting the bullwhip effect: how can AI smoothen the bullwhip phenomenon?, The International Journal of Logistics Management 34 (2023) 98–120. doi:10.1108/IJLM-02-2022-0078.

[9] Q. Li, W. Tian, H. Zhang, Digital transformation for sustainability in industry 4.0: alleviating the corporate digital divide and enhancing supply chain collaboration, Systems 13 (2025) 123. doi:10.3390/systems13020123.

[10] T. Ghosh, Combating the bullwhip effect in rival online food delivery platforms using deep learning, Arxiv (2025). doi:10.48550/arXiv.2503.22753.

[11] A. Sarıcıoğlu, G. E. Müjde, M. Cedolin, Analyzing one-step and multi-step forecasting to mitigate the bullwhip effect and improve supply chain performance, IEEE Access 12 (2024) 180161-180174. doi:10.1109/ACCESS.2024.3510175.

[12] Y. Yang, J. Lin, G. Liu, L. Zhou, The behavioural causes of bullwhip effect in supply chains: a systematic literature review, International Journal of Production Economics 236 (2021) 108-120. doi:108120. 10.1016/j.ijpe.2021.108120.

[13] K. el Fellah, I. el Azami, A. el Makrani, A comparative analysis of blockchain and Electronic Data Interchange (EDI) in supply chain: Identifying strengths, weaknesses, and synergies, Journal of Autonomous Intelligence 7 (2024). doi:10.32629/jai.v7i5.1481.

[14] L. Koh, A. Dolgui, J. Sarkis, Blockchain in transport and logistics – paradigms and transitions, International Journal of Production Research 58 (2020) 2054-2062. doi:10.1080/00207543.2020.1736428.

[15] IBM, What is electronic data interchange (EDI)?, 2025. URL:https://www.ibm.com/tr-tr/topics/edi-electronic-data-interchange

[16] E. Sahinaslan, O. Sahinaslan, F. Ilgun, EDI technology, an exemplary application developed on this technology and speed performance results, Proceedings of the 5th international conference of mathematical sciences, volume 2483, AIP Publishing, Paparazzi Press, Istanbul, Turkey, 2022, 070003. doi:10.1063/5.0115601.

[17] V. Velaga, K. Routhu et al. Transforming Supply Chain Performance Based on Electronic Data Interchange (EDI) Integration: A Detailed Analysis,. European Journal of Applied Science, Engineering and Technology (2025) 25-40. doi:10.59324/ejaset.2025.3(2).03.

[18] S. Leech, J. Dunne, D. Malone, A framework to model bursty electronic data interchange messages for queueing systems, Future Internet 14 (2022) 149. doi:10.3390/fi14050149.

[19] S. Gavrylenko, V. Zozulia and N. Khatsko, Methods for improving the quality of classification on imbalanced data, Proceedings of the 4th KhPI Week on Advanced Technology, KhPIWeek, Kharkiv, Ukraine, 2023, pp. 1-5. doi: 10.1109/KhPIWeek61412.2023.10312879.

[20] O. M. Nikulina, K. O. Khatsko, Method of converting the monolithic architecture of a front-end application to microfrontends, Bulletin of the National Technical University "KhPI", Ser. : System analysis, control and information technology 2 (2023) 79–84. doi: 10.20998/2079-0023.2023.02.12

[21] M. Zamkovyi, S. Gavrylenko, K. Khatsko and N. Khatsko, Algorithmic Support for Building a Distributed IoT System in a Cloud Service, Proceedings of the 4th KhPI Week on Advanced Technology, KhPIWeek, Kharkiv Ukraine, 2023, pp. 1-6, doi: 10.1109/KhPIWeek61412.2023.10312994.

[22] GS 1 - Global Location Number. URL: https://www.gs1.org/docs/idkeys/GS1_GLN_Executive_Summary.pdf .