DOI: http://dx.doi.org/10.26483/ijarcs.v14i3.6976

ISSN No. 0976-5697

Volume 14, No. 3, May-June 2023

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

STA-VISION19 ENABLING INTEROPERABILITY BETWEEN DISTINCT SYSTEMS

Syed Taimoor Ali
School of Computer Science and Engineering Central South University,
Changsha 410083, China

Abstract –For the past few years, technology has been steadily rising and leading to rapid change in the development process. Information & Communication Technology (ICT) approaches have been moving toward new development trends with a huge elevation of comprehension. Service providers are always seeking customer satisfaction and due to multiple essentials, clients prefer to migrate from one application to another. Newer software applications are being launched, but they are constricted within a defined scope and cannot perform functions of interoperable components. Therefore,the aim of this study is to facilitate clients with multiple services over a single platform and gain knowledge about cloud computing, service computing techniques to enable flexibility and interoperability between applications. This work proposes an STA-Vision19 cloud platform and Service Integration Bridge (SIB) as solutions. To validate methods,abasic prototype for the proposedframework STA-Vision19 andtwo applications E-Learning and Blogging System were implemented. In results, software evaluation parameters, and testing results are discussed, which indicates that the proposed system can collaborate two or more applications and it is beneficial to use as compared to an isolated system.

Key Words: –Service Orient Architecture; Cloud Computing; Service Computing; Integration; Web Service; Resource Sharing;

1. INTRODUCTION

Information Technology (IT) is a wider field that evolves with the advancement of knowledge. Over the years methods of developing applications have been evolved by adopting new Information & Communication Technology (ICT) techniques. Technology like cloud computing brought a massive change in the development process by increasing data storage and computing power.

Usually, two approaches, centralized and decentralized are considered to manage systems and their functionalities. Term centralized considers the whole system concerned with a central authority, while the term decentralized is not concerned with the central authority, it relies on all defined levels of an organization from the middle to low [1].

In the life of software systems data is to be stored on database servers, and most of the large-scale software systems use the CDB approach (Centralized Database), the database that is located, stored and maintained on a single location [2].

In the current climate, organizations are standalone and have defined domain boundaries. They are made to perform specific functionalities, like “YouTube” for video streaming, “Facebook”,” WeChat” social networks for communication purposes, and a lot of other systems with their own aspects.

An application is considered as a service and the service providers are always looking forward to accomplishing the goal that satisfies the end-users. The users over the web have multiple requirements, and they access several applications to come into their needs. Clients from one system cannotavail of services from another unless collaboration is made.

When it is about integration then cloud computing is the best solution. It is an environment that works with a remote server to maintain applications and data. In cloud computing, everything offered to the client is treated as service and it is considered as a utility computing model where services with a wide range can be offered to users on demand.

Cloud technology is an evolution of virtualization utility computing. It is a web space where computing terminologies are pre-installed and provided to the clients as a service which includes networks, servers, data, operating systems, applications, storage, services, and processing power. It can improve the availability of IT resources with huge advantages where clients can use infrastructure with pay per use on-demand and it saves the cost to buy required physical resources.

The responsibilities of service providers increase and become challenging when huge traffic of end-users comes over a network. Like responding to client requests within time, retrieval of accurate information, satisfying customers withthe availability of services, system downtime and so on. For that reason, numerous experiments and researches are being conducted in laboratories to evolve and make the efficient computational process, management of software architectures and networks.

As time passes, things are moving towards modern technology. The evolution of software architecture,

mechanism and maintenance have become a challenge for service providers. The application developed with traditional methods cannot compromise with the protocols of modern frameworks while making collaboration.

Previously applications were monolithic in nature and now techniques like web service and microservices are being used, which made the computational process more powerful, flexible and efficient than traditional.

Web service is a technology that works as a communication means to integrate multiple applications.

Basically, web services are programed methodologies implemented by programmers available at the service provider hub. Most of the available paid services are provided by AWS and Google. But programmers are focused touseopen source web services, or they are concerned to design own methodologies.

The trend of using web services is expanding, because it reduces the cost of expenditures and makes business efficient. That is the reason, big companies like Google & Amazon are always using their own services and not relying on others.

What else can be better if you get all in one place?It is integral to develop such kind of a network-based system, where all fully functional applications and their services can be placed. This study proposes a Cloud-based Content Management System (CMS) to integrate distinct applications and services, thereby clients can be served with available services from distributed servers over a network.

2. Research Challenges

Non-collaborative or Non-Integral frameworks have numerous difficulties. In the present atmosphere, the working frameworks are isolated and oversee process separately. They have separate data centers, equipment, teams, and operations.

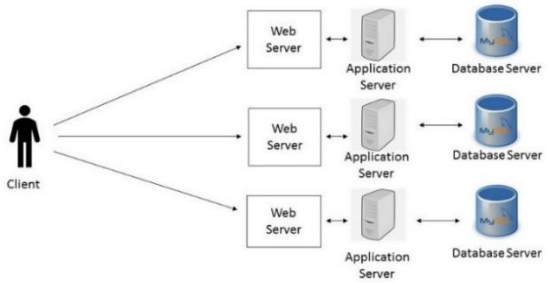

Fig. 1. Access of Client on Distinct Systems

The Scenario in figure 1 illustrates the access of a client over multiple applications. Each system is designed and implemented to provide specified functionalities. By the reason of requirement limitations, clients depart to different platforms.

Somehow, at the backend applications may have a monolithic structure that may face numerous challenges while having huge traffic of users over a network. Furthermore, the drawbacks with isolated systems are described as follow:

- **a.** Inflexible with demand. One system may not overcome all requirements of end-users. The limitation of services lets the user move toward other systems. This is often expressed when end-user meet with an external business view of applications.
- **b.** Investment of larger expenditure on infrastructure.
- **c.** The user registration process in an individual system may cause the storage of repetitive information.
- **d.** The systems are standalone and isolated. It is good to work with small business operations. But in large scale software, an isolated system may face challenges [3, 4].
- **e.** Extending a network will be challenging if isolated applications are monolithic in nature.
- **f.** The data of the separate system is localized within their defined scope that couldn't be used out of boundaries. The information available on one system cannot be accessed on others.
- **g.** None of the applications will allow privileges to use functions of one in another unless collaboration is made.
- **h.** Problems in the interoperability process due to different standards and protocols.

3. Related Works

Different approaches have been discussed for software development and system integrations. Some of these are discussed below:

Hohpe G. and B. Woolf [3], applications are isolated. But they need to be linked with each other because applications can work better if they are integrated. Since distinct culture applications may face some challenges while collaborating, they should reduce their restrictions to make integration without facing many problems. The applications are made with different technologies and programming languages, they must have the flexibility to make collaboration.

Dror G. Feitelson*et al* [5], Some software systems are determined and have specified scope. Their deployment is scheduled at the initial stage, but if it is about creating large scale applications, the development will be continued due to limitless scope.

Linux OS since 1994 and Google since 1997 [6, 7], Many software systems have been launched in the past and still being used. The applications with long term evolution are available in multiple fields, i.e. Facebook is being evolved since May 2004.

Mukhtiar Memon *et al* [8], proposed the solution for agriculture research organizations, which enables scientists to share scientific research information among distinct organizations. The author implemented a prototype for agriculture research organizations where scientists of different organizations are working in a collaborative environment.

Stephanie Lang *et al* [9], Companies are always looking for customer satisfaction. Initially, Amazon was implemented to sell books and today it's a second big e-commerce company trying to provide all user requirements over web-like hardware, media, services, technologies and many more.

John Jung WoonYoo*et al* [16], proposed a Service Oriented Architecture (SOA) framework to collaborate healthcare services. The proposed framework reduces the medical expenditure of clients and increases the revenue of the healthcare service provider.

Swati Gupta [17] discussed distributed database systems and mechanisms. Hardware, software placed at

different systems can make coordination and their stored data be accessed through network-based applications by passing messages. Moreover, the author also discussed the Pros and Cons of the distributed database system.

Yan Cao *et al* [18]., described that organizations have different environments and having isolated application systems. In this scenario sharing information in the business process is much important.

Cloud Computing and Big Data are widely used technologies to combine different platforms by virtualization. Feng Yu *et al* [19]., proposed an improved and efficient heterogeneous cloud architecture which includes multifunction of big data and cloud computing.

Abdulsalam Ya'uGital*et al* [20]., Proposed a cloud architecture framework for the collaborative virtual environment (CVE) to enable users from distinct systems to share internet resources and make cloud business process efficient.

Suhail Madoukh and RebhiBaraka[21]., purposed a system for data integration of Palestinian e-Government technical framework and transformed the central database model of Palestinian government into SOA framework focused on interoperability and flexibility using Web Services.

The contributions of this research are as follows:

- **a.** This work proposes an interactive SOA based cloud architecture STA-Vision19, that enables collaboration among distributed systems.
- **b.** Cloud infrastructure for STA-Vision19 and a basic prototype was implemented. Moreover, for the interoperability experiment, two more SOA based applications E-Learning & Blog Applications were implemented.
- **c.** This work discusses the given solution Service Integration Bridge (SIB) for cloud interoperability, SIB mechanism, service publishing, service consuming, trust mechanism and load balancing.
- **d.** In results, simulation for load balancing algorithm, general evaluation, functional, non-functional testing and real-time unit testing of implemented components are discussed.

At the end of this practice, a survey was done to evaluate the designed prototype. 15 students of the different level group by 5, including bachelors, masters, and Ph.D. were considered for testing.

4. Research Design & Methods

This research was conducted to gain knowledge about the adoption of modern technologies and propose a collaborative framework for non-integrated software systems. To overcome the challenges defined above in section 2, we have investigated some Information & Communication Technologies (ICT) approaches and proposed a novel SOA based collaborative framework "STA-Vision19".

As discussed before, various services are situated in various application areas, since we have proposed a framework and built up a prototype to incorporate different applications and their services which has brought numerous domains and related applications over a single platform. CMS (Content Management System) technology is adopted which facilitates end-users to perform CRUD (Create, Read, Update, Delete) operations. Simultaneously we have used cloud computing technology to design the infrastructure of the system and SOA Web Services to make integration possible among distinct applications. Furthermore, for interoperability experiments, we have designed and implemented two SOA based applications "E-Learning" and "Blog App".

4.1 Proposed Collaborative System (STA-Vision19)

The STA-Vision19 is a solution for the non-integrated application systems. It is a vision that brings all distinct applications and services at a single integrated environment, where defined domains and related applications are loosely coupled and can share information and services through the network.

Figure 2illustrates the architecture of STA-Vision19, where clients can access virtual & collaborative environments using an application framework. According to the architecture illustrated in figure 2, currently, we have multi-flavored domains and related applications. Each domain can have one or more sub-domains / applications, that can be configured and stored over virtualized servers by using an administrative tool of STA-Vision19. It facilitates clients with a variety of services that are loosely coupled.

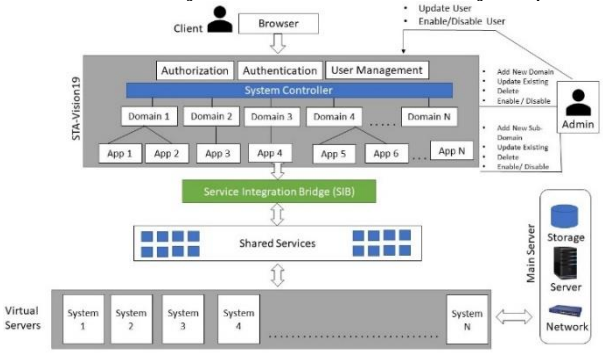

Fig. 2. Proposed Collaborative System (STA-Vision19)

All required applications and their services are subordinate to the system. A variety of domains like Entertainment, Social Networks, E-Business, E-Education, E-Billing, E-Medical, E-Reservations, Gamming, Blogging and many more can be embedded into this system and make the system pleasing to the users.

STA-Vision19 enables distinct applications to store their service descriptions into the repository of service integration Bridge (SIB), which enables integrity and interoperability between multiple applications.

STA-Vision19 works as a central authority and governs all subordinate applications that are linked with it. The proposed framework can be extended up to the Nth number of domains, applications, and virtual servers as illustrated in figure 2.

To overcome the challenges characterized in section 2, initially, we have implemented a fundamental model that is illustrated in figure 2.Itis a cloud-integrated platform that consists of several domains, applications, and cloud services. Moreover, we have additionally

implemented two SOA based applications "E-Learning"&"Blog App" to validate integration methods.

In the designed framework "STA-Vision19", users are divided into 3 categories.The privileges and access controls of the categorized users, that are defined as follows in section 4.2.

4.2 User Access Control

Applications contain some protected information. For the reason of information security and access control, authentication is necessary. Both terms authentication and access control are often used together in applications.

Access control is a technique that controls who can view or use the resources in a system. In the modern scenario, applications may have multiple types of users and each type of user has been assigned a role on what basis he performs the operation. It is a process to identify appropriate user and privilege right access to the right user. Thus, STA-Vision19 has categorized users into three different panels which are portrayed as beneath:

4.2.1 Super Administrator:

SuperAdministrator is capableof managing, creating, updating, and deleting users, defined domains and related applications and data. The super administrator works as a manager for STA-Vision19 and it can import and configure the new virtual host, application, related files and database over the system. An external application can also be embedded with the system "STA-Vision19".

4.2.1.1 STA-Vision19 App Management

The super administrator is the main authority of this system. Management operations of the proposed system are arranged into three distinct segments, that are discussed as follows:

(a) Domain Management

In domain management, a super administrator can search existing domains, add new domains, modify or update the information and delete existing domains.

(b) Sub Domain or Application Management

In this section, a super administrator can search existing applications, configure or add new applications in domains, manage application resources, update or modify existing applications, and delete applications. In this part, the external application can be embedded in the system that is located at distinct servers.

(c) User Management

In this section, a super administrator can manage the remaining categories of users, update or modify user information, enable, or disable users and create the new sub-administrators for sub-domains or applications.

4.2.2 Sub Administrator:

Sub-Administrator is responsible for managing sub-applications. Each application may have one administrative authority, as in the current scenario we have two applications "E-Learning" and "Blog app", thus we have two Sub-Administrators.

In this system, sub-administrators are enterprise managers for applications or sub-domains. They are enabled to perform the cross-platform operations and provide or avail cloud services to STA-Vision19 and to the other subordinate applications. (Managers of E-Learning and Blog Application).

4.2.3 Registered Client:

Clients can access multiple applications and services according to their needs. Available applications are loosely coupled so the information of one application can be shared in another, and that flexibility makes users attracted to the system.

The registered users from STA-Vision19 can migrate from one app to another with the same user credentials. Encoded JSON data is generated and sent through POST (HTTP Protocol), and at the receiving end system decodes the information and executes the authentication process.

4.3 Implementation of SOA based E-Learning Application

E-Learning involves the use of a computer or electronic devices to provide training, educational or learning materials to support education via the internet. It supports services for learning and teaching that are delivered and enhanced through digital technologies and electronic media. To analyze experimental results, the E-Learning application was implemented and configured in the proposed framework "STA-Vision19", the structure of the application is illustrated in figure No 3.

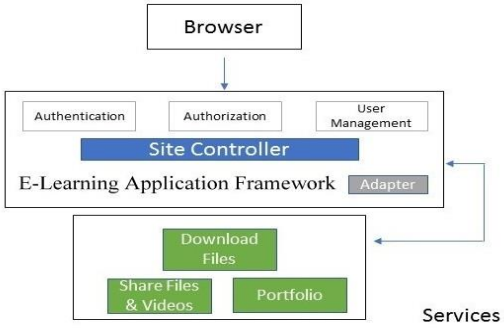

Fig. 3. E-Learning Application

Figure 3 illustrates an architecture for E-Learning Application, where services of E-Learning can be accessed by using an application framework. It consists of services that facilitate users to view, download and share electronic data and manage their personal portfolio. While sub administration service allows the application manager to upload, delete and modify the data.

Fig. 4. Graphical User Interface of E-Learning App

Figure 4 illustrates the GUI of the designed application "E-Learning" for central south university.

4.4 Implementation of Blog Application

The Blog application facilitates clients to create discussion forums where they can easily discuss different research issues, subjects, and related queries. It supports collaboration among multiple users based on the discussion forum in which information about multiple criteria can be shared over a network using a web browser. Structure of Blog application is illustrated as follow:

Figure 5 illustrates the structure of the blog application. It includes services that help to Create, Delete, Update, View and Manage discussion blogs, user portfolios, data sharing, and view shared repositories.

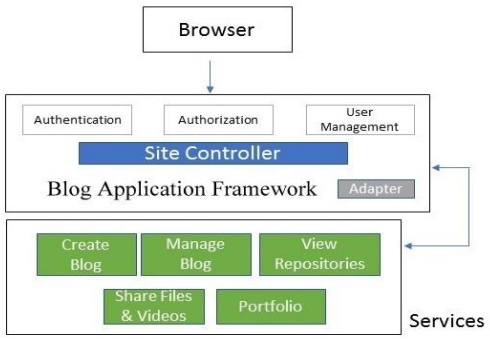

Fig. 5. SOA based Blog Application

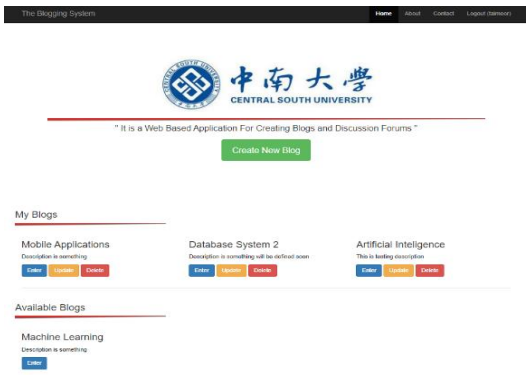

Fig. 6. Graphical User Interface of Blog App

Figure 6 is a graphical user interface of the designed application "Blog App" for Central South University.

4.5 Integration of Applications

The data frameworks and application systems normally refuse to compromise on the sharing of modules. But for integration, they have to compromise. Multiple services can be integrated over a common solution, which can guide users to use services with desired functionalities.

Service-Oriented Architecture (SOA) supports interoperability among distributed systems. It is a mechanism to bring distinct systems at a unique and integrated environment, where systems can be interconnected with a central entity so-called SOA framework.

This work proposes a SOA based framework "STA-Vision19" to interconnect distinct applications and servers, that support collaboration using SOA standard protocols.

STA-Vision19 was designed and implemented based on SOA & cloud computing technologies. It is an approach to deploy loosely coupled services and enables cross-platform integration over the web. As illustrated in figure 2, there are multiple domains and related applications, and we considered two distinct domains "Blogging" and "E-Learning" from the main system "STA-Vision19" and implemented their child applications along with service as illustrated in figure 3 & 5. After implementation, these are configured into the main system "STA-Vision 19" to enable applications to work in a virtual and collaborative environment.

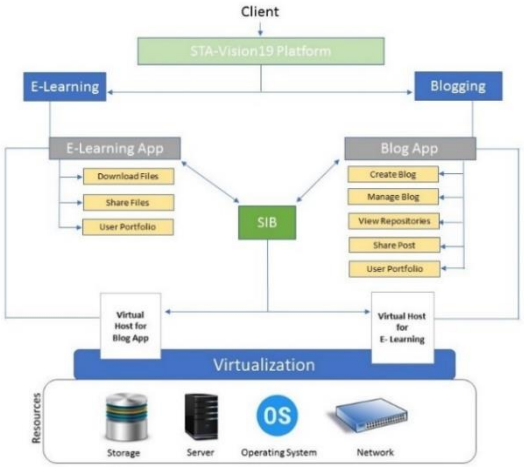

Fig. 7. Integration of Application

Figure 7 illustrates an integration mechanism for two applications in STA-Vision19. Components of applications are considered as services, in the above figure both applications have some shared services and STA-Vision19 performs a union operation between applications, that enables the user to access multiple services and desired functionalities through a single and central platform.

Services of both Blog app and E-Learning applications are coupled over a service-oriented architecture (SOA) platform. The integration of the applications can be done by using the proposed method "SIB" solution. What is SIB? and the mechanism of SIB is discussed next in section 4.6.

Equation:

$$EL = \{S1, S2, S3, S4\}$$
$$BS = \{S5, S6, S7, S8\}$$

Let's consider "***EL"*** as (E-Learning App) and "***BS"***

as (Blog App), both are two distinct applications along with services ***S1, S2, S3, S4*** and ***S5, S6, S7, S8***.

"ED" (E-Learning Domain) &**"BD"** (Blogging Domain) are two different Domains.***"ED"*** contains multiple sub-applications related to Electronic Education and **"BD"** contains applications related to Blogs.

Sub-domain ***"EL"*** (E-Learning App) is a member of domain ***"ED"*** (E-Learning Domain) and the sub-domain ***"BS"*** (Blog App) is a member of domain ***"BD"*** (Blogging Domain).

$$EL \in ED$$
$$BS \in BD$$

Each system may have both private and public services, the collaboration between the two systems is only possible through public services.

Let's consider***"ELx"*** as a set for private services of "***EL"*** having a service named "***S4",*** and ***"BSx"*** as a set for private services for ***"BS"*** having a service named "***S7"***.

$$ELx = \{S4\}$$
$$BSx = \{S7\}$$

To find out public services from EL and BS we will perform complement operations for ***ELx*** and ***BSx***.

For ***ELx'***

$$ELx' = EL - ELx$$
$$ELx' = \{S1, S2, S3, S4\} - \{S4\}$$
$$ELx' = \{S1, S2, S3\}$$

Now for ***BSx'***

$$BSx' = BS - BSx$$
$$BSx' = \{S5, S6, S7, S8\} - \{S7\}$$
$$BLx' = \{S5, S6, S8\}$$

Now for the integration of two applications and sharing public services over a central framework, we will perform union of ***ELx'*** and ***BSx'*** as:

$$SIB = ELx' \cup BSx'$$
$$SIB = \{S1, S2, S3\} \cup \{S5, S6, S8\}$$
$$SIB = \{S1, S2, S3, S5, S6, S8\}$$

As we have already described that *"**EL"*** is a member (∈) of *"**ED"***and *"**BS"*** is a member (∈) of *"**BD",*** while *"**ED"*** and *"**BD"*** are members (∈) of the main system "STA-Vision19".

$$"EL \in ED"$$
$$"BS \in BD"$$
$$STA\text{-}Vision19 = \{ED, BD\}$$

Hence, domains, both applications and their shared services are members to STA-Vision19 as illustrated in figure 7.

Cloud computing technology highlights the creation of services on demand and makes communication between SOA and cloud computing deep. When service is created, users do not depend on the knowledge of the service creation. SOA exchanges data deploying protocols [10,11].

4.6 Proposed Service Integration Bridge (SIB)

Service Integration Bridge (SIB) is a designed methodology paradigm, which works as a connector between distinct systems. It connects multiple applications with each other and is responsible to manage service transactions.

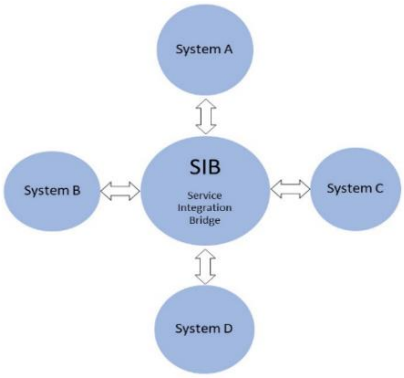

Fig. 8. Proposed Service Integration Bridge (SIB)

Figure 8 illustrates the proposed Service Integration Bridge (SIB). It is a methodology that enables services to communicate and interact using standard messaging. It works as middleware between applications to route service messages, transfer data, manage protocols and security for transmitted information.

Service Integration Bridge (SIB) works as a message carrier among distinct systems and it manages the mechanism of routing messages and delivery. The messages transferred between the systems are queued in Service Integration Bridge (SIB), individual systems are not responsible for message delivery, they are focused on the development of web services rather than managing the routing systems. The working mechanism of SIB is divided into six segments, that are service protocols, service publishing, service consuming, service retrieving, trust mechanism and load balancing. These terms are discussed as follows:

4.6.1 Service Protocols

The service-oriented architecture is implemented using internet protocols referred to as web services, and web services are often implemented using Representational State Transfer (REST), Simple Object Access Protocol (SOAP) or JavaScript Object Notation Web-Service Protocol (JSON WSP).

The Protocols support machine to machine interaction using standard messaging over HTTP. The conversion among servers follows the standard messaging protocol for invoking service using Universal Resource Located (URL). The Transferred message contains input, output to or from web service, it can also carry additional entities like security information for authentication.

We have used JSON-WSP (JavaScript Object Notation Web-Service Protocol) for service description request and response. Communication among clients and JSONWSP was done using an HTTP POST request and response methods.

JSON is a data representation format like XML, and protocols as SOAP that describes types and methods according to given services. JSON WSP not only supports the plain text but it can also transfer objects through HTTP over client application. The client sends a request to the server in text format and JSON WSP returns the object as a response. In our designed system, we have focused on designing our web services with JSON WSP because it is lightweight and easy to read/ write and supports easy integration with most languages.

4.6.2 Service Publication in Service Integration

Bridge Repository (SIBR)

Service-Oriented Architecture (SOA) supports interaction, interoperability among systems using standard messaging. The service provider publishes a service description at the service registry, that contains specifications about the usage of service. While a consumer can avail service after going through the steps which are described in the service description repository to embed web service into an application.

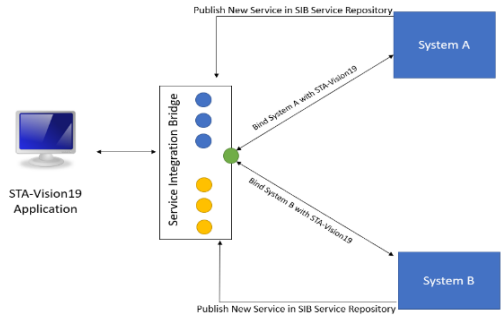

Fig 9. Service Registry

Figure 9 illustrates the mechanism of publishing services in the repositories. The blue circles show published services in service integration bridge (SIB) repositories by System A, while yellow circles are services that are published by system B.

A green circle is a service that is provided by the designed system STA-Vision19. It can be embedded in different applications and authenticates users by using credentials from STA-Vision19.

In experimental work, we have embedded service in both E-Learning and Blog Application for authenticating users and allow them access into the systems.

4.6.3 Service Consuming Through SIB

The mechanism of service consuming is illustrated in Figure 10. It defines how services of one system can be availed in another. There are two distinct applications (A & B), that are interconnected with Service Integration Bridge (SIB). SIB works as a service broker between system A and System B, both systems can work as either a service provider or a service consumer.

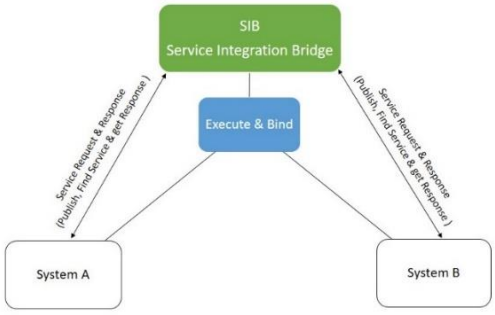

Fig 10. Service Consuming

Let's follow the steps and consider System A as a service consumer and System B as a service provider:

Step 1: System B registers service into SIB repository as discussed & illustrated in figure 9.

Step 2: System A finds service from Service Integration Bridge (SIB) repository and makes a request for service.

Step 3: Service publisher and consumer, both Parties are authenticated through the assigned token. If authenticated, it moves towards step 4.

Step 4: Service Integration Bridge (SIB) sends a response back to System A as a service description.

Step 5: System A follows instructions given in the service description file and proceeds request again to Service Integration Bridge.

Step 6: Service Integration Bridge (SIB) receives the request, then executes and binds a connection between System A and System B that allows both to be integrated.

We followed the above steps in our experimental work to integrate our implemented applications E-Learning & Blog Application or more, using our system STA-Vision19.

4.6.4 Service Retrieving between distributed systems.

In the distributed environment, data is stored on different virtualized servers and to retrieve information at a single point, there must be an integrated environment. Google search engine is the biggest example of distributed data retrieval. It is an integrated environment where multiple distributed and virtualized servers are interconnected, whenever a user searches any information using the graphical user interface (GUI) of google search engine, it retrieves data from multiple interconnected servers and presents it over browser.

The distributed servers are configured as virtualized servers which are often interconnected with a central entity. In our proposed work there are some virtualized servers that are connected to the central authority STA-Vision19. Service Integration Bridge (SIB) is the central entity in the proposed system, that connects multiple servers to each other and enables interoperability between them over the world wide web.

The main part of our designed system is Service Integration Bridge (SIB), which works as a medium among distinct servers as illustrated in figure 8. As designed framework STA-Vision19 is focused to embed multiple services over a single platform, so the information about shared web services from distinct systems ispublished over the SIB repository, which enables the users to search, embed and consume services from distributed servers through an integrated environment.

Embedded services authenticate administrator and execute the query to bring information over an integrated environment STA-Vision19 and make shared services available over an application for clients. At the same time service descriptions can be accessed for STA-Vision19 to Sub-App or Sub-App to Sub-App integration.

Every system has two kinds of information “shared” and “secure”. The collaboration among systems can be done only with shared services. In this research, we brought shared services of multiple systems over the STA-Vision19 platform. The process of service publishing and consuming is illustrated above in this section.

Figure 10 shows the collaboration of two distinct applications through Service Integration Bridge (SIB) this collaboration can be made up to the N^{th} node. The proposed system STA-Vision19 is not only restricted to make collaboration between two applications, but the N^{th} number of applications can also be integrated as illustrated in figure

2.

In STA-Vision19 relationships among applications are classified as follows:

4.6.4.1 Sub-App to Sub-App (One-to-One)

Fig 11 App to App Relation

Figure 11 shows the relation between 2 Sub-Apps. It is the collaboration of two applications using the SIB solution as illustrated in figure 10. Both systems P^1 and P^2 can work as either a service provider or a service consumer by going through the steps defined above in section **4.6.3**.

4.6.4.2 STA-Vision19 to Sub-Apps (One-to-Many)

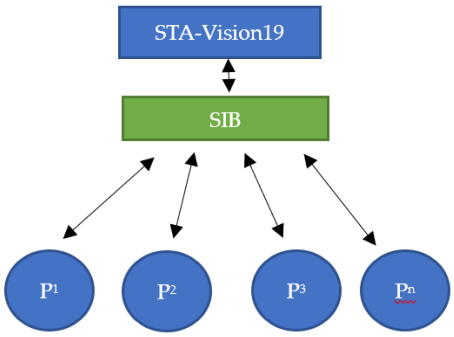

Fig 12 STA-Vision19 to App Relation

Figure 12 is the relation between the designed framework and multiple applications. Multiple applications are interconnected with STA-Vision19. Authentication service from STA-Vision19 can be embedded in P^1 up to P^n, which enables access and route users towards multiple applications. In the same manner, shared services from P^1 to P^n systems can be embedded in STA-Vision19, through which multiple shared services from distributed systems can be accessed over our proposed framework.

4.6.4.3 Sub-Apps to Sub-Apps (Many-to-Many)

Distinct applications can publish service descriptions over SIB repositories. At the other hand, distinct service consumer follows the description steps and consume available service over their applications.

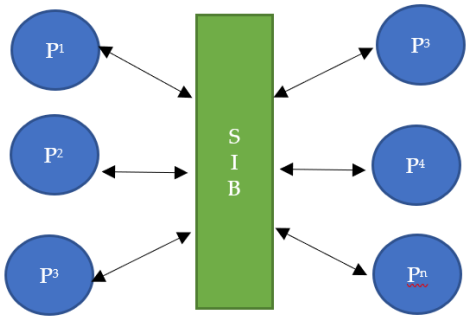

Fig 13 Sub-Apps to Sub-Apps Relation

Figure 13 represents relationships between multiple Sub-Applications through Service Integration Bridge (SIB).

4.6.5 Service Integration Bridge (SIB) Trust Mechanism

To enable collaboration among multiple systems, enterprise managers of distinct applications may make registration over STA-Vision19. At the registration time, STA-vision19 generates a unique code based on the current date and time and sends it to the user via email in the shape of token for confirmation and allows ownership of an account to users, which lets the user access the system.

When different parties share information over a network, an external source may try to temper or corrupt the shared information. To secure communication messages and resist vulnerability, encryption techniques are used. We have used hash 'encode64()' encryption for data encryption, that encrypted data is stored in temporary memory (variables) and furthermore it is encrypted into JSON encoded format, while the token is already generated with 'md5' encryption to secure communication among applications. The data transferred among distinct applications is transferred in an encrypted format and the only authority that can decrypt it is the one who has a secret key. The generated token is assigned to each user at registration time, which is used as a secret key whenever the system makes communication.

We considered date and time for generating token value because date and time are continuously changing and have unique specifications that can be calculated in seconds. So, the generated token is a unique entity with each enterprise manager that authenticates actual authority and allows communication among applications.

Service Integration Bridge (SIB) is responsible for managing all transactions of the system. The given unique keys to different enterprise managers are stored in the STA-Vision19 database. The integration of multiple applications through SIB is done in a secure manner because SIB authenticates both service consumer and service provider, every time they make collaboration.

For each message, Service Integration Bridge (SIB) authenticates both sides based on a given unique token and allows communication between different servers. At the time of integration, that token is passed through the parameters when service consumer is requesting for service, while token information of service provider is already available at SIB, which was authenticated at service publishing time. It makes integration possible only among valid and authenticated parties. Each embedded application will have a unique token provided by STA-Vision19 while making registration, that unique token is used every time when communication is made among distinct systems.

STA-Vision19 is a central authority that can be considered as a token distributer among multiple applications through which Service Integration Bridge (SIB) enables them to make interoperability and perform cross-platform operations over a web.

4.7 STA-Vision19 Cloud Environment

STA-Vision19 has focused on collaboration among applications, its flexibility, interoperability, agility and advance features can make it a priority for clients. If the system can overcome the multiple needs of the end-user, everyone will love to be a part of it, and that is the biggest advantage of this system. Larger the number of clients over applications includes a lot of benefits.

In STA-Vision19, information of the clients is stored once in a database, the client user has no need to make individual registration for each application. It resists the storage of repetitive information. STA-Vision19 cloud

service modules are further described as following:

4.7.1 Software as a Service (SaaS)

Applications are released in the hosting environment, which can be accessed by application clients through the network. It provides an application STA-Vision19 with GUI interfaces to the clients with a variety of sub-applications and their services, in STA-Vision19 Super administrator can host applications and make them available to the clients.

Two kinds of applications can be embedded in STA-Vision19. One that is launched by STA-Vision19 itself and the other is an external application that is hosted somewhere else but uses STA-Vision19 for service publication.

4.7.2 Platform as a Service (PaaS)

PaaS is a service for consumers through which they can create cloud services and make it available for clients. STA-Vision19 can facilitate vendor (Sub-Administrator) to manage applications without headache of maintaining the infrastructure associated with hosting or launching of an application. Consumers just need to make a subscription in the main system and manage their application.

4.7.3 Infrastructure as a Service (IaaS)

STA-Vision19 can provide access to fundamental resources; like server, virtual machines, storage, and network. Instead of purchasing hardware equipment, one just needs to pay for what your application delivery requires. STA-Visison19 is responsible to manage and govern subordinate applications and it allows the super administrator to host and manage an application in its own data center and make it available for multiple clients over the web. Any pre-launched application also can be a part of it by contracting authority.

IaaS provides an environment for running user-defined virtualized systems in the cloud. A cloud-based CMS prototype is implemented that facilitates service providers to upload configurations using GUI and it is deployed to the clients within the environment. The internal management of the system is managed by the super administrator.

4.7.4 Database as a Service (DBaaS)

Database services are accessible through the cloud platform. STA-Vision19 can allow vender to access databases available at the physical infrastructure of a service provider, by making a subscription and consuming service over their systems.

4.8 Virtualization

Virtualization in STA-Vision19 allows administrators, vendors, and clients to work on distinct applications using the same software (STA-Vision19 Prototype). It is a process which virtualizes one server into multiple servers to carry out results. It resists the need for physical hardware system for separate applications. In STA-Vision19, we have virtualized one resource server into two virtual servers as illustrated in Figure 14. One for blog application and other for E-Learning application.

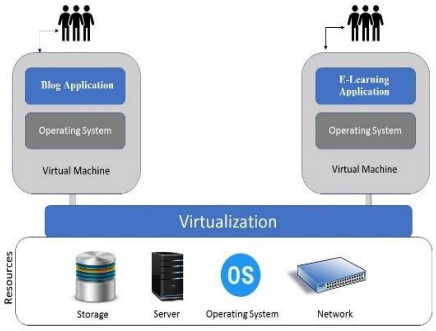

Fig.14. Virtualization

4.9 Load Balancing

Load balancing is a technique to manage traffic over a system. It is a middleware methodology between client and server which is responsible for managing transactions between networks and distribution of HTTP traffic. In the proposed system, the Service Integration Bridge (SIB) works as a broker among distinct cloud applications. It is a system controller that is responsible for the management of STA-Vision19's framework mechanisms like security, quality of service (QoS), inter-cloud operability, etc.

SIB mechanism governs requests and responses between STA-Vision19 and distinct applications. In distributed systems, servers are interlinked with each other through a network if they are in the same warehouse. In that scenario, data clones are stored on multiple servers such that one of these servers will respond to an application's request if the destination server is having huge traffic. Storing clone data on multiple servers may cause data duplication, but if there isn't any clone data stored, the fetching data will take internal path among servers which may consume more time, thus response time will be increased. In a scenario where applications located on different warehouses are interlinked with each other, they may never ever compromise on sharing data with other applications and for this reason, web services are used.

It is better to design a URL-based load balancing mechanism that can forward HTTP messages to destination servers. VM's specification may be configured in a way that can have less probability of server downtime instead of storing data clones over multiple servers, which may cause data duplication. Therefore, the request of the client can be sent directly to the appropriate server and get a response directly within time by applying efficient load balancing algorithms.

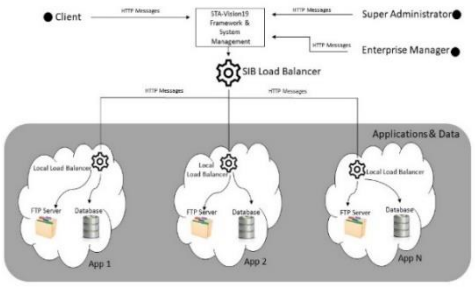

Fig.15. Inter-cloud load balancing

In STA-Vision19, the load balancer routes HTTP messages over distinct application servers as per request.

SIB repository makes the information available about using services that are located at different servers and published by distinct vendors. Users from different cloud apps utilize the services just by following procedures described in a published web service description file as illustrated in Figures 9 and 10.

Services that are published over the SIB repository have information about destination servers, so whenever a request comes to the SIB Load balancer, it is examined and forwarded towards destination servers to get a response.

There is a local load balancer at the application cloud that is responsible for managing transactions for a single application as illustrated in figure 15, Brokers contain only information about how a service can be embedded and used. The SIB load balancer is configured according to a client's request. Distinct cloud apps have their own load balancer located in their applications. A user's request from the STA-Vision19 framework is moved towards SIBLB (Service Integration Bridge Load Balancer), then it is forwarded to APP-LB (Application Load Balancer). This approach can be referred to as inter-cloud load balancing.

The published services over Service Integration Bridge (SIB) repository contain the address of actual servers, indicating where server data is located. Whenever a client requests access for services, the SIB examines the request and forwards it to the server using the URI (Endpoint-IP) defining a unique address of the server. Communication over the network is done using JSON-WSP web services or Restful APIs.

4.10 Advantages

4.10.1 Shared Resources

Shared resources are also known as network resources that can be referred to as computer data, information that is hosted by one host and can be accessed by others. In a collaborative system, distinct applications and resources are centralized over a single platform which enables clients to avail heterogeneous services.

4.10.2 Shared Services

The applications have some shared serviceswhich can be accessed by clients in the same or in other applications. The STA-Vision19 environment is a network-based structure where services of multiple applications can be shared with each other.

4.10.3 Data Sharing Web Service

This system has multiple applications,each has its own functionality and kind of shared information. Clients are enabled to share data from one application to another as per requirement, shared data can be referredto as education, scientific information, sports, news, business and so on.

Local Share: Term local sharing enables sharing within the application boundary; like in the Blog App, the system has multiple blogs inside. Data of one discussion blog can be shared in another blog.

Global Share: Term Global share enables sharing among distinct applications. Data of one application can be shared in another application.

4.10.4 Memory Sharing Web Service

The database is the backbone of the application. In a designed system, distinct applications and their databases are linked. The proposed system works as a communication bridge and allows distinct applications to share data from the memory of one application to another.

4.10.5 Device Sharing Web Service

Machines available at the client-side are not much efficient to perform high computation operations for data processing and so the execution process and analysis becomes slow and causes unnecessary delay. STA-Vision19 is a solution for that problem, where the client can send the request and distinct linked server machines are performing operations as per request.

4.10.6 Security

STA-Vision19 has secured infrastructure which does not grant access to data without authorization. A highly encrypted algorithm over the cloud is applied to minimize security risks.

4.10.7 Data Monitoring

STA-Vision19 enables the super administrator to monitor system and user activities by using monitoring services.

The design of the proposed system (**STA-Vision19**) is not limited.It is not based on specific functionality,but it has wider fields of interest.Multiple types of applications can be implemented as it's partas illustrated in figure 2. Initially, the structure of the system along with two applications; E-Learning and Blogging System, are implemented to validate integration methods. Furthermore, more applications will be added in the future.

5. Results

The objective of this study is to enable flexibility and interoperability among applications. As in the current climate, new tools and technologies like web services are being used, especially Google and Amazon are the companies that are trying to create varieties of web services to facilitate clients in web network. This research proposed the STA-Vision19 framework as a solution and Service Integration Bridge (SIB), that can overcome client needs from different aspects and enable integration between applications. A Service-Oriented Architecture (SOA) based Framework was proposed, and its basic prototype was implemented as illustrated in figure 2. Moreover, to integrate applications, Service Integration Bridge (SIB) was proposed which enables distinct applications to work in a collaborative environment and has the flexibility to share data, resources, and services. The mechanism of the Service Integration bridge (SIB) is also discussed in this work. The proposed system facilitates toadd additional components into the system and make it business efficient. In the current scenario, domains like social networks, entertainment, educational systems, billing systems, hospital systems, can be implemented in the proposed system and users can access a variety of domains and relate applications.

To validate results, multiple domains were added to the system STA-Vision19. Designed methodologies were tested on the implemented applications; E-Learning & Blog application, and the results were carried out that collaboration of two or more applications is possible in this system.

Having traffic of user's requests in a system is a challenge for the cloud atmosphere. In the designed methodology, we have applied a customized URL based load balancing algorithm, as discussed previously in section 4.9& figure 15.

Before implementing the given load balancing technique,for validation,a simulation was done. The

simulation parameters and performance analysis for load balancing are described as follow:

The applied load balancing algorithm was implemented using Cloud SIM 3.0, JDK 1.8.0_201 and Eclipse 2019-3. And for simulation, Cloud Analyst is used. Cloud Analyst is a GUI tool kit that works with Cloud SIM for testing and simulating cloud infrastructure [22].

For the simulation of the load balancing algorithm, we configured 4 user base stations as UB1, UB2, UB3, and UB4 at different geographical regions along with a central data center D1 using Cloud Analyst. The data center was configured as 8086 architecture, Linux operating system, Xen virtual machine manager, 204800 RAM (MB), 1000000 available bandwidths, TIME_SHARED VM scheduling policy. Each physical machine is integrated with 5 virtual machines. Each virtual machine is configured as follows: 10000 MB Image size, 512 MB memory, 1000 MB bandwidth.

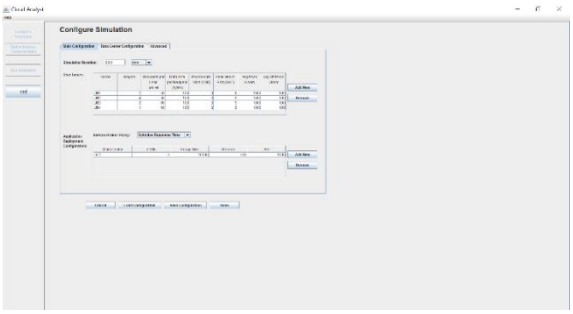

Fig.16 Configuration of Simulation Parameters

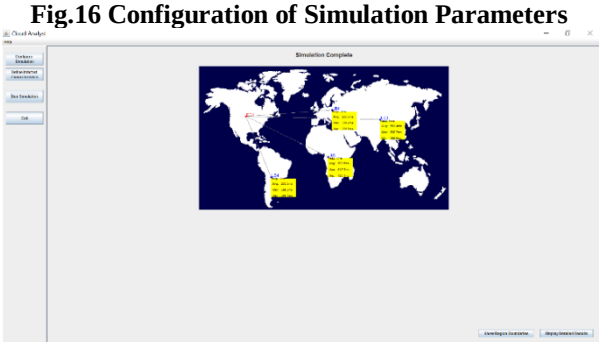

Fig 17 Simulation Process

The configuration parameters were set using Cloud Analyst GUI, as illustrated in figure 16,17 and the custom load-balancing algorithm was embedded from cloud SIM to get testing and simulation results of a cloud. After configuring simulation parameters, performance analysis and simulation results were carried out, that is described as following in table 1,2,3 and figure 18, 19:

TABLE 1
OVERALL RESPONSE TIME SUMMARY

Units	Avg (ms)	Min (ms)	Max (ms)
Overall response time	377.02	155.21	632.71
Data Center Processing Time	0.38	0.02	0.071

TABLE 2
RESPONSE TIME BY REGION

Userbase	Avg (ms)	Min (ms)	Max (ms)
UB1	299.78	235.71	354.22
UB2	502.77	425.09	612.72
UB3	503.32	382.54	632.71
UB4	200.34	155.21	237.22

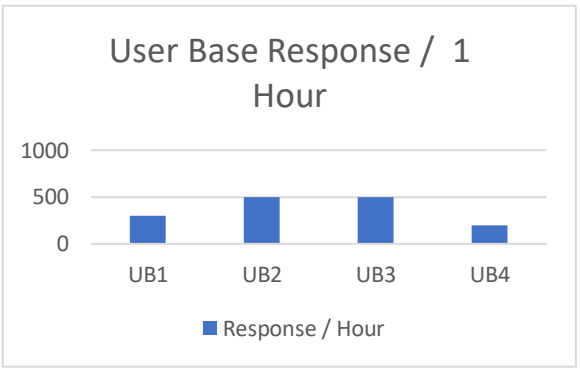

Fig 18. Userbase Response Time per Hour

TABLE 3
DATA CENTER REQUEST SERVICING TIMES

Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	0.38	0.02	0.71

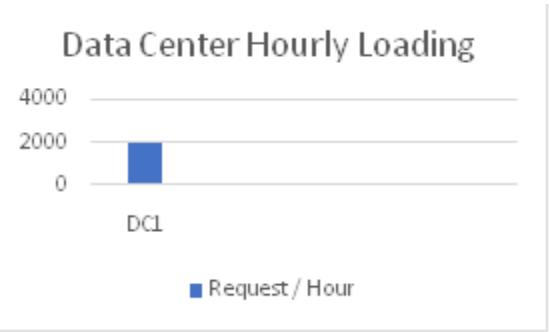

Fig.19 Data Center Request Per Hour (Hourly Loading)

According to simulation results, the data center can process almost 2000 requests in an hour.

Software Testing is necessary before the software is deployed. Various techniques are being used by researchers to conduct software testing [12,13, 14, 15]. After the simulation, we tested our designed systems and applications.

We created a local environment and virtual servers. The designed prototype, related applications, and databases were hosted over virtualized servers at the local area network. Application STA-Vision19 was operated from 15 different client nodes. Evaluation of system and surveys

were done by 15 students of the different level group by 5, including bachelors, masters, and Ph.D. Moreover, after examination, some quality frameworks were considered to evaluate results.

TABLE 4
GENERAL EVALUATION

Parameters	Isolate Application	STA-Vision19
Functionality	Isolate application has limited functions to perform.	STA-Vision19 has a variety of domains and related applications. Multiple functionalities
Resources	Resources are available at one host	STA-Vision19 have Shared resources.
Data Sharing	Isolated has local data sharing, sharing of data within the application boundary.	STA-Vision19 has both local & global sharing. because of a wider boundary, data can be shared with multiple applications.
Extensibility	The structure of the Isolate application is designed for a purpose.	STA-Vision19 can extend the system. More domains and related applications can be added.
Requirements	Application with a defined scope is not capable to come with all requirements of end-user	Varieties of applications are available, STA-Vision19 can come with most of the user requirements.
Memory Sharing	Less amount of memory	Large amount for data can be shared
Device Sharing	Linked with a single server machine	Multiple virtual servers are linked. Capable to work with High-performance Computing (HPC)

Furthermore, the designed prototype is compared with isolate system in the following tables and evaluated parameters for both are summarized among all 10 based sample space:

TABLE 5
FUNCTIONAL TESTING RESULTS

Parameters	Isolate System	STA-Vision 19
Units	8	9
User Acceptance	7	9
Regressions	6	9
System	9	9
Integration	6	10

Graphical summary for table 5 is illustrated as follows in figure 20.

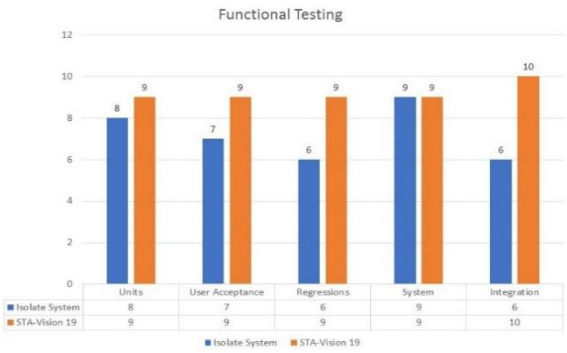

Fig.20. Functional Testing Results

In Table 5 & Figure 20, the system testing result is at the same level in both, while other parameters are at upper levels in STA-Vision19. After functional testing we conducted non-functional testing illustrated as following in table 6:

TABLE 6
NON-FUNCTIONAL TESTING

Parameters	Isolate System	STA-Vision 19
Availability	8	8
Efficiency	6	9

Reliability	8	8
Usability	5	8
Flexibility	4	10
Interoperability	4	10
Portability	7	7
Performance	9	9

Table 6 & Figure 21 illustrate a comparison of isolate systems and STA-Vision in the light of non-functional software testing parameters. Availability, reliability, portability & performance results are at the same level in both systems, while efficiency, usability, flexibility, and interoperability are more in STA-Vision19.

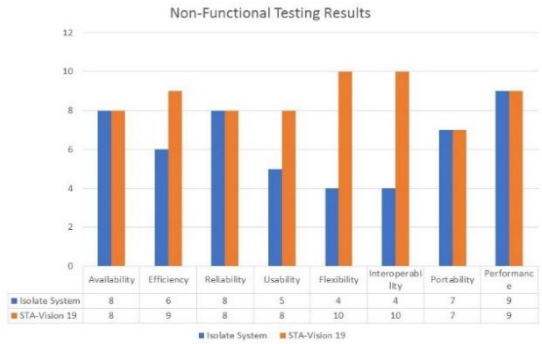

Fig.21. Non-Functional Testing Results

We have also conducted unit testing for STA-Vision19, which is illustrated as follows in table 7.

TABLE 7
REAL-TIME UNIT TESTING

Unites	**Successful**	**Failed**
Sign Up	Yes	No
Sign In	Yes	No
Form & Data Validation	Yes	No
Security & Access Control	Yes	No
Cross-Platform Integration	Yes	No
Crud Operations	Yes	No
Web Services	Yes	No

The result illustrated in table 7 was collected in the real-time environment, it indicates that units of STA-Vision19 are working properly. Furthermore, testing results were calculated according to human response, during the usage of the designed system.

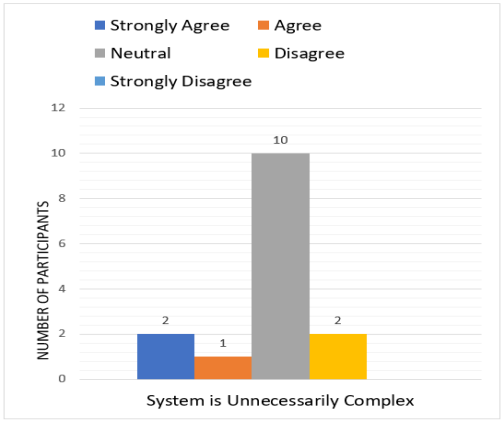

Fig.22 System is Complex

Figure 22 shows that 2 of 15 participants "strongly agreed" with the above statement, 1 was "agreed", 10 were "neutral", while 2 "disagreed".

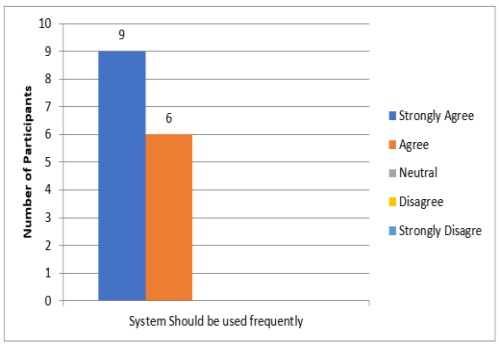

Fig.23. Frequent use of the system

Figure 23 illustrates that, 9 participants out of 15 "strongly agreed" regarding frequent use of the system, while 6 were marked as "agreed".

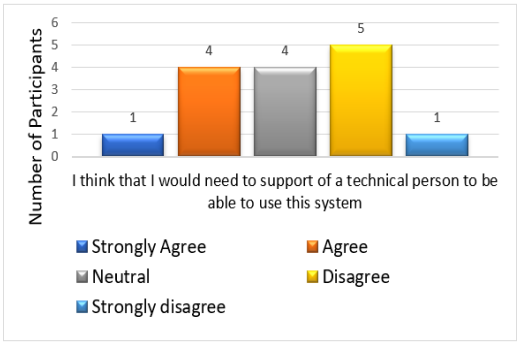

Fig.24. Easy to use

Figure 24 illustrates 7 participants out of 15 "strongly agreed" about the above statement and 8 were marked as "agreed".

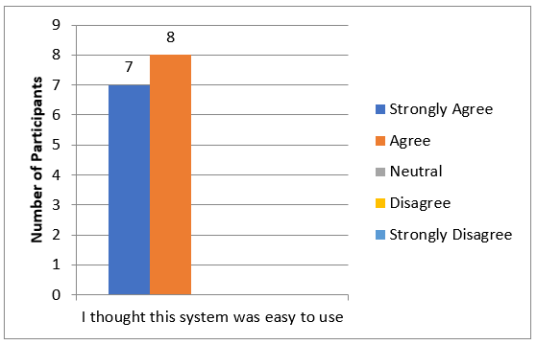

Fig.25. Need for Technical Person

Figure 25 illustrates 1 out of 15 participants "strongly agreed" with the above statement, 4 were marked as "agreed", 4 as "neutral", 5 as "disagreed", 1 as "strongly disagreed".

According to system functionalities, technical support is only needed when an administrator embeds a new application in the system using an administrative panel. Other processes like enterprise account creation, service publishing and consuming are done automatically with dynamic methods.

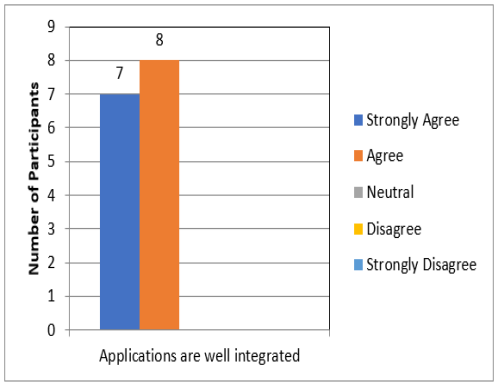

Fig.26. Integration of Applications

Figure 26 illustrates that 7 out of 15 participants "strongly agreed" about the statement that "Applications are well integrated", while 8 of them were marked as "agreed".

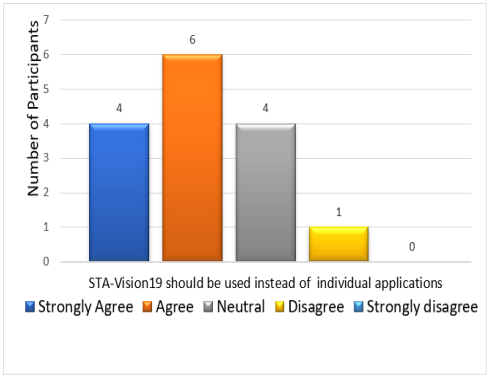

Fig.27. Use of the proposed system

This question was asked at the end of the evaluations. Figure 27 illustrates that, 4 participants out 15 "strongly agreed" with the above statement, 6 were marked as "agreed", 4 as "neutral" and 1 as "disagreed".

Conclusion

In this study, our focus was to propose a novel SOA and Cloud-based framework STA-Vision19 to bring multiple domains, related applications, and services over a virtual, collaborative and integrated environment. Our work proposes a new platform STA-Vision19 and Service Integration Bridge (SIB), a basic prototype of that system along with two sub-applications "Blog Application" & "E-Learning" that were implemented to perform experiments. The technical concerns like service publishing, consuming, trust mechanism and load balancing were discussed in this research. At the end of the study, designed methodologies were tested in a real-time environment and results were found that indicate: In STA-Vision19 two or more applications can be integrated and work in a collaborative environment. Different domains and sub-applications can be added to STA-Vision19 that enables clients to access desired services. Moreover, simulation, testing & survey results illustrate that the proposed system is much more beneficial to use as compared to isolated systems because it can support flexibility and interoperability among the applications which are heterogeneous by nature.

Future Work

As it is described above, we have implemented the basic structure illustrated in figure 2and have also implemented two distinct applications E-learning and Blogging System to validate integration among applications as our goal is to complete the whole project. Furthermore, applications will be implemented in the future.

We will extend the STA-Vision19 network by merging other networks with it. The design and management of a huge network will be challenging.

References

[1] Chantal (July 19, 2012) Centralized vs Decentralized organizations.

[2] Swati Gupta, A Study on Distributed Database System, Department of Computer Science, Amity University Haryana, Gurgaon, Haryana, INDIA, available at Http:// ib2012itgs.wikispaces.com/file/.../ITGS%20databasde %20homework.pdf. (IJETCAS)

[3] Addison-Wesley, HohpeG. and B. Woolf (2004) Enterprise Integration Patterns, Designing, Building, and Deploying Messaging Solutions.

[4] Robert S. Chen(2005) The GEO Data Sharing Challenge, Putting Principles into Practice, Center for International Earth Science Information Network, The Earth Institute, Columbia University, Palisades, New York, USA.

[5] Dror G. Feitelson, Eitan Franchtenberg, and Kent L. Beck (July 2013) Development and Deployment at Facebook, Hebrew University.

[6] D. G. Feitelson(Apr 2012) Perpetual Development, A Model for Linux Kernel life cycle.

[7] Charlene Croft (Dec 18, 2007) A Brief History of The Facebook.

[8] Mukhtiar Memon et al (Aug 2012) Towards an Approach of Agricultural Virtual Organizations (AgriVOs) for Sharing and Management of Scientific

Research Data, Information Technology Center, Sindh Agriculture University, Tandojam, Pakistan, (IJERT).
[9] Stephanie Lang et al (Dec 2012) Amazon Offering Everything from A to Z, The Robin School of Business, University of Richmond.
[10] Ayman G. Fayoumi (2018) From Service-Oriented Architecture to Cloud Computing, Global Journal of Computer Science and Technology: B Cloud and Distributed, Volume 18 Issue 1 Version 1.0.
[11] Ebrahim Shamsoddin-Motlagh (2012) A Survey of Service Oriented Architecture System Testing, Computer Engineering Department, Faculty of Engineering, Science & Research Branch, Islamic Azad University, Tehran, Iran, International Journal of Software Engineering & Applications (IJSEA), Vol.3, No.6, November.
[12] Mei, L., Chan, W.K., Tse, T.H. (2008, May 10-18). Data Flow Testing of Service-Oriented Workflow Applications, ICSE '08, Leipzig, Germany, 371-380. DOI: 10.1145/1368088.1368139.
[13] Lertphumpanya, T., &Senivongse, T. (2008, May) Basis Path Test Suite and Testing Process for WS-BPEL. WSEAS TRANSACTIONS on COMPUTERS. ISSN: 1109-2750, Issue 5, Volume 7, 483-496. Retrieved from www.wseas.us/e-library/transactions/computers/2008/26-156.pdf.
[14] Yuan, Y., Li, Z., Sun, W. (2006). A Graph-search Based Approach to BPEL4WS Test Generation. Proceedings of the International Conference on Software Engineering Advances (ICSEA'06). Doi: 10.1109/ICSEA.2006.261270
[15] Bartolini, C., Bertolino, A., Elbaum, S., Marchetti, E. (2011). Bringing white-box testing to Service Oriented Architectures through a Service-Oriented Approach. The Journal of Systems and Software, 84, 655–668. Doi: 10.1016/j.jss.2010.10.024.
[16] John Jung WoonYoo., KrishnaveniGnanasekaran., Chen-Yang Cheng (2014). A Collaborative Healthcare Service Framework and Performance Evaluation., American Journal of Industrial and Business Management, 2014, 4, 274-283.
[17] Swati Gupta (2015). A Study on Distributed Database System., International Journal of Emerging Technologies in Computational and Applied Sciences, 12(2), March-May 2015, pp. 185188.
[18] Yan Cao, Yan Chen and Yiting Shen (2007) The Research and Application of Web Services in Enterprise Application Integration.
[19] Feng Yu1*, Casey Stella2, Kriss A. Schueller (2014)., A Design of Heterogeneous Cloud Infrastructure for Big Data and Cloud Computing Services., OPEN JOURNAL OF MOBILE COMPUTING AND CLOUD COMPUTING Volume1, Number2, November.
[20] Abdulsalam Ya'uGital, Abdul Samad Ismail, Min Chen, and Haruna Chiroma (March 12 – 14, 2014) A Framework for the Design of Cloud-Based Collaborative Virtual Environment Architecture, Proceedings of the International MultiConference of Engineers and Computer Scientists 2014 Vol I, IMECS 2014, Hong Kong.
[21] Suhail Madoukh and Rebhi Baraka (January 2014) A SOA-Based e-Government Data Integration., International Arab Journal of e-Technology, Vol. 3, No. 3.
[22] Behal V. and Kumar A (2014) Comparative Study of Load Balancing Algorithms in Cloud Environment using Cloud Analyst, International Journal of Computer Applications, Vol. 97, No.1,201, 2014, pp 36-40.

Declarations:

Author's Information:

Syed Taimoor Ali received his BSIT(Hons) & MSIT(Hons) from Sindh Agriculture University Tandojam, Pakistan. He is currently pursuing a Ph.D. Computer Science & Engineering at Central South University, Changsha, China. His research interests include Software Engineering, Web Technology, Cloud Computing, Service Computing.

Availability of data and materials:

The implemented system has confidential data that cannot be shared. After publication,theproject data will be hosted at remote server of an authorized organization.

Competing interests:

I declare that authorhas no conflict of interest.