Heighten Science PUBLICATIONS Corporation

ISSN
2575-0186

Review Article

WMW: A Secure, Web based Middleware for C4I Interoperable Applications

Nida Zeeshan*

University of Sindh, Pakistan

***Address for Correspondence:** Nida Zeeshan, University of Sindh, Pakistan. Email: nidazeeshan1403@gmail.com

Submitted: 07 January 2017
Approved: 18 January 2017
Published: 19 January 2017

Keywords: Semantic interoperability; Enterprise architecture; Information exchange; Middleware; HTTP; Software engineering

ABSTRACT

Modern-day enhancements in Enterprise Architectures (EA) has increased the interoperability issues in almost all domains; these issues are increasing day-by-day as organizations are spanning and information is being exchanged between different platforms. Command Control Computer Communication and Intelligence (C4I) complex systems are also facing the interoperability issues due to highly classified and sensitive information being exchanged. In this paper we have discussed the integration of different C4I applications running under heterogeneous platforms by allowing them to communicate using a secure and ciphered web based middleware named as Web Middleware (WMW). This middleware is a client-server based web adaptor to achieve clean, systematic, secure and reliable communication. The main feature among many is the simple HTTP browser based customization that do not require any specific or special add-ons and controls to be installed on the client machine. Architecture usage, and initialization of the WMW middleware is discussed with security and performance discussion.

INTRODUCTION

Data integration is the main goal and task of different organizations working under different domains from last 02 decades. Every organization has different means of data transfer depending on the environment they are using. A small organization comprises of few number of computerized systems doesn't have that much critical interoperability issues. Big organizations dealing with huge integration between different channels have major integration issues. These issues get critical when applications are penetrating on different platforms. Changing the complete computerized structure of a huge organization for future interoperability is a massive task that requires cost efficient solutions and will affect the whole enterprise working and behavior.

If any large scale domain lacks structure, efficiency, communication and on top of all information security then this will not only affect that particular organization but it will mess up the functions of other organizations associated with it. In this research, we are stressing upon Command, Control Computer, Communication and Intelligence (C4I) interoperability, commonly known as System of Systems (SOS) or Command & Control (C2) [1].

The main interoperability issue of a complex C4I System is joint operations; these issues are already well defined and pointed out in many studies [1-5]. There may be different C2 systems and applications running under different C2 platforms, the problem gets more complex when different C2 platforms are exchanging information using a single platform. Every C2 application has its own crucial information which may be required to interoperate with another C2 application and that information

How to cite this article: Zeeshan N. WMW: A Secure, Web based Middleware for C4I Interoperable Applications. J Forensic Sci Res. 2017; 1: 010-017. https://doi.org/10.29328/journal.jfsr.1001002

should be same and un-forged as it was sent. Considering this scenario, there are many interoperability issues and hurdles which are observed e.g. platform dependency, security structures, terminology differences, metadata structuring and many more [6,7].

The best way to accomplish complete and secure interoperability is to introduce a middleware platform that act as a bridge to communicate different domains to exchange information without platform dependencies. In a small organization, the integration platform can be based on desktop application. But in a huge environment like C2 infrastructure, where a single system is controlling and interfacing many systems located remotely, the preeminent way to accomplish the interoperability task is to use secured web based middleware.

In this research we have proposed a Web Middleware (WMW). The rest of the paper structure is discussed as follows: Section II discusses literature review, Section III describes the WMW architecture, Section IV addresses its usage, Section V covers its core features, section VI covers the conclusion and future work.

LITERATURE REVIEW

Mainly, proxies are used for firewalling and other security related to networks. Although research is present on which the whole communication is dependable on proxies, however, very unassertive research is available that is identical to our research, especially for a complex system as C4I.

In 2013, Dickerson et al, discussed the ROSE methodology applied to a large scale SOS tactical data-link interoperability. A model-driven architecture is engaged using the ROSE method. This technique recommends a repeatable approach for the analysis performed by the Office of Chief Engineer of the U.S. Navy. This study concentrates on a framework and factorized SOS architecture for selection and evaluation of portfolio. This case study has analyzed the original data-link interoperability [8]. A general purpose proxy system and its usage have been discussed with respect to mobile environment in the paper title, “A General Purpose proxy filtering mechanism applied to mobile environment.” [9].

David. J and others, in their paper title “ANTS: A Toolkit for building and dynamically deploying Network protocols”, present a novel approach to build and deploying network protocols, without the need of coordination and without unwanted interaction between existing protocols [10]. Introduction of an adaptable interface, including adaptable protocols, to adopt the change in the resources and protocols is being presented by T. Kunz and J.P. Black, in their paper title “An Architecture for adaptive mobile applications.” [11].

Mika. L and others demonstrated their designed communication architecture that makes it possible to exploit the existing TCP/IP protocols. How the architecture is used to improve the WWW information browsing. In their paper title “Optimizing World Wide Web for Weakly Connected Mobile Workstations: An Indirect Approach” [12].

Above we have shortened our literature review posting due to the paper lengthiness, there were many reviews with respect to the settlements of remote adaptations of web information using proxies [13,14]. There were many papers related to time-outs and delays and performance. Example profile based pre-fetching with local and remote proxies to reduce the web surfing waiting time delay. As discussed in [15] the number of links diagonally on the wireless link can be reduced using remote processing.

WMW ARCHITECTURE EXPLANATION

There are numerous customs in which our research is different from other studies:

- WMW adapting system designed specifically for the WWW.

- We have used a simplified coding instead of using generalized coding, which we have discussed in depth in our forthcoming article.

- The prominent feature and utilization of Restricted Working (RW) that is capable of handling multiple Distant Working (DW) at a time.

The stimulation behind this research is to develop and program a web based middleware that acts as an overpass to help communicate dependable platforms together without changing their dependencies and without motivating the client to install additional plug-ins/add-ons or upgrade infrastructure resources which are usually require in many middleware based applications.

In our interoperability process, when a request is passed by any application that require specific data integration from another application, it routes to the specific middleware adapter. The request is being altered according to the requirement of that middleware adapter before the communication. As a result, the request is being forwarded to the requested server for response. As soon as the response reaches back to the adapter it transforms or alter the response and send it back to the originator.

Requester doesn't have to know who the responder is and vice-versa. This is because, for the requester the responder is the WMW adapter which is available in the form of a proxy and for the responder the requester is also the WMW adapter. WMW architecture contains two workings; Restricted Working (RW) and Distant Working (DW). The requester has to deal with (RW) named as Restricted Working Request (RWR) and the responder has to contract with Distant Working Response (DWR), as illustrated in the figure 1.

PRIMARY ROLES OF RW & DW

That WMW middleware adapter is the combination of two workings i.e. RW and DW. It is because the roles for both are different and distinct in nature. The browser codes cannot be altered because altering any browser code during any data transformation is in any way not a suitable task. RW works crucially as a browser extension. The RW runs on RWR i.e. on request server, RW contains the information of system situations and availability of supply. The RW have further communication with DW for performance improving and performance status. Another primary role of RW is to keep track of data encryption and decryption being done in DW working.

DW distant working as the name speaks for itself is mainly responsible for remote activities for being deployed near the server or responder i.e. DWR. Main role of DW is to perform site dependable(s) and data firmness before it broadcast on any bandwidth. There may be number of web servers planted, each web server is handled by his own RW. In the same way, one client can easily handle multiple DW and each acts as a combination of separate RW and DW. However, the main difference is the RW generated by the client, runs on the same client server i.e. RWR, handling different

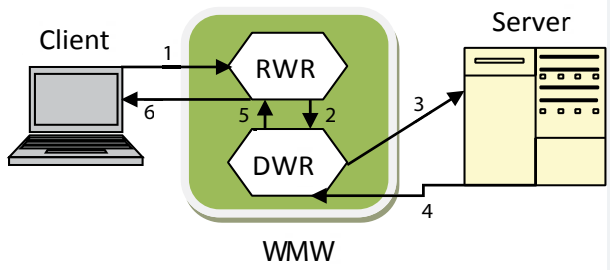

Figure 1: Restricted and Distant working of the adapter on RWR and DWR.

RWs within a single instance. DW is different against each RW and each DW is assigned to each DWR or Web server separately as shown in figure 2.

The RW is available in the form of proxy settings configured in the client's browser. The request automatically parses through the RW to their respective RWR which is then forwarded to the adapter. The client machine should be capable of handling such type of transactions; because these RWs can be located on different machines to distribute the load.

WMW USAGE

- The middleware adaptors are built to introduce a middleware that can be easily implemented and accessed on any client browser. Therefore, the RWs are available in the form of hyperlinks that are clickable to access the specified RW. Further, we discuss the following two significant behaviors of the WMW adapter workingInitializing WMW, and

- Query/Request execution

INITIALIZATION OF WMW

Browser should be configured to use the specified RWR Server. The proxy settings are the only thing which is required to do that. The set of instructions and information required to initialize the corresponding adapter by clicking the hyperlink is stored and contained in a file called Middleware Information File (MIF). Following information is stored in the MIF;

- The 'congregation machine name' running the DWR server that is responsible to execute the RW of the adapter.

- The verification certificates for the server that is running the RWR.

- Java Archiving file i.e. jar file that supports and includes the execution of the java classes.

- The java major class that is responsible for initializing the RW from the jar file.

- The java major class that is responsible for initializing the DW to send it back to the DWR server.

- Initializing configuration constraints for DW and RW.

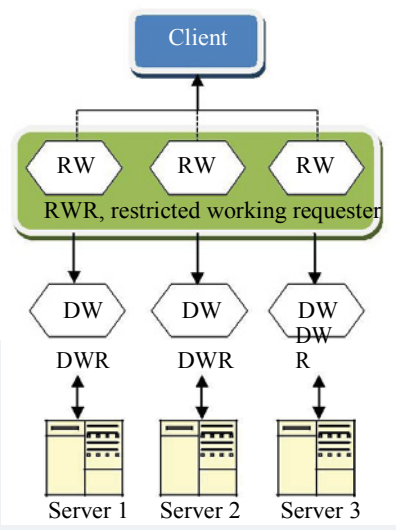

Figure 2: Several request handling.

- The relevancy of Domain, which spells out the sites to be operated by the adapter.
- Initialization and configuration page URLs.

The figures 3 and 4 are illustrating the above explanation clearly: and their appropriate MIF file.

The above illustration demonstrates that the client do not have to know any sort of knowledge about their server location; they don't require to store those DWs which are not requested by the client.

QUERY/REQUEST EXECUTION

The main WMW association is the repository of websites which we have named as Relevancy of Domains (RD). The adapter will only receive and send the query on the basis of these listed relevant websites. On the reception of request, the behavior of the RWR is as follow:

- Check the URL association within the RD domain.
- Send the query/request to RW and then to DWR Server.
- Choose the first initialized adapter if URL is common in multiple RDs.
- Send back the failure request to the Web Server if no URL is found in the RD.

All the above is illustrated in figure 5.

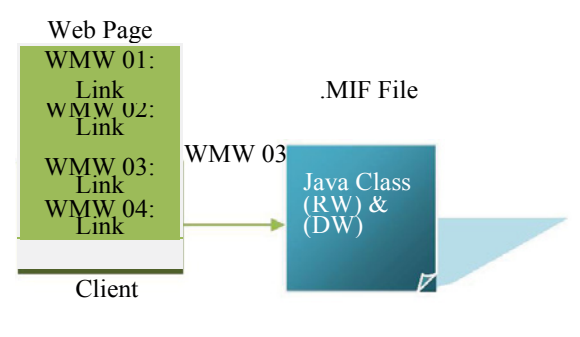

Figure 3: Client side webpage contains the list of adapters and their appropriate MIF file.

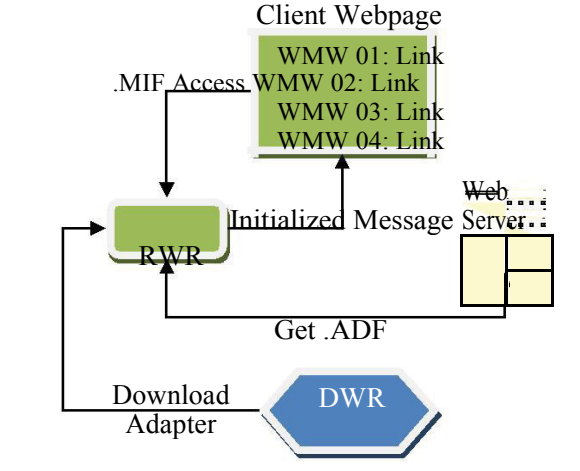

Figure 4: Initializing demo & message delivery to the client browser.

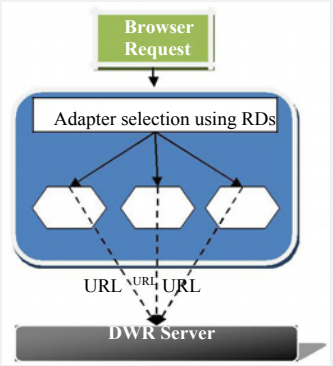

Figure 5: Adapter selection on the basis of RD.

INFORMATION SECUREITY & OTHER FEATURES & BENEFITS

Up-to this point we have discussed the architecture and behavior of our WMW. Further, we discuss the technical features of WMW. We have divided this discussion into following points:

- Operation Consistency
- Information Security & Confidentiality, and
- Performance

OPERATION CONSISTENCY

The reliability of our WMW middleware is unique in a sense that even on unreliable connections the data is not lost. The technique which we have used is simple; the client requests many or multiple documents while browsing normally. Each request contains a separate connection; in case of loss of connectivity the browser simply displays the broken placeholder that allows the user to re-send the request. In case of a weaker link, the DW stores the object temporarily until the request is successfully sent back to the client.

This all is done by the Operation Consistency Store (OCS), the OCS stores all the transactions records and flags only when there is a failure. The client operations are accessible to the client by accessing the non-completed requests. In this way there is no re-request generation.

INFORMATION SECURITY & CONFIDENTIALITY

There are two Coded/Encoded Mechanisms (CMs) available on both ends i.e. RW and DW. These CMs cipher the data from the requester and decipher it at the server end. The WMW middleware is basically content based, therefore; the CMs do not cipher/decipher sensitive information on detection.

The following is the flow of CMs during ciphering:

- Client browser request
- Ciphered by CM located in RW over an unstable and unsecured connection
- Query sent in ciphered form to DW
- Deciphering at DW using CM
- Non-ciphered query sent to the server
- Content received and sent to CM for ciphering

- Ciphered content sent to RW over non secured connection for deciphering
- Original query result provided to client

WMW is secured and reliable in a sense that the whole communication is taking place on proxies and whether how open the client side is, the (CMs) used at both ends will tighten the security. There are many other options available to increase the security walls that can also be taken into consideration i.e. by installing additional firewalls, filters etc.

PERFORMANCE

The performance factor is excessively vital for our WMW, if the performance is slow then nothing can be achieved and the concept of an efficient web middleware cannot be fulfilled.

The performance depends on the geographical locations of the contacted C2 servers, where the RWs are located. The web operations and WMW delay in response are the two important factors to be observed. These performance tests are planned to be covered in our forthcoming extended studies. Following listed issues are considered critical during operations, these issues are also highlighted in table 1.

- Operation request/response instant using adapter
- Client to RWR link transparency
- RWR to DWR link transparency
- Overall processing transparency
- Reply time from server to client
- Human sensitivity time-out
- Web operation time-out

CONCLUSION AND FUTURE WORK

This research discusses a thorough and detail functionality of WMW architecture, design, security and other benefits. The following conclusions have been made:

- WMW is flexible as compared to adaptation
- WMW provides the freedom of add-ons and pre-requisites that are the mandatory ingredients of most present middleware
- WMW is considered to be a very user-friendly and easy to implement middleware
- WMW is tightly interoperable to the web model, existing in present time

This research is an ongoing work therefore; in our forthcoming studies we demonstrate the WMW middleware transparency and information filtering testing on various domains to have the consistency usage and firm ability knowledge. Our future based on its implementation using a case-study and testing.

Table 1: Performance critical issues.

S.No	Performance	Issue
1.	Operations	Request & Response
2.	Transparency	Link/connection transparancy
3.	Reply query	Time-outs and delays
4.	Web & Human Operations	Time-outs and delays

REFERENCES

1. Alghamdi A, Siddiqui Z, Syed Quadri SA. A Common Information Exchange Model for Multiple C4I Architectures. Computer Modeling & Simulation (UKSIM). 2010; 538-542. Ref.: https://goo.gl/o4Niql

2. Ghamdi A, Siddiqui Z. Common interoperability Framework for defense Architectures, A web Semantic Approach. 16th Internationa Conference on Distributed Multimedia Systems. 2010; 14-16. Ref.: https://goo.gl/IpDqt1

3. Siddiqui Z, Abdullah A, Khan MK, Alghathbar K. Analysis of Enterprise Service Buses on Information Security, Interoperability and High-availability using Analytical Hierarchy Process (AHP) method. Physical Sciences. 2011; 6: 35-42. Ref.: https://goo.gl/MncrrU

4. Siddiqui Z, Abdullah AH, Khan MK. Qualified Analysis b/w ESB(s) using Analytical Hierarchy Process (AHP) Method. International Conference on Intelligent Systems, Modelling and Simulation. 2011; 100-104. Ref.: https://goo.gl/F9M44P

5. Siddiqui Z, Abdullah A, Khan MK, Ghamdi A. Node Level Information Security in Common Information Exchange Model (CIEM). College of Computer and Information Sciences. 2010; 21: 221-230. Ref.: https://goo.gl/rlZPma

6. Patrick E. Information breifing to M&S CoC. Simulation-C4I Interoperability (SIMCI); Overarching IPT (OIPT). 2008.

7. Tolk A. Beyond Technical Interoperability-Introducing a Reference Model for Measures of Merit for Coalition Interoperability. CCRTS. 2003; 47. Ref.: https://goo.gl/jIWlZS

8. Dickerson CE. A Relational Oriented Approach to System of Systems Assessment of Alternatives for Data Link Interoperability. Systems Journal IEEE. 2013; 7: 549-560. Ref.: https://goo.gl/kCegpM

9. Zenel B, Duchamp D. A general purpose proxy filtering mechanism applied to the mobile environment. ACM/IEEE. 1997; 248-259. Ref. https://goo.gl/D26blq

10. Wetherall DJ, Guttag JV, Tennenhouse DL. ANTS: A Toolkit for building and dynamically deploying network protocols. IEEE OPENARCH. 1998. Ref.: https://goo.gl/F3SnOJ

11. Kunz T, Black JP. An architecture for adaptive mobile applications. 11th International Conference on Wireless Communications. 1999; 27-38. Ref.: https://goo.gl/S4PJQD

12. Liljiberg M, Alanko T, Kojo M, Laamanen H, Raatikainen K. Optimizing World-Wide-Web for Weakly connected mobile workstations: An Indirect Approach. 2nd International Workshop on Services in Distributed and Networked Environments. 1999; 32-139. Ref.: https://goo.gl/j9vp8M

13. Buyukkokten O, Molina HG, Paepcka A, Winograd T. Power Browser: Efficient Web Browsing for PDA's. Proceedings of the SIGCHI conference on Human Factors in Computing Systems. 2000; 430-437. Ref.: https://goo.gl/Sd1rH8

14. Noble B. System support for mobile, adaptive applications. IEEE Personal Commu. 2000; 7: 44-49. Ref.: https://goo.gl/cbJDVR

15. Villate Y, Gil D, Goni A, Illaramendi A. Mobile Agents for providing mobile computers with data services. DSOM 98. 1999; 1-12. Ref.: https://goo.gl/hQj4Yu

16. Siddiqui Z, Abdullah AH, Khan MK, Alghamdi AS. Smart Environment as a Service: Three Factor Cloud based User Authentication for Telecare Medical Information System. Journal of Medical Systems. 2014; 38: 9997. Ref.: https://goo.gl/Klshmp

17. Siddiqui Z, Abdullah AH, Khan MK, Alghamdi AS. Cryptanalysis and improvement of 'a secure authentication scheme for telecare medical information system' with nonce verification. Peer-to-Peer Networking and Application. 2016; 9: 841-853. Ref.: https://goo.gl/XlV89E

18. Siddiqui Z, Alghamdi AS. SOA Based C4I Common-View Interoperability Model. Science International, Lahore. 2016; 26: 175-180. Ref.: https://goo.gl/VNW2Sj

19. Siddiqui Z, Alghamdi AS. A Universal View SOA Interoperability Framework for Multiple C4I Applications. Science International, Lahore. 2016; 26: 97-100. Ref.: https://goo.gl/QLF5aC