Managing Data Location in Cloud Computing
Evaluating Data Localization frameworks in Amazon Web Services

Vijay Kokkonda
Krishna Sandeep Taduri

School of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona
Sweden
This thesis is submitted to the School of Engineering at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 30 weeks of full time studies.

Contact Information:
Authors:
Vijay Kokkonda
Address: LGH 542, Annebo 3A, Gamla Infartsvagen, Karlskrona, Blekinge, Sweden
E-mail: vijaykokkonda89@gmail.com

Krishna Sandeep Taduri
Address: LGH 518, Annebo 3A, Gamla Infartsvagen, Karlskrona, Blekinge, Sweden
E-mail: sandeep.jusrock@gmail.com

University advisor:
Dr. Mikael Svahnberg
Senior faculty member in Software Engineering Research Lab (SERL)
School of Computing

School of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona
Sweden

Internet : www.bth.se/com
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57
Abstract

Context: Cloud Computing is an emerging technology where present IT is trending towards it. Many enterprisers and users are still uncomfortable to adopt Cloud Computing as it uncovers many potential and critical threats which remind the most concerned security issue to consider. As data is migrated between different data centers dispersed globally, security for data is a very important issue. In Cloud environment Cloud user should be aware of the physical location of the data to ensure whether their data reside within certain jurisdiction as there are different data privacy laws. Evaluating different data localization frameworks in Amazon Web Services by deploying web application in Amazon availability zones (US, Europe and Asia) is the main context of this study.

Objectives: In this study we investigate which strategic data localization frameworks have been proposed, can be used to identify data location of web application resource deployed in Cloud and validate those considered three frameworks by conducting experiment in a controlled environment.

Methods: Literature Review is performed by using search string in data bases like Compendex, IEEE, Inspec, ACM digital Library, Science Direct and Springer Link to identify the data location frameworks. Later these data location frameworks are evaluated by conducting a controlled Experiment. Experiment is performed by following the guidelines proposed by Wohlin, C [66].

Results: Three data localization frameworks out of ten, obtained from literature study are considered for the evaluation. The evaluation of these frameworks in Amazon Web Services resulted that replication of three data localization studies is possible, predicting the location of data in US, Europe and Asia close enough accurate and the factors considered from the frameworks are valid.

Conclusions: We conclude that from the identified ten frameworks, three data location frameworks are considered for evaluation in which one framework allows verifying the location of data by trusting the information provided by cloud provider and second framework is to verify the location of cloud resources without need of trusting cloud provider, finally the third framework is to identify the replicated files in cloud however this framework also does not need trusting the Cloud provider. These frameworks address the data location problem but in a different way. Now the identified three frameworks are validated by performing a controlled experiment. The activities performed from the three frameworks in the experiment setup allow identifying the data location of web application deployed in US, Europe and Asia. The evaluation of these frameworks in Amazon Cloud environment allowed proposing a checklist that should be considered to manage the web application deployed in cloud regarding data location. This checklist is proposed based on the activities performed in the experiment. Moreover, authors conclude that there is a need for further validation, whether the proposed checklist is applicable for real Cloud user who deploys and manages Cloud resources.

Keywords: Cloud Computing, Data Location, and frameworks
ACKNOWLEDGEMENT

Praise to Lord Krishna, The Almighty, whose blessings and kindness made us to complete the thesis.

We admire the industrious support of our supervisor Dr. Mikael Svahnberg, without his guidance and contribution the entire work seemed to be journey in ocean and his presence guided us like a map. It was through the assistance of our supervisor that enabled us to bring the tacit idea in to existence.

We express our gratitude to our beloved family members, especially our parents for providing their support and prayers throughout our efforts.

We would also like to thank our loving friends and well-wishers for their relentless support in extending their cooperation, knowledge and experiences with us that helped us to reach our goals.

Vijay Kokkonda,
KrishnaSandeep
# TABLE OF CONTENTS

ABSTRACT ........................................................................................................................... I
ACKNOWLEDGEMENT ...................................................................................................... II

1. INTRODUCTION ........................................................................................................... 1
2. THEORETICAL BACKGROUND ................................................................................... 2
   2.1. OVERVIEW OF CLOUD COMPUTING ................................................................. 2
   2.2. IMPORTANCE OF DATA LOCATION IN CLOUD COMPUTING ......................... 4
   2.3. DATA LOCATION PRIVACY LAWS ...................................................................... 5
   2.4. IDENTIFYING DATA LOCATION IN CLOUD COMPUTING ............................... 5
3. RESEARCH DESIGN & EXECUTION ............................................................................. 6
   3.1. AIMS AND OBJECTIVES ....................................................................................... 6
   3.2. RESEARCH QUESTIONS ......................................................................................... 6
   3.3. RESEARCH METHODOLOGY ............................................................................... 6
4. LITERATURE REVIEW .................................................................................................... 8
   4.1. STRUCTURE OF LITERATURE REVIEW ................................................................ 8
   4.2. SEARCH STRING & DATA EXTRACTION .............................................................. 8
   4.3. STUDY SELECTION CRITERIA ................................................................................. 9
      4.3.1. Study selection process .................................................................................. 9
   4.4. DESCRIPTION OF DATA LOCATION FRAMEWORKS .......................................... 10
   4.5. ANALYSIS OF CONSIDERED PRIMARY STUDIES ............................................... 12
5. BRIEF OVERVIEW OF AMAZON WEB SERVICES ..................................................... 13
6. EXPERIMENT ................................................................................................................ 15
   6.1. DEFINITION ........................................................................................................... 15
   6.2. PLANNING .............................................................................................................. 16
      6.2.1. Context Selection ......................................................................................... 16
      6.2.2. Hypothesis Formulation .............................................................................. 17
      6.2.3. Variable Selection ....................................................................................... 17
   6.3. DESIGN ................................................................................................................... 18
      6.3.1. Design Type .................................................................................................. 18
      6.3.2. Lab Environment .......................................................................................... 19
      6.3.2.1. Instrumentation ....................................................................................... 19
   6.4. OPERATION ............................................................................................................. 20
      6.4.1. Preparation (Experiment Environment Details) ............................................. 20
   6.5. EXPERIMENT -1 .................................................................................................... 21
      6.5.1. Framework – 1 Description ......................................................................... 21
      6.5.2. Experiment setup by replicating framework 1 ............................................. 22
      6.5.3. Execution ....................................................................................................... 23
      6.5.4. Data Analysis ............................................................................................... 27
      6.5.5. Weakness of Framework ............................................................................. 27
   6.6. EXPERIMENT -2 .................................................................................................... 28
      6.6.1. Framework – 2 Description ......................................................................... 28
      6.6.2. Experiment Set-up for replicating framework 2 ........................................... 28
      6.6.3. Execution ....................................................................................................... 30
      6.6.4. Identifying the data location of the deployed web application in AWS servers .................................................................................................................. 35
      6.6.5. Analysis of framework 2 whether satisfying in our experiment setup ........... 35
      6.6.6. Weakness of Framework ............................................................................. 36
   6.7. EXPERIMENT -3 .................................................................................................... 36
      6.7.1. Framework – 3 Description ......................................................................... 36
      6.7.2. Experiment Set-up for replicating framework 3 ........................................... 38
      6.7.3. Execution ....................................................................................................... 38
      6.7.4. Estimating the data location of the deployed web application in AWS .......... 38
      6.7.5. Estimating the geolocation by comparing latency values with haversine distance .................................................................................................................. 39
      6.7.6. Creating CloudFront nodes to the deployed web application ...................... 40
      6.7.7. Analysis of framework 3 whether satisfying in Experiment setup ................. 42
      6.7.8. Weakness of Framework ............................................................................. 43
1. INTRODUCTION

Cloud Computing is an innovative Information System architecture which is emerging in Information Technology field. Many users and IT technologies are shifting towards this trend as their data is growing bit by bit and wants to maintain their data on remote servers (Clouds), which can migrate from place to place that can be able to access via network. Now a day’s many small and medium business companies are increasingly realizing the benefits [13] [92] of the Cloud and are simply stepping into the Cloud environment. By using Cloud technology companies can simply gain fast access to the best business applications, dynamically boost their infrastructural resources at minimal costs.

Cloud computing in its essence has the capacity to point out the deficiencies of the traditional computing due to its unique characteristics but adopting this innovative technique may uncover many potential and critical threats which remind to consider the most concerned security issue. Due to its improved efficiency and reduced costs this service got a lot of popularity in the market. Authors in [93] [94] found that many enterprises and users are still uncomfortable with the Cloud delivery models due to the lack of visibility of the way data is stored and secured. There are many key security elements that need to be considered as an integral part of the Cloud development and deployment process which are like a) Data Security b) Network Security c) Data Localization d) Data Integrity e) Data Segregation f) Data Access g) Authentication and Authorization h) Data Confidentiality i) Web Application Security j) Data breaches k) Virtualization Vulnerability l) Availability m) Backup n) Identity management and sign-on process. Among the issues/challenges that are faced while using Cloud services, Data Localization (or) Data Visibility (or) Data Transparency is the important issue that needs to pay utmost attention for public cloud and private cloud [74][95].

Many research [95] [96] [97] done on security issues in relation to IT industry found that, data localization issue becomes a big problem for enterprises than normal users. Generally normal users do not worry about their data storage location because their data is not much confidential when compared to enterprise(s) they store their confidential data in Cloud by trusting the service provided by the Cloud Provider.

Data Localization is having high priority which is defined as “awareness of the data presence to the user which is distributed remotely across different Cloud zones due to compliance and data privacy laws in various countries” [98]. Feeling of data insecurity generally arises within the minds of enterprise or organization where they do not know if their data is safe without being modified or stolen from the Cloud and also they want to know what data laws are applicable if their data is stored in the region/country of strict data privacy laws [74]. The most important aspect for many enterprise architectures is that every Infrastructure as a Service (IaaS) model must be capable of providing security and reliability to the customer on the location of the data [98]. Data Localization stood on the second position from both users and enterprise point of view by empirically analyzing security and user experiences issues who are participating in Finnish Cloud software program consortium [99].

Different countries have different data privacy laws and for example in UK the laws are framed so that personal information of their country citizens should reside within their country boundaries. Also in European Union and USA, the data privacy laws differ from each other by restricting the data to reside and access from other countries [74]. Cloud provider, in order to reduce the costs and benefits may transfer the restricted data to other sites [74]. At present to overcome this issue many providers started providing the facility of choosing their own geographical location to store their data by expanding their locations to other parts of the world and stating legal and regulatory benefits to this facility [74][96]. As a preventive measure Amazon Web Service [37] implemented S3 (Simple Storage Service) and Microsoft Windows Azure stated legal and regulatory benefits for specification of the regions to allow the customers to choose their own geographical regions for storage and location of the data.

The main challenge of this study is to evaluate the existing different frameworks that allow managing and controlling Cloud resources regarding Data location. The rest of the sections are, Section 2 is Theoretical Background where overview of the Cloud Computing, its characteristics, importance of the security in Cloud Computing is discussed, this section also reveals the importance of data location in Cloud computing. Section 3, is Research Methodology where authors clearly mentioned the research questions for this thesis and the Methodology used in order to answer the research question is also discussed. Section 4 is Literature Review where authors identified the studies on verifying data location in Cloud computing. A clear description of the research done on data location is also provided. In this section authors considered three types of data location frameworks that need to be evaluated. Before evaluating the frameworks, Section 5 discusses a brief overview of Amazon Web Services which
is used in the Experiment. Section 6 is Experiment where authors performed a controlled experiment in Amazon Cloud environment to validate the data location frameworks. By replicating the data location frameworks authors conclude that the evaluated three frameworks are valid measures. After validating three different types of studies in a controlled experiment, in Section 7 a checklist for managing data location of web application deployed in Cloud Computing is proposed by considering the activities performed in experiment. Finally in Section 8 Summary and conclusion of this study is discussed.

2. THEORETICAL BACKGROUND

Before we dive into the Research Methodology and the main issues of this thesis, a detailed explanation is required about the Cloud Computing, its characteristics and data location issues that effect in Cloud Computing environment.

2.1. OVERVIEW OF CLOUD COMPUTING

Cloud Computing is an internet based development (Cloud) and use of computer technology (computing) where IT-related capabilities are provided as a service that allows users to access technology enabled services from the internet [82][84]. National Institute of Standards and Technology (NIST) [83] defined cloud computing as “Cloud computing is a model for enabling ubiquitous, convenient, on demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction ”[83].

![Figure 1: Paradigm Shift in IT [85](image)](image)

The above Figure 1 shows the trend towards Cloud Computing [85]. Figure 1 shows that the commoditization and the centralization of IT services have increased year by year. Previously many organizations used to have their IT within their own server rooms which is On-premise. But later on the servers were shared with other businesses in Shared Service Centers (SSC) then the trend of IT changed to hosting applications on external platforms and outsourcing/off-shoring of IT departments to low-wage countries. Many major companies like Amazon, IBM, Microsoft, Yahoo, have more than hundreds of thousands servers. Cloud with the large scale feature enlarges the user’s computing power [91]. According to the Gartner Hype cycle [86] the year 2011 is called as ‘the year of cloud computing’. It is called so because many organizations started using the phenomenon Cloud Computing very much to enhance their business needs. The road towards the Cloud Computing was started in late 1980’s which was Grid Computing. In Grid Computing many large systems were applied to single (hard, scientific) problems [88] [89].

Coming to service models Cloud Computing on a whole is classified into three different models like Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS) [16] [90]. Software as a Service (SaaS) facilitates the user to use the provider’s application service which runs on the Cloud infrastructure with the help of user interface (like web service or web based email). Here the user cannot control or access the underlying infrastructure which includes network servers, operating systems and storage but can do some configuration settings to some extent. Examples are like Google Docs, Mobile Me, and Zoho. Platform as a Service (PaaS) facilitates the user to deploy onto the Cloud infrastructure where they can create, acquire and produce the applications by using programming languages and tools provided by the provider. Like SaaS, user cannot control or manage the Cloud infrastructure. Examples are like Microsoft Azure, Force.com, and Google App Engine. Infrastructure as a Service (IaaS) facilitates the user to deploy and run discrete absolute software which includes operating systems and
applications. Users in this service can use and control processing, storage networks, other computing resources and other applications deployed. Examples are like Amazon S3 [57] and Sun’s Cloud service.

The deployment models in Cloud Computing are four types [90]. Public Cloud (external) enables its infrastructure to make available to general public or large industry group where the services provided by the provider are used by any organization and not proprietary to one organization [90]. Private Cloud (internal) is operated for a private organization where the Cloud may exist within or out of premises of the organization which is managed by a third party or organizations [90]. Community Cloud has its infrastructure shared by several organizations and supports a community that has concerns that are mutually shared between them like mission, security requirements and policies [90]. This Cloud is managed by the organizations or the third party and the cloud here may or may not be in the boundaries of the organization and Hybrid Cloud is a mixture of two or more clouds (i.e. private, community or public) and their combination is unique [90]. Here the cloud uses standardized or proprietary technology to enable the data and application portability. Cloud computing has different key characteristics [83] [87] [91] which are On Demand Self Service, Large-Scale, Versatility, Inexpensive, Resource Pooling, Broad Network Access, Measured Service, Multi-tenancy and Service oriented.

**Importance of Security in Cloud Computing**

Cloud computing and web server(s) both run on a network structure which are easily prone to network attacks. Distributed Denial of Service Attacks (DDoS) is one of the main network type attacks which means, when a user (say hacker) hijacks the server, the hacker could stop the web services from functioning and demand money to put the services back online [22]. Another attack is Man in Middle attack if the secure sockets layer (SSL) is incorrectly configured then the client and server may not behave properly as expected. So security is very important issue in cloud computing, as putting your data, running your software at someone else's hard disk using someone else's CPU appears scaring to many people who uses the Cloud Computing. Morrel et al. in article [53] gave some suggestions about how to get the cloud security right way. The suggestions are a) a company should understand about the storage requirements associated with the Cloud Computing b) should always be aware of the situations while moving to the Cloud. c) It is very important to note that data privacy guidelines by the European Union, or the Dutch guidelines ‘Wet Bescherming Persoonsgegevens’, are very specific and these are to be followed strictly when it comes to the responsibilities of data controllers when engaging with Cloud providers. Similarly different countries have different data location privacy laws [39]. Privacy and security are the main areas that have become the main focus of attention for many researchers in recent times [41] [48] [49] [50] [51]. It is because the cloud computing environment uses the internet as a communication media which brings a lot security issues and challenges to the sensitive data when the data is outsourced to be shared on the servers globally, which are not within the same trusted domain as data owners.

There are many issues in Cloud Computing which are discussed in detail by Kulkarni et al. in the article [33]. Some of the Service Provider (SP) Security Issues are: a) Identity and access management has the features like Authorization, Authentication and Auditing of the users accessing the Cloud [33]. b) Privacy is one of the important security issues in Cloud Computing. For a cloud provider dealing with personal information is really a tough job as the personal information regulations are strictly implemented and they vary from one country to another whether it is stored outside the country. As for the SP every single level of service is important and acceptable and he has to make proper contractual commitments of where the data should be stored in which country or region. So SP has to take care about this issue and implement effective assessment strategy for data protection, compliance, privacy, identity management, secure operations, and other related security and legal issues [33]. c) Cloud provider has to face problems during data transmission if the data is not properly encrypted [33].

The Cloud Infrastructure Provider (IP) who provides resources available to SP, to deploy and manage resources in Cloud also has some security issues that should be considered [34] a) In Cloud storage infrastructure there will be sensitive data present which has to be protected and isolated from others. It involves the way in which data is accessed and stored, audit requirements, compliance notification requirements, issues involving the cost of data breaches, and damage to brand value. There are many encryption tools to protect data privacy and manage the compliance in Cloud that can be used by the Cloud provider [34]. b) Virtual servers and applications exist as non-virtual counterparts; they have to be properly secured both physically and logically in the IaaS clouds. Especially in hybrid clouds, preventing holes or leakages between composed infrastructures could be of a major concern which thereby increases the complexity and diffusion in responsibilities. Infrastructure provider needs to be care with the
issues related to network and server as the Cloud purchasers will be interested to form certain agreement that every tenant domains are properly isolated that no probability exists for data or transactions to leak from one tenant domain into successive [34]. In Cloud infrastructures Loss of Governance becomes a problem as the client handovers control to SP on a number of issues which may affect the security [34]. At the same time the Cloud provider may take it as an easy way to violate the commitments and promising’s offered in the SLA’s to render the service as agreed before on the part of the cloud provider. These security issues should be considered by IP, SP and End User. Among these security issues this research study mainly concentrates on Data location issues of the Cloud resources.

Service Level Agreement (SLA) is a description of the agreed service, service level parameters, guarantees, actions and remedies for all cases of violations [64] [81]. SLA [64] [81] is an important element that provide a degree of assurance for CP’s and users. It defines the scope of usage and provision of resources. Cloud consumers need SLA to provide certainty regarding the resources provided and the ability to reach the desired level of productivity. Cloud providers need SLA to define the trust and quality of services they provide to the users as well as framework agreed to be rendered on cost and charges. M.Alhamad et al. in [64] specified the SLA parameters in terms of metrics which defines how the cloud service parameters can be measured and specify values of measurable parameters for the cloud service models. Here the availability of the resources is also included in the parameters of Cloud delivery models [64]. In addition to this data location legislation of the Cloud resources should also include in SLA [41].

2.2. IMPORTANCE OF DATA LOCATION IN CLOUD COMPUTING

Cloud computing has grown as a promising computing paradigm as it guarantees measurable and nonfunctional performance metrics like service availability or throughput. But it still lacks adequate mechanisms for guaranteeing certifiable and auditable security, trust and privacy of the application and the data they process [59]. Some important requirements like limiting the usage of submitted data only to the intended users by the submitting user or storage of data in certain geographical regions cannot be guaranteed all the time. It is also one of the most important common compliance issues faced by the organizations [41].

Some events that occurred recently are, the customers of the Amazon’s EC2 cloud service lost their data permanently [22] and also Hotmail users were upset when it appeared as though their accounts had been wiped [3]. Another incident is that in February 2011, 150,000 Gmail accounts were ‘accidentally reset’ by Google, leaving the users with all their previous messages wiped out from their mail [27]. The client has to make sure and know that the services paid for are the services provided as there is a risk that in order to cut costs, the service provider might make the data highly available and supply resources elasticity [72][73][74]. The cloud provider may claim one level of service but actually provide an inferior level. Every client has to know the information about not just the file is being stored but also to know where the data is being replicated, onto multiple disks distributed across different geographic locations [3] and also the location of data because different countries have various data privacy laws [3]. This is very useful for the advertising companies to verify that their ad content is displayed to a geographically diverse demographic [3].

Customer demands regarding data location of Cloud resources

Cloud customer should demand Cloud provider about the physical location of the data and whether the data reside under certain jurisdiction [11]. Some of the important requirements that Cloud Customer should demand are as follows:

• The Location of the Cloud Service provider should be known [11].
• Whether the Cloud Provider applies their own resources or uses other Cloud Provider resources [11] [74].
• The physical location of the Cloud resources created should be visible [11].
• Whether the data location jurisdiction are followed by Cloud provider [11].
• Whether the Cloud resources are outsourced to different countries [11].
• Whether the Cloud resources are maintained by other subcontractors [11].
• The procedure of collection, transfer and deletion of the Cloud resources should be understandable and clearly visible to the Cloud customer [11].
• The details of the terminated data in Cloud resources should also be encountered [11].

This section reveals the data location important issues that Cloud resources should be visible to the Cloud Users. However there are few studies that allow clients to verify the data location in Cloud.
The Expert group report [68] mentioned a number of issues in which the concerns over security with respect to valuable knowledge, information and data placed on an external service are given first priority. To provide Cloud characteristics of location independent Virtualization is used. Location independent is that customer generally doesn’t have control on the exact location of the resources instead they specify location of data present in country, state or data centers [31]. The data location privacy laws of few countries are briefly described in Section 2.3.

### 2.3. DATA LOCATION PRIVACY LAWS

*Compliance* is one of the most important privacy and security related issue that has long term significance not only for public cloud computing but also in many other Cloud computing service models [39]. *Compliance* refers to an organization’s responsibility to operate in agreement with established laws, regulations, standards, and specifications [39]. There are various types of security, privacy laws and regulations exist within different countries at the national, state, and local levels, making compliance a potentially complicated issue for Cloud Computing. A detail discussion on the rules and privacy laws are discussed below.

The National Conference of State Legislatures reported that at the end of 2010, 46 states have enacted legislation governing disclosure of security breaches of personal information and among them 29 states have enacted laws governing the disposal of personal data held by business and/or government [39]. However well-known cloud products like Amazon EC2 [38] and Goggle app engine [61] are providing some basic security, privacy and trusted mechanisms, but these are not customizable at every case. In general Cloud Computing offers high degree of data mobility in which users do not always want to know the location of their data [74]. This may not be much of importance as for example a user who uploads photos to Facebook does not bother about the stored location or in which region their files are situated at particular time, but for organizations who has sensitive data stored in cloud want to know the location of their data because there are many data privacy regulation acts present for each country [70] [71] (ex. US, UK, France, Netherlands, Germany and many European countries) and if their data is migrated to the country that has strict laws, leakage of data or modification of data might take place which leads to severe loss for the organization [1].

The general characteristic of Cloud Computing service is that the data is stored redundantly in multiple physical locations and detail information about the organization’s data or the personal data of the user cannot be known or unavailable at that moment and also the information is not disclosed by the SP. This situation makes it difficult to ascertain whether necessary safeguards are in place or whether legal and regulatory compliance requirements are met or not. When information crosses the borders the governing, legal and regulatory laws can be of uncertain and raise a variety of concerns [42] [43].

The main compliance concerned is whether the laws in the area where the data is collected permit the flow and whether the laws apply to the data post transfer and also whether the laws at the destination present any additional risks or benefits. For example, European data protection laws may impose additional obligations on the handling and processing of data transferred to the U.S [44]. European laws demand that organizations should be aware of the location of data stored in cloud all the time [3] and the data should be located within the country. Some of the clients were not aware how it affects when their data is deleted from application level or infrastructure level. Restricting some of the geographically area reduces the availability and increases the cost, Europe cloud service providers costs more than US for hosting their data due to high electricity costs [10]. According to IT company Scale up technologies Germany is the country with most restricted data privacy laws in the world [1].

### 2.4. IDENTIFYING DATA LOCATION IN CLOUD COMPUTING

The study of data location issues in Cloud Computing shows that Cloud user who deploys and manages the Cloud resources should be aware of the physical location of the Cloud resources provided by IP. Research on verifying the data location of the Cloud resources addressed different techniques, frameworks, schemes and architectures (Authors in this study considered techniques, architectures, schemes that are proposed to verify the data location of Cloud resources as ‘frameworks’ for identifying data location).

Cloud Computing need more importance towards security as it has become a problem for both enterprises and normal users to secure sensitive information from being modified or altered by unauthorized persons. Z. Mahmood in [74] discussed the different data security issues like data privacy, data protection, data availability, data location and data transmission has to be more concentrated. Due to the increase in Cloud computing virtualization, the
control of physical location of the Cloud resources is an important aspect which allows verifying the data location by implementing different techniques [100].

Cong Wang et al in [76] proposed a scheme which satisfies the integration of storage correctness and recognize data error localization. The proposed scheme is effective but this scheme is weak in investigating data error localization due to lack of achieving public verifiability and storage correctness in dynamic data [103]. However this framework allows identifying error localization but not the actual location of Cloud resources.

Currently cloud infrastructures are having opaque service offerings where customers cannot see or monitor the underlying physical infrastructure. When federated cloud infrastructures move across the borders of different countries, data migrate from place to place and it should be possible for the data owner to monitor the location of the data. Massonet et al. proposed a framework [1] which shows how the existing federated cloud monitoring infrastructure is used for data location monitoring without compromising the cloud isolation. The proposed logging architecture is capable of tracking the entire set of Cloud operations [1]. It shows the end users how Cloud IP’s manage their data and the execution location of this data during the entire lifecycle. However there is a need for rigorous validation of the proposed architecture [1], this motivated to evaluate the data location framework. In addition to this, it is challenging to distinguish different types of techniques on identifying data location and validating the efficiency of those techniques [1].

3. RESEARCH DESIGN AND EXECUTION

In this section the main aim of the thesis and the objectives to achieve the desired aim are described. In addition to this, the research questions formulated to achieve the aim are also described.

3.1. AIMS AND OBJECTIVES

The main aim of our research is to identify and evaluate the efficiency of Data Localization frameworks in Cloud Computing.

In order to achieve this aim the Objectives/Sub-goals are as follows:

- Identifying the architectures and frameworks for managing data location of web application deployed in Cloud.
- Conducting a controlled experiment by deploying a web application in Amazon Web Services and test Data Localization framework(s).
- Validating the efficiency of data location frameworks in experiment setup.

3.2. RESEARCH QUESTIONS

In order to achieve the aims and objectives the research questions formulated are as follows:

**RQ 1:** What are the existing frameworks that allow identifying data location of web application resources in Cloud Computing?

**RQ 2:** How efficient are the DL frameworks in identifying where data and services are stored and run?

The results of our research gives a clear knowledge about how efficient are the existing frameworks in identifying the Data Location when web application is deployed in AWS. And also facilitates in preparing a checklist to identify data location when a web application is deployed in Cloud. This checklist is prepared by considering the efficient factors that satisfy by evaluating three different frameworks in experiment setup.

3.3. RESEARCH METHODOLOGY

This research is a mixed methodology that consists of both qualitative and quantitative methodologies. For answering RQ 1 Literature Review (LR) is used as qualitative methodology and for RQ 2 authors conducted controlled experiment as a quantitative method.

In **Qualitative methodology** Literature Review is performed in order to identify existing frameworks that allow verifying the Data Localization in Cloud Computing. Though Data location in Cloud computing is addressed, authors followed some of the general principles that should be followed for systematic review in order to identify the existing studies effectively. However the main aim is to validate the existing data localization frameworks which are performed by conducting quantitative research. The considered frameworks for validation generalize the existing techniques for some extent.
In *Quantitative methodology* authors conducted a controlled experiment within a closed environment by using Amazon Web Services as Cloud environment and followed a systematic approach (i.e. considering the factors from frameworks) to evaluate the identified three data localization frameworks. To validate the existing data localization frameworks authors performed this experiment by following the guidelines of Wohlin, C et al. in [66] and conducted the experiment.

So replicating different data localization frameworks and evaluating it requires conducting a controlled experiment. Comparing various treatments like process, models, methods, techniques etc. by conducting a randomized or quasi experiment in which one or more software engineering tasks are performed is defined as Controlled experiment [46]. Wohlin, C et al. in [66] described few characteristics to perform experiment, these characteristics relating to the study is described as follows:

- To confirm that the existing data location frameworks are valid.
- Evaluate the accuracy of the data location frameworks.
- To validate that factors considered in the experiment are performing according to the framework.

The strength of the experiment is validating different data localization frameworks in a controlled environment. This quantitative methodology is conducted in a real cloud environment, i.e. Amazon Web Services [38]. *Figure 2* shows the two phases of the Research Methodology where Phase I is conducting Literature review to identify different existing data location frameworks. Now in Phase II the identified data location frameworks are evaluated by conducting a controlled experiment then results of the evaluation is discussed in final analysis and discussion.

*Figure 2: Brief Overview of Research Methodology*
4. LITERATURE REVIEW

To identify the techniques, frameworks, architectures that allow verifying the location of cloud resources this literature study is conducted. The structure of the literature study shows how authors identified the data location frameworks and considered for evaluating it in the experiment setup.

4.1. Structure of Literature Review

![Diagram of Literature Review Flow]

The above Figure 3 represents the structure of the Literature Study. Firstly authors created search string and used it in databases like Inspec, IEEE, ACM Digital Library, Springer link, Compendex, Inspec, Science Direct and Google Scholar. By analyzing the articles using inclusion exclusion criteria authors considered the articles that solve the issues of data location in Cloud Computing. Authors finalized the data location frameworks that need to be evaluated which helps in verifying data location of web application in Cloud Computing. Security issues should be considered when deploying web application in Cloud [74]; however identifying the data location of the web application resources in Cloud environment is also one of the important security issues [75].

4.2. Search String and Data Extraction

To identify the frameworks, architectures, scheme, techniques regarding verification of data location in Cloud Computing the literature study is applied. By using different databases to search the articles that solve the issue of data location, first step is to prepare a search string that allows getting complete articles. Authors by considering the synonyms and information prepared a search string and used it in six databases namely Inspec, IEEE, ACM, Springer link, Compendex, Science Direct and Google Scholar. The search string used here is as follows: ("Cloud Computing") AND Data AND (loc* OR security OR exist*) AND (framework OR model OR method* OR structure OR technique))

Using this search string authors searched in 6 databases by choosing the preference of publishing year from 2000-2012. The below Section 4.3 show the study selection criteria applied.
4.3. Study Selection Criteria

After retrieving the articles authors enforced study selection Criteria in order to achieve RQ1. According to RQ1 existing research is elucidated by considering three types of criteria namely a) Basic Inclusion Criteria b) Detailed Inclusion Criteria and c) Exclusion Criteria. A detailed description of these criteria is as follows:

a) Basic Inclusion Criteria: Studies published between 2000-2012 are considered as the main adoption rate of Cloud Computing is increased from 2000 [74] which is of English language text and are Journal or Conference or Workshop reports.

b) Detailed Inclusion Criteria: This criterion includes few different aspects that are considered to achieve RQ1.
   - Articles that comprises data location issues in Cloud Computing.
   - Articles that discuss importance of data location Cloud Computing.
   - Articles that proposes frameworks in order to verify data location of Cloud resources.
   - Proposed frameworks verify data location of web application content or data files deployed in Cloud.

c) Exclusion Criteria: Studies that propose technique to identify data location of end users who access Cloud resources are not considered as there are many existing tools and techniques to encrypt end user access [102] which comes under data integrity security issue [100]. As the main intention is to verify whether the geolocation of Cloud resources are at certain jurisdiction. However the studies that does not consists of the above detailed inclusion criteria are not considered. Number of articles retrieved in each database using the search string is presented in below Figure 4.

4.3.1. Study Selection Process

By considering the study selection criteria described in Section 4.3 authors extracted the studies in order to achieve RQ1 in five steps. Figure 4 shows the steps and procedure for study selection process.
The Figure 4 shows the steps involved in study selection process Step 1: Authors in this step removed the duplicates that occur in databases and 4180 initial studies are obtained. Step 2: By reviewing the title and abstract the studies which are related to basic and detailed inclusion criteria are considered this resulted with 218 articles Step 3: Here complete text in the articles are reviewed by authors and used the information regarding data location in Cloud Computing in providing Background. Step 4: Now the 10 frameworks regarding data location in cloud computing are considered. Step 5: The frameworks that solves the issues of verifying data location of web application or data files in Cloud resources are considered which resulted in three frameworks. However the three frameworks evaluation generalizes the existing data location frameworks to some extent.

4.4. Description of Data Location Frameworks

By extracting articles based on basic and detailed inclusion and exclusion criteria authors considered 10 data localization frameworks, architectures, schemes, that identifies data location in cloud computing. A brief discussion of the research done on data location in Cloud Computing is done in this section.

Gartner [77] described seven security issues that customers should ask cloud providers before moving to cloud. Data location is one of the security issue among the seven, in that customer should demand information about the location of data stored in cloud [77]. To secure data stored in cloud, providing a clear view on the location of data stored in cloud is also one of the important issues to allow capturing the misbehaving server and show reliability to the data stored. During the verification of storage correctness in cloud, Cong Wang et al. in [76] proposed a scheme which identifies the location where data error took place. It is crucial for users to check their data as data in cloud is not physically stored in with them, so Cloud providers should equip users with security concerns. As a data in cloud is saved in different locations it is also crucial to check whether the data is safe at all locations so a Distributed Multiple Replicas Data Possession checking scheme is proposed by Jing He et al. in [45] which checks geographically located replicated data. One of the important components in of IT auditing is to recognize the information of where data is stored and who is accessing the data in cloud. Once data in cloud transfers from one place to other it makes difficult to get an information of transferred data to resolve this problem tracking number are used that allows to get the IP address where data is transferred, this makes easier to get the location of the transferred data in Cloud [5].

To make the resource highly available Hadoop Mapreduce systems allows minimizing the routing distance and improves the performance. However to improve the performance of the existing methods in a heterogeneous environment, X. Zhang et al. in [105] proposed a data locality aware task scheduling method. The method allows mapping the task whose input data is nearest to the requesting node. An algorithm Balance Reduce (BAR) to reduce the task scheduling is proposed which manages large problem instances very efficiently [106].

Different new architectural models satisfying to government required security, privacy concerns are described in [6]. Government private cloud can utilize data from public cloud but these data centers must be located within the country according to security and privacy laws. As it requires to high cost to establish dedicated vendor very few uses this option [6]. Due to unequal security standards, data location of Cloud Provider and change of governance within in Cloud service enterprisers as well as governments were lacking behind to move their data to the Cloud resources. To address this problem of managing security in public Cloud, Rizwan Ahmad et al. in [4] proposed a government life cycle framework for securing users data in Cloud.

Cloud Traceback (CTB) is a cloud protector made within virtual machines in network structure to identify and locate the attacker. CTB prevents cloud data from attackers to access data [7]. To preserve user’s authority and location information of a mobile, a group signature scheme is proposed known as Data Anonymous Attestation (DAA) protocol [8]. Private Cloud infrastructure is managed and possessed by customer and is located in their own region where as public cloud infrastructure is managed and located by cloud service provider which is out of their region [9]. An architecture that uses trusted third party to supervise the compliance of user’s requirements is proposed which consists of three components Users interface, resource management and data distribution service. Users are allowed to select the options relating data retention i.e. security, geographic location, budget, availability and performance. Users can check the location of data when needed [10].

Many clients of storage as a service system such as Amazon S3 [57] want to be sure that the files they have entrusted to the cloud are available at any time to address this problem, methods like Proofs of retrievability and Provable Data Possession [54] helped the clients to verify that their files are really stored in the server and also to
check the integrity of the data is maintained at any time. But if the exact location of data is not known then the above methods cannot guarantee whether the files are replicated onto multiple drives or multiple data centers. So in order to solve the above problems authors in [20] have started studying the problem of verifying whether the cloud provider is replicating the data in diverse geo locations or not. After studying the problem they have also proposed a theoretical framework to verify this property.

There is need to provide location preference to Cloud user who deploys and manage cloud resource, where Cloud provider facilitates user to create their resource in preferred location. Ali Noman et al. in [21] proposed Data Location Assurance Service (DLAS) model based on two techniques cryptographic primitives and cipher text policy based encryption. DLAS model allows cloud users to select the preference regarding data location (i.e. keep the data in certain geographic location for example Europe) and also verify whether Cloud provider is maintaining the data at the preferred location [21]. This model is based on information provided by Cloud provider regarding data location [21].

There is a need for users to be aware of the exact virtual machines (VMs) and the physical locations of underlying physical machines (PMs) that they have stored data in. As an answer to the shortcomings Jagadpramana et al. in [23] proposed a solution called Flogger, a novel file-centric logger suitable for both private and public Cloud environments. Flogger records file centric access and transfer information from within the kernel spaces of both VM’s and PM’s in the Cloud, thus giving full transparency of the entire data landscape in the Cloud. The proposed technique satisfies the limitation of [24] [25] by addressing the need for logging within networks of PMs hosting multiple-folds of VMs. As data stored in Cloud is replicated at different geolocation, to verify the location of those replicated data Birjodh Tiwana et al. [75] proposed a framework for verifying caching data in Cloud resources.

In cloud computing, the need for data management services in public cloud is very much essential for sensitive enterprise data. Security to data stored in public cloud has been neglected [36] and it should be given at most importance. For this many techniques were proposed that provide data confidentiality, integrity and availability for processing queries on encrypted data. But integrating these techniques to form a practical secure data management service that works for most database queries has always been challenging. So Agrawal, et al. [35] reviewed all the relevant techniques and presented a framework based on the secure index for practical secure data management and query processing. They also discussed about how to enhance data confidentiality by providing practical access privacy for data in cloud.

Identifying the error localization of the data distributed in Cloud resources is one of the challenging issues to address this Hiep Nguyen et al. in [108] proposed a system called Propagation aware data anomaly localization, this system allows to point out the error location in distributed files. Integrity and the correctness of the user data when data is migrated in different geolocation is a security challenge Sravan Kumar R et al. [102] proposed a scheme which provides proof of the storage files in Cloud.

In order to provide the service worldwide, CSP generally has data centers spread across different locations around the world. Likely, many users also use the services who are located in different locations across the world. So the challenge comes in here is, providing best service to the user (i.e. finding data centers and physical machines that has lighter workload and geographically close to the user’s location). To answer this problem Gihun Jung et al. [47] proposed an adaptive resource allocation model for both users and providers to find proper data center and physical machine, geographically close to the user location for resource allocation. They used two evaluations for their proposed models that best matches the user requests which are a) The geographical distance between the user and data b) the workload of data centers.

In contrast, a large amount of data needs to be stored by an organization (customer), where all of them are not sensitive. As an answer to the problem Ganguly et al. [48] proposed an approach for data privacy of private sensitive data in hybrid cloud computing. This mechanism ensures that the private and sensitive data of the enterprise are to be stored and maintained within the enterprise itself when the enterprise is globally distributed. They implemented an authentication algorithm in a very small setup Cloud environment and conducted the experiment. A framework for improving management of data location is proposed by Jifeng Cui et al. [80] in addition to this an optimal scheme is proposed to merge the management of spatial data from different sources. However well-known Cloud products like Amazon EC2 [40] and Google app engine [61] are providing some basic security, privacy and trusted mechanisms, but these are not customizable at every case. To answer the above
problem Ivona Brandic et al. [62] proposed a novel approach for compliance management in clouds which is termed as Compliant Cloud Computing (C3). They proposed two things which are: a) proposing novel languages for specifying compliance requirements concerning security privacy and trust by leveraging domain specific languages and compliance level agreements b) Proposing C3 architecture middle ware that is responsible for the deployment of certifiable and auditable applications for provider selection, in compliance with the user request and also for the enactment and enforcement of compliance level agreement.

In order to reduce the cost Cloud resources are distributed at different locations, a formal mathematical routing problem named as Cloud Location and Routing Problem (CLRP) is proposed by Federico Larumbe et al. in [19] to define the data center location. CLRP allows emphasizing the importance of data location by showing the cost required for the data centers located in different locations. Apart from this framework, Aiiad Albeshir et al. in [78] proposed a protocol which allows verifying the data location of Cloud resources by using latency measurement; this protocol does not require trusting the Cloud provider.

To integrate the places of user centered platforms a prototype is proposed by Martin Gruhn et al. [79], this prototype is developed based on grassroots approach which allows personalizing, visualizing, classifying and verifying the location in online communities. Jifeng Cui et al. in [80] studied the management of large and small distributed files. They proposed a framework based on the Google file system that improves the management of geospatial data by accessing locality and geospatial relationship in distributed files. Now authors conclude that research on verifying the data location in Cloud resources is done in different ways. Discussion of the types of existing framework clearly defined in Section 4.5

4.5. Analysis of Considered Primary Studies

In order to validate the data location frameworks, different types of studies are considered which are clearly described in this section.

Figure 5: Analysis of considered primary studies

Figure 5 shows that there exist 10 different frameworks, architectures and schemes [1] [2] [3] [4] [75] [19] [78] [79] [80] to verify the data location of Cloud resources in which four frameworks [1] [4] [21] [79] allows identifying data location by using the monitoring information provided by Cloud Provider (Trusting Cloud Provider) and the remaining six frameworks [2] [3] [19] [75] [78] [80] are proposed to verify the data location without the help of Cloud Provider (Without Trusting Cloud Provider).

From the considered frameworks authors observed that framework [1] allows to identify data location However these frameworks are based on the performance measures i.e. based on RTT and latency values and managing geospatial data. From these studies [2] [3] [19] [75] [78] authors have observed that data location problem has been solved under two different scenarios 1) Identify the data location of Cloud resources by measuring Round Trip Time [2], 2) To identify the geolocation of the replicated files in Cloud resources by measuring Latency [3]. The
framework in [78] proposed to identify data location based on performance measurements. While the framework in [79] [80] are schemes that allow managing geospatial data.

However the data location is visible to Cloud customer by following the frameworks, there are two different types of frameworks i.e. with the help of Cloud Provider and without the help of Cloud provider.

**With Trusting Cloud Provider:** There exist four frameworks [1] [4] [21] [79] to identify data location with the help of cloud provider in which framework [1] is proposed to identify data location by monitoring information provided by CP, framework [4] is government life cycle framework which allows to manage users data security in public cloud computing, framework [21] is DLAS data localization assurance service which is based on cryptographic primitives that allows cloud users to select the preference regarding data location. And in framework [79] authors proposed a prototype to select user interface for localization i.e. to select preference regarding data location. As the main aim of this thesis is to validate the data location framework that allows to identify data location in cloud authors considered framework [1] whereas the remaining framework [4] [21] [79] are not considered as these frameworks allows how to consider preference regarding data location.

**Without trusting Cloud Provider:** There exists six frameworks obtained from literature [2] [3] [19] [75] [78] [80] to identify data location without the help of cloud provider. In [3] authors proposed theoretical framework that allows identifying data location of replicated files by measuring latency. In [2] authors proposed theoretical framework that allows to identify data location of cloud resource by measuring RTT (Round Trip Time) value, framework [19] authors proposed mathematical solution to define data center by considering budget and network performance. In framework [75] authors proposed a contour system that allows identify location of cloud files in different zones, this system is based on latency measurement. In [78] authors proposed a geo proof protocol to identify the location of files, this protocol is based on latency measurement. In [80] authors proposed a geospatial file system to manage geospatial data in cloud. Among these frameworks [2] [3] [19] [75] [78] [80] authors considered only theoretical framework for validating, so authors considered framework [2] [3] which are the theoretical frameworks and the remaining frameworks [19] [75] [78] [80] are schemes and prototypes.

By following this criteria authors considered three frameworks which allow identifying data location with the help of CP i.e. by trusting the Cloud Provider i.e. [1], and the two frameworks [2] [3] which does not require trusting Cloud Provider. However the remaining frameworks allows managing data location of Cloud resources in two different ways and evaluation of the three different data location frameworks generalize the remaining frameworks to some extent i.e. factors considered from the three frameworks [1][2][3] also included in remaining data location frameworks that are not considered for evaluation.

Identifying data location of Cloud resources when a web application is deployed in Cloud is possible through the proposed framework [1]. There also exist another framework [2] to verify the data location of Cloud Resources without the help of Cloud Provider and however files are replicated in Cloud, the data location of the web application content when the files are replicated in different geolocation is also a data location issue which is solved by Karyn Benson et al. in [3]. This motivated authors to perform a controlled experiment and validate the efficiency of three data location frameworks [1] [2] [3]. The replication of these three frameworks [1] [2] [3] is performed by controlled experiment is clearly described in Section 6. Before discussing about the evaluation of experiment a clear view on Amazon is discussed in Section 5.

5. **BRIEF OVERVIEW OF AMAZON WEB SERVICES**

**Amazon AWS:** Amazon Web Services (AWS) is known as the top innovative leaders of all cloud providers because its computation power is very high when compared with other cloud providers [37] [38]. Amazon AWS was started in the year 2006 which offers Infrastructure as a Service and Platform as a Service to the public cloud users (i.e can be accessed by anyone from anywhere through internet by all enterprise applications, big data projects, social games and mobile applications [37] [38]) spread over 190 countries and data centers located in US, Europe, Brazil, Singapore and Japan. One of the key benefits of Cloud Computing (i.e opportunity to replace up-front capital infrastructure expenses with low variable costs that scale with your business) made Amazon to offer low cost, instantly elastic, open & flexible and secure web service which is Amazon Web Service (AWS) [38].
Figure 6 shows the principle followed by Amazon Web Services to render their services to the users of the public Cloud. Amazon initially started the service Amazon Simple Queue Service (SQS) and later on they eventually increased the services. Out of many services author selected important services required for deploying the web application in AWS. The services used for the deployment of web application are: Amazon Elastic Compute Cloud (EC2), Amazon Simple Storage Service (S3), Amazon Cloud Front (CF), Amazon Relational Database Service (RDS) and Amazon Cloudwatch.

Amazon Elastic Compute Cloud (EC2): Amazon Elastic Compute Cloud (EC2) is a web service provided by Amazon to provide a resizable compute capacity in the cloud [37]. In simple words it offers virtual servers running on remote locations on a rental basis [37] [40]. These virtual servers function over Amazon Data center. EC2 can allow us to do the following things [14] [40].

- It provides complete control own computing resources and effectively use the Amazon proven computing environment.
- It reduces the time required to obtain and boot new server instances to minutes, allowing quickly changing or modifying your capacity up and down according to the change in requirements.
- EC2 enables us to use web service interfaces to launch instances of any O.S, Load them with custom applications, manage the network access permission and run the image in few or many systems of choice.

Amazon Simple Storage Service (S3): Amazon Simple Storage Service (S3) is a service offered by Amazon that provides a simple web service interface to store, retrieve any amount of data, at anytime from anywhere on the web. It gives highly scalable, reliable, secure, fast, inexpensive infrastructure for the developers to store their data [14] [37] [57].

Amazon Cloud Front (CF): Amazon Cloud Front (CF) is a service offered by Amazon which can be used for content delivery for global users. In simple words by using this service we can deliver our web site including dynamic, static and streaming content using globally spread cloud front edge locations (i.e 24 places spread across US, Europe, Asia and South America) [37] [60]. Here the requests for the content are best routed to the nearest edge location so that content is delivered with best possible performance because amazon has its data centers spread in different geographical locations across the globe and as well as the users too. Cloud front works with other AWS services like S3, EC2, Load Balancing and Route53 [37] [60]. CF utilizes S3 by replicating the buckets across multiple edge servers. Amazon charges only for the amount of the content actually delivered through the service [60].

Amazon Relational Database Service (RDS): Amazon Relational Database Service (RDS) [67] is a web service that helps to easily setup, operate and scale relational database in the cloud. It minimizes the managing time for database administration tasks but allows us more to concentrate on the applications and the business [37]. RDS provides access to different database engines like MySQL, Oracle or Microsoft SQL Server databases which allows all our code, applications and tools to be readily used with the database engines existing in RDS. RDS offers the flexibility of patching the database software, backup of the database, storing backups for user-defined retention period and enabling point-in-time recovery automatically by itself [67]. RDS contains API calls that provide RDS with flexibility to scale the computing instances linked to RDS. RDS follows pay-as-you-go model for pricing [67].

Amazon Cloudwatch: Amazon Cloudwatch [32] [37] is a web service that enables the user to monitor the AWS Cloud resources and applications starting with Amazon EC2. The details which we can monitor are: resource utilization, operational performance, overall demand patterns such as CPU utilization, disk reads & writes and network traffic [32]. The monitoring information can be seen with the help of statistics, view graphs and set alarms of the metric data [32]. To use the Cloud watch one has to select an Amazon EC2 instance, business or application
metric data of choice that needs supervision [32]. Amazon cloud watch uses API’s or command line tools for aggregating and storing of monitoring data.

**Amazon Cloud Elastic Beanstalk (EBS):** Elastic beanstalk [63] is a service offered by Amazon which provides block level storage volumes to be used with Amazon EC2 instances. It is mainly useful for the applications that require a data base, file system or access to raw block level storage volumes. The storage volumes provided by EBS are highly available, highly reliable and predictable storage volumes which can be attached to a running Amazon EC2 instance. EBS allows creating storage volumes from 1 GB to 1 TB which can be mounted as device by EC2 instance. Volumes of EBS can be stored in a specific availability zone which later on can be added to instances present in the particular availability zone. EBS [63] prevents the data loss due to failures of any single hardware component by replicating the storage volume within same availability zone.

**Research done on Amazon Web Services**

Many researchers have done lot of research on or using Amazon Web Services. Lingli Fu et al. in the article [30] discussed the importance of Cloud Computing, its resources, computing layers and its architectures. Later on the proposed a scheme discussing on how to use cloud computing to design, build and deliver enterprise application with Amazon Web Services provided by Amazon and also the security requirements that are to be considered by enterprise while adopting cloud computing. Li Zhang et al. [29] emphasize the need for software testing especially for a distributed system using web service technique. As research on testing problems is still going on they propose a framework which integrates cloud computing and performance testing technologies. The evaluation of caching and storage options of Amazon Cloud is done by David Chiu et al. in the [28]. They have evaluated various cache storage options in Amazon cloud projecting the cost, performance and persistence tradeoff of each. Liang Zhao et al. [26] distinguish that horizontal scaling of Amazon is comparatively high when compared with Microsoft Windows Azure and Google App Engine.

Unavailability of cloud is an issue that depends on a lot of factors. Evangelinos and Hill [18], Hill and Humphery [17] evaluated the 32 bit and 64 bit of Amazon EC2 using MPI and memory bandwidth benchmarks respectively to explore the feasibility of running High Performance Computing (HPC) applications on cloud. Hoffa et al. [15] and Deelman et al. [12] also observed the data intensive workload cases on Amazon Simple Storage (S3) which mainly focused on analyzing cases of scientific applications rather than business applications.

Multi Cloud Computing is emerging Cloud model that needs to pay utmost importance. Multi Cloud is an improvement model of the hybrid Cloud Computing that is combination of one or more deployment models (like public Cloud, Private Cloud etc.) where Cloud resources are distributed across different geographical locations. The performance of different Amazon Cloud resources is evaluated by conducting an experiment, which concludes that performance of Cloud resources depends on Load balancer as well as location of the server [69]. However there is need to identify the geographical location of Cloud resources. This motivated to evaluate the frameworks [1] [2] [3] in different Amazon availability zones located across US, Europe and Asia.

### 6. EXPERIMENT

Literature study shows that Data Localization (DL) is one of the most important security issues in Cloud Computing (CC). There exist many DL Frameworks, Architectures and models that satisfy DL issues. To evaluate the efficiency of DL frameworks identified from Literature review, authors performed a controlled experiment by following the guidelines proposed in [66]. This section presents a clear vision of the experiment performed in Amazon Web Services.

#### 6.1. DEFINITION

**Object of the study:** The objects studied are three Cloud Computing Data Location frameworks [1] [2] [3] obtained from literature study, which satisfy the data location issue.

**Purpose:** The purpose is to evaluate the efficiency of three different data location frameworks by replicating the study using Amazon Web Services (AWS).

**Quality focus:** The quality focus is efficiency of the identified DL frameworks.

**Perspective:** The perspective is from Cloud user (deploys and manages the web application) and end user (access the web application/ use the services) point of view.
Context: Context here is the environment in which the experiment is run, authors acted as Cloud user and evaluated three DL frameworks in Amazon Web services (AWS) environment. The study conducted here is a multi-object variation study as objects here are the three different types of data localization frameworks.

Summary of Definition
To replicate the three DL frameworks, for the purpose of testing it in all possible AWS regions in order to improve the efficiency of DL frameworks, the experiment is carried out. This research study also helps Cloud users to deploy and manage web application in cloud regarding data location.

6.2. PLANNING
This chapter describes the way authors have conducted experiment. Overall planning phase is described in below four steps.

6.2.1. Context Selection
This section describes about the environment created for evaluating the three DL frameworks, in detail. The selected data localization frameworks are:

Framework 1: “Monitoring and Audit logging architecture for data location” [1],
Framework 2: “A framework for verifying the location of Cloud resources” [2] and
Framework 3: “A framework for identifying the location of Cloud Files and the replicated data in cloud” [3]

The above frameworks are proposed in order to identify the data location (data location here includes cloud resources, Cloud data files as well as total web application deployed) in Cloud Computing. Here the three frameworks show three possible scenarios to verify the data location in Cloud Computing, even though their cloud environments are different. Before replicating the study, authors have assumed that data in this study includes cloud resources, cloud data files as well as total deployed web application content. Now the work done in the three frameworks is replicated (i.e. complete framework [1] [2] [3] activities) in Amazon cloud (i.e. in their respective availability zones such as US, Europe and Asia).

Motivation for selecting the Amazon cloud environment: AWS is leading cloud service provider [65] that provides regions and availability zones to deploy and manage the application, data, and resources [1]. Evaluating the frameworks by utilizing the possible availability zones (US Virginia, US California, US Oregon, Europe Ireland, Asia Singapore and Asia Tokyo and South America Sao Paulo) contribute in exploring the behavior of DL, when Cloud resources are distributed throughout the globe. The availability zones selected for the experiment are shown in the below Figure 8. Thus authors used Amazon Web Services for the evaluation of the DL Frameworks [1] [2] [3].

![Figure 7: Amazon Availability zones [56]](image)

Figure 7 [56] shows the availability zones of Amazon web services the whole experiment is conducted from a single internet location Karlskrona, Sweden BTH which is represented as ‘A’ in the map. The remaining points ‘B’, ‘C’, ‘D’, ‘E’, ‘F’ & ‘G’ represents different amazon resources located at different continents. The locations of points ‘B’, ‘C’ & ‘D’ belongs to United States continent, ‘E’ belongs to Europe continent and ‘F’ & ‘G’ belongs to Asian continents.

US: ‘B’ represents the Virginia State, ‘C’ represents the state California and ‘D’ represents the state Oregon.
Europe: ‘E’ represents the location Ireland
Asia: ‘G’ represents the location of Singapore and ‘F’ represents Tokyo in Japan.

As deploying and managing a web application in Cloud requires many resources like data base, application server, etc., there is a need to manage and identify the data location whether a Cloud provider shows compliance [1]. Different countries have different data location issues that, data should reside within the country [1]. A typical .Net web application is developed using .Net framework 4.0 with SQL server 2008 as a backend and tested it locally. After successfully testing the web application locally, the application is transferred to AWS by using Amazon Elastic Beanstalk (EBS) [63] connecting to the Relational Database (RDS) [67].

Before deploying the application to AWS, RDS with ‘SQL Server Express Edition’ consisting of 30GB storage is created. RDS is Relational Database Service acts similarly as SQL Server database engine [67]. Now the database tables and content of the developed web application in SQL server 2008 is exported to the RDS database. The main aim of this experiment study is to deploy and manage the web application in possible Amazon availability zones, and identify how Amazon Cloud is offering the data location compliance feature that serves the content of deployed web application.

6.2.2. Hypothesis Formulation

To replicate and evaluate the efficiency of three DL frameworks authors formalized two hypotheses for each DL framework.

Null hypothesis H0: Verifying the DL of the deployed web application in US, Europe and Asia is not possible through the frameworks [1] [2] [3]. Here the null hypothesis is formulated into two sub hypothesis.

Null hypothesis H10: Replicating the data location frameworks [1] [2] [3] is not possible in Experiment setup.

Null hypothesis H20: Activities performed from framework [1] [2] [3] in the experiment does not make any change in identifying data location in the experiment setup.

Alternative Hypothesis H1: Verifying the DL of the deployed web application in US, Europe and Asia is possible through frameworks [1] [2] [3]. Here the alternative hypothesis is formulated into two sub hypothesis.

Alternative Hypothesis H11: Replicating the data location frameworks [1] [2] [3] is possible in Experiment setup.

Alternative Hypothesis H21: Activities performed from framework [1] [2] [3] in the experiment allow identifying data location in the experiment setup.

6.2.3. Variables Selection

The experiment was conducted in a controlled environment where a web application was deployed and managed by the authors. The effect of changes occurred during the evaluation of three frameworks in AWS environment was traced. The dependent and independent variables in this experiment are described below.

Independent variables:
• Selecting the availability zone in Amazon and creating instances at US, Europe and Asia.
• Deployed a sample web application.
• Activities performed in Framework 1 [1].
• Activities performed in Framework 2 [2].
• Activities performed in Framework 3 [3].

Dependent Variables:
Identifying the data location of deployed web application, where Amazon virtual resources US, Europe and Asia reside within the country jurisdiction. The value of dependent variable is the output came from evaluating three DL frameworks, the following measurements are used to get output:
• Capturing the data location events provided by AWS [38].
• Measuring actual distance between from the user location to Amazon AZ (Availability Zones).
• Measuring the RTT value of deployed web application.
• Measuring latency of deployed web application.
• Calculating the number of hops travelled to reach the AWS resources from internet location Sweden BTH.

As it is more time consuming process to study and collect data from real Amazon cloud users who are managing data in Amazon cloud, authors who conducted the Literature Review and identified the three DL frameworks
conducted the experiment in Amazon by deploying a web application. Deploying a .Net web application in Amazon cloud environment requires a complete awareness of the services (Amazon EC2, S3, and RDS). Before conducting the experiment authors have studied the user guide of AWS [38], which describes in detail about the procedure and services required to deploy and manage the web application. Authors faced some technical problems while creating the instances and deploying the web application, these technical problems were resolved by studying the Amazon user guides.

After getting a clear vision on how to use the services, web application is developed locally and deployed in Amazon cloud environment by the authors. Here authors acted as a cloud user who deploys and manages the application in cloud. The evaluation of the DL frameworks [1] [2] [3] in Amazon cloud was done in this experiment, which helps a new or existing Cloud application developer to deploy and manage a web application in Cloud regarding data location. Even though the experiment setup consists of micro instances and a sample web application, the evaluation of the three frameworks represents the general factors that should be considered by the Cloud user while managing the web application in real time. Authors conclude that evaluation of the three frameworks [1] [2] [3] in Amazon regarding DL is possible by deploying a web application and observing how effective and efficient are the frameworks regarding data location issue and also how is the data location feature being offered by Amazon Cloud in a generalized and improved manner.

6.3. DESIGN
To evaluate the three DL frameworks [1] [2] [3] in the experiment, a series of steps are planned and designed in a proper way. This chapter clearly describes in detail about the factors and treatments of the experiment

6.3.1. Design Type
The design type of the experiment involves one factor with three treatments. Here the factor is to identify the data location (by deploying the web application in Amazon Cloud and using the studies of the three frameworks in the experiment setup) and the treatments are three DL frameworks (by replicating the studies of the three DL frameworks).

<table>
<thead>
<tr>
<th>Factor</th>
<th>Treatment 1</th>
<th>Treatment 2</th>
<th>Treatment 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identify DL of the Cloud resources and data files created in US Virginia, California, Oregon, Asia Tokyo, Singapore and Europe Ireland</td>
<td>✓ Framework 1</td>
<td>✓ Framework 2</td>
<td>✓ Framework 3</td>
</tr>
</tbody>
</table>

The Table 1 shows that the three frameworks [1] [2] [3] are evaluated in experiment setup, where frameworks act as treatments. Hypothesis of the three Frameworks are tested and represented by using descriptive statistics.

Below Figure 8 represents design of the experiment setup. Before replicating the study authors created a Cloud environment by deploying a typical .Net web application to Amazon Web Services. The web application is deployed in all possible AWS availability zones one after the other, then activities of the three DL frameworks [1] [2] [3] considered for evaluation are performed to identify the DL of servers created in US, Europe and Asia. After performing the activities authors validated the results of each framework with experiment setup. By considering the three framework activities authors prepared a checklist for managing data location in Cloud Computing.
6.3.2. Lab environment

The experiment is conducted at BTH from a single Computer by accessing AWS resources [38] located in US, Europe and Asia through a student account.

6.3.2.1. Instrumentation

The design of the experiment setup shows that to identify DL in Cloud there are three DL frameworks [1] [2] [3] (refined from LR) that are to be evaluated in Cloud environment. As authors are familiar with developing ASP.Net web application, they developed a sample web application. The following are the tools used to while developing:

- Visual Studio 2010 using C#.Net framework 4.0 and ASP.Net.
- SQL server 2008 as a backend.

Web application

The web application is an (online shopping) typical .Net web application, which consists of homepage allowing the end user to register or login to the website. After signing up, the user is presented with some sample products like Cars, mobile phones and clothes etc. When an end user clicks on the products to buy, the details of products are added to the cart. The payment is made at the end of the process. Database in this application reads and writes the login credentials of the end user. After testing the functionalities of the developed application locally, the entire application is deployed to the Amazon Web Services [38]. To deploy the web application in Amazon an ‘AWS software development toolkit’ (SDK Toolkit) is added to the Visual Studio 2010 which allows .Net developers to deploy and manage the data in all Amazon Availability zones. There are few steps performed to deploy web application in AWS.

In order to publish the web application onto the Cloud, authors used AWS Visual Studio toolkit to create resources and export the web application data [107]. To deploy the web application the following services are required:

- AWS management console [52]
• Amazon Relational Database Service (RDS) of 30GB storage memory as an SQL server [67].
• Amazon Elastic Beanstalk as an application server (EBS) [63]. EBS automatically creates and performs auto scaling for two services namely;
  (i) Amazon Elastic Cloud Computing (EC2) as 64bit windows server 2008 r2 running IIS 7.5 with micro instance [40].
  (ii) Amazon Simple Storage Services S3 [57].

Amazon Elastic Beanstalk service is used to auto scale the resources like EC2 and S3 [63]. Now web application is deployed in all the Amazon Availability Zones (AZ) which are US (Virginia), US (California), US (Oregon), Europe (Ireland), Asia (Singapore) and Asia (Tokyo) and there is also another availability zone at South America (Sao Paulo) which is not considered in our experiment setup because EBS service is not available in this region. The overall goal is to manage data location, when an application is deployed in all AWS regions which are done by assessing the three flavors of DL frameworks. This allows controlling the experiment setup and monitors the web application regarding DL. Below Figure 9 is a reference diagram provided by Amazon [63] of how to deploy web application to AWS.

![Figure 9: Procedure followed for deploying the web application in AWS cloud](image)

AWS consists of different availability zones to run and manage our data in cloud they are:
1) US East VG
2) US West Oregon
3) US West N California
4) Europe West Ireland
5) Asia Pacific Tokyo
6) Asia Pacific Southeast Singapore

A sample view of the deployed web application in AWS is shown in Appendix E Now the web application is deployed in all the AWS availability zones one after the other to verify whether data location of each AWS [38] resource is showing the compliance to data location regulation acts or not. Authors used the architecture and activities proposed in [1] [2] [3] as treatments. Authors considered security policies and cloud compliance requirements [1] for the deployed web application that the application content in AWS should be visible and reside within the selected location.

6.4. OPERATION

The experiment planning and design phases are carried out to accomplish the object of the study in operation. This section describes how the DL frameworks [1] [2] [3] are carried out in the experiment setup. The operation phase consists of three steps preparation, execution and data validation.

6.4.1. Preparation (Experiment Environment details)

As described in the planning and design phase the web application is developed locally and deployed on all possible AWS availability zones one after the other. It took more than 50 minutes to create instances and deploy the web application in each Amazon region. Before conducting the experiment, authors discussed the three structures of DL frameworks [1] [2] [3], as the replication of the exact architecture differs i.e. while conducting in Amazon Cloud, due to lack of tools and change in the environment. The main concepts that are useful for identifying the data location from the three frameworks [1] [2] [3] are considered. Factors considered for each treatment (for each framework) are shown in the Tables (3), (6) and (9). The activities performed in the experiment resulted only in two
ways one is activities performed allow identifying data location, or does not allow. Authors evaluated the efficiency of each framework(s) by considering the activities in it respectively and used the rubrics for the evaluation of each framework [1] [2] [3] during the experiment. To control the results of activities performed in the experiment authors assigned rubrics. A brief description of rubrics provided is as shown in the below Table 2.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Rubrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>If the activity performed is satisfied in the experiment</td>
<td>1</td>
</tr>
<tr>
<td>If the activity performed does not satisfy in the experiment</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2: Rubrics for evaluating experiment activities with frameworks

A detailed description of preparation and execution of each treatment (i.e. each framework) is described in below Sections 6.5, 6.6 and 6.7.

6.5. EXPERIMENT- 1:

After deploying the application in US, EU, Asia locations provided by Amazon [38], authors by considering the framework 1 [1] seeks to identify the DL of web application content. Before going to evaluate the framework 1 [1], a brief description of the framework is shown in below Section 6.5.1.

6.5.1. Framework – 1 Description [1]

This article [1] shows how to verify the data location of the Cloud resources with the help of monitoring information provided by Cloud Provider (CP) [1]. In the proposed logging architecture collaboration is needed between Infrastructure Provider (IP), Service Provider (SP) and the user of the cloud [1]. Service Provider (SP) verifies the data location compliance with the help of logging architecture provided by CP, in federation [1]. Federated Cloud here is combination of different Cloud Computing services to manage the migrated Cloud [1]. To ensure the transparency of the processing data in Cloud to the CP, this monitoring and audit logging architecture is proposed. It uses a reference architecture proposed by [55]. To get total control over the Cloud resources regarding data location the proposed architecture allows tracking the complete Cloud operations [1].

“Reservoir is a virtualized infrastructure layer on the physical infrastructure” [1] that consists of three levels where firstly the Service manager instantiate the service application in Cloud. Secondly Virtual Execution Environment Manager (VEEM) manages the placement of Virtual Execution Environment (VEE) on to the VEE hosts. The third level is the Virtual Execution Environment host (VEEH) which represents a virtual resource that hosts a specific type of VEE [3]. Virtual environment is the Virtual Machine (VM) created and deployed in Cloud.

![Figure 10: Proposed monitoring and audit architecture [1]](image)

In this architecture [1] they used an XML based service definition which is used for specifying the requirements of a Service Provider regarding the placement of the VM and the monitoring information of the VM [1]. The monitoring information that SP verifies include the following events: Memory allocated for the resource created, placement information regarding VM [1], details of the time created the VM [1], details of the events provided when the VM is deleted and details of the VM when migrated from one location to the other [1] etc. These monitoring and audit logging information allows SP to acquire complete information about the data when migrated to the cloud [1].
Figure 10 is the Reservoir architecture [55] that is used as reference architecture for monitoring and audit logging [1] of data location in Cloud (framework 1). The proposed monitoring and audit logging architecture [1] is evaluated by performing case study in the French e-government application [1]. Authors considered data location constraint requirements, these requirements are considered in the experiment setup and evaluated the possibility in Amazon Cloud environment. Now a detail description of the experiment environment is shown in Section 6.5.2.

6.5.2. Experiment Setup for replicating framework 1 [1]
The main factor is to identify the location of data in AWS [38]. Authors used the architecture as a treatment to identify the data location in AWS. Amazon architecture is used to capture monitoring events [38].

Figure 11: AWS web hosting architecture [38]

Figure 11 shows the Web hosting architecture of AWS [38]. The AWS management console present in the Amazon provides the events that were happened in the form descriptive logs regarding the data location. These monitoring information regarding data location were captured by the authors. The following are the monitoring events captured while deploying the web application in US, Europe and Asia availability zones.

EC2:
- Instance zone created
- EC2 instance IP address
- DNS name of the load balancer
- Availability zone of the load balancer
- Hosted zone ID of the load balancer

S3:
- S3 bucket name
- Standard Region of S3
- http link of S3

RDS:
- End point server name
- Zone details of RDS

Factors are considered from framework 1 [1] and these are evaluated in the experiment setup are shown in the below Table 3. Some of the activities that are not possible in the experiment setup show cross (‘×’) mark and the brief description of how the experiment setup tweaks, in order to perform the activity are shown in the experiment section. Activities that are performed and considered in experiment similar to framework shows tick mark (✓).

Table 3: Factors considered in the experiment setup

<table>
<thead>
<tr>
<th>S.no</th>
<th>Framework</th>
<th>Experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Application server i.e. VM should be located and stay in the given country [1].</td>
<td>✓ Online shopping web Application deployed in AWS availability zones was verified by capturing the monitoring events.</td>
</tr>
</tbody>
</table>
Compliance with legislation is possible through the proposed monitoring and logging architecture, preventing any violation [1].

Reservoir based cloud monitoring and logging architecture is proposed to meet compliance requirements [1].

The logging information of VM should be captured [1].

We used AWS management console to capture logging events.

The monitoring events of VM should be guaranteed [1].

Monitoring events of EC2, RDS, S3 and EBS is captured in AWS management console.

The Cloud application in federation i.e. when created in different locations must logged [1].

All application events regarding data location were captured.

Monitoring information should be performed even when the resources are created in different locations [1].

Monitored information is provided and managed by Amazon [38].

Changes in VM operations should be monitored by Service Provider (SP) [1].

As a SP in our experiment we were able to record all VM operation of AWS resources.

Logging details should be flexible and information should be up to date [1].

AWS provided complete monitoring information of each and every resource, distributed in different locations with a most feasible manner.

Isolation, access to the VM resources should only be possible for SP to monitors information VM [1].

AWS availability zones provide information about each zone separately.

Data location of application server should be visible and reside within the given preference for location [1].

Table showing the captured events of each availability zone allows identifying the location of data.

The proposed logging architecture captured the monitoring events by using a French e-government cloud web application [1].

Deployed typical .Net web application developed by authors in AWS and captured the monitoring events [38].

In framework XML based service definitions are used to express the requirements of the SP [1].

Here instead of XML based service definition, the requirements of authors are directly selected from the drop down menu (i.e. choosing the desired location of AWS resources from list of available zones).

<table>
<thead>
<tr>
<th>6.5.3. Execution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon Web Services [38] not only provides the monitoring information regarding data location but also health of the instance and count on number of input/output requests and other details about the resources [52]. By using Amazon Cloudwatch [32] service users are allowed to monitor the information of the resources created in Amazon cloud [38]. As object of this experiment study is to monitor the information regarding data location (i.e. the location of resources holding web application content) the information regarding the web application content other than DL is not considered. However authors examined the environment details shown in Appendix F i.e. for example.</td>
</tr>
<tr>
<td>• Successfully deployed online shopping web application using Elastic Beanstalk in us-west region.</td>
</tr>
<tr>
<td>• Environment is healthy at <a href="http://onlineshopping.elasticbeanstalk.com">http://onlineshopping.elasticbeanstalk.com</a>.</td>
</tr>
<tr>
<td>• Incremental deployment of application is done i.e. information when any changes made in application.</td>
</tr>
<tr>
<td>• Terminating the environment completed successfully.</td>
</tr>
</tbody>
</table>

An Amazon Cloud service is an IaaS provider [38] which provides different regions in the same location to replicate the data within the location, authors deployed in each of the regions. The monitoring information of the events observed when an application is deployed in different availability zones is shown in the below Table 4. This table shows that the web application content is resided according to the selected country preference (input).
<table>
<thead>
<tr>
<th>INPUT</th>
<th>AMAZON WEB SERVICES DATA LOCATION MONITORING EVENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RDS</td>
</tr>
<tr>
<td></td>
<td>Instance</td>
</tr>
<tr>
<td></td>
<td>Zone</td>
</tr>
<tr>
<td>US East VIRGINIA (VG) Monitoring Events</td>
<td></td>
</tr>
<tr>
<td>US West OREGON (OG) Monitoring Events</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>US West CALIFORNIA (CA) Monitoring Events</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>EU West IRELAND (IR) Monitoring Events</td>
<td></td>
</tr>
<tr>
<td>RDS</td>
<td>EBS</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Con</td>
<td>Cou</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>EU</td>
<td>West IR</td>
</tr>
<tr>
<td>EU</td>
<td>West IR</td>
</tr>
<tr>
<td>EU</td>
<td>West IR</td>
</tr>
<tr>
<td>EU</td>
<td>West IR</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
</tr>
<tr>
<td>ASIA</td>
<td>NORTHEAST TK</td>
</tr>
<tr>
<td>ASIA</td>
<td>NORTHEAST TK</td>
</tr>
<tr>
<td>ASIA</td>
<td>NORTHEAST TK</td>
</tr>
</tbody>
</table>

### ASIA Pacific TOKYO (TK) Monitoring Events

<table>
<thead>
<tr>
<th>ASIA</th>
<th>SOUTHEAST SGP</th>
<th>NO</th>
<th>ASIA PACIFIC SGP</th>
<th>AP Southeas t1a</th>
<th>175.41.179.71</th>
<th>AP South east 1a</th>
<th>AP South east-1.elb</th>
<th>AP South east 1a</th>
<th>Z1W18VXHPB1R38</th>
<th>EBS AP South east1</th>
<th>ASIA SGP</th>
<th>AP Southeas t1</th>
<th>AP South east 1.rds</th>
<th>AP Southe ast 1b</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASIA</td>
<td>SOUTHEAST SGP</td>
<td>1a</td>
<td>ASIA PACIFIC SGP</td>
<td>AP Southea st1b</td>
<td>46.137.227.145</td>
<td>AP South east1b</td>
<td>AP-south east-1.elb</td>
<td>AP-south east-1b</td>
<td>Z1W18VXHPB1R38</td>
<td>EBS AP South east1</td>
<td>ASIA SGP</td>
<td>AP Southeast 1</td>
<td>AP South east 1.rds</td>
<td>AP South east 1a</td>
</tr>
<tr>
<td>ASIA</td>
<td>SOUTHEAST SGP</td>
<td>1b</td>
<td>ASIA PACIFIC SGP</td>
<td>AP Southeast 1b</td>
<td>54.251.10.83</td>
<td>AP South east1b</td>
<td>AP South east-1.elb</td>
<td>AP South east 1b</td>
<td>Z1W18VXHPB1R38</td>
<td>EBS AP South east1</td>
<td>ASIA SGP</td>
<td>AP Southeast 1</td>
<td>AP South east 1.rds</td>
<td>AP South east 1b</td>
</tr>
</tbody>
</table>

*Table 4: Data location monitoring events captured*
Con: Continent, Cou: Country
6.5.4. Data Analysis

The Table 4 represents the captured events regarding data location when a web application is deployed in AWS [38] availability zones (US, Europe and Asia). It shows that Amazon creates and maintains the resources according to the user preference i.e. if a Cloud user prefers to deploy application in US Virginia then the monitoring information of EC2, RDS and S3 resources shows that services are created and running at the US Virginia. The monitoring information from Table 4 shows that the location of all AWS resource events except IP address and Hosted zone ID can be identified by simply observing the name (for example: when the application is deployed in US the monitoring information of EC2 instance zone observed is US-east 1b which thereby confirms that it is located in US). The IP addresses and Hosted zone ID for sub regions are similar (i.e. US 1b, 1c, 1d) but different for other countries. Table 4 shows that when deploying an application in Amazon Cloud [38] the captured monitoring events allows us to identify the information of each and every service (i.e. EC2, RDS and S3 regarding data location) [38]. By implementing the framework 1 [1] identification of data location of Cloud resource is possible in Amazon Cloud environment. Authors identified that deployed application in USA (Virginia, California and Virginia), Asia (Singapore and Tokyo) and Europe (Ireland) reside in the given country by capturing the events of resources EC2, S3 and RDS. By following the rubrics authors evaluated the framework 1 [1] with experiment setup. A detailed description of the comparison of rubrics and activities is shown in the below Table 5.

### Table 5: Evaluating the framework with experiment

<table>
<thead>
<tr>
<th>S.no</th>
<th>Activities regarding Data Location</th>
<th>Framework</th>
<th>Experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Application server migrated to the cloud environment should be located and stay in the given country [1].</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Compliance with legislation is possible through the proposed monitoring and logging architecture, preventing any violation [1].</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Reservoir based Cloud monitoring and logging architecture proposed meets data location compliance requirements [1].</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>All VM monitoring events are logged [1].</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>All application events regarding data location must be recorded in when resources are place in different countries [1].</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>Monitoring must be performed when application is deployed and incremented in different locations [1]</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>All VM operations must be monitored by Service Provider (SP) [1]. (As a SP in experiment setup are authors who are allowed to record all VM operation of AWS resources [38].)</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>Flexible log format (AWS provides complete information of each and every resource, distributed in different locations with a most feasible manner).</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>Isolation SP monitors information of his own VM [1] (AWS availability zones provide information about each zone separately.)</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>Data location of application server is visible [1] (Table showing the captured events of each availability zone, allows identifying the location of data).</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>Capturing the IP address of the Cloud resources [1].</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>Monitoring the information of Cloud servers [1] residing US, Europe and Asia.</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>Identifying the data location without the need Cloud Services Provider.</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The Table 5 represents the evaluation of the framework 1 [1] with the experiment setup by providing rubrics to the factor whether satisfied or not. However authors were able to satisfy more data location activities than the framework.

6.5.5. Weakness of the framework

By following the framework 1 [1] activities authors were able to identify the data location of web application deployed in US, Europe and Asia of AWS effectively and efficiently. However there are certain factors that may affect the identification of data location. The following are the factors that this framework [1] does not satisfy:
• It is not certain that a Cloud provider may provide monitoring information regarding data location exactly where the data reside. There is threat that Cloud provider may provide and maintain the data at some other location by giving the monitoring information at another location.
• The location of resources that utilize for deploying web application is identified, but the files and data transferring in the resources are not identified.
• Some of the IP address may give wrong information regarding data location. As the records may not be updated [3].

This framework 1 allows Cloud user to identify the data location of their resources created in different locations. Some of the data location factors were not satisfied as the framework 1 [1] is concentrated mainly on the monitoring information regarding data location. There is another framework 2 [2] that authors considered as the treatment that which allows identifying the data location without the help of Cloud provider.

6.6. EXPERIMENT- 2
The web application deployed in all regions of AWS availability zone is now evaluated by performing the activities of the framework 2 [2], a brief description of the framework 2 [2] is discussed.

6.6.1. Framework – 2 Description [2]
To identify the data location of Cloud resources without the help of Cloud provider Thorsten Ries et.al [2] proposed a technique using network coordinate system (VCS) [2]. The evaluation of this VCS is done on three prevalent systems Vivaldi, Pharos and Phoenix [2]. The proposed technique also allows verifying Cloud resources when a Cloud Provider changes the location of the resource to other region [2].
• The proposed geolocation technique is performed by measuring RTT (Round Trip Time) value between internet locations to the virtual Cloud resources. Through Planet Lab Node the RTT value from different internet locations to the Virtual resources is measured [2].
• The framework 2 [2] assumes two topologies for verifying data location of Cloud resources. 1) Cloud resource created and located at the given preference [2]. 2) Due to the heavy load in Virtual Resource Cloud provider changes the location of resource by placing a relay node at the actual location providing the transparency to CP [2].
• The RTT value is measured from Planet Lab Nodes located in 120 nodes at 28 different countries. The measurements show that it correctly estimated the geolocation of Cloud resources [2]. The verification of Cloud resources is done by calculating a set of landmarks distributed in different locations [2].
• By evaluating the three prevalent systems in VCS Thorsten Ries et al. [2] conclude that Phoenix performs best when compared with other two systems in identification of Cloud Resources.

6.6.2. Experiment Setup for replicating framework 2 [2]
• The same environment deployed in Amazon availability resources [38] that was used for evaluating framework 1 is again evaluated in this experiment by following the framework 2 [2] activities to identify the data location.
• RTT measurement is used in order to identify the location of AWS server. RTT value is the total time taken for a specific file or data to send a request to the server and get response from the server. RTT values depend on the distance between user locations in the internet to the server [2].
• In the experiment setup before measuring the RTT values. Authors first calculated the distance from user’s internet location i.e., Karlskrona, Sweden to all Amazon availability zones US (VG, California, Oregon), Europe (Ireland), Asia (Singapore and Tokyo). Figure 12 shows the actual distance measurement and comparison of the distance between the servers. Secondly calculated the total number of network hops travelled to get response from AWS resources [38].

Table 6 shows the factors that are considered from framework 2 [2] to evaluate it in the experiment setup. Some of the activities that are not possible in the experiment setup show cross (‘✗’) mark and the brief description of how the experiment setup tweaks, in order to perform the activity are shown in the experiment section. Activities that are performed and considered in experiment similar to framework shows tick mark (‘✓’).
<table>
<thead>
<tr>
<th>S.no</th>
<th>Framework 2 [2]</th>
<th>Experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Performed test by using Cloud resources to identify Data location [2].</td>
<td>✓ Authors used Amazon Web Service as Cloud resource.</td>
</tr>
<tr>
<td>2</td>
<td>To identify data location without the need of trusting the cloud operator [2].</td>
<td>✓ Deployed a web application in AWS availability zones and verified the location of AWS [38] resource by following the proposed geolocation approach.</td>
</tr>
<tr>
<td>3</td>
<td>Used both measurement based and semantic based approach to identify the location of virtual resources [2].</td>
<td>✓ Successfully identified the location of AWS resources by using both measurements based and semantic based approach.</td>
</tr>
<tr>
<td>4</td>
<td>Through Round Trip Time (RTT) values estimated the geolocation of virtual resources [2].</td>
<td>✓ Through RTT values and actual distance to the AWS resource authors estimated the geolocation.</td>
</tr>
<tr>
<td>5</td>
<td>Measured RTT values from different locations in the internet to the virtual resources.</td>
<td>✗ Measured RTT values from single location i.e., Karlskrona, Sweden to the AWS availability zones.</td>
</tr>
<tr>
<td>6</td>
<td>Considered Virtual resource as a Virtual machine which is able to respond to RTT request</td>
<td>✓ Authors used RDS, EC2 and S3 resources for deploying the application as these are able to respond to the RTT request.</td>
</tr>
<tr>
<td>7</td>
<td>Identified that Cloud resource created through the internet [2].</td>
<td>✓ Authors created and managed AWS resources [38] by directly connecting through the internet.</td>
</tr>
<tr>
<td>8</td>
<td>To increase the transparency and prove the geolocation of the virtual resource in Cloud [2].</td>
<td>✓ Objective of the Experiment study is to Identify and prove the geolocation of each AWS resource created in different locations (USA, Europe and Asia)</td>
</tr>
<tr>
<td>10</td>
<td>Distance between the Virtual resource and internet node is similar to the RTT value between them [2].</td>
<td>✓ Actual distance and RTT values from the user location to AWS availability zones are calculated.</td>
</tr>
<tr>
<td>11</td>
<td>The computation of Vivaldi coordinate system is based on RTT time between different internet nodes to the virtual resources [2].</td>
<td>✓ RTT values from single node to different servers are measured to estimate geolocation.</td>
</tr>
<tr>
<td>12</td>
<td>Identified the location of single Virtual resource [2].</td>
<td>✓ The Identification of six different AWS resources located in US, Europe and Asia is considered.</td>
</tr>
<tr>
<td>13</td>
<td>Used triangle inequality equation to identify the location of virtual resource [2].</td>
<td>✗ Triangle inequality equation requires measuring RTT values from more than three internet nodes to server [2], which is not possible in the experiment setup. However Comparing the actual distance and RTT values with IP address of the AWS resources is performed to allows identifying the location.</td>
</tr>
<tr>
<td>14</td>
<td>Through different internet nodes of Planetlab measured RTT to the Virtual resources [2].</td>
<td>✗ Planetlab node access is not involved in the experiment setup instead authors measured the RTT values of 15KB file/data from single location to AWS resources.</td>
</tr>
<tr>
<td>15</td>
<td>If the virtual resource is moved from one location to the other, RTT values are changed. RTT value increased if the distance between new locations is increased and decreased if the distance is decreased [2].</td>
<td>✓ RTT value of each AWS resources is calculated and compared with each other. Network hops travelling are also considered.</td>
</tr>
</tbody>
</table>
### 6.6.3. Execution

**Actual distance measurement:** As the whole experiment is performed from Karlskrona, Sweden BTH, the node from which the server is created and tested is a single computer that authors developed and deployed web application. Actual distance from the single computer node to the amazon availability zones is calculated in Kilometers. To differentiate the distance between each location a statistical graph representing the Actual distance from BTH to Amazon [38] Availability Zones is shown in KM.

![Actual Distance From Sweden to Amazon Availability Zones in KM](image)

*Figure 12: Comparison of Actual distance from BTH to all AWS Availability Zones*

The above Figure 12 clearly shows which server is closer and far away from BTH, Karlskrona, Sweden. According to the framework 2 [2], the distance should be directly proportional to the RTT responding from server [2]. So the prediction of the RTT value of Europe Ireland should be low when compared to the others servers. Similarly Singapore Asia server responding time should be higher when compared to other servers.

Network hops are the networks passing through, when a request is sent from user location to the server, Traceroute of the server shows the number of networks passing through. As the number of network hops between the server increases RTT value increases [2]. So authors considered that time taken by the server located in US and Asia will be different and depends on the network hops travelling.

Through ping test or traceroute it is possible to capture the response time. Authors captured the RTT value of 30 request count from web browser Google chrome network performance. Now the RTT value of the deployed web application is calculated from BTH, Karlskrona, Sweden. To evaluate the RTT value, a 15KB of file block is accessed from the http link is captured every time. In order to test the accuracy and effectiveness, RTT value for

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Considered/Not Considered</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>Accessed Planetlab nodes and measured RTT between 120 nodes distributed in 28 countries and 75 internet networks [2].</td>
<td>X</td>
</tr>
<tr>
<td>17</td>
<td>Performance depends strongly on location of the involved nodes [2].</td>
<td>✓</td>
</tr>
<tr>
<td>18</td>
<td>Identified the location of virtual resources accurately [2].</td>
<td>✓</td>
</tr>
<tr>
<td>19</td>
<td>Connecting and blocking values in measuring RTT is not considered [2].</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>As the experiment setup does not include planet lab node. Using Google chrome authors captured RTT of 30 request count of deployed application for each AWS resource distributed.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Evaluation of performance is measured by comparing the RTT values internet hops and distance.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>To Identified the location of web application content created AWS resources accurately authors calculated RTT values, IP address, network hops, and actual distance from single location.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Authors consider 30 requests of each AWS resource where there is no packet loss or connecting problem.</td>
<td></td>
</tr>
</tbody>
</table>
15KB of file block is measured for 30 request counts which allow testing the frequency and accuracy of the RTT value.

This measurement depends on the performance of internet due to internet connection problem. The RTT value shows more than expected which on considering may allow identifying location in a wrong way. So RTT value of each count is considered only when the connecting and blocking time is less than or equal to 1 millisecond (ms). A 1ms of time taken for connecting or blocking value may not change the accuracy of identifying data location. The RTT values of 6 amazon availability zone are shown in the Appendix C.

Before deploying the application to Amazon [38] it allows the authors to select the region in the country, i.e. if US Virginia is selected then an option for selecting each region 1b, 1c, 1d is provided, Amazon Web Services [38] also provides no preference which on selecting will create instance automatically in at 1b or 1c or 1d. The regions in each availability zone were allowed to improve the performance of the instance. Authors in this study considered regions as sub regions. Following are the regions provided in each AZ where authors deployed the application and captured the RTT value.

- US East Virginia (1b, 1c, 1d)
- US West Oregon (2a, 2b, 2c)
- US West California (1a, 1b)
- Europe Ireland (1a, 1b, 1c)
- Asia Pacific Tokyo (1a, 1b)
- Asia Pacific Singapore (1a, 1b)

With the help of RTT values authors have identified the data location of the Web application content, by capturing the 15KB of file block each time. There are many files in the deployed web application that can be measured but, capturing a single file block that is accessed from database is considered which holds the web application. Here accessing the database is logging in with user details in the application or signup for the application, because the data reads or writes from the RDS server.

Before verifying the location of the resources authors with the help of RTT values confirmed that the above sub regions are located in the same location i.e. for example US East Virginia 1b, 1c and 1d sub regions are located in the same region by measuring the RTT value. To verify this assumption authors measured and compared RTT values of sub regions within the location and proved it by statistical measurement. All the RTT values of sub regions are compared and shown in the below Figures 13-18.

**US East Virginia (1b, 1c and 1d)**

Below Figure 13 shows the RTT value of 30 request count of each sub region of US Virginia 1b, 1c and 1d responded with more or less nearly at a same time, this shows that the sub regions reside on the same location.
Above Figure 14 shows that the sub regions of California 1a and 1c respond nearly at same time, this allows to know that both the sub regions reside at the same place.

Above Figure 15 showing the RTT value of US west Oregon 2a, 2b and 2c respond at the same time, hence it is proved that the sub regions located at the same location.

Above Figure 16 shows the RTT value of Europe Ireland 1a, 1b and 1c respond nearly at the same time, this allows to know that sub regions are located in the same location.
The above Figure 17 shows that the RTT value of two sub regions resides at the same location as the RTT values are nearly similar.

The above Figure 18 shows that RTT value of both the sub regions is between 1000ms to 1500ms; this shows that the sub regions of servers in Singapore located at two different places.

Now authors came to a conclusion with the above six (Figures 13-18) that all the sub regions are located within the same location as the performance of each sub regions are similar. As the RTT value also depends on the networks hops travelling to reach the server, authors also calculated the network hops of each server. The below table shows the number of hops travelled when Average RTT value of 30 request count is calculated from BTH, Karlskrona, Sweden to AWS availability zone. In addition to this IP addresses for all AWS availability zones are calculated and the Whois IP [104] which shows the information regarding the location of the IP address is also identified.

**Table 7: Average RTT values, network hops and actual distance of each AWS server**

<table>
<thead>
<tr>
<th>Input Region</th>
<th>Average Output response time of 15kb (RTT) in ms</th>
<th>Actual distance to the server in KM</th>
<th>No. of Network hops travelled</th>
<th>IP address</th>
<th>Whois IP information [104]</th>
</tr>
</thead>
<tbody>
<tr>
<td>US</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oregon</td>
<td>582.6667ms</td>
<td>8129KM</td>
<td>20</td>
<td>50.112.105.171</td>
<td>US San Jose</td>
</tr>
<tr>
<td>California</td>
<td>524.8333ms</td>
<td>8787KM</td>
<td>18</td>
<td>184.72.56.25</td>
<td>US California</td>
</tr>
<tr>
<td>Virginia</td>
<td>390.1ms</td>
<td>6870KM</td>
<td>17</td>
<td>50.19.225.216</td>
<td>US Virginia</td>
</tr>
<tr>
<td>Tokyo</td>
<td>949.7ms</td>
<td>8530KM</td>
<td>21</td>
<td>176.34.48.73</td>
<td>Asia Tokyo</td>
</tr>
</tbody>
</table>
When resources are located nearer to each other it is not possible to identify the data location of particular server. To overcome this issue authors considered the network hops when the sites are nearer to each other and the RTT values are nearly similar. In the above table network hops travelled are considered when the resources are placed nearer to each other i.e. nearly 500KM. Network hops also respond depending on the country travelling i.e. the network hops responding to the request in US will be different with the network hops responding to the Asia.

Now to prove that whether each server located in US, Europe and Asia are running in the same location, the RTT values of each server are compared and shown in the below Figure 19. Here Authors considered one sub region from each location i.e. US Virginia has sub regions 1b, 1c and 1d, so authors considered 1b values from US Virginia. Similarly the first sub region of each server values is considered. The graph in Figure 19 represents the frequency of each server responding for 30 request count.

<table>
<thead>
<tr>
<th>Region</th>
<th>Location</th>
<th>RTT (ms)</th>
<th>Distance (Km)</th>
<th>Average RTT Values</th>
<th>IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia</td>
<td>Singapore</td>
<td>1084.5ms</td>
<td>9781KM</td>
<td>26</td>
<td>46.137.222.70</td>
</tr>
<tr>
<td>Europe</td>
<td>Ireland</td>
<td>160.3667ms</td>
<td>1550KM</td>
<td>19</td>
<td>54.247.167.7</td>
</tr>
</tbody>
</table>

Figure 19: Graph representing RTT values of all AWS resources

The comparison of the RTT values and the actual distance is shown in the below Figure 20.

![Figure 20: Estimating geolocation by comparing actual distance with RTT values](image-url)
6.6.4. Identifying the data location of the deployed web application in AWS servers
A detailed description of how authors identified the data location of deployed web application in US, Europe and Asia is as follows:

**Europe:** The expected response time of the *Europe Ireland* server should be very low when compared with all the remaining servers as the distance to Europe Ireland is very less when compared to the remaining servers. It is clearly shown in the above *Figure 19* and *Figure 20* that the Europe server on an average takes RTT value of 160.3667ms. It is certain that the file block tested is responding from Europe Ireland.

**USA:** The expected response time of the *US Virginia* server should be less when compared to US Oregon, US California, and Asia Tokyo and Singapore server. But the RTT value should be more than Europe Ireland server as the distance from BTH to Ireland is low when compared with distance from BTH to US Virginia. It is clearly shown in above *Figure (19, 20)* that US Virginia server on an average takes RTT of 390.1ms which is less when compared to all the servers except Europe. And more when compared with Europe Ireland server so authors conclude, it is certain that the file block tested is responding from US Virginia server.

The expected response time of the *US California* server should be more when compared with Virginia server. But here there is another server US Oregon which is located approximately 658KM away from California. Identifying the data location of server located within 500KM distance response similarly [2]. Authors expect the RTT values of these two servers would be nearly same. Now to identify the DL of each server authors considered the network hops travelled for each server and expected that the RTT value depends on the network hops travelled i.e. Oregon server travelled 20 network hops while California travelled 18 hops, so the expected RTT value of the server Oregon should be more when compared with California although the distance is more. The average RTT value of California is 524.8333ms and Oregon is 582.667ms which shows that Servers are located in California and Oregon.

**Asia:** The expected RTT value of Servers located in Asia are, Singapore server response time should be more when compared with Tokyo as the distance from Tokyo and Singapore is more than 1000KM. Authors also considered network hops for both the servers where Singapore server travels 26 hops while Tokyo travels 20 hops. This clearly allows expecting the RTT value of Tokyo to be less when compared with Singapore. The Table 7 shows that the average RTT value of Tokyo is 949.7ms whereas the RTT average value of Singapore is 1084.5ms, which shows that both the servers located at Tokyo and Singapore. Authors estimated and identified the DL of servers at Singapore and Tokyo with the help of measuring RTT values [2] and network hops and actual distance.

6.6.5. Analysis of the framework whether satisfying in our experiment setup
Authors by following the framework activities identified the DL of deployed application in Amazon. The activities satisfied in the experiment are now assigned with rubrics and shown in the below Table 8.

<table>
<thead>
<tr>
<th>S.no</th>
<th>Activities considered regarding Data Location</th>
<th>Framework</th>
<th>Experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Performed test by using cloud services to identify data location [2].</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>To ensure without the need of trusting the cloud operator, identified data location [2].</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Successfully identified the location of Amazon Cloud resources.</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>Through Round Trip Time (RTT) values the estimated the geolocation of virtual resources [2].</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>Measured RTT values from internet locations to the virtual resources [2].</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>Considered Virtual resource as a server which is able to respond to RTT request [2]. Virtual resources include RDS, EC2 and S3 service.</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>Identify that cloud resource directly connected to the internet [2].</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>Identified and proved the geolocation and transparency of each AWS resource placed in different locations (USA, Europe and Asia)</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>The change of location of server is identified through the RTT measurement [2].</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>Distance between the virtual resource and internet node is directly proportional to the RTT value between them [2].</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
The computation of Vivaldi coordinate system is based on RTT time between different internet nodes to the virtual resources [2].

Identified the location of six different AWS resources located in US, Europe and Asia.

Comparing the actual distance and RTT values with IP address of the AWS resources to identify the location.

Measured the RTT values of certain file/data from single location to AWS resources [2].

If the virtual resource is moved from one location to the other, RTT values are changed. Here RTT value increased if the distance between new locations is increased and decreased if the distance is decreased [2].

Accessed Planetlab nodes and measured RTT [2].

Performance depends strongly on location of the involved nodes [2].

Identified the location of all AWS resources by calculating RTT values, IP address, network hops, and actual distance from single location.

Accurately identified the data location servers located nearer to each other [2].

Connecting and blocking values is not considered [2].

6.6.6 Weakness of the Framework

By following the framework activities, authors identified the data location of the deployed web application from the end user perspective i.e. without the need of cloud provider for example Amazon. However this framework also has some drawbacks for identifying data location of Web application content, they are as follows:

- Identified location of web application content by measuring a 15KB of file from the Web application. The identification of each and every file block is not considered.
- Framework does not allow to identify each resource separately i.e. EC2, RDS, S3.
- RTT value depends on the performance of internet which may sometime give late response due to internet connections [2].

This framework 2 [2] allowed identifying the data location of the server created at US, Europe and Asia but there is a threat that data may replicate in different locations though the server is located in one location. Identifying the data location of replicated data is done by following the framework 3 [3] activities, where latency is measured to identify the data location [3].

6.7. EXPERIMENT - 3

The web application deployed in six AWS availability zones (US Virginia, California, Oregon, Europe Ireland, Asia Singapore and Tokyo) is again evaluated by performing the activities of the framework [3]. The main difference between the experiment 2 and this experiment 3 here is latency values of the AWS resource is measured instead of measuring the RTT values. In addition to this authors investigated the data location of files replicated in different geolocation. Before performing the framework activities a brief description of the existing framework is shown below Section 6.7.1.

6.7.1. Framework - 3 Description [3]

Karyn Benson et.al in [3] proposed a theoretical framework to identify the origin of data and also verified the geolocation when Cloud provider replicated the data in different location. This framework 3 [3] verifies the data location Amazon CloudFront nodes where data replicated in different locations in US are identified by measuring the Latency values from different Planet Lab nodes. The framework 3 [3] is proposed based on the previous work which allows to verify the data location of the server [20]. The framework 3 [3] allows verifying the file block replicated in US. Amazon Simple Storage Service S3 is selected as Cloud resource and connected it to the Amazon CloudFront [60], and verified the geolocation of CloudFront edge locations where data is replicated. The theoretical framework 3 [3] is proposed by considering the assumptions that data centers of the replicated data is known before estimating and the internet connections between data centers are not established and each data center is accessed with access machines near the data center [3].
Unidirectional network latency is calculated from Planetlab nodes to the AWS resources [3]. The main concepts proposed in this framework are construction of the verification system [3], identifying the geolocation of data by measuring latency [3], evaluating geolocation estimating techniques [3], distance estimation techniques and effect of the network nodes travelled between server and internet location [3]. The estimation of the sites located within 500KM distance was proved [3].

To evaluate this framework in experiment setup some of the factors may differ from the existing framework. The factors that are considered in the experiment setup is as shown in the below Table 9, comparing the factors considered in the framework. Factors considered similar in the experiment are marked as (‘✓’) and factors that are not possible in the experiment setup are marked as (‘✗’), however authors describe the factors consider similar to the framework in experiment section.

<table>
<thead>
<tr>
<th>S.no</th>
<th>Framework 3</th>
<th>Experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Verified the location of data in Amazon cloud services [3].</td>
<td>We setup Amazon cloud service to test data location.</td>
</tr>
<tr>
<td>2</td>
<td>Verified the geolocation of S3 service in Amazon [3].</td>
<td>Verified the geolocation of RDS, S3 and EC2 services provided by Amazon.</td>
</tr>
<tr>
<td>3</td>
<td>Framework helps cloud users to verify the property themselves without CP [3].</td>
<td>Authors followed the theoretical framework of the deployed web application content in AWS resource without the use of AWS functionality.</td>
</tr>
<tr>
<td>4</td>
<td>The time taken by the server to respond a request sent depends on the distance between the sites [3].</td>
<td>This assumption, distance between sites should be directly proportional to latency of the server is considered.</td>
</tr>
<tr>
<td>5</td>
<td>To overcome complications related to variance in network latency authors verified a random block of 64KB in the request [3].</td>
<td>As the network latency of each and every file blocks in the deployed web application is more complicated and vague to test, a network latency of 15KB random file block request is captured.</td>
</tr>
<tr>
<td>6</td>
<td>Proposed framework uses the idea of imposing time constraint but in a different way [3].</td>
<td>Using time constraint for identifying data location experiment is conducted.</td>
</tr>
<tr>
<td>7</td>
<td>Identified the data replication in cloud [3].</td>
<td>Identification of data replicated when CloudFront service is connected to the web application created in Europe Ireland.</td>
</tr>
<tr>
<td>8</td>
<td>Geolocation of data can also be identified by domain registration record or IP address [3].</td>
<td>Identified the geolocation of the Amazon RDS, S3 and EC2 by IP address. For example domain name of RDS located in US shows(amazon.ciaxf0hp1xbw.us-east-</td>
</tr>
<tr>
<td>9</td>
<td>Using Traceroute per-hop latencies are also captured to geolocate the target and intermediate routers [3].</td>
<td>Using traceroute authors captured No.of hops passing through to reach the server.</td>
</tr>
<tr>
<td>10</td>
<td>The locations of cloud data centers are well known and data is stored within the given region [3].</td>
<td>Locations of all AWS resource US, Asia, Europe are well known.</td>
</tr>
<tr>
<td>11</td>
<td>Compared distance and network latency of the two servers located in different geolocation and estimated the location [3].</td>
<td>Compared the actual distance from internet to all AWS resources and latency of each server and estimated the geolocation of each AWS resource accurately.</td>
</tr>
<tr>
<td>12</td>
<td>It is not possible for estimating data replication of servers located close to each other but still able to identify the location [3].</td>
<td>Identification of the location of servers located close is possible. For example Servers located in Oregon and California is very close.</td>
</tr>
<tr>
<td>13</td>
<td>Used 40 planetlab nodes to identify two Amazon S3 servers located in US [3].</td>
<td>Experiment setup does not consist of Planetlab node access. Verified the location of Amazon servers EC2, S3, and RDS located is Us VG, US</td>
</tr>
</tbody>
</table>
### 6.7.2. Experiment Setup for replicating framework 3 [3]

The same environment deployed in Amazon availability resources [38] that was used for evaluating framework 1 and 2 is again evaluated in this experiment by following the framework 3 [3] activities to identify the data location. Latency values of the AWS server are considered in this experiment in order to identify the data location and replicated data. Latency is the time taken for the server to respond the request i.e. one way latency value is considered [3]. The geolocation of the server is also estimated based on the latency values [3]. After verifying the location of all AWS resources, authors changed the experiment setup slightly by connecting CloudFront service [60] to the Europe Ireland server and identified the replicated data.

### 6.7.3. Execution

By following the proposed framework 3 [3] authors measured the haversine distance (internet locations Sweden, BTH to Amazon availability zones), server name and latency of the 15KB file block from deployed web application. According to the framework, latency and distance should have positive relationship [3]. The expected Latency from Sweden to the AWS server should be high if the distance between sites is more i.e. latency value of Singapore (9781KM)>California (8787KM)>Tokyo (8530KM)>Oregon (8129KM)>Virginia (6870KM)>Ireland (1550KM). The actual distance measurement and latency (30 request counts) of each Amazon availability zones is shown in the Figure 12 and Figure 21.

### 6.7.4. Estimating the data location of deployed web application in AWS

The deployed web application in six AWS resources is well known [38]. To verify the location of the web application content of each resource in Amazon, authors measured the latency value of 15KB of file block for 30 request count. The latency values of each request count to AWS resource is shown in Appendix D. The frequency of latency value of each resource located in different regions is clearly shown in the below Figure 21.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>Haversine distance gives the best prediction of geolocation [3].</td>
<td>✓</td>
</tr>
<tr>
<td>15</td>
<td>Errors occurred in capturing latency sometimes gives an unpredictable value, this latency value is due to the internet connection problem [3].</td>
<td>✓</td>
</tr>
<tr>
<td>16</td>
<td>Measured latency of HTTP request of a file block from reliable planetlab nodes to the server [3].</td>
<td>x</td>
</tr>
<tr>
<td>17</td>
<td>Identified the location of data within the country (US) [3].</td>
<td>✓</td>
</tr>
<tr>
<td>18</td>
<td>Used Latitude and Longitude of the target location to estimate distance and latency [3].</td>
<td>✓</td>
</tr>
<tr>
<td>19</td>
<td>Detected changes in the location of the server [3].</td>
<td>✓</td>
</tr>
<tr>
<td>20</td>
<td>The prediction is always correct when the distance is more than 1500KM [3].</td>
<td>✓</td>
</tr>
</tbody>
</table>

Oregon, Us California, Europe Ireland, Asia Singapore and Asia Tokyo.

Haversine distance is used to predict the geolocation and this satisfies in our experiment setup.

We excluded the latency values when the DNS lockup and blocking values are more than 1.

Measured latency of 30 HTTP request count from one single location Sweden, BTH.

Identified the location of data present in Asia, Europe and US

Latitude and longitude of the AWS resources and single node from where authors test (Sweden, BTH) is calculated and estimate the distance between sites.

Detected changes in the location when comparing the latency values of all the AWS resource located in different locations.

The latency of servers located far away from each other considered and compared.
Figure 21: Comparison of Latency values of all AWS resources

Above Figure 21 shows that frequency of latency values of Europe Ireland server is very less when compared to other servers, as the distance from request sent i.e. Sweden BTH is nearer when compared with other country. This allows estimating that the web application deployed is located in Europe Ireland. Similarly servers located in Tokyo and Singapore respond with high latency as the distance is more and Virginia server respond to less latency when compared with Oregon and California as the distance to Virginia is less. The frequency of Oregon and California servers respond with nearly same latency value as they are located close to each other i.e. less than 1000KM. By considering the latency value of each server for 30 request count allows to estimate that Amazon resources located in three different continents. However there is not much difference between identifying the data location of server through RTT and Latency values, the frequency of values changes more with RTT values as it depends on the total time taken. But the latency values provide fewer changes in the frequency of the request which helps to identify data location of servers accurately.

6.7.5. Estimating the geolocation by comparing latency values with haversine distance [3]
The average latency value from 30 request count is calculated and compared with the haversine distance of each AWS resource. “Haversine distance is the distance between two points on a sphere” [3]. The average latency values of each resource is shown in the below Table 10.

<table>
<thead>
<tr>
<th>Regions</th>
<th>Average latency values of 30 request count</th>
<th>Haversine Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Singapore</td>
<td>322.8</td>
<td>9781</td>
</tr>
<tr>
<td>Tokyo</td>
<td>278.4667</td>
<td>8530</td>
</tr>
<tr>
<td>Virginia</td>
<td>136.133</td>
<td>6870</td>
</tr>
<tr>
<td>Oregon</td>
<td>192.1</td>
<td>8129</td>
</tr>
<tr>
<td>California</td>
<td>183.7667</td>
<td>8787</td>
</tr>
<tr>
<td>Ireland</td>
<td>57.733</td>
<td>1550</td>
</tr>
</tbody>
</table>

Now the average latency values and haversine distance is compared to estimate the geolocation. Below Figure 22 represents the haversine distance in Kilometers at left side representing with bar graphs and latency values at right side representing with red mark points for all the AWS resources.
Figure 22: Estimating the geolocation by comparing distance with latency values

Figure 22 shows that all six Amazon servers located in different regions are responding according to the distance. The blue bars represent haversine distance from BTH to US continent, red bar represents the haversine distance to Europe and the green bars represents the distance to Asia continent. The graph represents that California and Oregon respond nearly same as they are located in the same country US. When compared to distance and latency of California and Tokyo. The distance is similar but the latency value of Tokyo is more than California, because the servers located at different country respond in different way. The graph allowed in estimating the geolocation of the web application content deployed in different Amazon availability zones.

6.7.6. Creating CloudFront nodes to the deployed application

The performance of the server depends on the distance between the server and endpoint [3]. In order to improve the performance by reducing the latency values, Amazon provides CloudFront service [60] where the cache files of the web application will store at the nearest node to the user [60]. Here files from the main server are replicated at nearest edge location nodes, so it is necessary to identify the data location of files replicated at different nodes. To verify the replicated files authors used the CloudFront service to the deployed application in Europe Ireland. The verification of the replicated data from deployed web application is performed by capturing the following values that are captured in framework 3 [3]:

- Latency
- network hops
- Host name
- IP address
- Distance

Here Europe Ireland server is considered for evaluating the replicated data, as the CloudFront node routes to the nearest location even though the servers are located at different locations. Authors captured the events two times to verify the replicated data of deployed web application.

(a) Without connecting CloudFront Node to the deployed application

Here the web application deployed in Europe Ireland is considered and captured the events without connecting CloudFront services [60]. Below Table 11 represents the latency, hostname, Whois country [104], IP address and haversine distance when web application is deployed at Europe Ireland server. Here latency value is captured by using ping and traceroute test.
### Table 11: Events captured without connecting CloudFront

<table>
<thead>
<tr>
<th>Deployed Region</th>
<th>Haversine distance</th>
<th>Network Hops</th>
<th>IP address</th>
<th>Whois Country</th>
<th>Host name</th>
<th>Latency (avg/max/min)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>79.125.1.98</td>
<td>Europe Ireland</td>
<td>ec2-54-247-109-248.eu-west-1.compute.amazonaws.com</td>
<td>61ms/81ms/56ms</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>178.236.0.221</td>
<td>Europe Ireland</td>
<td>Amazon Data Services Ireland Ltd</td>
<td>60ms/62ms/59ms</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>178.236.0.202</td>
<td>Europe Ireland</td>
<td>Amazon Data Services Ireland Ltd</td>
<td>60ms/62ms/59ms</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>178.236.0.204</td>
<td>Europe Ireland</td>
<td>Amazon Data Services Ireland Ltd</td>
<td>60ms/62ms/59ms</td>
</tr>
</tbody>
</table>

The above table 11 shows that the captured IP address of the server are located at Europe Ireland and Host name of the server shows the server is in Europe west i.e. Ireland, this allows estimating data location of web application content reside in Europe Ireland. Total number of network hops travelled is 19, and the latency of each amazon host IP address shown in the above table is nearly similar. In addition to this IP addresses for all AWS availability zones are calculated and the Whois IP [104] shows the information regarding the location of the IP address. Now authors created a CloudFront node [60] to the deployed web application to identify the replicated data.

### (b) With CloudFront Node to the deployed application:

Authors created a CloudFront distribution to the load balancer of the deployed web application; load balancer [58] here automatically scales incoming application traffic across multiple Amazon EC2 instances. Creating amazon CloudFront [60] distribution from load balancer took nearly 10 minutes. After successful deployment of CloudFront node [60] authors verified the new distribution link. Now the latency, network hops and IP address of the new distribution link is captured.

Amazon provides the global edge networks throughout the world to reduce the latency and improve the performance [60]. Amazon provides 12 CloudFront locations at United States i.e. (Ashburn VA, Dallas/Fort Worth TX, Jacksonville FL, Los Angeles CA, Miami FL, New York NY, Newark NJ, Palo Alto CA, San Jose CA, Seattle WA, South Bend IN, St. Louis MO), 8 edge locations at Europe (Amsterdam, Dublin, Frankurt, London, Madrid, Milan, Paris and Stockholm), 5 edge locations at Asia (Hong Kong, Osaka, Singapore, Sydney and Tokyo) and one edge location at South America Sao Paulo.

As the latency values and traceroute of the deployed application connected to CloudFront [60] distribution is captured from a single internet location Sweden BTH, authors expect the web application files should store at the nearest edge location Stockholm.

### Table 12: Events captured after connecting CloudFront

<table>
<thead>
<tr>
<th>Deployed Region</th>
<th>CF Location</th>
<th>Haversine distance</th>
<th>Network Hops</th>
<th>IP address</th>
<th>Whois Country</th>
<th>Host name</th>
<th>Latency (avg/max/min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europe Ireland</td>
<td>Best performance node (US or Europe or Asia)</td>
<td>1550KM</td>
<td>12</td>
<td>205.251.219.171</td>
<td>US Washington, Seattle</td>
<td>Server-205-251-219-171.arin1.r.com</td>
<td>15ms/18ms/15ms</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>194.68.123.47</td>
<td>Sweden, Stockholm</td>
<td>Netnod-ix-ge-a-sth-1500.amazonaws.com</td>
<td>15ms/17ms/14ms</td>
</tr>
</tbody>
</table>
The Table 12 represents that request travelled through 12 network hops. The IP address and Whois country [104] information shows that the request sent is responding from the node located at US Washington, Seattle and the next network passing the IP address located at Stockholm, this is because when the CloudFront distribution is created at first the location of the created CloudFront node is at US and then based on the requests sent from the destination (BTH Sweden) the location of CloudFront node is shifted to the nearest Amazon CloudFront location of Europe i.e. Stockholm. This allows estimating that the data is replicated at US Washington, Seattle and Sweden Stockholm. Here the latency values of the web application (average latency 15ms) with CloudFront are very less when compared with the latency values of web application without CloudFront distribution (average latency 62ms). The distance from Ireland to Sweden BTH (1550KM) is far away, when compared with distance from BTH to Stockholm (381KM). CloudFront Node created at US Washington is far away from Europe Ireland, the files of the web application are redirected to the Stockholm to improve performance and reduce the latency. Authors estimated data location of replicated data files when the web application deployed is connected to CloudFront service [60].

6.7.7. Analysis of the framework 3 [3] whether satisfying in our experiment setup

By following the framework 3 [3] activities in the experiment setup authors estimated the data location of the web application content deployed in Amazon availability zones [38]. The activities satisfied in the experiment setup are now assigned with rubrics as shown in the below Table 13.

Table 13: Activities satisfied regarding data location in framework 3 [3] and experiment setup

<table>
<thead>
<tr>
<th>S.no</th>
<th>Activities considered regarding Data Location</th>
<th>Framework</th>
<th>Experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Performed test by using cloud services to identify data location [3].</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Verified the geolocation of web application content RDS and EC2 resources</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Verified DL without the use of Amazon functionality [3].</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>The latency values of the server to respond a request sent depends on the distance between the sites [3].</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>The variance in network latency is verified through a fixed random block for each the request [3].</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>Geolocation of data can also be identified by domain registration record or IP address [3].</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>Identify the data location of replicated data [3].</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>Using traceroute per hop latency and number of networks nodes passing through is calculated [3].</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>Comparing latency values with haversine distance identify the data location [3].</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>Evaluated the performance and identified location Amazon CloudFront edge location at Europe.</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>Measured the latency values of certain file/data from single location to AWS resources [3].</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>Access Planetlab nodes and measured latency of Amazon [3].</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>Performance depends strongly on location of the involved nodes [3].</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>Accurately identified the geolocation of servers located nearer to each other [3].</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>Connecting and blocking values in measuring latency is not considered [3].</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Authors identified the data location of replicated files, when the web application deployed is connected to Amazon CloudFront distribution [60]. The files replicated are located at the nearest edge location Stockholm.
Authors in this study identified the DL of the web application content deployed in US, Asia and Europe by following the activities of the framework 3 [3]. Identifying the data location of the replicated data when web application created in Europe server is connected to the CloudFront Distribution [60] is also proved in the experiment setup by following the framework activities.

6.7.8. Weakness of the framework
The data location of the web application content is identified by following the framework 3 [3] activities. However there are some drawbacks that effect identifying the data location of web application content. The following are the weakness of the framework activities that authors observed while evaluating it in the experiment setup:

- Latency value of the deployed web application depends on the internet which allow in estimating wrong location [3].
- Capturing the latency values of 30 request count helps to identify the data location within a specific period of time.
- When a request sent to the server, the server may respond by passing different internet routes located at different places. Here the network passing through to respond a request may not be equal to haversine distance.

Authors conclude that data location of servers located in US, Europe and Asia of AWS are identified by following the framework 3 [3] activities. In this Experiment 3 authors also identified the location of replicated data of the deployed web application.

6.8. DATA VALIDATION
The values and events captured from three experiments are considered only after verifying that the web application performing all the activities. Some times after deploying the application authors faced some problems while fetching data from Database i.e. monitoring information shows that environment is healthy but when signing into the web application shows error while connecting to data base. Authors have redeployed and verified that the web application functionalities are performing well and there is no error connection problem.

More than 60 request counts were performed to capture RTT and latency values, authors considered only the values where the connecting and blocking values are less than 1ms. However authors examined the values of each server and monitoring events 30 days, to validate the results in statistical measurement 30 request counts were considered.

6.8.1. Analysis and Interpretation
Data localization of the deployed web application in US, Europe and Asia AWS resource [3] proved that Amazon is maintaining the resources according to the given preference. By performing three treatments (Framework 1, 2 and 3 [1] [2] [3] to the experiment setup authors were able to identify the DL of the deployed web application efficiently. Here the efficiency of identifying the DL of the framework activities is measured by assigning rubrics to the activities performed for each and every framework [1] [2] [3] in Table 5, 8 and 13. Authors conclude that the three framework [1] [2] [3] activities are satisfied in the experiment setup to identify data location. However the three frameworks help in identifying the DL of the deployed web application in three different ways.

6.8.2. Hypothesis Testing
This chapter describes in detail about the power of hypothesis formulated for conducting the experiment. The experiment setup consists of both descriptive and numerical results, to evaluate the efficiency of DL frameworks [1] [2] [3] by identifying the data location of deployed web application in US, Europe and Asia. Authors assigned rubrics by performing each and every framework [1] [2] [3] activities to test the hypothesis.

Null hypothesis $H_0$ The two sub hypothesis formulated resulted in power hypothesis after conducting the experiment setup. Replicating the data location frameworks [1] [2] [3] is possible in Experiment setup by considering the factors from framework, here $H_{1b}$ null hypothesis is rejected and the alternative hypothesis $H_{1a}$ is accepted. $H_{2b}$ activities performed from framework [1] [2] [3] in the experiment setup does not make any change in identifying location of data, but after conducting experiment the activities performed in the experiment setup allowed to identify data location hence $H_{2b}$ null hypothesis is also rejected and alternative hypothesis $H_{2a}$ is widely accepted in the experiment setup.

Authors assumed the null hypothesis $H_0$ before conducting the experiment. Here the null hypothesis is rejected as the data location of the web application deployed in US, Europe and Asia is visible by following the three
framework [1] [2] [3] activities. The tables 5, 8 and 13 shows the comparison between the satisfied framework activities and experiment setup, here the comparative results show that experiment setup performed activities are effective and satisfied with the framework 1, 2 and 3 [1] [2] [3] activities respectively close enough regarding data location. Finally authors conclude that the null hypothesis $H_0$ is rejected and the alternative hypothesis is accepted $H_1$.

**Alternative Hypothesis $H_1$:** By performing the activities of framework 1, 2 and 3 [1] [2] [3] allows identifying the data location of web application deployed in US, Europe and Asia where framework activities are compared with the experiment satisfied activities.

### 6.9. VALIDITY THREATS

Many security issues should be considered when data is migrated to the cloud [1]. Identifying DL of the migrated data is one of the security measure that Cloud user should monitor and manage [1]. In order to answer the Research Question RQ2 the experiment is carried out where authors acted as Cloud user and AWS [38] as a Cloud Provider (CP). This section describes how experiment results were established to have decent validity. There are four types of classification themes for experiment setup and they are: conclusion validity, internal validity, construct validity and external validity [66]. Even though the authors performed the experiment and identified the data location of AWS resources located in US, Europe and Asia, there are some threats while conducting each framework [1] [2] [3] activities. However authors have reduced the threats to minimum, a detailed description of each threat and its mitigation strategies are explained in this chapter. The threats of three experiments performed in this study are as follows

#### 6.9.1. Conclusion Validity

Validating the relationship between, evaluating DL frameworks in all possible AWS availability zones, and the outcome of the results observed regarding DL here shows the conclusion validity. Authors considered the factors from each framework [1] [2] [3] and executed it in Amazon Cloud environment.

- Some of the framework [1] [2] [3] activities may not be possible in the experiment setup, which may result in poor identification of data location by performing the activities. However authors performed the activities similar to the framework activity and identified the DL efficiently.
- The efficiency of DL frameworks [1] [2] [3] evaluated in the experiment is measured by providing rubrics to the satisfied factors in experiment.
- The activities considered from framework [1] [2] [3] in the experiment shows the replication of study but not the replication of the results. As the environment of the experiment is different from the frameworks [1] [2] [3].
- The study of these frameworks [1] [2] [3] in single controlled experiment environment shows the consolidated results of the three DL studies.
- Identification of data location of servers created in US, Europe and Asia also depends on the performance of the server created and the monitoring information provided by Amazon [38]. Authors however used micro instances for all the servers created, so the assumption of estimating data location is also similar.
- After terminating the environment the data from web application is not completely deleted from cloud. The data may reside in that location even after termination. Authors after evaluation of the three frameworks deleted even the files and terminated environment.

#### 6.9.2. Internal Validity

By deploying the web application at three different continents US, Europe and Asia provided by AWS, evaluating the framework [1] [2] [3] activities however shows that problem with internal validity in the experiment setup is minimum.

- Identifying the frameworks [1] [2] [3] for data location in Cloud Computing by conducting a Literature Review misses some of the data location frameworks. However authors followed a formal approach by using search strings on different data bases and also the evaluation of the three data location frameworks generalizes the existing data location frameworks for some extent. Authors in addition to this also identified the most recent research done on data location in Cloud Computing i.e. considered the frameworks that are published after the year 2011. This allows minimizing the threat on conducting Literature Review.
• Elastic IP (Static IP address) is preferred before creating server which allows managing the server only from a single IP [101]. If the server is created from one IP then modification of the server is not possible from other IP. This shows that authors have performed the experiment from a single internet location.

• Deploying web application in all AWS availability zones and testing the efficiency of DL frameworks [1] [2] [3] may allow to evaluate the performance of AWS resources regarding DL and also shows the effectiveness and efficiency of DL frameworks [1] [2] [3].

• Each and every value and information acquired by conducting the experiment are captured and then the values are noted.

• It is necessary to monitor the information of deployed web application when the application is updated and terminated. Authors however monitored the information of the deployed web application when application is redeployed and even when the application is terminated.

• Even though the connecting and blocking values of the request count are less than 1ms, the latency and RTT values still change (i.e. first two request count is 604 and 560). For this authors considered the average of 30 request count.

• A latency value also depends on the internet connection and response time took for each node. However authors considered the latency values when the packet loss of each node is less than 1ms.

6.9.3. Construct Validity

Even though the experiment setup consists of web application developed by authors, but not the real time Cloud web application, the behavior and the results of the experiment environment actually represents the real time Cloud web application regarding DL. The major construct validity here Amazon provides Availability zones to create resources. The location of the deployed web application is known before conducting the experiment. Merely there is need to verify the DL, while a Cloud user migrate the data to Cloud [2] [3]. However authors could verify that whether the DL of the resources created in three continents reside in given preferences.

• Conducting the whole experiment from a single location may not get a complete view on data location of AWS resource. Authors however estimated the geolocation of AWS resources by performing the activities several times to different servers.

• The evaluation of three framework [1] [2] [3] activities may result in different way for different resources created in AWS. However authors considered whether the servers created in US, Europe and Asia reside in the respective region.

• The behavior of servers created at three continents US, Europe and Asia from the Sweden BTH is allowed to estimate the performance of servers located in different places.

• To control the web application deployed in AWS availability zones the experiment is performed but the framework [1] [2] [3] activities are different from each other which may not hold the complete web application content regarding data location. However by implementing the three frameworks [1] [2] [3] of three different scenarios may improve the control on web application regarding data location.

• Activities performed for evaluating framework 1 [1] are captured by accessing Management console provided by AWS, however the monitoring information provided by Amazon [52] contain latency and performance of the server, the accuracy of identifying the latency values and performance may not be measured. So authors performed activities by measuring RTT and Latency value from web browser for evaluating framework 2 [2] and framework 3 [3].

• Identifying the DL by replicating study depends on the activities performed from the framework [1] [2] [3], i.e. the result of three frameworks may not be similar but however the results prove the estimating DL.

• Some of the activities in the framework 2 [2] and framework 3 [3] are not considered in the experiment setup, due to lack of resources in the experiment setup. This may not allow identifying the DL of the deployed web application efficiently. However authors could identify the DL of AWS resources by performing the activity related to the framework. One of the activities that were not performed in the experiment setup is verifying the DL by measuring response time from Planet Lab nodes distributed in different locations. However authors in this experiment study considered response time from single location to different cloud resources, the comparison of the response time came from different resources allows identifying the geolocation of servers.
The data location of the deployed web application is verified by capturing the monitoring events provided by Amazon management console. Here the monitoring information should be examined from start of deploying the application to termination. Authors have examined the events during the experiment which took nearly 30 days.

6.9.4. External Validity

The results are based on the deployed sample web application in AWS resources. But these results are likely to generalize for the Cloud users to manage and deploy web application in different locations. The importance is to evaluate the three frameworks in a cloud environment and validate the efficiency of DL frameworks.

- The frameworks [1] [2] [3] are validated in real Cloud environment i.e. Amazon Web Services and the efficiency of identifying the DL can be generalized in Cloud Computing.
- The evaluation of framework [1] [2] [3] may not be confirmed to a single test for a specific period of time. Authors however to prove the effectiveness of DL frameworks [1] [2] [3], performed the activities several times and performed the experiment for 30 days.
- The evaluation of frameworks [1] [2] [3] is done on resources EC2, S3 and RDS only. However the results of evaluation of these services show that Amazon performs all remaining the services similarly regarding data location.
- RTT values depend on the internet connection, by considering this it may affect in estimating the geolocation of the AWS resources. Authors have considered the RTT value of the 30 request count for 15KB of file block when the blocking, connecting and packet loss values are less than 1ms.
- Capturing the latency values from the web browsers depends on the performance of browser, Authors considered the latency values by using a single browser Google chrome.
- The latency and RTT values of the server also depend on the performance of the server created in Cloud. However authors used micro instance with internal memory 512mb for creating the resources to deploy web application in US, Europe and Asia. So the performance of the AWS resources is compared to estimate the geolocation (performance here is measuring the latency and RTT value of the server).
- The performance of the internet bandwidth also affects the latency values. Authors measured a constant bandwidth i.e. 10mbps while conducting experiment.

Finally authors conclude that identification of data location of the web application deployed in US, Europe and Asia is visible by minimizing threats in the experiment setup.

7. Checklist for Managing Data Location of Web Application

By performing the activities of three frameworks [1] [2] [3] in the experiment setup authors identified the DL of the server created at US California, Virginia, Oregon, Europe Ireland, Asia Singapore and Tokyo. The replication of the three framework activities in the experiment setup shows that when a new or existing Cloud user deploys and manages the web application in cloud one should verify the activities performed in this experiment to identify the DL. However there are many existing guidelines and architecture to manage the DL in Cloud Computing there is still a need to improve the DL. Authors prepared a checklist which helps a Cloud user or end user to manage the web application content. The activities performed and satisfied in the three experiments are considered; here some of the activities performed in one experiment setup are repeated in other. The duplicates of the activities considered are removed and a unified checklist for managing data location of web application is prepared. This checklist is prepared based on Amazon Web services but also valid for other cloud services as the checklist consists of basic requirements that every cloud service provides. This checklist consists of two sections one is (i) verifying the data location of web application with the help of monitoring information provided by Cloud Provider [1] (ii) verifying the data location of web application without the help of CP i.e. verifying by considering semantic based and measurement based approach [2] [3].

(i) Verifying the DL with the monitoring information provided by CP

Monitor the events regarding data location when web application is deployed. As deployed web application requires services EC2, S3 and RDS. This monitoring information is performed by following framework I [1]

EC2:
- Verify that name of Instance zone created is at the given region.
- Verify the IP address of the instances created by using Whois IP location.
- Verify that the DNS name of the load balancer shows the region where the server is created.
The name of the Availability zone of the load balancer provided should be at region where the server is created.

The Hosted zone ID of the load balancer should be similar with other sub region but should be different with other region.

**S3:**
- The name of S3 bucket created should be at the place where the server is created.
- The Standard Region of S3 should be at the given preference.
- The http link of S3 service should include the name of the region created.

**RDS:**
- The End point server name should be the name of the given region.
- The Zone details of RDS should be at region where server created.

(ii) **Verifying the data location without the need of CP [2] [3]**
- Calculate the actual distance from internet location to the servers and compare the distance from internet location to the servers created [2] [3].
- Measure the average RTT and latency values of the deployed web application http link which is allowed to use for end user. By comparing the RTT and latency values with the distance. Estimate positive linear relationship between distance and latency [2] [3].
- When the resources are located close to each other the RTT and Latency values should respond with nearly same values [2].
- Measure the number of network hops travelled, if resources are located near to each other. Estimate that the latency of the server travelling through depends on the number of network hops travelled [2] [3].
- Server located in different countries with same distance from internet location may not be equal [2] [3].
- Capture the IP address of the http link and verify the location with Whois IP location [3].

### 8. Summary and Conclusion

#### Answers to Research Questions

**RQ 1: What are the existing frameworks that allow identifying data location of web application resources in Cloud Computing?**

There exist ten data location frameworks, architectures, techniques, models and guidelines [1][2][3][4][19][21][75][78][79][80] which allow managing data location in Cloud Computing identified from literature study. These frameworks consists of two categories 1) identifying data location of Cloud resources with the help of information provided by Cloud Provider. 2) Identifying data location without the help of Cloud provider i.e. by measuring the performance estimating geolocation. Here the framework solves two issues a) identifying data location of the Cloud resources by measuring RTT b) Identifying data location of the Cloud resources and the replicated data in Cloud Computing by measuring latency. Finally in order to validate the data location frameworks three data location frameworks [1][2][3] are considered for evaluation, in which one frameworks solve data location issues with 1) i.e. [1] and other two frameworks solves the issue from 2) i.e. [2] [3]. These generalize validation of the data location frameworks for some extent.

**RQ 2: How efficient are the Data Localization framework(s) in identifying where data and services are stored and run?**

By replicating the data location frameworks [1][2][3] in Amazon Cloud environment authors evaluated the efficiency of data location frameworks by performing activities considered from frameworks in experiment and providing rubrics to the activities performed in experiment as shown in table (5) (8) and (13). The three data location frameworks [1][2][3] allowed estimating the geolocation of AWS resources located in US Virginia, California, Oregon, Europe Ireland and Asia Tokyo efficiently. This proves that the three data location frameworks are effective and are valid data location frameworks that allow estimating geolocation of resources, files located in Cloud. Moreover evaluating three different types of frameworks [1] [2] [3] to identify data location allowed proposing guidelines for managing data location in Cloud Computing.
Summary and Conclusion

Cloud computing is an internet based development and use of computer technology, where IT provides software, platform as well as infrastructure as a service [1]. Most of the business and IT are presently trending towards cloud computing and using the Cloud services, similarly most of them are lacking behind due to some security reasons. When a data is globally distributed, different countries have different data privacy laws [2]. So the physical location of the data should be within the country jurisdiction, data security issues arises if the physical location of the data is migrated towards other country. Cloud customer when migrating their data to the Cloud, should demand Service provider about the physical location of data. To verify the data location without taking help from Cloud provider allows cloud customers to manage complete control of their physical location of data. So identifying the data location when data is migrated to the cloud is an important issue that should be considered by Cloud user. There are ten existing data localization frameworks that satisfy in identifying the data location of Cloud resources. To evaluate the efficiency of DL frameworks authors considered the three main data localization frameworks [1] [2] [3] from literature study.

A web application is developed using C#, ASP.Net and SQL server as backend and deployed in US, Asia and Europe regions of Amazon Web Services. Now to verify the location of web application content, the three framework activities are performed in the deployed application. Here the three frameworks allow estimating the geolocation in three different ways. Identifying the DL of the deployed web application in AWS resources using the three framework activities allows evaluating the efficiency of the DL frameworks in the experiment setup. By conducting the three framework activities authors came to a conclusion that Amazon maintains the resources according to the preference given for creating the resource.

The evaluation of three frameworks from a single internet location Sweden BTH, allows estimating the location of servers created in US Virginia, California, Oregon, Europe Ireland, Asia Singapore and Asia Tokyo. The servers located close to each other (less than 600KM) i.e. Oregon and California are also verified in the experiment.

Authors in this experiment study also identified the replicated data of the deployed web application. Here CloudFront service [60] is connected to the web application deployed in Europe Ireland to improve the performance which reduces the latency. The files of Europe server are saved to the nearest Amazon edge location Sweden Stockholm is identified in experiment 3.

The experiment results shows that the three frameworks satisfy in identifying the DL of web application content, even though the experiment setup is different from the framework. Authors prepared a checklist that should be considered by new or existing Cloud user or End user to manage the data location when a web application is deployed to the Cloud resources. The checklist consists of verifying the data location web application with the monitoring information provided by CP i.e. AWS and without the need of verifying the data location i.e. verifying by using measurement based and semantic based approach.

The main contribution of this research study here is we can trust Amazon regarding data location as the cloud resources reside according to the considered preference for location. But Cloud users who deploy and manage cloud resources have to make sure that files may be replicated in different locations when cloud users use the services like CloudFront. This shows that even though service provider follows the data location laws that data reside in the selected preference Cloud user should be aware of the location of the cloud resources.

Finally authors came to a conclusion that by evaluating three different frameworks in the experiment setup allows to control and manage the web application content regarding data localization. However there are few studies that improve the performance of verifying data location of Cloud, some future study regarding data location is, verifying the data location when an application is maintained at one location and the database is maintained at different location still improves identifying DL in Cloud Computing. In addition to this, validating the proposed checklist whether applicable for real cloud user who deploys and manages data in Cloud resources improves the efficiency of identifying data location Cloud Computing.
9. REFERENCES


[55] Rochwerger et al. (Eds.), RESERVOIR High Level Architecture (release1.2), Project Deliverable D.1.1.1, 2008.


Appendix

Appendix A: Primary Studies


Appendix B: Web References


Appendix C: RTT values of 30 request count

<table>
<thead>
<tr>
<th>Request count</th>
<th>RTT value of 30 request count from BTH to AWS AZ in Milliseconds</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>United States</td>
</tr>
<tr>
<td>1</td>
<td>392ms</td>
</tr>
<tr>
<td>2</td>
<td>395ms</td>
</tr>
<tr>
<td>3</td>
<td>390ms</td>
</tr>
<tr>
<td>4</td>
<td>391ms</td>
</tr>
<tr>
<td>5</td>
<td>397ms</td>
</tr>
<tr>
<td>6</td>
<td>384ms</td>
</tr>
<tr>
<td>7</td>
<td>400ms</td>
</tr>
<tr>
<td>8</td>
<td>389ms</td>
</tr>
<tr>
<td>9</td>
<td>384ms</td>
</tr>
<tr>
<td>10</td>
<td>391ms</td>
</tr>
<tr>
<td>11</td>
<td>441ms</td>
</tr>
<tr>
<td>12</td>
<td>392ms</td>
</tr>
<tr>
<td>13</td>
<td>389ms</td>
</tr>
<tr>
<td>14</td>
<td>394ms</td>
</tr>
<tr>
<td>15</td>
<td>387ms</td>
</tr>
<tr>
<td>16</td>
<td>517ms</td>
</tr>
<tr>
<td>17</td>
<td>387ms</td>
</tr>
<tr>
<td>18</td>
<td>386ms</td>
</tr>
<tr>
<td>19</td>
<td>398ms</td>
</tr>
<tr>
<td>20</td>
<td>437ms</td>
</tr>
<tr>
<td>21</td>
<td>418ms</td>
</tr>
<tr>
<td>22</td>
<td>411ms</td>
</tr>
<tr>
<td>23</td>
<td>407ms</td>
</tr>
<tr>
<td>24</td>
<td>399ms</td>
</tr>
<tr>
<td>25</td>
<td>397ms</td>
</tr>
<tr>
<td>26</td>
<td>397ms</td>
</tr>
<tr>
<td>27</td>
<td>387ms</td>
</tr>
<tr>
<td>28</td>
<td>400ms</td>
</tr>
<tr>
<td>29</td>
<td>392ms</td>
</tr>
<tr>
<td>30</td>
<td>391ms</td>
</tr>
</tbody>
</table>
## APPENDIX D: latency values of 30 request count

<table>
<thead>
<tr>
<th>Request Count</th>
<th>United States</th>
<th>Asia</th>
<th>Europe</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Oregon</td>
<td>Virginia</td>
<td>California</td>
</tr>
<tr>
<td>1</td>
<td>175ms</td>
<td>133ms</td>
<td>175ms</td>
</tr>
<tr>
<td>2</td>
<td>177ms</td>
<td>141ms</td>
<td>177ms</td>
</tr>
<tr>
<td>3</td>
<td>182ms</td>
<td>136ms</td>
<td>182ms</td>
</tr>
<tr>
<td>4</td>
<td>196ms</td>
<td>136ms</td>
<td>196ms</td>
</tr>
<tr>
<td>5</td>
<td>180ms</td>
<td>140ms</td>
<td>180ms</td>
</tr>
<tr>
<td>6</td>
<td>178ms</td>
<td>133ms</td>
<td>178ms</td>
</tr>
<tr>
<td>7</td>
<td>212ms</td>
<td>143ms</td>
<td>212ms</td>
</tr>
<tr>
<td>8</td>
<td>175ms</td>
<td>132ms</td>
<td>175ms</td>
</tr>
<tr>
<td>9</td>
<td>190ms</td>
<td>131ms</td>
<td>190ms</td>
</tr>
<tr>
<td>10</td>
<td>176ms</td>
<td>135ms</td>
<td>176ms</td>
</tr>
<tr>
<td>11</td>
<td>175ms</td>
<td>133ms</td>
<td>175ms</td>
</tr>
<tr>
<td>12</td>
<td>196ms</td>
<td>131ms</td>
<td>196ms</td>
</tr>
<tr>
<td>13</td>
<td>182ms</td>
<td>136ms</td>
<td>182ms</td>
</tr>
<tr>
<td>14</td>
<td>185ms</td>
<td>131ms</td>
<td>185ms</td>
</tr>
<tr>
<td>15</td>
<td>178ms</td>
<td>131ms</td>
<td>178ms</td>
</tr>
<tr>
<td>16</td>
<td>188ms</td>
<td>314ms</td>
<td>188ms</td>
</tr>
<tr>
<td>17</td>
<td>179ms</td>
<td>131ms</td>
<td>179ms</td>
</tr>
<tr>
<td>18</td>
<td>183ms</td>
<td>133ms</td>
<td>183ms</td>
</tr>
<tr>
<td>19</td>
<td>180ms</td>
<td>139ms</td>
<td>180ms</td>
</tr>
<tr>
<td>20</td>
<td>177ms</td>
<td>137ms</td>
<td>177ms</td>
</tr>
<tr>
<td>21</td>
<td>176ms</td>
<td>146ms</td>
<td>176ms</td>
</tr>
<tr>
<td>22</td>
<td>186ms</td>
<td>148ms</td>
<td>186ms</td>
</tr>
<tr>
<td>23</td>
<td>179ms</td>
<td>145ms</td>
<td>179ms</td>
</tr>
<tr>
<td>24</td>
<td>176ms</td>
<td>135ms</td>
<td>176ms</td>
</tr>
<tr>
<td>25</td>
<td>176ms</td>
<td>138ms</td>
<td>176ms</td>
</tr>
<tr>
<td>26</td>
<td>180ms</td>
<td>135ms</td>
<td>180ms</td>
</tr>
<tr>
<td>27</td>
<td>201ms</td>
<td>133ms</td>
<td>201ms</td>
</tr>
<tr>
<td>28</td>
<td>181ms</td>
<td>140ms</td>
<td>181ms</td>
</tr>
<tr>
<td>29</td>
<td>196ms</td>
<td>134ms</td>
<td>196ms</td>
</tr>
<tr>
<td>30</td>
<td>198ms</td>
<td>136ms</td>
<td>198ms</td>
</tr>
</tbody>
</table>
APPENDIX E: Sample view of the deployed web application

APPENDIX F: Captured Monitoring events