International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

SOA Concepts for Distributed Computing Applications in Telecommunications

M. Mohammed Thaha[1], Jastini Mohd Jamil[2], Sumit Kumar[3], T. Jackulin[4], D. V. Lokeswar Reddy[5]

Submitted: 21/09/2023 **Revised**: 24/10/2023 **Accepted**: 12/11/2023

Abstract: Scalable systems allow for the cost-effective deployment of both big and small versions of telecommunications networks. The connections, hardware, software, and traffic factors will vary among the configurations. In the future, distributed software will be used to implement many systems. By picturing an end-to-end platform that links end-user clients to telecommunication core networks and the Internet, this article suggests a way to close this gap. By integrating application store application distribution concepts, telecommunication SDP architecture concepts, and SOA platform principles, all platform stakeholders' needs are identified and met (like Google Market and Apple AppStore). For service delivery platforms, state-of-the-art is now regarded as SOA. The intelligent network and object-oriented programming interfaces have given place to more modern Web-services-based platforms as these value-added service platforms have developed. They have implemented an open set of service components using the newest information technology. Further driving the need for telecom operators to create an open service market based on an open pool of service components and enabling services is the emergence of Web 2.0.

Keywords: *Intelligent network (IN), telecommunications, deployment, computing applications, Service-oriented architecture (SOA)*

1. Introduction

SOA is one approach that is becoming more and more popular in practically all software applications and systems. With the use of accessible, loosely coupled, and interoperable services, this approach seeks to facilitate communication between clients and servers. Under this architecture, published interfaces enable service users to locate and connect to available services. Service users are also known as clients and invokers. Both formal and informal descriptions of the services that are given are made possible through these interfaces [1]. Formal information could include terminology related to quality of service (QoS), parameters for input and output as well as descriptions of the available activities. On the other hand, a casual description can provide SLAs, contact details, and details about the service creators. What makes the SOA paradigm special is that service users are unaware of the underlying platforms, programming dialects, and other technical details involved in creating services.

Put another way, invokers are completely shielded from all technical and intricate information. As a result, service invokers just need to locate and utilize services that satisfy their technological and business requirements.

Owing to the benefits of the SOA model, it has been believed that the field of grid computing may effectively utilize it. This would make it possible for clients to quickly and readily access grid resources regardless of the technology used to build them. By providing a high-level design for SOA-based grid computing, we aim to apply SOA principles to grid computing in this article. Through the use of flexible, interoperable, and user-friendly services, this architecture is intended to facilitate the sharing of core assets and skills. Grid middleware, or the software parts and protocols that provide needed regulated access to resources, is what makes grids viable. Up to this point, most grids have been built with either private equipment or ad hoc public components. Though unique commercial and public solutions have been effective in their specific financial markets, they all have benefits like drawbacks, and their potential as the foundation for future grids—which must be highly scalable and interoperable to meet the demands of global corporations—is limited. This is because Web

[1]*Assistant Professor (Sr.Grade), B.S.Abdur Rahman Crescent Institute of Science and Technology, GST Road, Vandalur, Chennai - 600 048, Tamilnadu, INDIA*
Email: mohammedthaha@crescent.education
[2]*School of Quantitative Sciences, Universiti Utara Malaysia, 06010UUM Sintok, Kedah, Malaysia*
Email: jastini@uum.edu.my
[3]*Associate Professor& Head, Dept of Artificial Intelligence & Machine Learning, COER University, Roorkee.*
Email: mail2dr.sumit@gmail.com
[4]*Associate Professor, Panimalar Engineering College, Chennai, India.*
Email: karthijackulin@gmail.com
Orcid: 0000-0003-4015-7718
[5]*Assistant Professor, Humanities and Social Sciences Department, JNTU College Of Engineering, Pulivendula-516390, Kadapa (D), Andhra Pradesh, India*

services technology and standards have advanced so quickly that it is now conceivable to transition from the "stovepipe" design of existing layouts to the standardised, service-focused, enterprise-class grid of the future. In Figure 1.1, a basic grid that uses services for even virtualization of resources and other grid tasks is used to demonstrate how a service-based grid may be constructed.

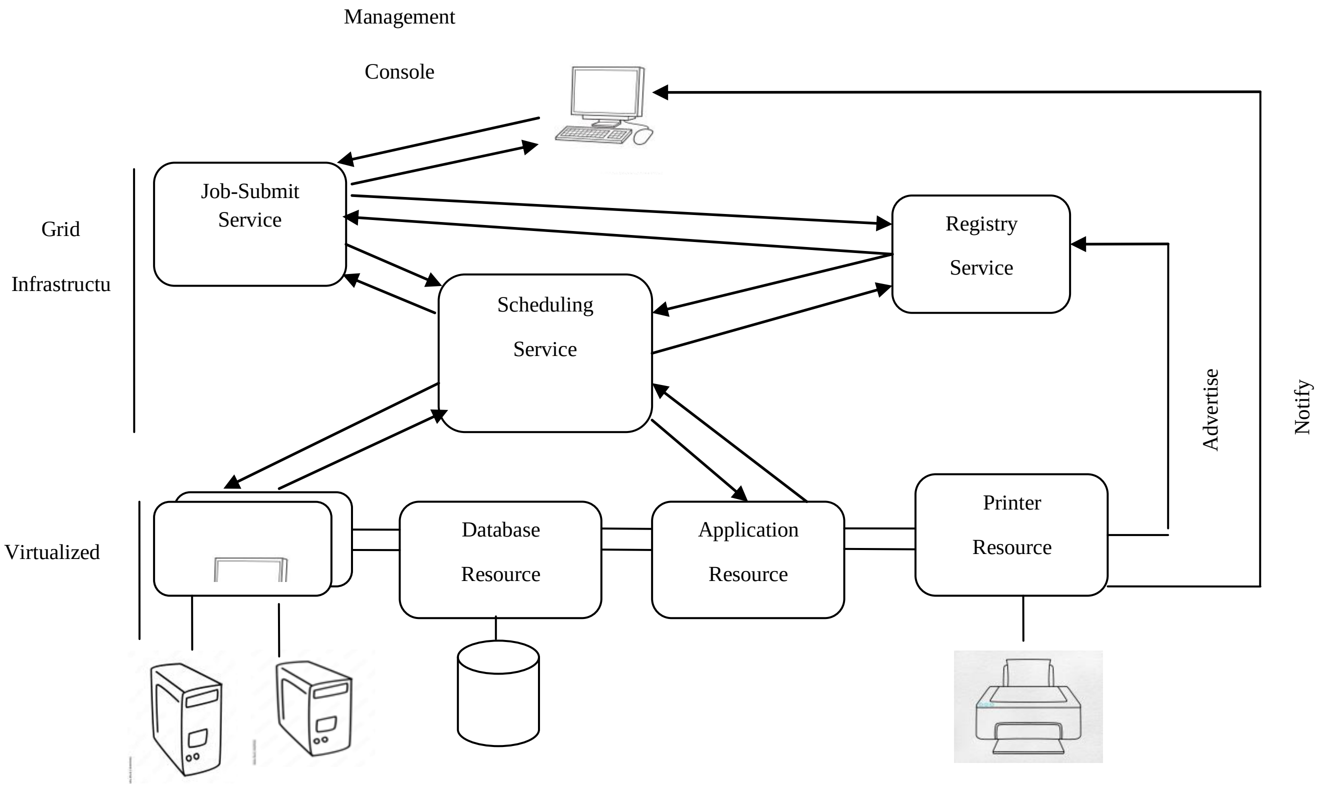

Fig. 1.1. A basic grid of SOA

For submitting jobs and managing the grid's resources, the image depicts a single terminal. When a resource becomes unavailable or overloaded, for example, the management software on the console notifies the representing service of that resource. It also interacts with the registry help to find out which grid resources are available. At last, it notifies the representing service of each resource to request performance data regularly.

The job-submit service takes a job request from the user, finds the grid's scheduling service, and forwards the request. Identifying the service that represents the requested application, the scheduler gets in touch with it to get knowledge of the resources required to complete the assignment. After checking the database for all relevant resources, it makes individual contact with each one utilizing the services that are connected to it to find out if it is accessible. If the event that there are enough resources available, the scheduler chooses the most suitable grouping and gives the application service a request to begin the job's execution [2]; if not, it queues the task and executes it when the necessary resources are available. The application service tells the scheduler, who then notifies the task submit service, of the result of the work. The user gets a notification from the job-submit service.

A genuine enterprise-class grid will involve far more complex operations than is demonstrated in this example, with the ultimate objective being a high level of automation and optimization in the usage of resources throughout the grid's domain. Please note that this example was simplified for clarity.

The order of the essay's subsequent sections is as follows. Section 2 presents the research on the relevant previous work. Section 3 describes the characteristics of the suggested system, such as its suggested design, implementation methodology, graph-based approach components, and data analysis. Section 4 explains the implementation environment and evaluates the effectiveness of the system. In Section 5, the resolution is provided.

2. Literature Review

Ahmed, W., et al., [3] The dependability of software has long been a source of criticism for all parties involved, particularly for sellers and users of the system. Reliability has a bigger impact on software longevity than other aspects like usability, adaptability, and maintainability because it can throw a programmer out of order while it's operating. Since the term "reliability" is so wide, different software applications operating in dispersed environments may have different meanings of it.

Tsai, W. T., et al [4] The second distinctive characteristic of SOA-based application architecture is its capacity to incorporate lifecycle management into the operating infrastructure to facilitate dynamic software building. This allows for the integration of the SOA operating and application development infrastructures into a single, cohesive SOA infrastructure. The second distinctive characteristic of SOA-based application architecture is the capacity to incorporate lifecycle management into the operating infrastructure to facilitate dynamic software building. This allows for the integration of the SOA operating and application development infrastructures into a single, cohesive SOA infrastructure.

Jorstad, et al., [5] The crucial aspect to remember is that a service ought to be independent. Because of this, it might always provide the same functionality independent of other services. Distributed systems have historically been prone to tight coupling, or intimate linking. The distributed system would not be able to offer its complete service if one of its components failed. Closed would also be the case with previously distributed systems. Only the system developers were aware of their internal, non-exposed interfaces.

Moulick, et al [6] The key component of SOA is the service. A service is described as "a mechanism to enable access to one or more abilities, where the access is exercised consistent with the limitations and policies as specified by the service's description and is provided via a prescribed interface." The following qualities can be present in a service: Similar to a software component, a service offers a contract that is specified by one or more interfaces. This permits modifications to the service implementation without requiring the client to be rebuilt, provided that the contract remains unchanged. The service requestor is not concerned with the implementation details of the service (such as operating systems and programming languages).

Griffin, D., et al [7] Using research studies and real-world implementations as examples, we outline the main advantages of implementing SOA in communications. We go into detail about event-driven architecture (EDA) and how a true SOA implementation requires both EDA and SOA. We outline the technology that makes Web services in telecoms possible and explain SOA middleware, specifically mentioning the enterprise service bus (ESB). Lastly, we summarise the findings and discuss in depth the future orientations of Web services in the telecommunications sector.

Duan, Q., et al [8] A service is a self-contained (self-maintaining state) and platform-independent module in software architecture; that is, the interface of the service is independent of the platform on which it is implemented. Standard interfaces and messaging protocols allow for the description, publication, location, orchestration, and programming of services. Because all services in SOA are independent of one another and external services view service activities as opaque, it is ensured that external components are unaware of and unconcerned with how services carry out their duties. The service interface conceals the technologies that enable the intended functionality of the service.

Papazoglou, et al., [9] Web services are one way that SOC (including a few SOA components) is applied on the Web. Web services are a subset of services that have unique URIs and use open Internet standards for both their transport and service description. Web services usually communicate with one another via SOAP calls that contain XML data. Web services interface descriptions are expressed using the Web Services Definition Language (WSDL). Web service descriptions are contained in directory services that adhere to the protocol defined by the Universal Description, Discovery, and Integration (UDDI) standard. Web service clients can find and learn about candidate services with the help of UDDI.

Kirkham et al., [10] Traditionally, achieving agility has mainly affected automation systems at the level of manufacturing devices, and SOA is best used at this level using a modular method. However, there is a greater push in SOA systems to enable system agility at higher application levels; to do this, it is crucial to seamlessly integrate the computational solutions at different levels. For example, these more widely used IT-focused solutions ought to help the company identify and anticipate future production trends. Establishing a system that facilitates planning entails connecting data from the production floor to advanced management information kept in computing systems connected to programs such as Enterprise Resource Planning.

3. Methods and Materials

3.1 SOA Architecture

SOA is a loosely linked, coarse-grained system-building approach. Platform compatibility, business-orientedness [11], transparency, loose coupling, and service security characterize its architecture. The most fundamental component of the SOA architecture is the service layer, which calls and applies services to control how the system and proxy components interact. A service is described as follows: the experience provider gives the required results to the service customers after completing a certain set of responsibilities. These services come together in concrete form through deployment and combination, and these service entities then go on to use and supply services to users through various service contracts, publications, enrolment, and delivery. Since each offering functions independently and there is no communication between them, one service's operation has no bearing on another's. At the same time, certain services may remain available to users even after they are controlled. This feature considerably improves the system's flexibility and maintainability by allowing the architecture to continually and dynamically extend its present-day services. Application systems benefit from the SOA structure's role as a kind of middleman, guiding them in understanding when and why business logic is applied. This knowledge helps administrators and analysts improve business processes in particular manners.

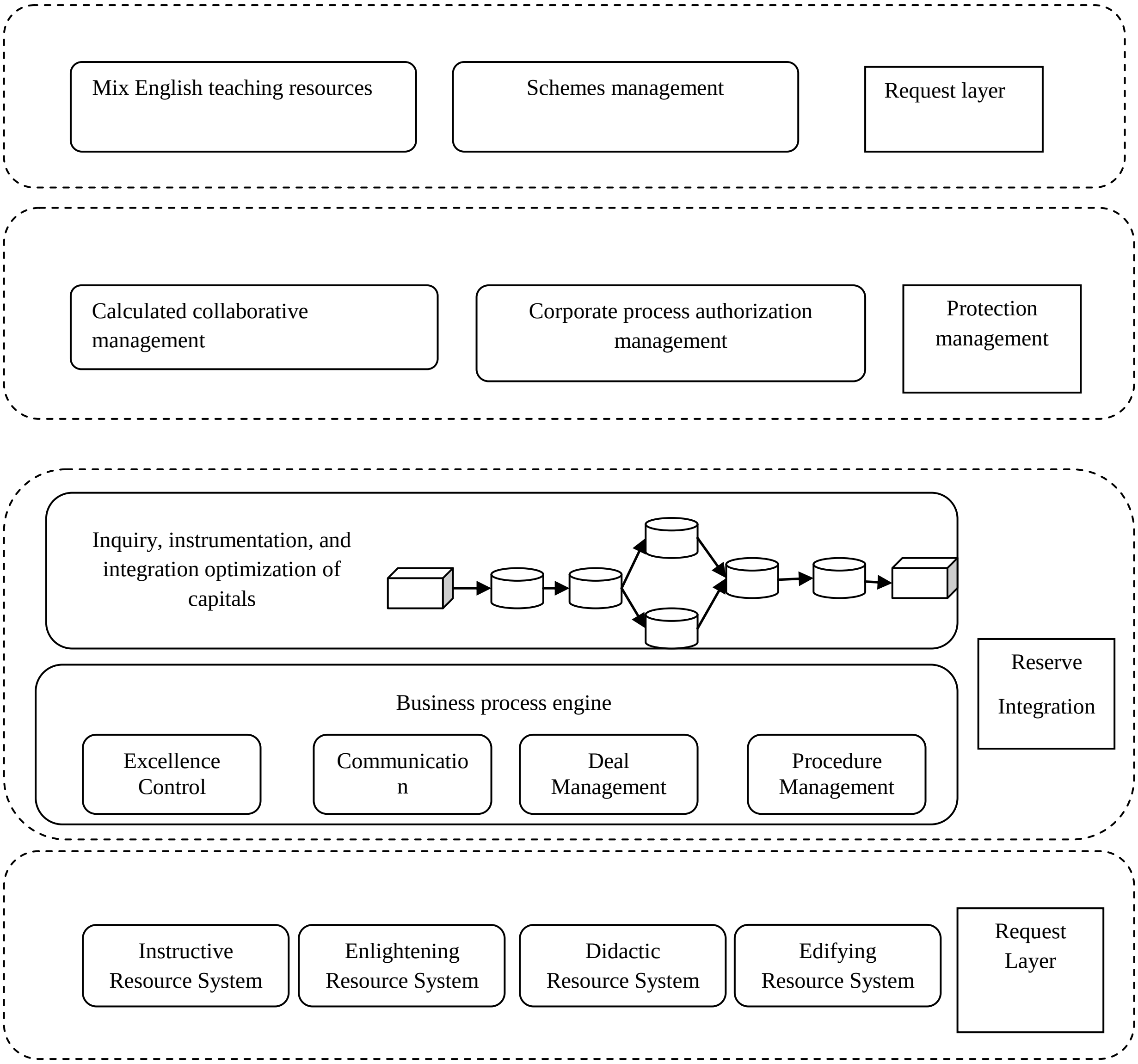

Fig. 3.1. SOA-based framework models of English teaching resources integration and optimization

3.2 Scheme of Teaching Resources Integration System Based on SOA

The cost of building the resource pool can be substantially reduced by the reusability of services. Reuse helps protect users' initial investments in information [12]. To guarantee the execution of service reuse, the service must be carried out inside the confines of a certain process during the construction phase, or the reasoning can be split over multiple services.

$$e = \sum_{I \in K} e(1) I. \quad (1)$$

Between the full user interest vector C and the document feature vectors s of the new instructional resource, the cosine correlation is determined. The stronger the cosine value, the greater the relationship Sim (V, S), and vice versa. Computed in this way:

$$\mathrm{sim}(V, S) = \sum_{l=1}^{m} \frac{(v_l \times s_l)}{\left(\sqrt{\sum_{l=1}^{m} v_l^2} \times \sqrt{\sum_{l=1}^{m} s_l^2}\right)}. \quad (2)$$

It is a given that every educational material has a special identifier. The learning resource collection has been designated as KO.

$$ko = \{ko_i | \leq i \leq n\}. \quad (3)$$

The node community j (denoted as C_i) is the set of nodes that conform to the community rule s_j which is

$$c_j = \{p_i | p_i \in p, p\_Belong\_to\_c(p_i s_j) = TRUE\}. \quad (4)$$

The object's community classification and description of its domain or category also satisfy the following criteria: meets

$$\nabla ko_i \epsilon LO, \exists s_i \; ko\_Belong\_to\_c(ko_j s_i) = TRUE \quad (5)$$

A directed weighted graph G (U, F) can be used to illustrate the Text Rank model. In (u_j) is the collection of points referring to the point u_j, and ω_{ij} is the edge weight between the two points u_j& u_i in the graph. The collection of points that point u_j points to is called Out (u_j). The following defines the Text Score value of point u_j:

$$WR(u_i) = (1-d) + d * \sum_{v_j \epsilon Im(u_j)} \frac{\omega_{ij}}{\sum_{u_{k\epsilon Out(u_i)}} \omega_{ik}}, \quad (6)$$

The modeling impact of the model is evaluated utilizing the root average square error index and the model judgment rate. Formula (3) gives the formula for determining the model accuracy index.

$$PNR = \frac{\sum_{t=1}^{T} \sum_{i=1}^{N} I^{(t)} \left(\overline{\alpha_k = \widehat{\alpha_k}} \right)}{T * N} =. \quad (7)$$

$$SNRF = \sqrt{\frac{\sum_{t=1}^{T} \sum_{k=1}^{2^l} \sum_{i=1}^{M} \left[P^{(t)} (\overline{s_i = s} | \alpha_k) \right]^2}{2^l * T * N}}. \quad (8)$$

There is an important moment for the telecom sector. To handle the convergence of cable networks, the Internet, and fixed and mobile telecommunications, operators have to set up global IP-based networks [13]. Platforms for providing services have to enable the effective development, planning, implementation, provisioning, and administration of unified services across a range of business models and access networks.

One of the primary benefits of telecommunication platforms for a long time has been the ability to create fresh applications by leveraging an extendable set of present-day service components. In a similar vein service-oriented design concepts have been used by the telecom sector for a long time, albeit under different names.

3.3 Telecommunications Services

The telecom sector has met people's desire for communication over long distances for over a century. But there have been major changes during the past thirty years. Mobile, data, and services were added to the industry's first offering of value-added services which included costs, routing, and interaction capabilities.

3.4 Service-driven Tradition

For any fresh service, an architecture must be developed that describes how different functional parts interact with one another through a protocol and sets up the control and content exchange techniques needed for offering the service to customers. The telecommunications environment has always been, in this regard, service-driven or -oriented.

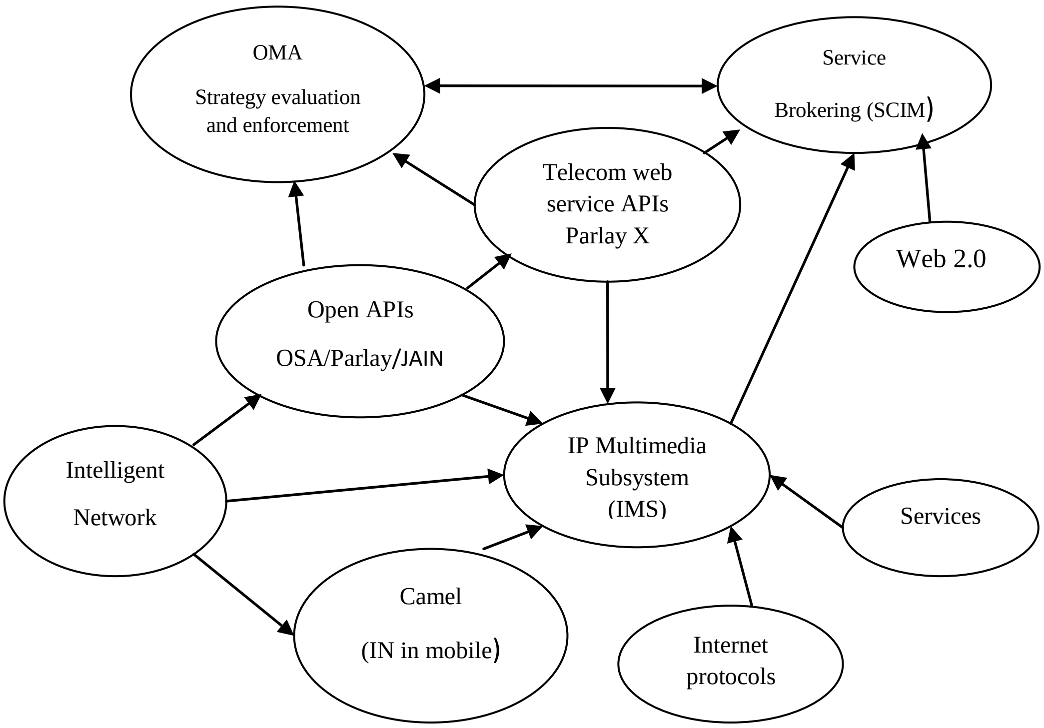

Fig. 3.2. Platforms of SOA

For platforms that supply products and services, state-of-the-art has become known as SOA. Figure 3.2 illustrates how the platforms used for services with value have changed over time [14], shifting from the Intelligent Net

(IN) and OOPS interfaces to a variety of modern Webservices-based platforms. Employing the newest information technologies, they created an open set of customer service components. Meanwhile, the arrival of Web 2.0 has increased pressure on telecom providers to establish an open service industry built on a wide range of supporting technologies and service components in Figure 3.3.

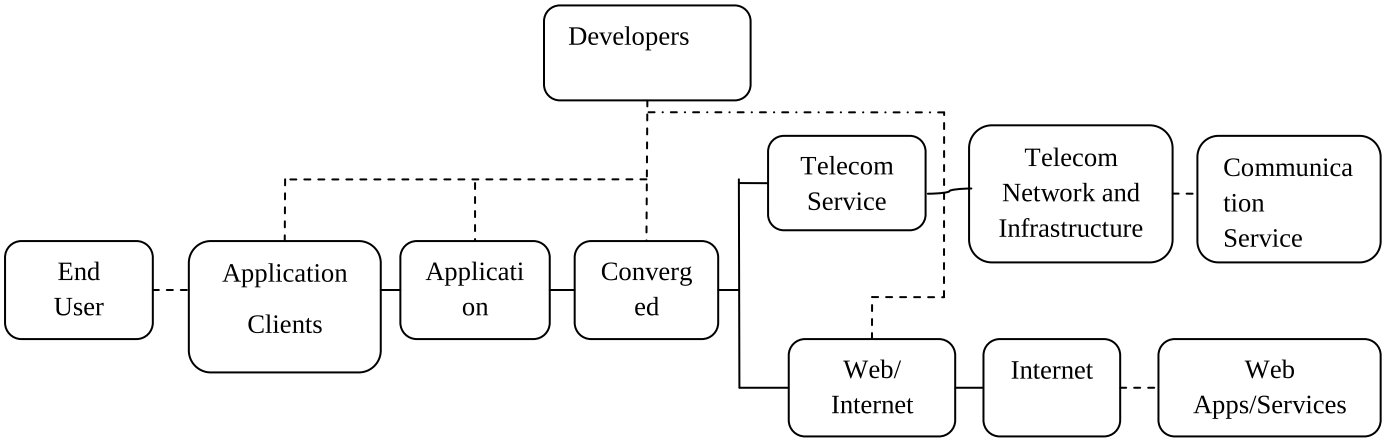

Fig. 3.3. Distributed Web-Telecom-Converged application and services

3.5 IP Multimedia Subsystems

MS facilitates the deployment and reuse of applications and services while decreasing the time needed for application development. IP is the network protocol that underlies IMS. As a signaling protocol, SIP is text-based, readily accessible by developers, and based on an easy request-response interaction paradigm. Developing and using standard interfaces across an IP-based infrastructure makes it possible for IMS apps to interact in the future with any Internet service or application.

3.6 Service Delivery Platforms

SDPs specify how Web/Internet and telecommunications network phrases should be utilized to give CSPs architecture for service delivery that supports the creation of offerings, service exposure, specified procedures, and session control. By giving third parties access to service infrastructures and facilitating the quick development of new service and application types, SDPs allow CSPs to alter their business models. Innovation either inside or outside of CSPs' network structure and services is helped by SDPs. Over the last five years, SDPs have grown more prevalent, and their origins may be seen in mobile content delivery and management systems. These included efforts for swapping out proprietary Intelligent Networks (IN) platforms for Java environments based on standards, and also more contemporary initiatives to use technologies such as W3C Web Services to open up service infrastructures.

Although there isn't a single definition for SDPs, the Service Shipping Framework (SDF), Service Shipping Environment, and OMA Service Supplier Environment (OSPE) specifications published by the TM Forum, Alcatel-Lucent, and OMA are important resources for knowledge regarding SDPs in the telecommunications industry. The Telecommunication SDP principles of this study effort are based on the OSPE, which has become the most extensively used in projects for research and development in the Telecommunications area.

3.7 Web-Telecom-Convergence Services and Applications Constructions

A SDP typically offers services in one area, such as the telephone area which includes, for example, the SDPs. The gap is filled via a Web-Telecom-Convergence architecture between the Web/Internet domain and the communications domain, moving beyond the abilities of an SDP. Diverse research initiatives put forth different Web-Telecom-convergence architectures at various abstraction levels and scopes. He offers a prototype solution and laboratory deployment jointly with a suggested Service Delivery Framework built on SOA concepts. Using SOA service definition and orchestration ideas, their technology enables the independent creation of telecom services inside the telecom domain. The convergence of web-based telecom services and applications occurs on the amount above the Telecommunication services layer and isn't covered by their platform.

In order to facilitate the integration of IMS-enabled telecommunication services into Web/Internet Mashups and other applications used by other organisations, he proposes a policy-based service broker. Broker policies act as either an illustration of user/service-specific capabilities and regulations or as a means of establishing service access between third-party applications and real-time telecommunication services.

A technique for fusing Web 2.0-enabledenabled gadgets with IMS-exposed telecom services was offered. They propose a novel abstraction layer that offers access to telecommunication services through interfaces and a gadget engine. In scenarios of usage where widgets are appropriate for end-user clients, it is restricted to how fast Web 2.0 widget applications can be created.

It suggests employing an Open Service Accessing Gateway (OSAG) to enable telecommunications companies to use W3C Web Services to offer integrated services. It also describes how applications may utilize these services via defined interfaces and how Web-Telecom-Converged Services can be developed on the OSAG through the combination of Telecommunication and W3C Web Services. This research project does not cover the architecture of applications or end-user client support.

Hybrid services are defined as those that are offered through multiple networks, particularly IP-based ones like mobile devices, Internet, and phones. They face an architecture that is used for offering this type of service. Their architecture enables it possible for developers to write platform elements to create new services that can be used in apps. In addition, they offer the concept of orchestration for telecommunication functions, exposing them to higher-level services that can be reused, such as call control, security, and address translation. Finally, they build a prototype service or application on top of a Java-based platform prototype.

It suggested a single platform for delivering telecom and IT services by combining W3C Web Services, underlying services, and IMS services. To accomplish the orchestration, they recommend employing a Service Bus in conjunction with their SIP-based Micro Services Orchestration solution. Finally, they show a multimedia service implementation prototype and speak about the way their service delivery platform may help with the quick creation and the introduction of fresh multimedia services merged with Web and telecom.

It suggested describing an SDP idea that is more expansive than the ones covered. They used Web/Internet and W3C Web Services technologies, like Parlay X, the execution of business processes. They sought to design an architecture that could live with both corporate and internet infrastructures, therefore they used a language (BPEL), corporate Services Bus (ESB), and Web Services Description Language (WSDL) to realise their SDP. They enhance the technologies accessible to meet the distinctive requirements of telecommunication services as determined by network service suppliers.

In addition to laying the foundation for this research project, the most pertinent connected publications are shown above. Their investigation revealed that no concept of an end-to-end SOA-based reference architecture based on ADP for Web-Telecom-Converged applications and services has been proposed that satisfies all the requirements stated in the part that follows.

- A structure is a formal representation of a system that outlines its objectives, operations, features that are visible from the outside, and interfaces. Together with the rules guiding the system's development, functioning, and evolution, it also explains the internal parts and how they interact.
- A service is a network-accessible computer component that provides functionality to a resource requester.
- A dependable distributed system planning approach that offers functionality as services and puts extra focus on the loose coupling between interacting services is known as service-oriented architecture.
- It's important to go over the definitions of the important terminology while discussing service-oriented architecture.

Therefore, in a technical sense, the word "SOA" refers to a system's design rather than its implementation. In this study, we avoid using the term "building a SOA," even though it is commonly used to refer to an implementation. On the other hand, when we use the word "service-oriented," we mean "service-oriented grid" or "service-oriented environment." In our view, SOA encourages the deployment of components as modular services that users can find and utilise. In general, services possess the following attributes:

- Higher-level services can be provided through the integration of services or they can be helpful on their own. This promotes the reuse of already-existing functionality, among other advantages.
- Provision of Services communicates by sending and receiving messages with their clients; their ability to receive signals and respond to them determines who they are.
- Services can interact with one another using workflows, where the order in which data is sent and received establishes how every service conducts its tasks. "Service choreography" is the term used to describe this concept.
- Services might be independent, or they may rely on the presence of a resource like a database or the accessibility of other services. In an ideal scenario, a service would have all the data it needs pre-loaded or it could be able to do a computation, such as finding the root of a cube of a given integer, without referencing any additional resources. On the other hand, to provide accurate results, a currency conversion service would require real-time availability of exchange-rate data.
- Services promote specifics including their interfaces, regulations, supported communications protocols, and capabilities. Clients have nothing to do with

implementation specifics, such as hosting platform and programming language, and these details are kept hidden.

4. Implementation and Experimental Results

The following features were examined through the simulation: the SOA system's efficiency, the likelihood of request loss due to overflowing network and server buffers, and the intensity of the flow of output of serviced requests. System performance has been taken into account at both high and exceptionally high lead levels [15]. Table 1 displays the input data used in the simulation.

Table 1. Input Data Used In the Simulation

The average of the service requests coming in from one access server's input flow intensity, λ_{IN}, req./sec.	1…10
The amount of time that passes in the input flow between subsequent requests	random, exponentially distributed
The average throughput of the telecom network linking the server-providers' servers, C, Mbps	2…10
Duration of message delivery over the network	random, exponentially distributed
Communication size, Kbytes	126
The average server productivity of the service providers, expressed as m requests per second.	60
The duration of a request's processing at the server	random, exponentially distributed
Buffer size, messages, and network and server	50…150

A modelled distributed network is composed of six servers that are owned by two different service providers and has the same architecture as earlier efforts each providing three sets of services that are comparable to the five servers that allow access to services. A distributed system with a model has the same design as previous works; it is constructed from up of six servers belonging to two distinct service providers, each offering three sets of services that are identical to each other, and five servers that allow access to facilities. All of the servers that were available were thought to have the same levels of service query intake flows, host productivity, and production capacity across the communications networks that connected them. Figure 4.1 presents the findings from the analysis of the output flows.

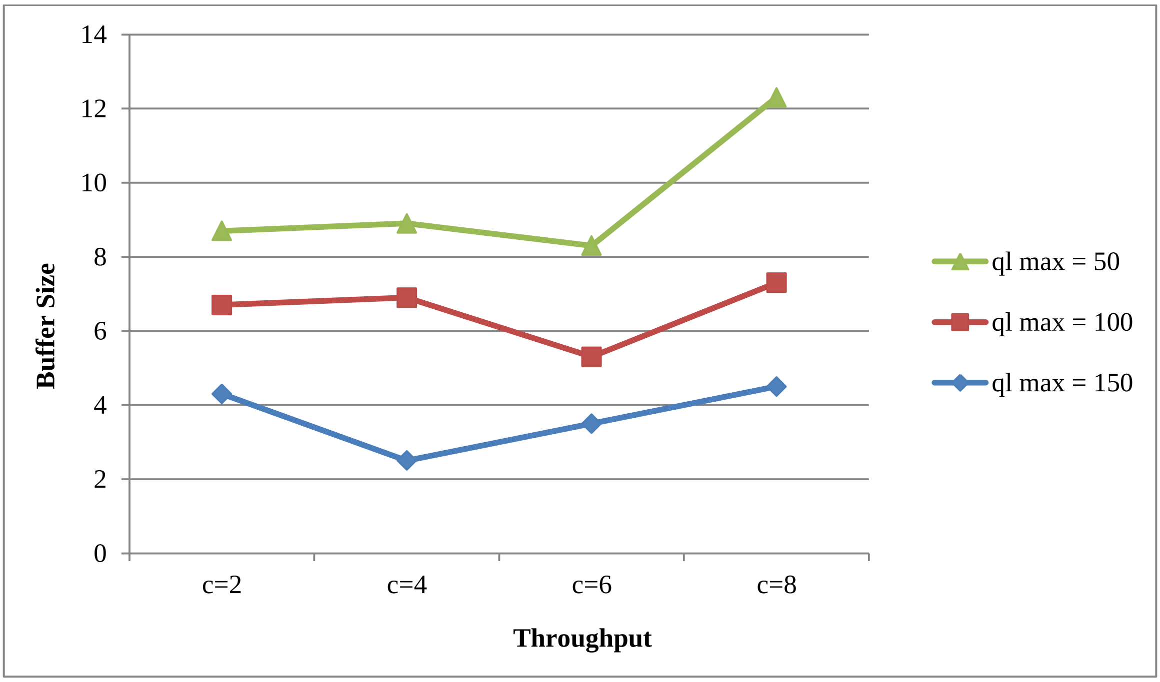

Fig. 4.1. Dependency of the input flow intensity at various levels on the output flow intensity

At certain levels, the output flow strength is more dependent on the network's performance than the buffer size. SOA system attributes. The intensity of the output flow rises with the strength of the input flow. Further rises until a predetermined system productivity limit value is reached. The SOA system is visibly overwhelmed when it operates at extremely high load stages, as seen by the reduced output flow intensity and rising input flow intensity.

Based on the simulation results, requests are lost due to network buffer overflow only when there is extra input data, but requests are not lost due to server buffer overflow. We can deduce from the graphics in this figure that the chance of a request losing its validity relies on network performance, which is far greater in significance than the amount of buffer space at an individual input value.

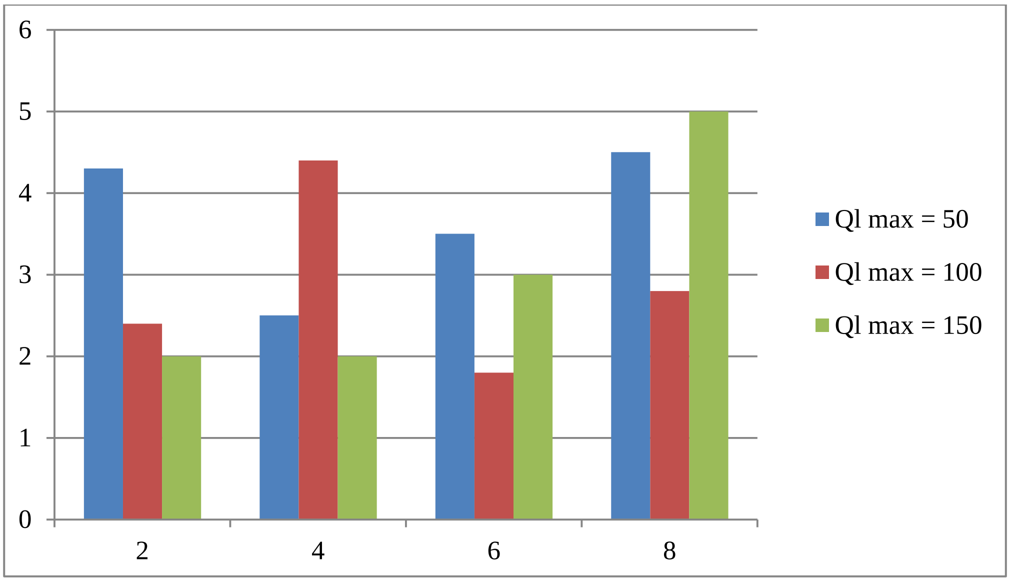

Fig. 4.2. Efficiency of the SOA framework

Figure 4.2 displays the histogram that demonstrates how the productivity of the distributed SOA system under investigation depends on the buffer size and network performance. Assuming a particular network throughput and the size of the buffer, a maximum amount of the output flow intensity of supplied requests was used for calculating the system productivity. As the results shown in Figure 8 demonstrate, there is very little variation in the dependence of SOA system efficiency on buffer size for a given input data when compared to the dependency on system productivity on network throughput.

5. Conclusion

Several software architectures for SOA-based applications were presented in this article. Applications based on SOA differ significantly from traditional software architecture in that they feature a dynamic architecture as opposed to a static one. The architecture of SOA-based apps was then classified using a system that this study presented. This classification enabled new understandings of runtime behaviours of SOA-based systems, including dependability, performance, dynamic reconfiguration, and runtime validation and verification.

This paper presents the study's findings regarding the performance of the distributed SOA system under constrained link and buffer resources of the system's fundamental communications network. A collection of models that replicated the functioning of a distributed SOA system were employed for that reason. Colored Petri Nets served as the foundation for the model construction. The preceding works' hierarchical structure was employed, but modules corresponding to the server and network resources were changed to enable accounting for the constrained sizes of the server and network buffers.

Acknowledgment

The authors wish to acknowledge the financial support provided by the Geran Penjanaan (SO code: 14587), Universiti Utara Malaysia, Sintok, Kedah, Malaysia.

References

[1] Riad, A. M., Hassan, A. E., & Hassan, Q. F. (2010). Design of SOA-based Grid Computing with Enterprise Service Bus. *Adv. Inf. Sci. Serv. Sci.*, *2*(1), 71-82.

[2] Srinivasan, L., & Treadwell, J. (2005). An overview of service-oriented architecture, web services and grid computing. *HP Software Global Business Unit*, *2*, 1-13.

[3] Ahmed, W., & Wu, Y. W. (2013). A survey on reliability in distributed systems. *Journal of Computer and System Sciences*, *79*(8), 1243-1255.

[4] Tsai, W. T., Fan, C., Chen, Y., Paul, R., & Chung, J. Y. (2006, April). Architecture classification for SOA-based applications. In *Ninth IEEE*

International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC'06) (pp. 8-pp). IEEE.

[5] Jorstad, I. (2005, July). A service-oriented architecture framework for mobile services. In *Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources Conference/E-Learning on Telecommunications Workshop (AICT/SAPIR/ELETE'05)* (pp. 65-70). IEEE.

[6] Moulick, H. N., &Ghosh, M. (2013). Medical image processing using a service-oriented architecture and distributed environment. *American Journal of Engineering Research (AJER)*, *2*(10), 52-62.

[7] Griffin, D., &Pesch, D. (2007). A survey on web services in telecommunications. *IEEE Communications Magazine*, *45*(7), 28-35.

[8] Duan, Q., Yan, Y., &Vasilakos, A. V. (2012). A survey on service-oriented network virtualization toward convergence of networking and cloud computing. *IEEE Transactions on Network and Service Management*, *9*(4), 373-392.

[9] Papazoglou, M. P., &Georgakopoulos, D. (2003). Service-oriented computing. *Communications of the ACM*, *46*(10), 25-28.

[10] Bih, J. (2006). Service-oriented architecture (SOA) is a new paradigm to implement dynamic e-business solutions. *ubiquity*, *2006*(August), 1-1.

[11] Kirkham, T., Savio, D., Smit, H., Harrison, R., Monfared, R. P., &Phaithoonbuathong, P. (2008, July). SOA middleware and automation: Services, applications, and architectures. In *2008 6th IEEE International Conference on Industrial Informatics* (pp. 1419-1424). IEEE.

[12] Du, J., Wang, H., Du, P., & Chen, H. (2022). An Optimization Model of English Teaching Resources Based on Distributed SOA. *Mobile Information Systems*, *2022*.

[13] Kovalenko, T. N. (2012). Analysis of productivity of distributed systems with service oriented architecture under conditions of limited link and buffer resources of telecommunication network.

[14] Magedanz, T., Blum, N., & Dutkowski, S. (2007). Evolution of SOA concepts in telecommunications. *Computer*, *40*(11), 46-50.

[15] Chen, R., Guo, J., & Bao, F. (2014). Trust management for SOA-based IoT and its application to service composition. *IEEE Transactions on Services Computing*, *9*(3), 482-495.