ISSN 0974-763X
UGC-CARE Listed Journal

SOUTH ASIAN JOURNAL OF MANAGEMENT RESEARCH (SAJMR)

Listed in UGC-CARE
Special Issue

Volume 14, No.4 November, 2024

Chhatrapati Shahu Institute of Business Education & Research (CSIBER)
(An Autonomous Institute)
University Road, Kolhapur - 416004, Maharashtra State, India.
website : www.siberindia.edu.in
E-mail : editorsajmr@siberindia.edu.in

Chhatrapati Shahu Institute of Business Education and Research (CSIBER)

South Asian Journal of Management Research (SAJMR)

Special Issue

Volume 14, No. 4, November 2024

Editor: Dr. Pooja M. Patil

Publisher

CSIBER Press

Central Library

Chhatrapati Shahu Institute of
Business Education & Research (CSIBER)
University Road, Kolhapur – 416004, Maharashtra, India.
Phone: 91-231-2535706/07, Fax: 91-231-2535708,
Website: www.siberindia.edu.in
Email: csiberpress@siberindia.edu.in
Editor Email: editorsajmr@siberindia.edu.in

Address:
CSIBER Press
Central Library Building
Chhatrapati Shahu Institute of Business Education and Research (CSIBER),
University Road Kolhapur, Maharashtra - 416004, India.

ISSN: 0974-763X

Price: INR ₹ 1,200/-

Editor: Dr. Pooja M. Patil

Distributed By

CSIBER Press

Central Library

Chhatrapati Shahu Institute of
Business Education & Research (CSIBER)
University Road, Kolhapur – 416004, Maharashtra, India.
Phone: 91-231-2535706/07, Fax: 91-231-2535708,
Website: www.siberindia.edu.in
Email: csiberpress@siberindia.edu.in

South Asian Journal of Management Research
(SAJMR)

Special Issue

Volume 14, No. 4 **November, 2024**

C O N T E N T S

A Method for Improvisation of Electronic Data Exchange in E-Commerce Applications

Dr. Mohammed Shameer M C
Assistant Professor, Dept. of Computer Science, Farook College(Autonomous), Kozhikode, India.

Miss. Mubeena V
Assistant Professor, Dept. of Vocational Studies, Farook College, Kozhikode, India.

Abstract

Increasing popularity of communication technology and services has revolutionized the way business operates. Many businesses has migrated to new platform and incorporated online services like e-com services, mobile banking, mobile payments and m-commerce. These migrations exponentially increased the network traffic and bulkiness of data. This created bottlenecks in data exchange. So there is a demand for strong and secure means of business communications. As complete technological revamp is not possible, efficient use of existing resources is essential. Reducing the amount of data exchanged can also positively impact the efficiency of data transfer e-commerce applications. This research aims to integrate the LXML data interchange format into e-commerce applications. LXML is a lightweight data interchange format proposed for environments with resource restrictions. The performance of the new model is evaluated and found that it has clear advantage over the existing conventional data interchange formats.

Keywords: Electronic Data Exchange, E- Commerce, Data Exchange Format, LXML

Introduction

Electronic Data Exchange refers to the way to exchange of structured business data between organizations in a reliable and secure manner. This process facilitates automating business data transfer without human intervention, enabling businesses to communicate efficiently. At the same time eliminates the manual tasks required with traditional data exchanges. Advancement in field of internet and communication technology has profoundly transformed the way businesses operate by offering enhanced communication, innovations and global reach. This nurtured the growth of ecommerce and mobile commerce platforms.

As more businesses move online platforms, network traffic rises significantly. This requires robust infrastructure to handle the huge volume of data and user interactions. At the same time, heavy use of rich media led to the increased bandwidth consumption as well. With the rise of mobile commerce, networks must accommodate mobile traffic efficiently [1]. Businesses may need to enhance mobile responsiveness and ensure their infrastructure supports mobile users. High traffic can lead to latency, affecting the user experience. Businesses need to optimize their networks to minimize delays and ensure fast responses. As a result, a simple and yet feasible means of data exchange is essential to support the growing need [2].

As complete revamp of existing technology and infrastructure is seldom possible, effective use of existing resource is mandatory. Reducing the amount of data to be transmitted in business communication and effectively managing existing resources is crucial in this scenario. Use of lightweight and standardized data exchange formats streamline the business communications.

This research integrates the lightweight LXML data exchange format proposed in [3] in e-commerce transactions and evaluates the performance of the model in business data exchanges.

The later sections of this paper are organized as follows. Section 2 reviews the existing data interchange formats. Section 3 discusses the methodology used for the research. Section 4 explains the details of how the new LXML format in integrated and implemented. Section 5 highlights the experimental set up and evaluation performed. Section 6 analyses the result and section 7 concludes the research findings.

Literature Review

XML

XML is the most widely used data interchange format especially in business to business (b2B) applications. XML is a mark up language derived from SGML and is used to describe the syntax and semantics of structured and semi-structured data. It is a standard data interchange format for heterogeneous environment. XML supports a hierarchical organization data using custom tags. XML documents are human readable, dynamic and extensible in nature.

The wide spread used of XML in ecommerce application is mainly due to the following reasons [4][5]: (a) Human readable and self describing format suitable for storage and transmission of structured data, (b) platform independent and support interoperability among heterogeneous platforms and applications (c) inherent support for complex data structures, (d) open standard that make it easier to generate, parse and transmit.

Though XML is considered as the de-facto standard in ecommerce applications, it posses many drawbacks especially in resource constraint environments like m-commerce platforms [5][6][7][8].

- Document verbosity: XML make use of custom tags for representing data. Due to the use of large number of descriptive custom tags the documents size increases. Even though the descriptive tags improve the readability of the document, it shrinks the amount of data contained in the documents
- Heavily descriptive and redundant tags increase the transmission cost and other processing overheads.
- Complexity of data structures used for handling XML documents
- Resource intensive nature
- Security issues

JSON

JSON is another lightweight, platform independent data exchange format used in e commerce applications. It is an alternative format popularly used in RESTful APIs for web services [9]. JSON is a text based format allowing seamless communication between front-end applications and back-end services. It is comparatively easier to parse and extract data contained in the document. They are supported by many programming languages and databases.

The demerits of JSON as a format for data exchange in ecommerce applications includes [5][7]: (i) Lack of schema: A schema support to structuring and organizing the document and helps in parsing and checking the document validity. Absence of a schema makes the format more generic and error prone, (ii) JSON is a text based format with support to few data types. It does not support date time and other complex types. Hence it is very difficult to represent non text types; (iii) does not support namespaces and comments (iv) verbose and non extensible, (v) vulnerable to security breaches as sniffing JSON objects can easily identify object properties and alter its values.

CSV

CSV files are simple text based files commonly used for handling bulk amount of data uploads and downloads in ecommerce applications. They are ideal for representing structured data and are supported by many programming languages. The document sizes of CSV files are much smaller than XML and JSON making it suitable for bulk data handling. The disadvantages of this format includes: (i) Lack of standardization: it is an inconsistent format for data representation. This inconsistency leads to compatibility issues, (ii) Inability to represent hierarchical data due to data structure limitations, (iii) not suited for complex types, (iv) limited data types, (v) scalability and security concerns [10][11].

YAML

YAML is a data serialization and transmission standard utilized by various programming languages. It is lightweight and human-readable format, making it ideal for applications that store, process, and transmit data in distributed applications. YAML are human readable format that inherently support complex data structures [12]. They are mainly used in configuration files and data serialization. YAML is not a popular format like XML and JSON and every language has different comfort level in using this format. It has a complex structure and requires much expertise to write policy constraints. As the format heavily depend on indentation formatting, a minor change in intent may cause in processing failures [12][13].

Protocol Buffers

Protocol Buffers is a precise, language-neutral, platform-independent and extensible mechanism for serializing structured data developed by Google Inc. Protobuf is a binary data format that can handle versioning issues in server during system development. This format is similar to XML and JSON in terms of serialization but more precise and faster [14]. This format is considered as very effective for systems that communicate large amount of data in constrained environments. The disadvantages of protoBuf are [15]: (i) it is not a human readable format making it difficult for debugging and manual data inspection, (ii) Generating such a format requires

specific tools to compile schema files into code, making it complex to handle, (iii) No flexibility for applications with highly dynamic or unpredictable data structures, (iv) overhead in serialization and deserialization of data, (v) supported by limited programming languages.

Choosing the right data exchange format in e-commerce depends on specific requirements like data complexity, interoperability, performance, and ease of use [16][17]. Literature survey underlines that existing data exchange mechanism employed in e-commerce and m-commerce application are verbose in nature leading to larger payload size. Larger payload size increases the processing overhead and slower parsing which adversely affect real-time processing in high-traffic business environments. This performance bottleneck impact user experience as the amount of data communicated in ecommerce applications varies.

Modern business applications that give emphasis to speed and efficiency require a paradigm shift on existing data exchange mechanisms. The trend is shifting towards more lightweight, flexible, efficient and developer-friendly formats.

Methodology

LXML is a lightweight data interchange format proposed as an alternative for existing data interchange formats like XML and JSON [3]. The format is an extensible, schema aware and less verbose format suited for environments that posses inherent resource limitations and constraints. This format is much concise when compared to XML and JSON.

In this research work, different e-commerce applications are considered as candidates for experimentation. The pattern of how the data communicated between the front end and back end system in these applications are studied. The first candidate platform exposes XML based web services and the second application uses RESTful web services where JSON is used as the payload.

The payloads with XML and JSON format are converted to LXML formats are the transmission is simulated. The content density and transmission time required for each is evaluated and analysed. The results obtained in these experiments definitely provide quantitative and qualitative insights to the performance of data exchange formats employed in e-commerce application.

Implementation

Most of the e-commerce and m-commerce applications follow a multi-tier distributed architecture [16]. The architecture of the e-commerce system is shown in Figure 1.

The client side: The client side constitute web or mobile application where users interact with the system. This is the place where user searches for products or items, add to their cart and make payments. This layer should be attractive and elegantly presented for better user experience. Client side of mobile applications are designed for touch interactions, optimized for smaller screens.

Server-Side: The server side of the distributed application handles incoming requests from clients and provide the responses. The web server interacts with a layer that runs the business logic.

Database Layer: This layer holds the data and popular choices include relational databases, NoSQL Databases or CRMs.

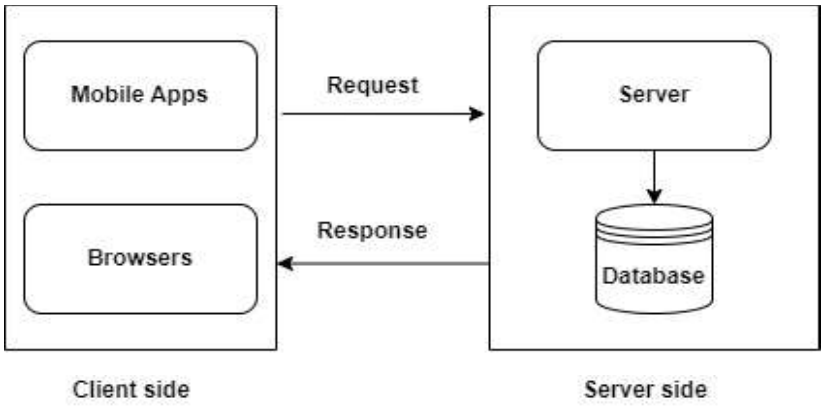

Figure 1: Architecture of an e-commerce system

LXML Format

The LXML is a hierarchical representation of data similar to XML. It is a format derived from XML. It uses level number instead of user defined tags. Each data is associated with its level number in its hierarchy. Since LXML uses level numbers instead of custom tags, this representation is very concise in nature.

XML and JSON messages can be converted to LXML format. A sample XML and JSON messages along with its equivalent LXML message is shown in the Figure 2.

```
<device>
   <category>
      <name>laptop</name>
      <series>A01</series>
      <model>M1</ model >
   </ category >
   <products>
      <product>
         <name>L01</name>
         <price>34000</price>
         <cpu>i3</cpu>
         <memory>8</memory>
      </product>
      <product>
         <name>L02</name>
         <price>44500</price>
         <cpu>i5</cpu>
         <memory>8</memory>
      </product>
   </products>
</device>
```

(a) XML Message

```
{ "device":
    { "category":
        { "name": "laptop",
          "series": "A01",
          "model": "M1" },
"products":
 { "product":
   [
     { "name": "L01",
       "price": 34000,
       "cpu": "i3",
       "memory": 8 },
     { "name": "L02",
       "price": 44500,
       "cpu": "i5",
       "memory": 8 }
   ]
 }
}
}
```

(b) JSON Message

```
device
<00>
<01>laptop
<01> 01A
<01>M1
<00>
<01>
<02>L01
<02>34000
<02>i3
<02>8
<01>
<02>L02
<02>44500
<02>i5
<02>8
```

(c) LXML Message

Figure 2: A sample XML, JSON messages and its equivalent LXML message

The LXML messages are created according to a schema. This schema provide information about the root tags, container tags, level of nesting and other meta data information about the document. This schema is responsible for parsing and extracting data out of the LXML document. The schema of the LXML message is shown in figure given below.

```
<device>
      <category>
            <name>laptop</name>
            <series>A01</series>
            <model>M1</ model >
      </ category >
      <products>
            <product>
                  <name>L01</name>
                  <price>34000</price>
                  <cpu>i3</cpu>
                  <memory>8</memory>
            </product>
            <product>
                  <name>L02</name>
                  <price>44500</price>
                  <cpu>i5</cpu>
                  <memory>8</memory>
            </product>
      </products>
</device>
```

(a) XML Format

```
device
<00>
<01>laptop
<01> 01A
<01>M1
<00>
<01>
<02>L01
<02>34000
<02>i3
<02>8
<01>
<02>L02
<02>44500
<02>i5
<02>8
```

(b) LXML format

```
<device>
<category>
   <name></name>
   <series></series>
   <model></model>
</ category >
<products>
   <product>
      <name></name>
      <price></price>
      <cpu></cpu>
      <memory></memory>
   </product>
 </products>
</ device>
```

(c) LXML Schema

Figure 3: XML message, LXML message and its schema

In this application, the client side make a HTTP request for data from the server. This request is generated according to the filter conditions and preferences given by the user. When the request reaches the server, the business logic layer extracts the data from the storage and creates an LXML message and transmitted to the client as an HTTP response. This LXML response is parsed to extract the data.

Experiment Setup and Evaluation

A test bed is configured using a laptop and an android mobile phone. The laptop has a core i3 processor with 4GB RAM and Microsoft Windows 10 as operating system. The mobile phone has Android version 12 as the operating system. The laptop has JDK version 1.8 installed. SoapUI tool is used for simulating servers. DOM parsing method is used for parsing XML and JsonParser is used for parsing JSON.

The responses generated from applications have a structure shown below. The root tag of the document is Products. This tag encloses many categories of products; say for example, mobile phone, laptop, desktop etc. Each category is further divided into sub categories. Sub categories are again classified into sub-sub categories. Each sub categories enclose many items. Each item contains 18-20 tags to describe the attributes of the item. There are thousands of such items in each document.

In this research, XML and JSON files generated from two different e-commerce application are considered. Here XML and JSON have almost similar structure. XML uses container tags where as JSON uses array objects. The structure of XML document is as shown

- Root Element: <products> contains multiple <product> elements.
- Product Element: Each <product> has a <productName> and a <categories> section.
- Categories: Each <category> has a <categoryName> and a <subCategories> section.
- Subcategories: Each <subCategory> has a <subCategoryName> and an <items> section.
- Items: Each <item> has a <name> and a <details> section, which contains multiple <detail> elements describing the item.

The structure of JSON file is

- Root Object: The root is products, containing a product array.
- Product Structure: Each product has a productName and a categories object.
- Categories: Each category has a categoryName and a subCategories object.
- Subcategories: Each subcategory contains a subCategoryName and an items object.
- Items: Each item has a name and a details object, with a detail array holding multiple descriptions.

The LXML format is compared with two prominently used data exchange formats- XML and JSON against three parameters. These three parameters are content density, parsing time and transmission time. These parameters are crucial for any data exchange applications especially in ecommerce field due to the bulkiness of data exchanged and processed [9][20][21].

Content Density

Content density refers to the amount of data contained in the format. It is measured as the amount of data contained against the total document size. The content density of a compact representation will be close to 1 and is considered as most suited for electronic data exchange. The content densities of XML, JSON and LXML are tabulated in Table 1.

XML	JSON	LXML
0.420	0.531	0.859

Table 1: Content densities of XML, JSON and LXML messages

Parsing Time

Parsing time is the time taken to process and extract data contained in the document. Parsing is an important performance criterion as it impacts the overall system performance. Validity of the document should be ensured before parsing. Parsing time depends on the technology used.

In this experiment, XML is parsed using DOM parser and JsonParser is used for parsing JSON. Custom APIs are developed to parse LXML messages. Here direct mapping to Data objects (DO) are performed to parse LXML. The LXML schema is used to validate LXML message before parsing. Parsing time of XML, JSON and LXML documents are tabulated in Table 2 and plotted in Figure 4.

No of Records	XML	JSON	LXML
1000	128	92	78
5000	536	326	260
10000	1318	779	632

Table 2: Parsing Time for XML, JSON and LXML messages for various number of objects

Transmission time

In any e - commerce or m-commerce applications, large volume of data is communicated between the server and the client device. Such bulk data is retrieved in pages and transactions spans multiple pages due to network or bandwidth constraints. In such cases, the total transmission time is the sum total of the time taken for each individual transaction and the time taken to process the messages.

For experimental purpose, documents having 1000 objects (items in sample transactions) are considered as a page. The download bandwidth is5.1 Mbps. The results obtained are tabulated in Table4 and plotted in Figure 5.

Pages	XML (in s)	JSON (in s)	LXML (in s)
1	2.231	1.351	0.642
2	4.465	2.685	0.642
3	6.974	2.685	0.642
4	9.861	5.231	1.277
5	12.197	5.231	1.277
6	15.360	7.332	1.277
7	18.948	7.332	2.614
8	22.476	9.742	2.614
9	26.258	9.742	2.614
10	29.544	12.591	5.546

Table 3: Page wise transmission time required for XML, JSON and LXML

Result and Analysis

From the table, it is clear that LXML is the most compact representation. A compact representation consumes less network bandwidth and other resources when compared to other bulky formats. The content densities of other formats is around 0.5 indicates that around 50% of the document size is used for meta-data like information and is not actual data. Such meta-data are forced to include either due to format specific constraints or to improve document readability.

From the result obtained in Table 2, It is very clear that LXML message has least processing overhead. Lower processing overhead considerably reduces the processing at the client side of the application. This enhances the end user experience and satisfaction.

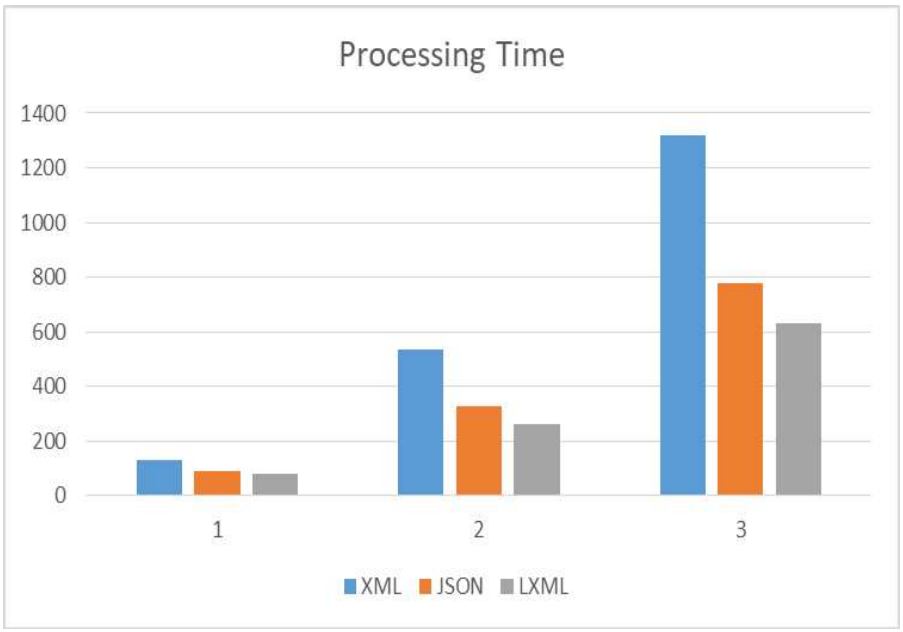

Figure 4: Parsing Time for XML, JSON and LXML messages for various number of objects

The cumulative transmission time for XML, JSON and LXML is presented in figure. The page wise processing and transmission of LXML message is the least among the three formats. This advantage is clearly due to the concise and compact nature of LXML format.

Figure 5: Page wise transmission time required for XML, JSON and LXML

Conclusion

There are many advances in e-commerce and m-commerce transactions to make it more convenient, secure and versatile in providing seamless interactions. Mobile computing helped financial sector to mobilize more financial services beyond payments. This includes bill payments, fund transfer to investment management. At the same time, mobile payment system also made huge transitions. Recently mobile payment applications and payment gateway interfaces gained much popularity.

Advances in the field exponentially increased the network traffic and the amount of data to be transmitted. Web services based on XML and JSON are the most commonly used approaches in ecommerce application where bulk amount of data is communicated. As the verbosity of XML and JSON is quite higher, data transmission using these conventional technologies results in wastage of network resources.

From the evaluation, it is clear that LXML is the most concise format for data transmissions. The content density of LXML is 104.5% higher than XML and 61.8% denser than JSON. Formats with higher content density consume less network resources for transmission, reduces the transmission cost and hence improve transmission efficiency.

Paring and processing time is crucial for any data communication system. It is found that the parsing overhead is the least for LXML and is considerably higher in XML. LXML format has 38-52% advantage over XML and 8-20% advantage over JSON in parsing.

The research highlights that transmission of LXML message has considerable advantage over its counter parts. The page wise transmission time of LXML messages consume only 1/3rd of the time taken for XML messages and there is 52% advantage over JSON.

Higher content density and advantages in processing and transmission of LXML message makes it promising format for data exchange in business application especially in m-commerce applications with resource limitations.

Reference:

Ben Kiki O. YAML Ain't Markup Language (YAML) Version 1.2. Available athttps://yaml.org/spec/1.2/spec.html, (Accessed March 2021).

Chen, B., Linz, D. D., & Cheng, H. H. (2008). XML-based agent communication, migration and computation in mobile agent systems. Journal of Systems and Software, 81(8), 1364–1376. doi:10.1016/j.jss.2007.10.026

Grzegorz Chodak, Grażyna Suchacka, Yash Chawla, HTTP-level e-commerce data based on server access logs for an online store, Computer Networks, Volume 183, 2020, ISSN 1389-1286, doi:10.1016/j.comnet.2020.107589.

G. Kaur and M. Fuad. An evaluation of protocol buffer. In Proceedings of the IEEE Southeast Con 2010 (SoutheastCon), pages 459–462, 2010.

I. Tatarinov, S. D. Viglas, K. Beyer, J. Shanmugasundaram, E. Shekita, and C. Zhang, "Storing and querying ordered xml using a relational database system," in Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data. New York, NY, USA: Association for Computing Machinery, 2002, pp. 204–215.

Jackson JSON Processor Available at http://jackson.codehaus.org/ (accessed on Sept 2024).

Jorstad, Ivar & Bakken, Elias & Johansen, Tor Anders. (2008). Performance Evaluation of JSON and XML for Data Exchange in Mobile Services. 237-240

K. Maeda. Comparative survey of object serialization techniques and the programming support. Journal of Communication and Computer, 9:920 – 928, 2012.

K. Maeda, "Performance evaluation of object serialization libraries in XML, JSON and binary formats," 2012 Second International Conference on Diital Information and Communication Technology and it's Applications (DICTAP), Bangkok, 2012, pp. 177-182, doi: "10.1109/DICTAP.2012.6215346"

M. C. Mohammed Shameer, P. P. Abdul Haleem, Yazik K. Puthenpediyakkal, "A lightweight Data Exchange Format for Mobile Transactions", International Journal of Computer Network and Information Security(IJCNIS), Vol.15, No.3, pp.47-64, 2023. DOI:10.5815/ijcnis.2023.03.04

Nicola, Matthias & John, Jasmi. (2003). XML parsing: A threat to database performance. International Conference on Information and Knowledge Management, Proceedings. 175-178. 10.1145/956863.956898.

P P Abdul Haleem and M P Sebastian (2014). An alternative approach for XML messaging, International Journal of Advanced Research ISSN NO 2320-5407, Volume 2, Issue 1, 251-294

Philipp Wehner, Christina Piberger, Diana Göhringer, "Using JSON to manage Communication between Services in the Internet of Things" 9th International Symposium on ReCoSoC, Montpelier, France 2014, pp 1-4.

Ramon Lawrence,The space efficiency of XML,Information and Software Technology,Volume 46, Issue 11,2004,Pages 753-759, ISSN 0950-5849, doi:10.1016/j.infsof.2004.02.003.

Siau, Keng & Lim, Ee-Peng & Shen, Zixing. (2001). Mobile Commerce: Promises, Challenges and Research Agenda. J. Database Manag.. 12. 4-13. 10.4018/jdm.2001070101.

Saurabh Zunke, Veronica D'Souza, JSON vs XML: A Comparative Performance Analysis of Data Exchange Formats, International Journal of Computer Science and Network, Volume 3, Issue 4, August 2014

Sol Chen, (2023). CSV Files: Use cases, Benefits, and Limitations, Available at https://www.oneschema.co/blog/csv-files. (accessed on Sept 2024)

Soap ui, Available at https://www.soapui.org/tools/soapui/ (accessed on Aug 2024).

White G, Kangasharju J, Brutzman D, Williams S (2007). Efficient XML interchange measurements note. Available at http://www.w3.org/TR/exi-measurements// (accessed on May 2020).

W. Li and Y. Liu, "Application of XML in E-commerce," 2009 WRI World Congress on Software Engineering, Xiamen, China, 2009, pp. 262-264, doi: 10.1109/WCSE.2009.36.

X. Zhu, Z. He, and S. Shi, "On an enhancement of xml applied for mobile e-commerce," Journal of Electronic Commerce in Organizations, vol. 10, no. 1, pp. 113–26, 2012.