ISSN 22773061

Representation of cloud ecosystem using engineering methodologies

Manu AR[1],Dr. Vinod Kumar Agrawal [2], Dr. K N Bala Subramanya Murthy [3]
[1].Research Scholar, CORI Lab, ISE Dept., PES Institute of Technology, Bangalore, Karnataka, India.
Manu.a.ravi@gmail.com, manu_ar@nitk.ac.in
[2].Director and Professor, CORI Lab, ISE Dept., PES Institute of Technology, Bangalore, Karnataka, India.
vk.agrawal@pes.edu
[3]. Prinicpal and Director, PES Institute of Technology, Bangalore, Karnataka, India.
principal@pes.edu

ABSTRACT

Cloud Computing has fascinated massive consideration for business in spite of lot of technologies and business models in the market. The operational particulars within the cloud are not coherent enough to customers. Hence, this paper provides the method to derive trusted cloud computing scaffold. This scaffold is represented using the engineering methodologies. In this work we use elementary engineering principles of computation to represent the cloud eco-system, comprising of complete cloud system from the end-user perspective.

Indexing terms/Keywords: Cloud System Life Cycle, Cloud eco-system, elementary engineering principles of computation, Work Breakdown structure.

Council for Innovative Research
Peer Review Research Publishing System
Journal: INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY
Vol 7, No 1
editor@cirworld.com
www.cirworld.com, member.cirworld.com

1. INTRODUCTION

Cloud computing is a computing service model enabled on convenience, on-demand shared network access to a divisible group of configurable computing assets (e.g., networks, servers, storage space, appliances, and services) which can be quickly provisioned and freed with least executive effort or service supplier-consumer interaction. [1]. Cloud computing is a broad term for everything that engrosses hosted and accessed services over the Internet. These cloud services are generally divided into three broad spectrums of basic categories [1]: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). There are 4 common types of cloud deployment models, namely [1]: Public cloud, Private cloud, Community cloud, and Hybrid cloud. Cloud Computing brings with it many benefits to the end user respectively:

- Accessing the massive choice of applications over the Internet devoid of having to download, install, purchasing and maintaining anything in house and also software updates etc.
- Scalability on demand, elastic, stress and strain resistible / flexible to the business needs.
- Reduction in the assets, including the hardware and infrastructure costs.
- Reduction in IT recruiting, administration and management costs, usage of automated methodologies to handle computational services.
- Increasing the data-centre storage capacity based on business requirement.
- Accessing the applications/resources using any computer, notebook, desktop, and mobile phone located in different geographical locations across the world.
- Reduction in spending on hardware, related software utilities, updates, patches, maintenance by consumers and use them whatsoever they need, when they need based on requirement.
- Organization can share resources and assets situated in one place, which is accessed at any place over Internet.
- Resources consumed is billed and charged as a utility based on metered/measured usage with minimal forthright costs.

The Cloud ecosystem co-exists where all the participants in a marketplace comprising of integrated business processes and apply universal standards for exchange of information, products, and services. This cloud ecosystem is the versatile community of individuals and its related environment operating as an ecological/natural unit. In terms of cloud computing, it includes not only traditional elements of cloud computing such as software and infrastructure but also various players, including consultants, cloud brokers, integrators, partners, third parties, vendors, customers, and anything in their business environments. Business processes management (BPM) has been a foremost step toward standardized business services and processes automation. With cloud computing, standards and technological developments come together to create an environment in which integrated business processes are supported by software services performed within and between enterprises. Cloud Ecosystem features are: it benefits large amount of players, heterogeneity of cloud platforms, interoperability and portability, no vendor-lock-ins and geographical distribution. With variety of providers, their capabilities and services, discoverability and matching of cloud services, service visibility [2].

There are various issues adjoining cloud adoption for end users in the cloud ecosystem.They are: cloud security issues, privacy issues, combining virtual resources with existing systems resources and assets in the cloud setup, failure or fear of losing the control over data associated with business and personal information, service availability issues, cloud performance issues, regulatory- governance- and compliance issues, cost- pricing issues, measuring - usage issues, portability - customization issues, credibility issues, ROI/profitability issues etc.Technology issues, computers need to be protected from well-known threats, such as, phishing, downtime, data loss, password weakness and compromised hosts running bot-nets, virtual machine vulnerabilities etc to name a few.

In this paper we propose a method to represent cloud computing eco-system using key engineering principles of computation, which may benefit the various actors/players, academicians and industrial researchers to run their job/business smoothly/effectively and efficiently .This work is the effort in that direction to achieve practical efficiency using the unified modelling of cloud system as demonstrated in the following sections. In this paper, we present cloud system life cycle model and its various phases, starting with requirement collection / planning phae till the end of phase of de-commissioning / retirement from the opted cloud service. We also present cloud system project charter- its components including cloud requirement specification (CRS), functional requirements and non-functional requirements, feasibility study and its types employed commonly in practice, also we mention various cloud enterprise wide ecological factors, and sample representational output format of the project charter is envisioned, it also proposed project and product stake holders registry and management- its representational table format, respectively, the abstract view elements in terms of the actors, class view with its representation in the cloud ecosystem and explicitly decompose cloud services using use work breakdown structural representation of cloud project and cloud product respectively. We have also presented cloud project management scope (CPMS) in terms of cloud project class diagram, product class diagram, and cloud state chart diagram.

The remainder of this paper is organized as follows: Section 2 enumerates cloud system life cycle mode, Section 3 covers cloud computing project charter and its components, requirement specifications for cloud ecosystem/setup, Section 4 presents the cloud system project and cloud product - work breakdown structural representations, Section 5 highlights the cloud system project management scope and its representation using principles of the engineering thru computational approach - represented by cloud product class diagram, and the cloud project class diagram, cloud system state chart diagram, etc. Section 6 concludes the presented work and outlines its ongoing extension.

2 Cloud system life cycle.

The cloud system life cycle (CSLC) includes the set of phases/steps mandatory to fabricate and execute the cloud service deployment, and its correlated architectural representational agenda which describes the perception of the group of actors in the cloud eco system, and the things they will see from those perspectives. The phases of CSLC includes [2]:

i) **Cloud planning, requirement gathering, market survey & fabricating strategy**: The view of the organization as a whole/intact, by and large cloud eco-system design development and improvement effort look like, designing of the components of the trade and business should deal with new information structure, assigning priorities to possessions etc related cloud eco system services and documented using the blueprint design / scheme/ architectural pattern. Planning engrosses category of project plan, resource plan, economic plan, quality plan, hazard plan, protection plan, acceptance plan, communication plan, procurement plan, disaster recovery plan etc.

An organization needs to consume service from the cloud service provider, needs to define completely via the project champion/victor, with the detail list of the category of service required, time period and ability of the required service, efficient project supervision and reputation, good service milieu and reputation of the organization offering the service, their effectual project management, security services offered, and parameters to define the service level agreements, well known implicit set of steps, and developing the special genus of relationship with the service vendor/ customer. There are abundant ways to characterize the type of service required by a venture using the, cloud ecosystem data representation models, cloud data flow diagrammatic representations (DFD's), state/transitions diagrams, and so forth.

ii) **Requirement analysis and feasibility study**: This is the second phase in the cloud system development lifecycle where we analyze the project requirement that use cloud system service. The requirement analysis is the procedure of crafting the scope for a system effort, with its technique, assign responsibilities to each player to perform their roles. Here, first we do the feasibility study where we analyze the existing methodologies and infrastructure, based on the analysis for procuring new service in the cloud, we document and represent using the engineering approaches which is a novel tactic style, is viable and advantageous to implement. We also gather all the requirements and group them and prepare a list of services that our system is expected to serve and behave in cloud ecosystem.

iii**) Design:** The application of knowledge and technology to deal with the gaps is recognized all through the requirements analysis phase. Represented by data structures become database designs or objects classes and the functional descriptions happen to be program specifications. At this point, in the interest of defining the actions of the potential system, concentration is also paid to the human interface, architecting the blueprint/road map to the cloud ecosystem and deployment represented using the existing engineering and computational approach. The cloud system design can have flexibility, elasticity and stretch-ability, designed into them should not mislead anyone into the belief that the design process itself can be one of endless modification.Victorious cloud service deployments require two main things: purchaser who can describe what kind of work desires doing, and increasing trade commerce needs, who can deliver/ render that will do the job/service requisite, at slightest cost doing the job is significant –whilst.

iv) **Construction**: The actual building and assembling/ fabrication of the system with existing infrastructure in the cloud setup using the related design and its representations.

v) **Transition**: The implementation and operation of the system is to make it part of the new infrastructure of the institution. This employs teaching, guidance, description of plan of the organizational arrangement and task representation using the work breakdown structures etc, and the translation of existing information and infrastructure based on system design. Implementation and management/administration involves – time period management, price management, excellence management, transformation management, threat management, Issue management, acquisition management, approval management, communications supervision, Security and privacy administration, violation and compliance management, assets management, contract and service management, employee management etc.

vi) **Production**: The ongoing supervision of the system to ensure that it continues and persists to meet the needs of the organization.

vii) **Closure/Retirement**: Retirement, perform project closure, review project completion since de-commissioning, verify and identify, security compliance and violations, document the same using suitable representations. The phases of CDLC, are shown below in fig. 1.

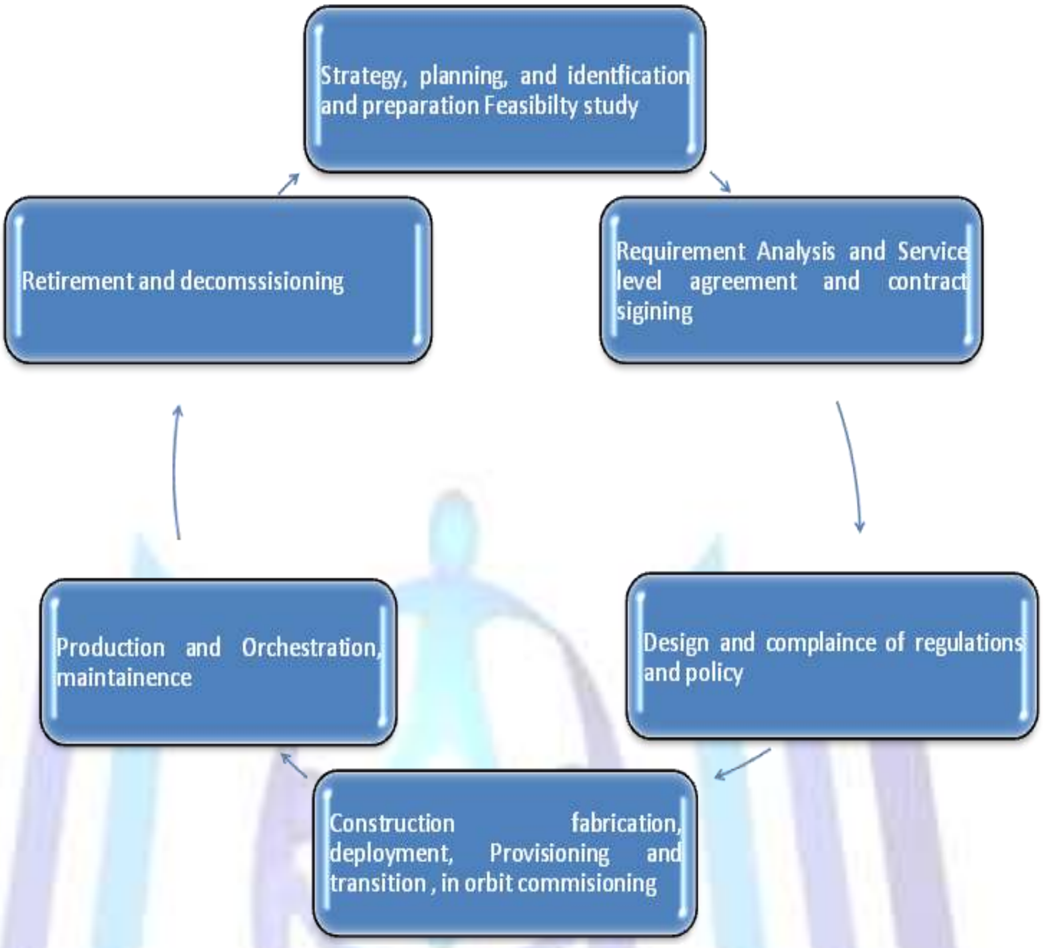

Strategy, planning, and identfication and preparation Feasibilty study
Requirement Analysis and Service level agreement and contract sigining
Design and complaince of regulations and policy
Construction fabrication, deployment, Provisioning and transition , in orbit commisioning
Production and Orchestration, maintainence
Retirement and decomssisioning

Figure 1: Stages/phases of Cloud system development life cycle

3 Cloud Project Charter

The Cloud Project Charter for cloud computing services will afford as an in-house internal document that covers representation of the cloud ecosystem with high-level-planning information design and documentation(defining the scope and goal of the cloud service, abstractions, deliverables, security safeguards, postulations, hypothesis, and attributes of the cloud service level agreement, etc.) about the project/service which is ready for the cloud deployment and defines the roles and responsibilities of the champions/players involved for the particular project in the cloud ecosystem. The project architect, who is the champion, creates the project charter in the initiation phase of the project, in consultation with the business associate. Its purpose is to identify the continuation of the project and to begin of the planning method required to achieve the project goals, as set by the organizational standards, and it reflects the readiness of the corporation which is set ready to cloud service deployment setup. It is not anticipated to be shared with the purchaser as a formal contract or legal document represented in the form SLA [3].

The project charter forms the basis for complete and meticulous project forecast. It is a vital element for beginning, instigating, scheduling, forecasting, accomplishing, executing - implementing, controlling and monitoring till retirement/closure of the project from the cloud setup. It is the supreme master manuscript documental representation for the project and as such it should be maintained under the single point of contact and reference on the project for aim and objectives, scope, estimates, approximations, end deliverables, and budget transactions. The items contained by the charter serve as project control documents. The project charter defines overall goal, readiness of the organization to the cloud deployment. Also, it defines the threats, security and privacy issues that are likely to happen, appear and breaches which may happen along the way, and necessary guidance on how to continue things on board , survive, supervise and monitor the progress [4].

3.1 Cloud requirement specification (CRS)

It comprises of the functional and non-functional requirements in the form of engineering representations for the apps to be developed in particular cloud deployment type. The functional requirement defines how the apps should do (behave and perform at every execution and input/runtime) and non-functional requirements consist of the constraints and restrictions on the blue print, design or implementation. Requirements must be characterized as per cloud features namely, scalability, measurable, elastic, testable, flexible to changes, modification, related and mapped to recognized requirements as documented needs or prospects.

The scripting of CRS lessens implementation effort, as cautious review of the article reveals omissions, slips, misunderstandings, misrepresentations, misinterpretations, missing features and inconsistencies early in the initial stages of the cloud system life cycle. The CRS discusses the artefacts/products cloud service types produced but not the project that developed and implemented it. Hence, the CRS serves as a basis and fundamental reference document for future continued production evaluation, enhancements, changes or modifications of the service product deployment for increased demand [3, 4, 5].

CRS includes a) The purpose and scope, defines overall scope and goal of the project, stake holders within the scope, out of scope, b) Terms used, glossary of the terminologies used, c) The use cases defines the primary actors of cloud eco system, their associated goals, the business use cases. The system use cases, the technology used for this system, and systems interfaces, d) Other requirements – defines i) who are all the project participants/stakeholders, values reflected (simple, soon, fast, or flexible), feedback or project visibility to the end users/ customers and needs of the sponsors, supplies to be purchased, what must we build, who are our competitors in the market, other process requirements (testing, installation, etc), dependencies does the project operate under ii) business rules, iii) performance iv) operations, security, documentation, v) use and usability, vi) maintenance and portability vii) unresolved or deferred scenarios e) Human resource backup, legally recognized, politically biased, organizational concerns, guidance requirements, postulations, dependencies and related representational documentations [6]. CRS includes:

3.1.1 Functional requirements and Non-functional requirements: A well-designed requirement specification does not define the detailed internal mechanism of the proposed cloud eco system; it does not include the specifications representations how the system utility will be implemented. Instead, it focuses on what diverse external agents (for example; end users using the program, computer hardware peripherals, or other computers,) might "observe" when interacting with the system.

3.1.2 The non-functional requirements: Specify the qualitative traits, representations such as, user-friendliness using GUI & system performance that are critical for the increased user-recognition/ authentication of the application. An organization has attributes that emerge from the combination of its parts. This emergent property is a matter of catastrophe, not design, if the non-functional needs, or organization merits, are not meticulous in advance.

3.1.3 Hardware Requirement: Processor, speed, RAM, hard disk, floppy drive, standard windows, keyboard, mouse, monitor, servers and work stations, data centres, networks, and storage space are needed. **Software requisite:** OS, apps server, front end, J2EE, database, database connectivity, JDBC [8 910].

3.1.4 Institutional and location Requirements: Without providing a mechanism for safe computing and i.e., to protect the confidential input and output data of the workloads/computing and to validate the integrity of the computational outcome, it is difficult to expect cloud end users to loop over control, their assignments from confined machines to cloud solely based on its cost-effective reserves and reserve elasticity.

3.1.5 Standard requirements - include:

a) Interfaces: This section describes the interfaces in detail about the user interface, using GUI and command line interface, hardware, software interfaces etc.

b) User Interfaces: A user- friendly GUI representational system, graphical editor- to write Java Script, to write the XML file, console for executing program and user commands virtually, GUI should provide graphical and command line interface to test the system virtually hosted at distant locations.

c) Network and Communication Interfaces: implemented by the system, TCP/IP communication protocol is used in order to establish a connection to the server which is present in the cloud of servers.

d) Constraints include: Time: Identify the time schedule, at which the product deployment must be accomplished, through the success & quality of the development depend on the precise time bound inhibited ? to complete the project within the time ?schedule as defined in SLA.

3.2 Feasibility Study

Feasibility is the analysis of whether or not a product or service is worth for outsourcing to the cloud ecosystem. The course of action comprehended in determination of this grit is called feasibility study. In the feasibility study for cloud, the analyst will usually consider several distinct features and it is documented, but inter-related types of feasibility study in practice is as follows [5]:

a) Technical feasibility: This is considered with specifying equipment and system requirement that will successfully satisfy the user prerequisite, the technological needs of the cloud eco-system representational structure may differ significantly also might comprise a variety of facilities to produce required outputs in a given time, with constraints, precision and acceleration of expected facility [6 7].

b) Economic Feasibility: Economic investigation is the system for evaluating the effectiveness of the deployed cloud system.,more commonly known as cost incurred - benefit analysis. This engineering method is to establish the profits and economy expected from cloud service and a compare them with costs incurred. An assessment to design, document and implement the system will have to be made if it is to have a chance of being approved. There is an ongoing effort that improves accuracy and consistency at each phase of the system life cycle [7].

c) Operational Feasibility: It is primarily associated to human association and political attributes. These attributes are considered and documented are [7]: System, transforms ought to be fetched within the organizational representations, secretarial patterns, novel proficiencies required, the prevailing system- human resources/players (cloud system actors) have these-expertise, if not, how can they be trained in the course of time for cloud players.

4. Cloud Enterprise Ecological Factors

Communications and interactions among players with goals, we know performers have goals, the actors have the accountability, responsibility and liability, The cloud system setup also have the responsibility to catalogue/representation list of services offered, needs to register for the same, schedule and initiate the service appeal, it actually has the duty to protect the interests of all the stake-holders, the primary actor to carry out its tasks, the structure formulate and devise sub goals using the WBS representation, to clutch sub-goals internally needs the aid of the supporting actor, may be from within or another organization, such as, a collaborator company or government bureau. The supporting actor usually carries away its guarantee, assurance given in cloud SLA and conveys the sub-goals to the cloud system under treatise, interacts with external peripheral actors, it achieves its sub-goals in some series-succession until it finally delivers responsibility that is its service promise, as documented and signed in SLA/contract.

Sub-goals can be broken down into sub-sub-goals indefinitely through compartmentalization and modularization WBS representation / approach. If we want to breakdown the performance of the actors finely adequate, the actors and goals abstract model representation is handy, since it applies equally to trade, commerce and computer systems. The actors can be individual people, organization or computers, mixed system consisting of people companies and computers of cloud ecosystem [6 7]. The system under design controls an agreement between stakeholders, through the use cases detailing the actuarial and behavioural element of that agreement/documented. Not all of the stakeholders are present whilst the system is running, so we call them as offstage actors, the system should satisfy including gathering information, running validation checks, and updating logs etc. To satisfy the interests of the stakeholders we need to describe three sorts of actions, a) an interaction between two actors (to further a goal), b) A validation (to protect a stakeholder), c) An internal state change (on behalf of a stakeholder) [8][9].

a) Organizational Assets: Each corporate ethnicity diverges from the next, and we need to make sure we are captivating this into consideration when we systematize a scheme /engineering representation for cloud setup.

b) Tools: It's important to understand the types of tools your organization uses to administer the cloud ecosystem so the team can leverage the assets that are already in place. Queries to consider about tools include [18]: Project administration tools are accessible to team players, is there any explicit equipment you ought to track time and assets, organizational trade partners utilize a precise tool to execute their job, orchestrator, provisioning, de-commissioning etc.

c) **Infrastructure [18]:** To identify whether the opted service is precinct to our infrastructure,whether we are exceeding out of space on owned/allocated servers, in that situation, how elongated time does it take to procure new hardware, does the communications network is capable to grip the extra capacity you're going to append to the existing infrastructure, in the cloud system.

d) **Marketplace conditions [18]:** Identify, if there is any authorized inference on the project and to look for governmental rules or restrictions that could impact the project.

4.1.1 Output: The output of the cloud project charter is as shown below in table 1.

Project Justification	A Project Representation of Cloud Computing ecosystem : A Practical Approach using the Engineering Methodologies
Purpose	To design practically efficient mechanisms for outsourcing of computational service and its representations
Cloud Software Requirement	Operating System: Windows 95/98/XP/2007/2010, Linux, Unix Application Server: Tomcat Front End: HTML, Java Scripts: JavaScript Server side Script: Java Server Pages. Database: MySQL. Database Connectivity: JDBC.
Objectives	1. Explicitly decompose the computational service outsourcing into public cloud running on the cloud and private owned by the customer to achieve practical efficiency. 2. Problem transformation techniques allow customers to transform by protecting sensitive input/output information. 3. To validate the computation result, derive the necessary and sufficient conditions that correct result must satisfy.
Milestones	1. To mire the CRS, design Control Flow Graph and Data Flow Transition diagrams. 2. To resolve the issues involved in Cryptography and Secure Computation.

	3. To perform Project work break down structure. 4. To identify the activities and combine the concurrent activities to form the process. 5. To design a practically efficient mechanism for outsourcing of cloud computations. 6. To derive a set of necessary and sufficient conditions that the correct result must satisfy.
Assigned Project stakeholder	Mr. XYZ

Table 1: Enterprise Environmental Factors.

Tools and Techniques: Automated cloud setup.

4.1.2 Cloud Project and Product stake holders

Input: Project charter (refer: Cloud requirements Spec, External Interface Requirement, and Organizational Process Assets)

Output: i. Stake holder register, as shown in table 2.

ii. Stake holder management strategy, as shown in table 3.

In this section we identify the persons associated with the project.

i. Cloud Stake Holders Register

Stake Holder	Stake Holder interest in project	Assessment Impact
Evaluating system	Evaluate and validate in depth	Very High
Developer	Identify the problem and solving problem	Very High
Cloud service vendor Organization consultant, cloud broker	Determine the mistakes in each level and provide guidance to overcome them. Verify each deliverable.	Very High
Cloud service owner/ service provider	Determine the mistake in each level and gives the description to overcome mistakes	Higher
cloud customers	They include clients	High
Cloud service user group	Various users who use the product and work on it	High
Cloud deployment service Testers	Test the system with different inputs	Very High
Head of the department/project section	Verifies each deliverable	High

Table 2: Stake Holder Register

ii. Stake holder management strategy

Name	Designation	Location	Role of stack holder	Contact info	Requirements	Expectations
Mr. ABC	Software deliver manager primary stakeholder	Bangalore	Developer	111-111-222-222-22	Absolutely necessary	High
Mrs. PQR	Delivery head	Bangalore	Vigilance	022-23333333	Absolutely necessary	High
External stakeholder	Auditor	Bangalore	Evaluate	-	Necessary	Medium

Table 3: Stake Holder Management Strategy

4.1.3 Abstract View Elements of cloud ecosystem.

Input: : Project Charter (refer: CRS, Organizational Process Assets)

Output: Referenced Items, Defined Items, cloud actor's name as shown in table 4, Actor interface attributes with class names as shown in table 5, Class attributes with employed algorithm as shown in table 6.

Actors	Attributes
Customer	Customer ID, Password
Cloud Server	Server ID

Table 4: Cloud actor's name with related attributes.

Classes Names	Attributes
Random Key Generator	Size of Problem, Random-Key
Encryption Scheme	Cloud service, Problem ID, Key
Decryption scheme	Encrypted text Solution, Problem ID, Key

Table 5: Classes of Actor with interface attributes

Algorithm name	Input	Output	Objective
Key-gen	Problem ID	Secret Key	To generate a Secret key for Encryption and Decryption
Prob-Enc	Secret Key	Encrypted text	To Encrypt using Secret Key
Result-Dec	Secret Key, Encrypted solution	Decrypted solution	To Decrypt the obtained solution

Table 6: Class Attributes with employed algorithm

4.2 Cloud Project Work break down structure

Input: 1. Project charter (Refer: CRS)

2. View elements (Refer: Output of Abstract View Elements activity)

3. Organizational process assets (Refer: Project charter)

Output: work break down structure for project/product design as shown in fig 2.

Tools and Techniques: cloud management/orchestration tool, Knowledge areas, cloud project life cycle, and cloud implementation life cycle stages, cloud provisioning, esilience.

4.3 Cloud Product Work breakdown structure (WBS):

Work Break down structure is a deliverable oriented decomposition of a cloud project into minor modules. It describes and clusters a project's isolated work elements in a way that helps systematize and describe the totality of the work extent of the project in the cloud ecosystem. A WBS element may be a product, data, or type of service, or any combination mutually. A WBS also endows with the obligatory framework for meticulous cost estimating and control along with on condition that guidance for schedule development and manages.

Input: 1. View elements (Refer: Output of Abstract View Elements activity)

2. Product stake holders (Refer: Output of Identify Project and Product stake holders)

Output: 1. Cloud Product work break down structure as shown in fig 3.

Tools and Techniques: use case, work procedure, effort. The WBS of project is as shown below.

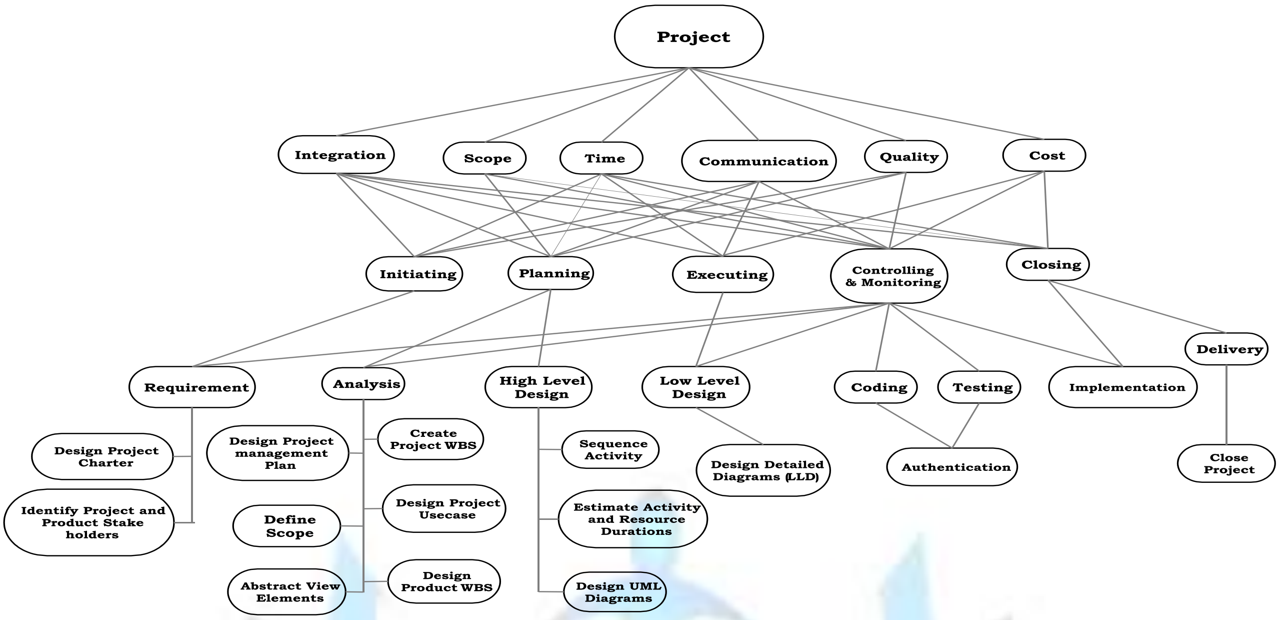


Figure 2: Project Work break down Structure for cloud services.

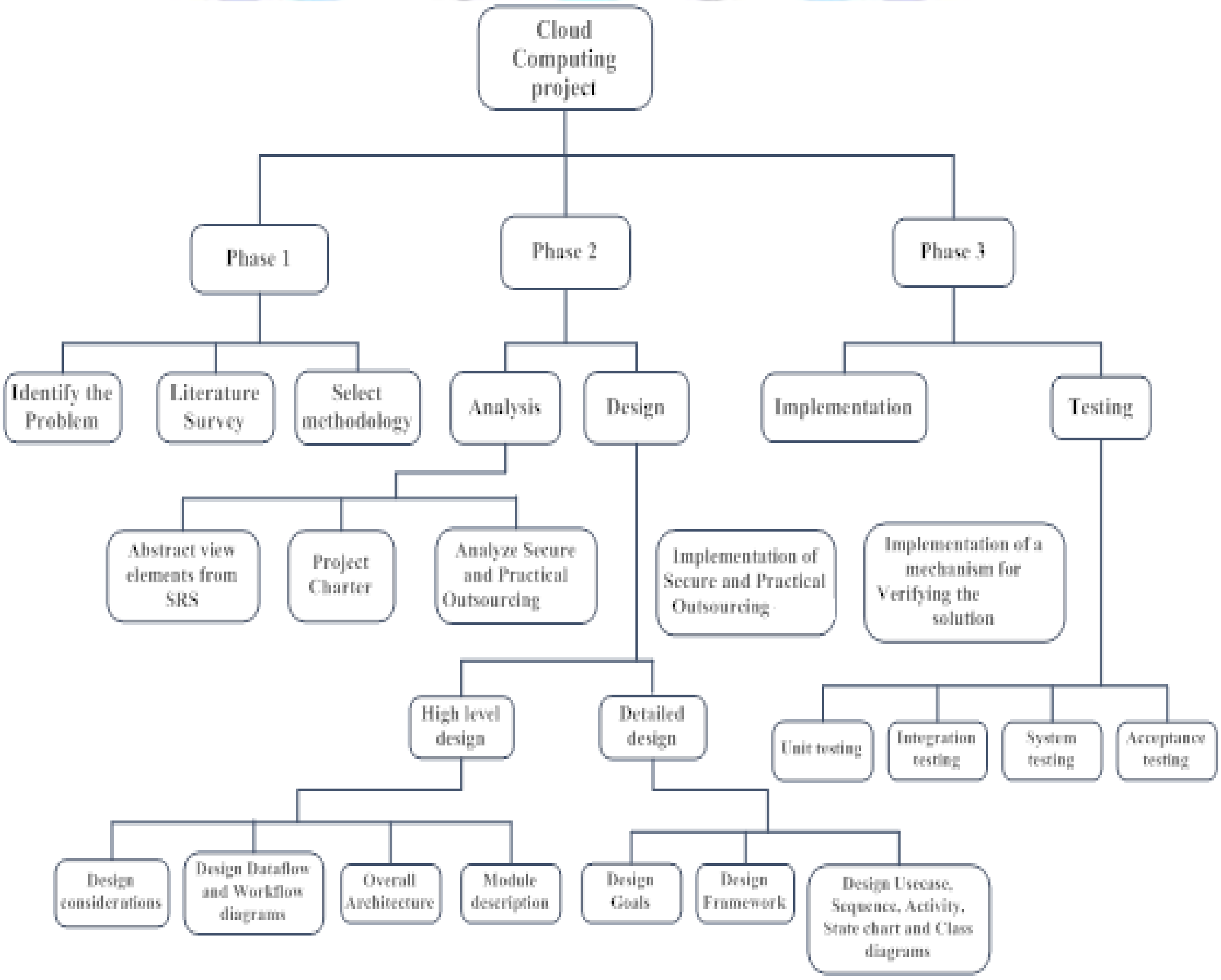


Figure 3: Product Work break down Structure

4.4 Cloud Implementation Methods

We are following an Iterative and Incremental cloud development in building the cloud application because Iterative and Incremental development is at the heart of a cyclic cloud apps development process developed in response to the limitations of the waterfall model. This starts with a preliminary forecast and concludes with implementation in cyclical communications in between. Phases involved in repetitive and progressive improvement practice is as follows:

- **Initiation:** Inception determines project span, hazards, and obligations (functional and non-functional) at a high level but in enough detail that work can be approximated.
- **Embellishment:** Elaboration conveys a working architecture that alleviates the zenith risks and performs the non-practical requirements.
- **Creation:** Construction accumulatively plugs-in the architecture with fabrication-ready code produced from study; devise blue print, execution, and testing (trial and error) of the serviceable requirements.
- **Transition:** evolution conveys the structure keen on the production in cloud commission environment.

4.4.1 Cloud Data Flow diagram of the Project: The dataflow can be shown at two different levels:

1. Level 0 DFD, which is also called as Context Diagram
2. Detailed DFD

The level 0 is the initial level Data Flow Diagram and it's generally called as the Context Level Diagram (A diagram giving an entire system's data flows and processing with a single Process (circle) is called a context diagram.) Later, the context diagram is expanded into a number of interconnected processes. Each method may be further lengthened into a set of interrelated sub processes. This method of expanding a DFD is known as levelling.

The dataflow of cloud system is shown as below in fig 4.

Level 0 – Context Diagram

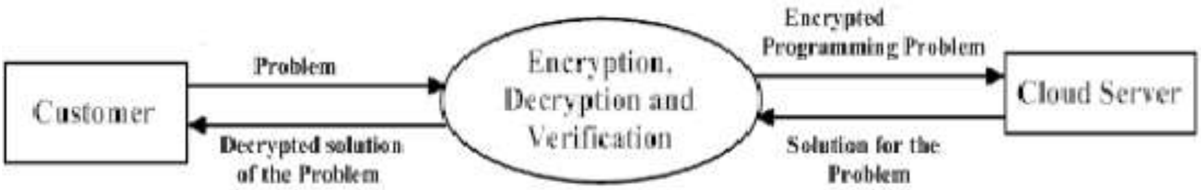

Customer
Problem
Decrypted solution of the Problem
Encryption, Decryption and Verification
Encrypted Programming Problem
Solution for the Problem
Cloud Server

Figure 4: Level 0 Context Diagram for Dataflow of cloud system.

Context diagram gives an overview of our project and shows the various external actors and existing methodologies that are used in cloud setup. Here, we have a customer who sends a computing problem to the Encryption and Decryption process where it encrypts the problem and sends it to the cloud system. Cloud system inturn finds the solution using the anticipated methodology. The implemented methodology provides the required and correct solution for the problem to the cloud setup and then cloud sends that solution to the encryption and decryption process where the solution will be decrypted and it will be sent to the requested customer in a decrypted format [23].

5. Cloud Project Management Scope (CPMS)

The Cloud project capacity characterizes all certified work of the project. The project scope statement is foundation on facet and scope of the product and project management scope as shown below in fig 5.

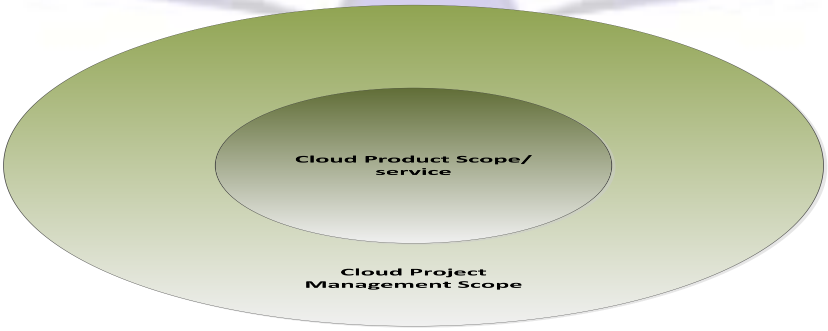


Figure 5: Cloud Project Scope

The product scope includes all technological outputs and work actions fashioned during the product development phases of the cloud systems life cycle (CSLC). The CPMS comprises of the sketching (planning), supervising, managing-controlling and quality assertion work activities produced throughout the cloud system project management life cycle. The project's management scope includes the planning and designing the blue print, monitoring, metering, controlling, authorizing and quality assurance work activities produced throughout the project management life cycle.

Input:

- Project charter (Refer: SRS)
- Stakeholder register
- Stakeholder management strategy
- Work break down structure
- View elements

Output:

- Project class diagram as shown in fig 6.
- Product class diagram as shown below in fig 7.
- State chart diagram

5.1 Cloud Project class diagram

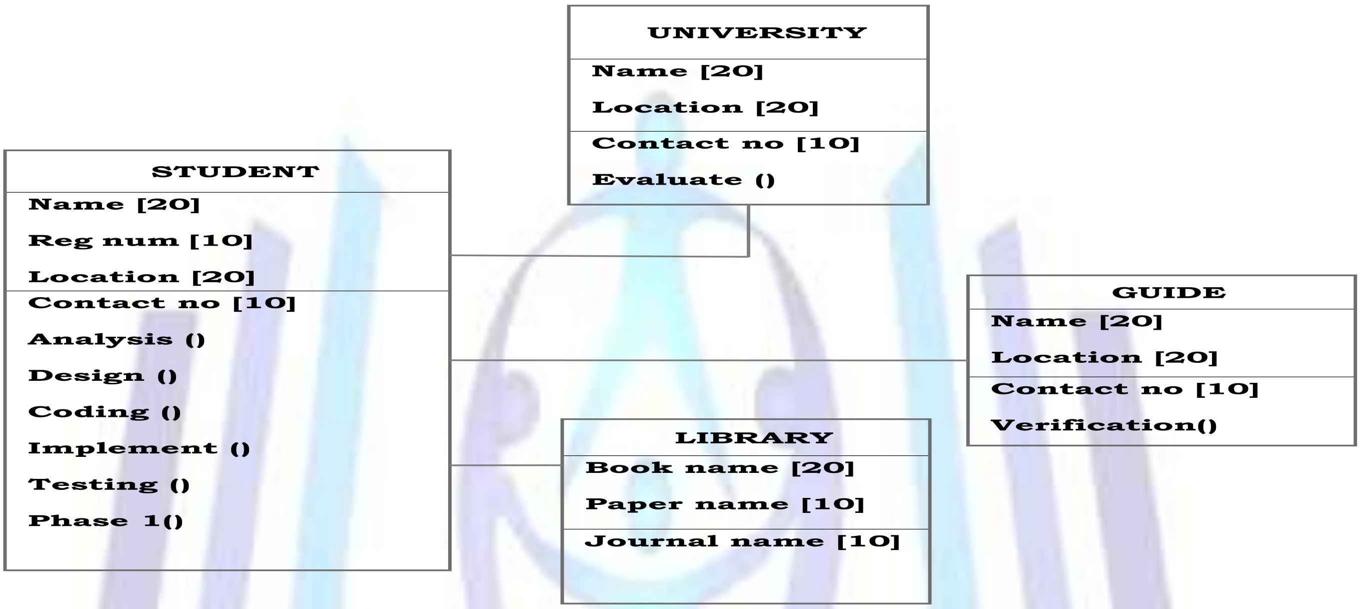


Figure 6: The Analogy Cloud Project class Diagram

5.2 Cloud Product class diagram

Product class diagram shown below gives the description of various classes involved and also their attributes. There are five Classes namely:

1. **Cloud Administrator Class:** Administrator performs various methods like login, editing user profile, updating the profile, selecting the server, changing passwords etc. It includes various attributes like username, password, address, server name etc.
2. **Cloud Client Class:** Client or a user performs methods like register, login, edit profile, upload file, down load file, change password etc. It includes various attributes like user name, password, address, phone number, filename etc.
3. **Cloud Data Access Object Class:** CDAO class involves various methods like admin DA where the data related to admin will be accessed from the database, user DA where data related to user will be accessed, common DA where any common data related to admin or user will be accessed. It includes various attributes like user name, password, address, phone number, filename etc.
4. **Cloud Utility Class:** Utility class mainly involves various methods like read file, write file, upload file, download file, delete file, get result, encrypt and decrypt. It involves attributes like file name, key, file ID etc.

Cloud Product Class diagram is as shown below in fig 9:

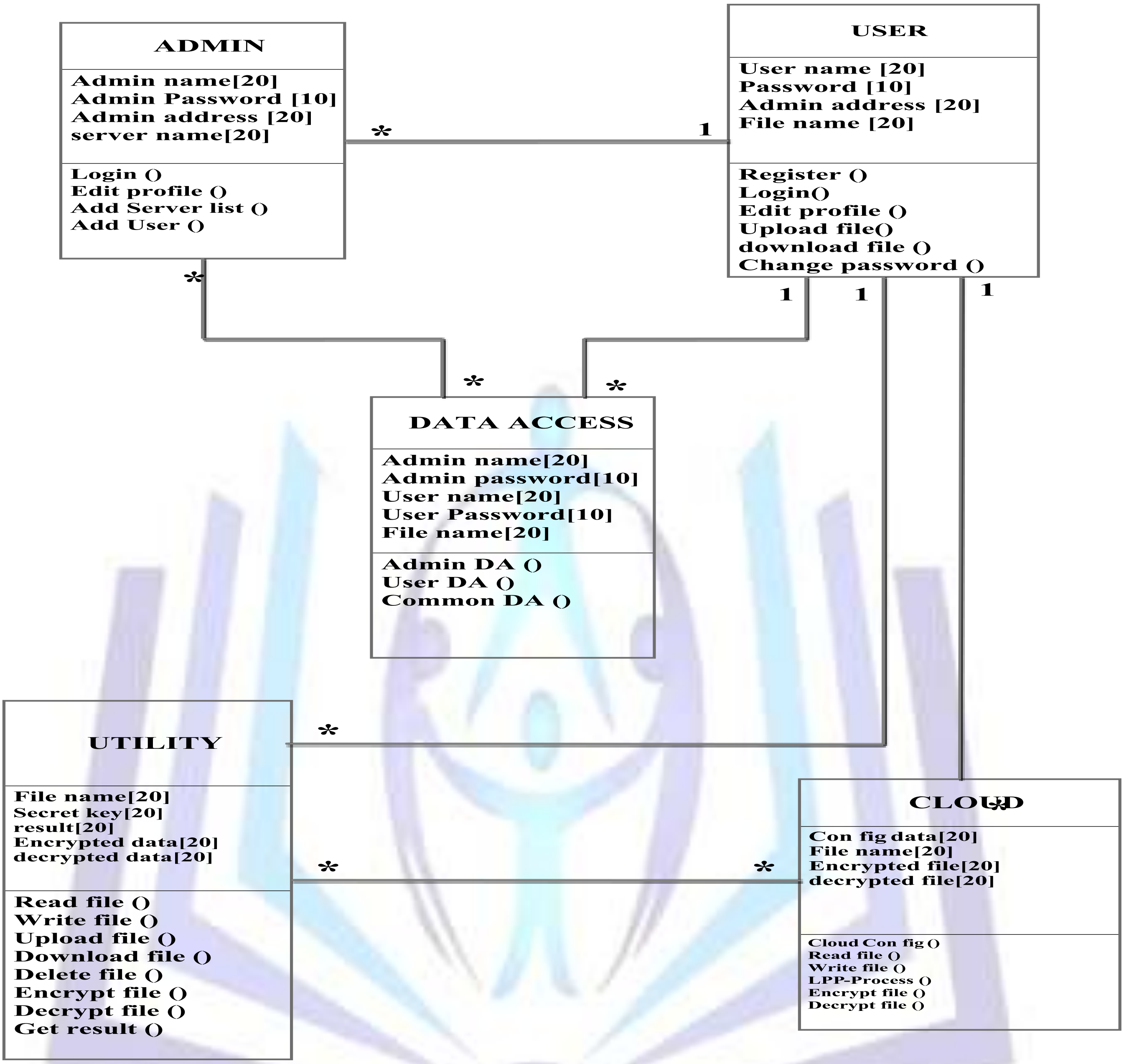


Figure 7: Cloud Product Class diagram

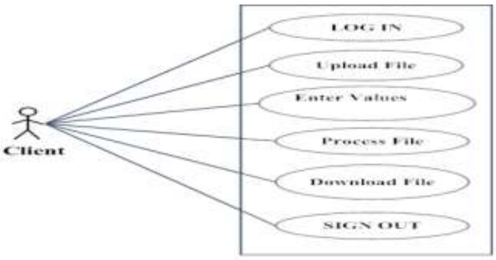
5.3 Cloud Service UML Diagrams

Input: Project and Product Class diagrams, Use case, Actors and Actor interface, State Chart Diagram, Signature, Messages and their types, Object states, Deliverables, Activity chart(Critical path), Work flow graph.

Output:

- Product use case diagrams as shown below in fig 8, 9, 10.
- Sequence diagrams as shown below in fig 11, 12, 13.
- Activity diagram

5.3.1 Use case diagram: A use case is a deposit of circumstances that recites an interface between a customer and an organization. A use case chart exhibits the association amongst actors and use cases. Its principle is to present a graphical synopsis of the utility provided by a structure in provisions of actors, their targets (embodied by use cases), and any adjuncts amid those use cases. The foremost rationale of a use case diagram is to illustrate what system functions are executed for which actor. Tasks of the actors in the scheme can be depicted. The major principle of a use case graph is to demonstrate what scheme functions are accomplished for which actor. Errands of the actors and their association amongst the use cases in the system can be portrayed. The use case depicts an order of actions that afford incredible quantifiable value to an actor and is drawn as a parallel ellipse. An actor is a personality, institute, or exterior organization that participate a role in single or additional communications with the structure. Associations amongst the use cases and actors can be illustrated as given below.



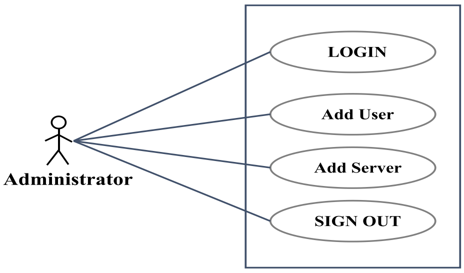


Figure 8: Client Use case Diagram

Figure 9: Administrator Use case Diagram

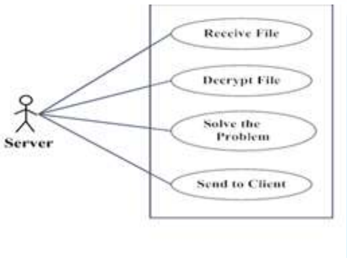
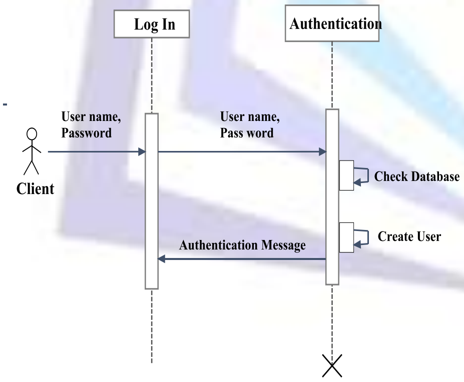


Figure 10: Server Use case Diagram

5.3.2 Cloud service Sequence Diagram

Unified Modelling Language series charts models the stream of logic within the structure in a diagram approach, facilitating together to manuscript and authenticate logic, and are universally used for both investigation and design rationale. Sequence diagrams are the primarily trendy UML relic for vibrant modelling, which aims for recognizing the protocol contained by system. So let us start with sequence diagram of the structure. It is a kind of communication diagram that demonstrates how methods function with one another and in particular fashion. Sequence diagram for each Use case is as shown below [24]:



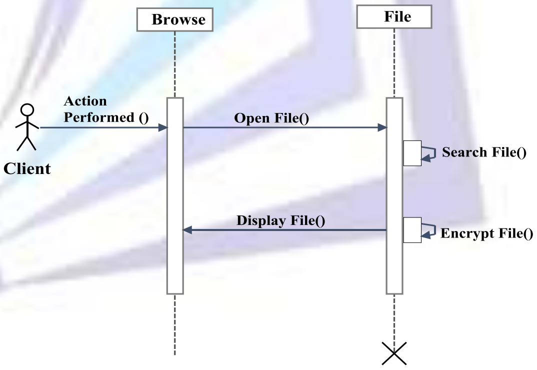
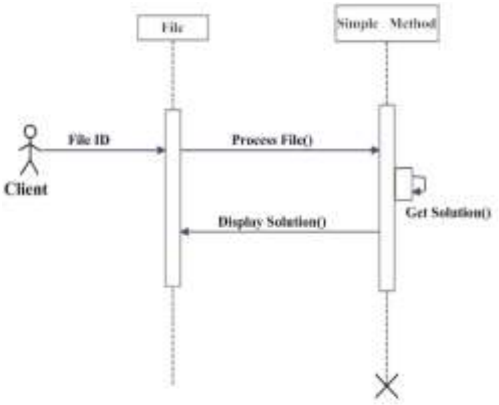


Figure 11: Sequence diagram for Client Log In

Figure 12: Sequence diagram for Client uploading a file



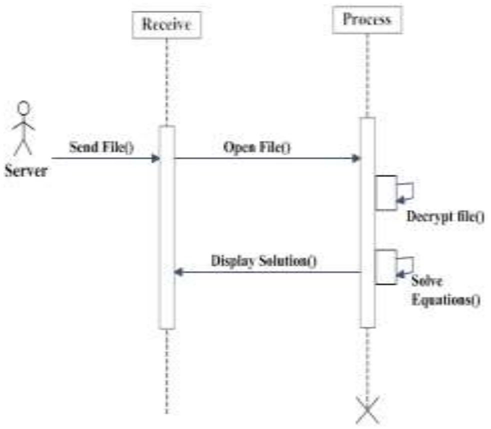


Figure 13: Use case diagram for client Processing a file and Downloading the Solution

Figure 14: Use case diagram for Server receiving the file, decrypting and storing

6. Conclusion and Future Enhancement

There is still a lot of research work going on to completely nullify the issues related to outsourcing the services to Cloud computing. Though the proposed mechanism is better than the existing schemes, there is still lot of scope for enhancements We further plan to investigate some interesting future work as follows:

- Devise robust algorithms to achieve stability and confidence in the customers to use the cloud service
- investigate the sparsely structure of predicament for further competence improvement;
- Establish strict security framework;

Acknowledgement

Our, sincere thanks to **Dr. Shylaja S.S.** , HOD, Department of Information Science and Engineering, PESIT, Bangalore, for their constant encouragement.

References

1. "NIST definition of Cloud Services and technology", National Institute of Science and Technology.
2i. Margaret Rouse, "Cloud ecosystem", Search – cloud – provider – tech - target, October 2011.
2. David C. Hay, "Requirements analysis: from business views to architecture", Upper Saddle River, N.J. ; London : Prentice Hall PTR, 2003.
3. **"**Project Charter Template", cdc – gov - templates- CDC - UP.
4. P Saravanan, BalajiMurugan, Lilima James, "Project-lifecycle", Slideshare.net.
5. Eng. Riham Z. Mqat, "How to write a Software Specification", Islamic University of Gaza Faculty of Engineering, CE Department ECOM, 3324 Software Engineering
6. Alistair Cockburn and Jim Highsmith, "Writing effective use cases – Agile software development", series editors,-Highsmith series editors, low price edition, 2010 Pearson Education, Inc.
7. Steve McConnell, "Software Project Survival Guide", - Mircosoft, Low cost edition, 1998.
8. "Scribd_com_doc_39584009_Synopsis-on-e-music-project", Scribd Docs.
9. "Consult-smp_SMP_policy_Code_of_Conduct.d", STRATEGIC MANAGEMENT PARTNERS (SMP), CODE OF CONDUCT FOR CERTIFIED MANAGEMENT CONSULTANTS, (CMC).
10. "ARCHITECTURE RESOURCES FOR Enterprise ", Advantage_bredemeyer, BREDEMEYER CONSULTING.
11. "Confidentiality: A Serious Business", Efoli Ekot, 04 October 2010.
12. "Demetersolutions.fr_ppolicy", Blog.
13. "Matrix Consulting code of ethics and compliance", Blog.
14. "Code of Ethics and Corporate Conduct", Blog.
15. "Imcsa_org_za_fileticket_1135", Certified Management Consultant (CMC) Application Pack.
16. "Techrepublic_blog_tech-manager_consider-enterprise-environmental-factors-when-planning-it-projects_552", Blog.
17. "Confluence_cornell_edu_display_CUSDSIG_Customer_Bill_Of_Rights", Cornell university
18. "Investorsinpeople_co_UK_Documents_CODE+OF+CONDUCT+FINAL", Oct 04.
19. "Faculty_csuci_edu_minder_chen_Project_slide_PM_Initiation_ppt", CSUC university.
20. "docstoc_Data-Flow-Diagrams-Worked –Example", Docstocs.
21. "capitaemployeebenefits_ What we do _Administration", blog.
22. "Docstoc_docs_6573493_UML-Diagram", DocStocs.
23. Manu A. R, V. K Agrawal, K N Bala Subramanya Murthy, "A Study on Security and Privacy Issues of Cloud Eco-System", International Journal of Applied Information Systems, April 2013. Published by Foundation of Computer Science, New York, USA.
24. Manu A R, Dinesha H A, Manoj Kumar M, "An Approach to Enhance Security of Cloud Computing Services using Software Engineering Model", IJCA April 2013 Edition, April 18, 2013.

Author's biography

Dr. Vinod K Agrawal, is currently Professor (ISE) & Director CORI Lab, in the Department of Information Science and Engineering, PESIT, Bangalore, with Ph.D. (Indian Institute of Science, Bangalore), he as worked as scientist for about 30 years with ISRO Satellite Centre, Bangalore, and 2 years with GM R&D Centre, Bangalore. He has **expertise in** Fault-tolerant Computing, Embedded Systems, Distributed Computing, Cloud Computing, His Research interests includes Software Engineering, Formal Methods in Software Engineering, Reconfigurable, Cloud computing, Networks etc.

Dr. K N Balasubramanya Murthy, is the Principal & Director, of PESIT, and Professor in ISE Dept. PESIT Bangalore with, Ph. D from IIT-Madras, M.E. in Indian Institute of Science, Bangalore,, post-doc fellowship - New Mexico State University, Las Cruces, USA He has Has 30 years of experience in teaching, Training, Industry, Administration, and Research. His area of Research interest is Image Processing, Parallel computing, Cloud Computing, and Computer Networks.

Mr. Manu A R has completed Masters in Physics, from university of Mysore, AMIETE in CS and Eng,IETE, New-Delhi With M.Tech in CS and Eng, from University of Mysore, He has research experience in NITK surathkal, Mangalore, Karnataka, India, past he was worked in IT industry with various organizations, Currently he is Pursuing his P.hD. research as a FullTime scholar, under TEQIP, in PESIT, Bangalore, VTU-Belgaum, under the guidance of Dr. V K Agrawal, and Dr. K N Balasubramanya murthy, his area of research is Cloud Computing and its security, He is interested in Operating Systems, Software Engineering, Networks , and algorithms.