ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 1, January 2018

IMPLEMENTING ONLINE DRIVING LICENSE RENEWAL BY INTEGRATION OF WEB ORCHESTRATION AND WEB CHOREOGRAPHY

[1]RJegadeesan, Assistant Professor, RMK Engineering College, Chennai, [2]Dr N Sankar Ram, Professor, Sriram Engineering College, Chennai, [3]J Abirami, Research Scholar, Anna university, Chennai.
Email id: [1]ramjaganjagan@gmail.com, [2]n_sankarram@yahoo.com and [3]abi.parimala@gmail.com

ABSTRACT:

The Web services architecture defines separate specifications for the composition and the coordination of Web services orchestration. Development of distributed systems on the Internet in web services community is to define coordination mechanisms to execute collaborative tasks involving multiple organizations. In this paper to implement online driving license renewal, this work presents a dependable infrastructure for cooperative web services coordination which is based on the tuple space coordination model as stored procedure, provide an infrastructure with decoupled communication, implement several security mechanisms that allows reliable coordination even in presence of malicious components and investigate the costs related to the use of this infrastructure and possible web service applications that can benefit from it.

INDEX TERM Web service, Tuple space model, distributed system, service-oriented computing, web service orchestration and web service choreography.

INTRODUCTION:

The introduction of web services to the World Wide Web has made it possible to build new applications that integrate information from different web services and web sources. Recently, several tools have been introduced to rapidly develop and deploy web services. The Web services architecture defines separate specifications for the composition and the coordination of Web services. Development of distributed systems on the Internet in web services community is to define coordination mechanisms to execute collaborative tasks involving multiple organizations.

The infrastructure proposed in this paper, called Web Service Dependable Space(WSDS), The Web Services technology, an instantiation of the service oriented computing paradigm, is becoming a defacto standard for the development of distributed systems on the Internet. The attractiveness of web services is its interoperability and simplicity, based on technologies widely used in the web, like HTTP and XML. Its attributes have been the motivation for many industrial and academic efforts for developing concepts and models for distributed applications based on the service-oriented paradigm. Also, many efforts have appeared to define adequate forms of services composition in order to execute complex tasks. These compositions aim the cooperation in the execution of tasks

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 1, January 2018

involving multiple organizations. Specifications like WS-Orchestration and WS-Choreography address exactly this problem, specifying mechanisms for definition and execution of tasks that involve several web services. The main contributions of this work are: the design and implementation of the WSDS infrastructure, which is the first dependable web services data-centered coordination service, to evaluation of the cost to access this type of infrastructure through an analysis of the operations latency and its causes, to analysis of some real applications where this type of infrastructure can be used, as well as its relation with the main standards for cooperation among web services.

Tuple spaces are a popular data-driven coordination model. In the tuple space model, processes interact through a shared memory abstraction, the tuple space, where generic data structures, the tuples, are stored and retried. The basic operations supported by the tuple space are insertion, reading and removal oftuples. The main attractiveness of this model is its support for communication that is decoupled both in time (communicating processes do not need to be active at the same time) and space (communicating processes do not need to know each others locations or addresses). Moreover, this model provides some synchronization power through special operations provided by the tuple space that allow applications to solve important problems like leader election, consensus and mutual exclusion.

Web Service and Ws-Specification:

The Web Services technology, an instantiation of the service oriented computing paradigm, is becoming a defacto standard for the development of distributed systems on the Internet. The attractiveness of web services is its interoperability and simplicity, based on technologies widely used in the web, like HTTP and XML.

RELATED WORKS

Supporting Web-based collaboration between virtual enterprise partners:

Many of the processes in a dynamic virtual enterprise (VE) are collaborative in nature and require consideration of cross-organizational issues. We present a way of supporting Web-based collaboration between organizations within a dynamic real-time virtual enterprise. In our approach we focus on a service-oriented view of collaboration that is compatible with emerging Web service industry standards based on XML, SOAP, and WSDL. We also introduce a collaboration system, which uses multiagent system (MAS) technology and specialized tuplespaces to realize many of the desired categories of automated support.

Context-based Semantic Mediation in Web Service Communities

Communities gather Web services that provide a common functionality, acting as an intermediate layer between end users and Web services. On the one hand, they provide a single endpoint that handles user requests and transparently selects and invokes Web services, thus abstracting the

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 1, January 2018

selection task and leveraging the provided quality of service level. On the other hand, they maximize the visibility and use rate of Web services. However, data exchanges that take place between Web services and the community endpoint raise several issues, in particular due to semantic heterogeneities of data. Specific mediation mechanisms are required to adapt data operated by Web services to those of the community. Hence, mediation facilitates.Interoperability and reduces the level of difficulty for Web services to join and interact with communities.

On self-coordinating Web services using similarity and neural networks

It support the self-coordination of Web services. Usually coordination is a designer-driven activity where, for instance, the designer clearly indicates the actions that Web services have to perform during conflict resolution. Web services are embodied with the mechanisms, which allow them to coordinate themselves during run-time. These mechanisms are encoded using control tuples that Web services post on tuple spaces. Web services consult a tuple space and consume the control tuples, which are relevant to their coordination work.

Secure Biddings

An application to manage biddings can be easily implemented over WSDS. Using WSDS, a consumer interested in some service (or product) inserts a tuple in the space, describing the characteristics of the service to be contracted (or product to be bought). Then, service providers) also insert a tuple, describing its proposal for service execution.

WS SPECIFICATION:

There are many specifications developed (or in development) that aim the integration of web services. This section presents the relationship between WSDS and some of these specifications.

WS-Coordination: This specification defines a generic model that can be used by several web services in order to coordinate the execution of some distributed task. The main component of this model is the coordination context, an abstraction responsible for managing information about the coordination of participants. At present, this model already was used to implement distributed transactions involving web services. The tuple space abstraction, provided by WSDS, can be implemented as an instance of WS Coordination, where the coordination context is the logical tuple space and services registered in the context can be understood as the clients able to interact with the space.

WS-Orchestration: Orchestration of web services consisting the coordination of these services by using an orchestration engine. The basic idea is to define the invocations flow (i.e., the sequence that services will be invoked), using a coordination language, as WSBPEL (Web Services Business Process Execution Language). The travel agency is an example of task that can be implemented using orchestration. Currently, there is no standard based on shared data repositories to be used by cooperating web services being

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 1, January 2018

orchestrated. In this way, all information required by services must be maintained by the orchestration engine and sent to each service as a parameter of operations. If the amount of shared data is large, this approach can be very inefficient. Thus, a dependable data repository is useful and its integration with WSBPEL language is easy, i.e., all orchestrated services access shared data on WSDS.

WS-Choreography: Choreography is a composition of web services that allows: a formal definition of the interactions among the web services; the verification of the correctness (e.g., if it is deadlock-free); and code generation for the interactions. The choreography differs from the orchestration in at least two aspects: it is distributed (while the orchestration has a coordinator – the orchestration engine); and the choreography languages (as the WSCL – Web Services Choreography Language) are used to specify the interactions required during the tasks execution, and not to execute the coordination (differing from the orchestration).

WS-Policy Framework: The *Web Services Policy Framework (WS-Policy)* defines a general-purpose model and syntax for expressing functional or non-functional properties of a Web service in a declarative manner. A policy is an XML-expression that logically combines one or more assertions which specify concrete or abstract service characteristics such as a required security authentication scheme or a desired quality of service. Policies can flexibly be attached to various Web services definitions, including WSDL type definitions, as described in the *Web Services Policy Attachment* specification.

WEB SERVICE ARCHITECTURE

The Web services architecture defines a set of specifications that provide an open XML-based platform for the description, discovery, and interoperability of distributed, heterogeneous applications as services. Included are specifications for business process management and various quality-of-service protocols supporting, for example, transactions, reliable messaging, and security.Web services are application components. Web services communicate using open protocols.Web services are self-contained and self-describing. Web services can be discovered using UDDI. Web services can be used by other applications. XML is the basis for Web services.

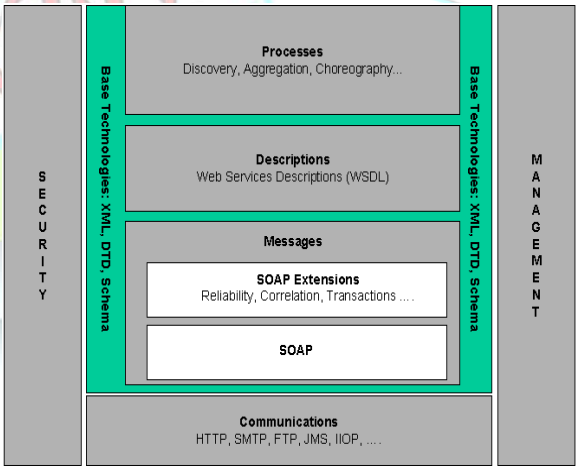

Figure1: Web service architecture

The architecture does not attempt to specify how Web services are implemented,

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 1, January 2018

and imposes no restriction on how Web services might be combined. The WSA describes both the minimal characteristics that are common to all Web services, and a number of characteristics that are needed by many, but not all, Web services. The Web services architecture is interoperability architecture: it identifies those global elements of the global Web services network that are required in order to ensure interoperability between Web services. [16] discussed about a method, Wireless sensor networks utilize large numbers of wireless sensor nodes to collect information from their sensing terrain. Wireless sensor nodes are battery-powered devices. Energy saving is always crucial to the lifetime of a wireless sensor network. Recently, many algorithms are proposed to tackle the energy saving problem in wireless sensor networks. There are strong needs to develop wireless sensor networks algorithms with optimization priorities biased to aspects besides energy saving. In this project, a delay-aware data collection network structure for wireless sensor networks is proposed based on Multi hop Cluster Network. The objective of the proposed network structure is to determine delays in the data collection processes. The path with minimized delay through which the data can be transmitted from source to destination is also determined. AODV protocol is used to route the data packets from the source to destination.

WEB SERVICE COORDINATION

The main component introduced in this architecture is the gateway web service that works as a bridge among clients (of the coordination service) and the dependable tuple space. The gateway receives SOAP messages sent by clients and transforms them in DEPSPACE requests. Thus, before accessing the tuple space, a client must find one or more gateway address on a UDDI service, Thereafter, the client sends its requests to one of the gateways, which, in turn, forwards it to the DEPSPACE using a total order multicast protocol. DEPSPACE servers process the request and send the response to the gateway, which collects responses from different servers and forward this set of responses to the client. The client determines the response of its request by verifying which response was replied by at least fr+1 servers (at least one correct server).

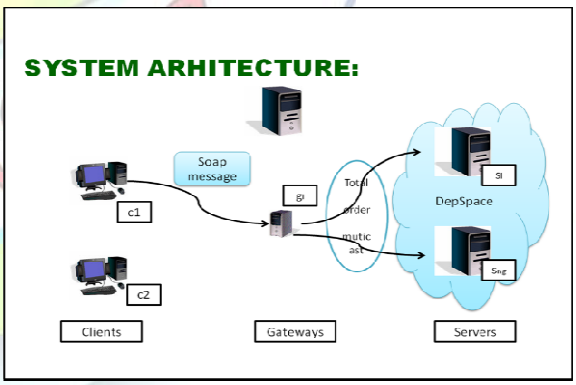

Figure2: System Architecture.

The gateway does not execute any processing on the content of requests or responses, only the transformations between the two "worlds" SOAP and DEPSPACE. Moreover, this service is stateless, i.e., it does not have state and thus there is no need of any synchronization among the gateways of the system.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 1, January 2018

SERVICE ORIENTED ARCHITECTURE

A Service Oriented Architecture (SOA) is a form of distributed systems architecture that is typically characterized by the following properties:

Logical view: The service is an abstracted, *logical* view of actual programs, databases, business processes, etc., defined in terms of what it *does*, typically carrying out a business-level operation.

Message orientation: The service is formally defined in terms of the messages exchanged between provider agents and requester agents, and not the properties of the agents themselves. The internal structure of an agent, including features such as its implementation language, process structure and even database structure, are deliberately abstracted away in the SOA: using the SOA discipline one does not and should not need to know how an agent implementing a service is constructed. A key benefit of this concerns so-called legacy systems. By avoiding any knowledge of the internal structure of an agent, one can incorporate any software component or application that can be "wrapped" in message handling code that allows it to adhere to the formal service definition.

Description orientation: A service is described by machine-process Meta data. The description supports the public nature of the SOA: only those details that are exposed to the public and important for the use of the service should be included in the description. The semantics of a service should be documented, either directly or indirectly, by its description.

Granularity: Services tend to use a small number of operations with relatively large and complex messages.

Network orientation: Services tend to be oriented toward use over a network, though this is not an absolute requirement.

Platform neutral: Messages are sent in a platform-neutral, standardized format delivered through the interfaces. XML is the most obvious format that meets this constraint.

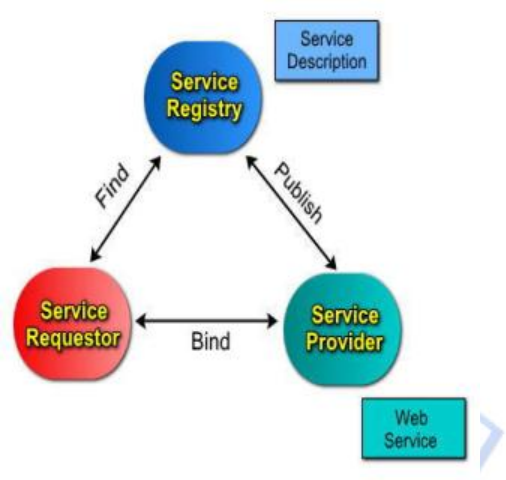

Figure 3 Service Oriented Interaction Cycle

Figure illustrates a simple service interaction cycle, which begins with a service advertising itself through a well-known registry service (1). A potential client, which may or may not be another service, queries the registry (2) to search for a service that meets its needs. The registry returns a (possibly empty) list of suitable services, and the client selects one and passes a

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 1, January 2018

request message to it, using any mutually recognized protocol (3). In this example, the service responds (4) either with the result of the requested operation or with a fault message.

TUPLE SPACE COORDINATION MODEL:

A tuplespace is a shared associative memory used by processes to communicate. Processes can post and read tuples using three basic operations: out to insert a tuple into the tuplespace, in to remove a tuple from the tuplespace and rd to check if a tuple is present in the tuplespace (without removing it). Tuples are anonymous and are extracted from the tuplespace by means of pattern matching on the tuple contents. Tuplespace communication is decoupled both in space and time: processes do not have to know each other beforehand nor to be available at the same time since tuples can be inserted and extracted independently. These forms of decoupling make the model highly suitable for mobile networks. However, maintaining a globally shared tuplespace is not feasible in a mobile setting. Within the domain of tuplespaces targeting the mobile environment, we have explored three main software engineering concerns:

Coordination in face of intermittent connectivity: How to allow applications to react to network disconnections.

Scopemechanism for tuplepropagation: How to delimit scope of a tuple preventing to be propagated to unwanted locations.

Abstractions for context-awareness: how to describe tuple perception without relying on network connectivity.

Some adaptations of tuplespaces targeting the mobile environment, such as LIME, have extended this model with the notion of *federated tuplespaces*. In this model every node in the network keeps a local tuplespace. The tuples in the local tuplespace of all devices in range are conceptually merged into a federated tuplespace. Nodes can post and read tuples from this federated tuplespace by means of the typical tuplespace operations. When devices move out of range their tuples are no longer shared and removed from the federated tuplespace. [9] discussed about a method, This scheme investigates a traffic-light-based intelligent routing strategy for the satellite network, which can adjust the pre-calculated route according to the real-time congestion status of the satellite constellation. In a satellite, a traffic light is deployed at each direction to indicate the congestion situation, and is set to a relevant color, by considering both the queue occupancy rate at a direction and the total queue occupancy rate of the next hop. The existing scheme uses TLR based routing mechanism based on two concepts are DVTR Dynamic Virtual Topology Routing (DVTR) and Virtual Node (VN). In DVTR, the system period is divided into a series of time intervals. On-off operations of ISLs are supposed to be performed only at the beginning of each interval and the whole topology keeps unchanged during each interval. But it has delay due to waiting stage at buffer. So, this method introduces

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 1, January 2018

an effective multi-hop scheduling routing scheme that considers the mobility of nodes which are clustered in one group is confined within a specified area, and multiple groups move uniformly across the network.

Other tuplespace systems, such as , have adopted a *replication*model where tuples are replicated amongst collocated devices in order to increase data availability in the face of intermittent connectivity. Rather than merging local tuplespaces when devices discover, tuples are equipped with a propagation rule that determines how a tuple hops from one tuplespace to another one. These propagation rules can be exploited to achieve context-awareness based not only on connectivity but also on semantic information. On the hand, replicated-bases system do not guarantee atomicity for remove operations which is an essential feature to support synchronization between applications.

IMPLEMENTATION

This project entitled as "ONLINE DRIVING LICENSE RENEWAL" has been developed using ASP.NET as front end and SQL SERVER as back end. The major scope of the project is "renewal driving license through online", The Driving License is the software to renew Driving License of the each and every people. Our software gives the how to renew the driving license. Initially, this software has verify that the license id checking. After the Id Checking, the End user can enter their own driving license registration number into that system. The Online license renewal system will again verify their information and display all of the information related to the respective candidate. Because, already available candidate only can enter into that system. The authentication module provides to give the security to each and every and module. Because end user can create the new account and they can delete the account and modify their own account using this module. The RTO License database module handled by the administrator only. The Admin can login into this software and they can edit, insert delete the RTO License Database. And they provide the verification process for the each and every security for each and every data. The RTO License Id is checked by the end user after the login verification. Because they want to proceed with the next process they should be an authorized candidate. The Renewal availability helps the end user to check if the candidate is having already their own account in RTO office therefore, this process is going to be implemented in this module. The Renewal displays the information while they are entering into the RTO ID field in the respective Place. Then software will display all of the information. Then, Candidate can apply for renewal and then it will provide the renewal providing date. And the candidate proceeds with the next module. The renewal receipt receivable module will provide the bill for the respective candidate. The session concept has been implemented here. This will generate the print out of the bill by end user.

Modules:

1. Authentication Module
2. RTO License Database By Admin

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 1, January 2018

3. RTO License Id Checking
4. Renewal Availability
5. Renewal Main Module
6. Renewal Receipt Receivable Module
7. Medical fitness
8. Banking
9. Police enquiry

Data Sharing among Services

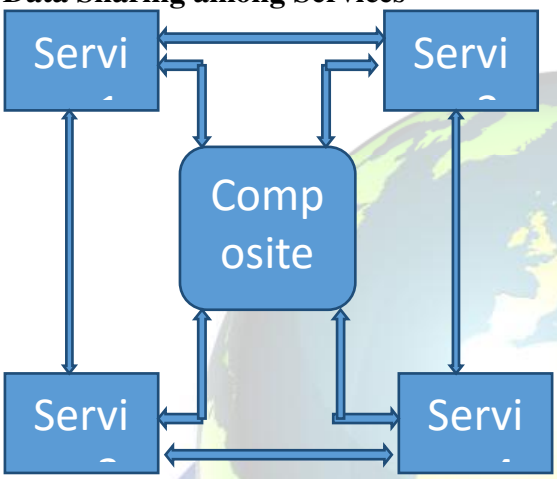

Figure 4:Data Sharing among Services

The applications that invoke several web services, in order to execute complex tasks, need to share information among these services. These applications may use a shared database (accessed by several services) or need the session information of each task to be included in all messages exchanged during the task execution.

CONCLUSION

This paper described WSDS, a dependable coordinationservice that can be used by web services to share dataand synchronize their actions. The proposed architectureis based on a dependable tuple space and stateless webservices (gateways) that provide access to it. An important feature of this system is that it is completely fault and intrusion-tolerant. This architecture integrates several dependability and security mechanisms in order to enforce dependability properties like integrity, confidentiality, and availability in a modular way.

FUTURE ENHANCEMENT

Grid concepts and technologies are transitioning from scientific collaborations to industry. We believe that the Open Grid Services Architecture will help accelerate that transition by recasting the Grid technologies that the Globus Toolkit provides in a uniform service-oriented architecture and integrating those technologies with emerging Web services standards. OGSA thus representsa natural evolution of both Grid technologies and Web services. By integrating support for transient, stateful service instances with existing Web services technologies, OGSA extends the power of the Web services framework significantly, while requiring only minor extensions to existing technologies. OGSA facilitates the realization of Grid concepts in practical settings by adopting an industry-standard interface definition language and enabling the use of Web services tooling. Consequently, grid computing is of course a multi user environment, and for this reason, secure authorization methods are crucial before enabling computer resources to be managed by remote end users.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 1, January 2018

References

[1] A. Alves et al. Web services business process execution language, version 2.0. OASIS Public Draft. Availiable at http://docs. oasis-open.org/wsbpel/2.0/, Nov. 2006.

[2] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. IEEE Transactions on Dependable and Secure Computing, Mar. 2004.

[3] U. Bellur and S. Bondre. xSpace: a tuple space for XML and its application in orchestration of web services. In Proceedings of the 21st ACM symposium on Applied computing – SAC'06, pages 766–772, 2006.

[4] A. N. Bessani, E. P. Alchieri, M. Correia, and J. S. Fraga. DepSpace: A Byzantine fault-tolerant coordination service. In Proceedings of the 3rd ACM SIGOPS/EuroSys European Systems Conference - EuroSys' 08, pages 163–176, Apr. 2008.

[5] A. N. Bessani, M. Correia, J. S. Fraga, and L. C. Lung. Sharing memory between Byzantine processes using policy-enforced tuple spaces. In IEEE Transactions on Parallel and Distributed Systems,
2008. to appear.

[6] M. Bichier and K.-J. Lin. Service-oriented computing. IEEE Computer, 39(3):99–101, Mar. 2006.

[7] D. Bright and G. Quirchmayr. Supporting web-based collaboration between virtual enterprise partners. In Proc of the 15th International Workshop on Database and Expert Systems Applications, 2004.

[8] D. Burdett and N. Kavantzas. The WS-Choreography model overview. W3C Draft. Available at http://www.w3.org/TR/ ws-chor-model/, Mar. 2004.

[9] Christo Ananth , P.Ebenezer Benjamin, S.Abishek, "Traffic Light Based Intelligent Routing Strategy for Satellite Network", International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST), Volume 1,Special Issue 2 - November 2015, pp.24-27

[10] G. Cabri, L. Leonardi, and F. Zambonelli. Mobile agents coordination models for Internet applications. IEEE Computer, 33(2):82–89, Feb. 2000.

[11] N. Carriero and D. Gelernter. How to write parallel programs: a guide to the perplexed. ACM Computing Surveys, 21(3):323–357, Sept. 1989.

[12] M. Castro and B. Liskov. Practical Byzantine fault-tolerance and proactive recovery. ACM Transactions Computer Systems, 20(4):398–461, Nov. 2002.

[13] T. Dierks and C. Allen. The TLS Protocol Version 1.0 (RFC 2246). IETF Request For Comments, Jan. 1999.

[14] C. Dwork, N. A. Lynch, and L. Stockmeyer. Consensus in the presence of partial synchrony. Journal of the ACM, 35(2):288–322, Apr. 1988.

[15] F. Favarim, J. S. Fraga, L. C. Lung, and M. Correia. GridTS: A new approach for fault tolerant scheduling in grid computing. In Proceedings of the 6th IEEE International Symposium on Network
Computing and Applications - NCA'07, pages 187–194, July 2007.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 1, January 2018

[16] Christo Ananth, T.Rashmi Anns, R.K.Shunmuga Priya, K.Mala, "Delay-Aware Data Collection Network Structure For WSN", International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST), Volume 1,Special Issue 2 - November 2015, pp.17-21

[17] M. Herlihy and J. M. Wing. Linearizability: A correctness conditionfor concurrent objects. ACM Transactions on Programing Languagesand Systems, 12(3):463–492, July 1990.

[18] L. Lamport, R. Shostak, and M. Pease. The Byzantine generals problem.ACM Transactions on Programing Languages and Systems,4(3):382–401, July 1982.

[19] R. Lucchi and G. Zavattaro. WSSecSpaces: a secure data-driven coordination service for web services applications. In Proceedings of the 19th ACM Symposium on Applied Computing – SAC'04, pages 487–491, Mar. 2004.

[20] Z. Maamar, D. Benslimane, C. Ghedira, Q. H. Mahmoud, and H. Yahyaoui. Tuple spaces for self-coordination of web services.In Proceedings of the 20th ACM Symposium on Applied computing – SAC'05, pages 1656–1660, 2005.

[21] N. H. Minsky and V. Ungureanu. Law-governed interaction: a coordinationand control mechanism for heterogeneous distributed systems.ACM Transactions on Software Engineering and Methodology,9(3):273–305, July 2000.

[22] R. R. Obelheiro, A. N. Bessani, L. C. Lung, and M. Correia. Howpractical are intrusion-tolerant distributed systems? DI-FCUL TR06–15, Dep. of Informatics, University of Lisbon, September 2006.

[23] Object Management Group. The common object request broker architecture:Core specification v3.0. OMG Standart formal/02-12-06,Dec. 2002.

[24] G. Papadopolous and F. Arbab. Coordination models and languages.In The Engineering of Large Systems, volume 46 of Advances inComputers. Academic Press, Aug. 1998.

[25] C. Peltz. Web services orchestration and choreography. IEEE Computer,36(10):46–52, Oct. 2003.

[26] F. B. Schneider. Implementing fault-tolerant service using the statemachine aproach: A tutorial. ACM Computing Surveys, 22(4):299–319, Dec. 1990.

[27] E. J. Segall. Resilient distributed objects: Basic results and applications to shared spaces. In Proceedings of the 7th IEEE Symposium on Parallel and Distributed Processing – PDP'95, pages 320–327, Oct. 1995.

[28] P. Verissimo, N. F. Neves, and M. P. Correia. Intrusion-tolerant architectures: Concepts and design. In Architecting Dependable Systems, volume 2677 of LNCS. Springer-Verlag, 2003.

[29] V. Yegneswaran, P. Barford, and S. Jha. Global intrusion detection in the DOMINO overlay system. In Proceedings of the 11th Network and Distributed Security Symposium – NDSS 2004, Feb. 2004.