

# My Second Coursework

Apurv Deshpande  
15001687  
MSc in Cloud Computing

11th April 2016

## Abstract

Cloud Computing provides scalable, reliable, pay per use and resource pooling features that will benefit organisations by migrating their applications to the cloud. But the migration process is complex hence different patterns has been proposed to facilitate this process. In this research a model driven approach is proposed to recommend design patterns that will help in software migration to cloud The research is divided into following three tasks. First task is extraction of design pattern from the on premise application ,second task is to mapped this design patterns with cloud based design patterns and then this application is deployed on cloud using cloud deployment models.

## Contents

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Introduction</b>  | <b>1</b> |
| <b>2</b> | <b>Literature Review</b>   | <b>2</b> |
| 2.1      | Research Papers on Software Migration to Cloud. . . . .                  | 2        |
| 2.2      | Review of design pattern research papers . . . . .                       | 5        |
| 2.3      | Multi-tenancy Patterns: Review of Database Driven applications . . . . . | 6        |
| <b>3</b> | <b>Research Method and Specification</b>                                 | <b>6</b> |
| 3.1      | Basic Architecture . . . . .   | 6        |
| 3.2      | Research Specification . . . . .   | 7        |
| 3.3      | Tools used . . . . .   | 7        |
| 3.4      | Activity Diagram . . . . .   | 9        |
| 3.5      | Alternative Approach . . . . .   | 9        |

## 1 Introduction

Cloud migration is evolving field in cloud computing. Migration of applications to cloud is not easy. Applications build in the cloud has different requirements as compared to application build on-premise. Before migrating their application to cloud users should take into account the various pros and cons of their software migration. The basic questions will be the migration process easy, will it be beneficial to migrate their application

to cloud. For example consider this three types of applications. Database driven applications, service oriented applications and legacy applications. Database driven applications will have special challenges like storage space required, migrating database from on premise to cloud environment or migration of data if database migration is not possible. Service oriented applications has same properties as cloud environment hence some of this applications can be directly migrated to cloud without any re-factoring or redesigning. For legacy applications changes in the source code is needed before migrating the application to the cloud. As there are design patterns for software development which can be reused to facilitate the development of software there are design patterns for migration of applications to the cloud. Although they are not globally defined but research work is going on to finding the common patterns that can be used in cloud deployment. By using this design patterns along with the modelling techniques migration of application to the cloud can be a easy process. Various research is done in this field but very few of them are model based. Hence I recommend a model driven approach to recommend suitable patterns for software migration to cloud.

## 2 Literature Review

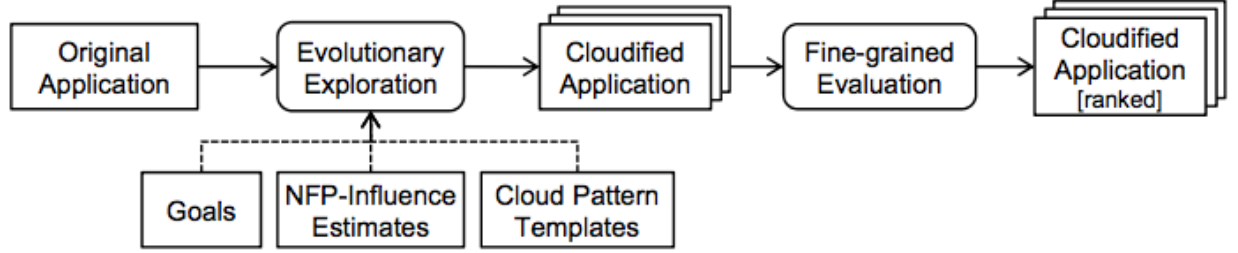
In this section we discuss about the research carried out in field of software migration to the cloud.

### 2.1 Research Papers on Software Migration to Cloud.

Many Patterns has been proposed for software migration to cloud but selection of pattern that will ease the developers in migration of applications is still a major Challenge.

In this research paperFleck et al. (2014). A model based optimization technique is proposed to help developers to select the most appropriate patterns for migrating their application to cloud.This research is carried out by using non functional properties of an application. The first approach to this research is goal modelling.This research is carried out in two phases. In the first phase a genetic algorithm called NFPDeb et al. (2002) which uses estimate to find out a set of solutions. In the next phase model simulation is used to support solution i.e to find out the most appropriate pattern. Caching,Horizontal Scaling and Auto scaling patterns were used to prove this. This patterns will help the application to achieve scalability. In this research the base architecture is considered suitable for the cloud. This is not the case with all the applications. Some applications needs to rebuild and re-factor before moving to cloud. Also for simulation e-motions were used which develop problems when application is complex and it also has some scalability issues. Also this research has limitations when there are large number of patterns and when the application is complex.

This paperFuhr et al. (2013)proposes migration of legacy systems into Service Oriented Architecture(SOA). It was implemented using Model Driven Techniques. This research is close to research I am conducting as migration to SOA are closely related to migration to cloud systems and they have used model driven techniques. A model driven service oriented modelling and Architecture(SOMA) is used which uses incremental and iterative phases to identify,specify and implement services.The next research is about migration of service based applications to Cloud Environments. The limitation of this research is that it is written only in Java. Hence this application is platform dependent. In my Research the application migrated to the cloud will be platform independent.

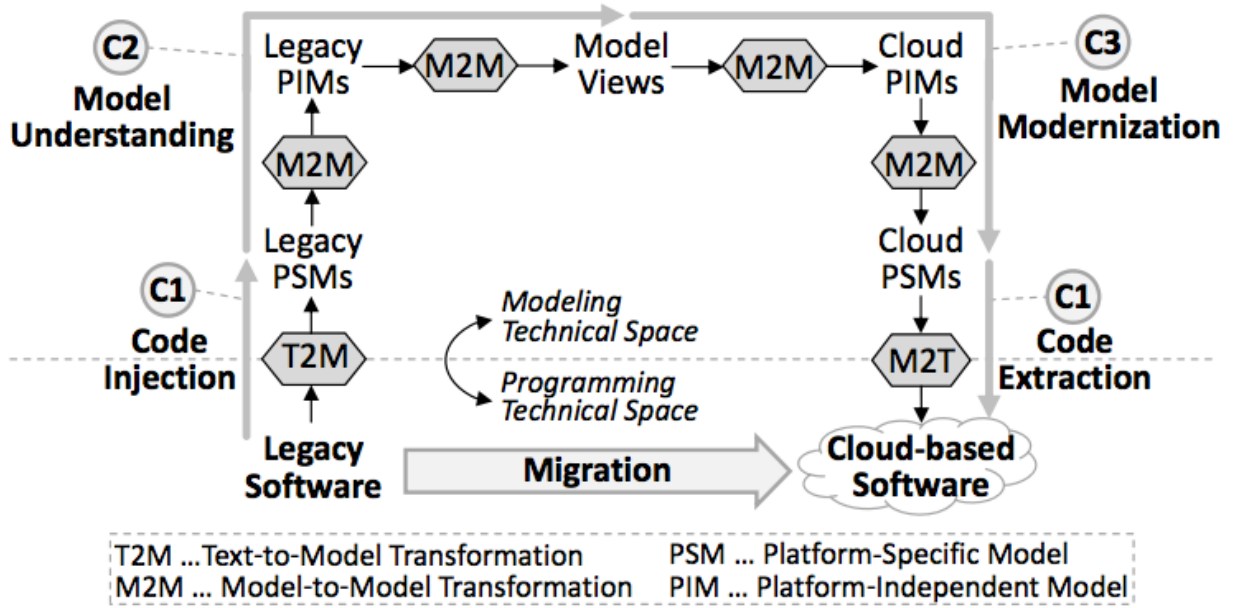


**Fig. 1. Approach Overview**

The paper Fehling et al. (2013) discusses migration of service based applications to cloud using service oriented architectures. Service oriented Architectures closely resembles to cloud applications as both have common functionality as both uses loosely coupled integration to form a distributed application. In this research mapping of architectural principles of service based application and cloud based applications are done which help service based applications to easily migrate to cloud. The advantage of this research and its use in my research is that the migration patterns like Forklift Migration, Stateless Component Swapping and Hypervisor swapping can be used in the form of reusable migration patterns to migrate similar applications to the cloud. The shortcoming of this research is link between the semantics is not clearly defined. The next paper discusses about more about methodology and tools for model driven migration.

This paper Mohagheghi and Sæther (2011) given an idea about the complexity in software engineering and challenges it possess while migrating to service cloud paradigm. It also discusses about the challenges in cloud environment and various things that developer should take into consideration before migrating their application to cloud. The first step before migrating the application to cloud is that the application should be decomposed, decoupled and be made scalable. The next step is that the application should be modernised. The modernisation can be achieved either by using black box modernisation or white box modernisation. Black box modernisation is relevant to my research as it concerns only with the input and output. The application migration carried out by modernising the architecture of application, then by modernising the data and finally by testing the application after migration to cloud. This research can be further enhanced if performance testing can be done and load balancing can be done of migrated applications. This research has limitations if the applications has scalability issues and storage issues. The storage issues can be solved by using cloud deployment pattern like multi-tenancy patterns. This research is flexible as it is platform independent. Application developed in any programming language like VB, .NET can be easily migrated using this approach.

This paper Bergmayr (n.d.) discusses the shift of applications from on premise to cloud using model transformation techniques. The first step is to convert the code of application into platform specific model that will tell the contents of legacy code. Then this platform is converted into higher platform which is independent from the legacy code platform. Then this platform will use UML modelling and create UML based models which will be modernised and then migrated into cloud. The main challenges for this research is Code Injection and Extraction, Model Understanding and Model Modernisation. In contrast to this research I will be using reversing engineering for code extraction. This paper can be further enhanced if the models can be generated from grammar using the lexical analysis. The next paper is about the use of UML diagrams for modelling the applications.



In this researchGuillén et al. (2013) the application modelling of applications is discussed by the use of UML diagrams which will help to develop applications in platform independent manner. By using the model driven development application will not be dependent on any particular platform and it will not be coupled to specific platforms. In this approach the application is divided into various components and different classes and this classes are modelled differently. This research suggests a MULTICLAPP Framework. It is designed in three stages. In the first stage the cloud application are modelled then application functional behavior is encoded in the second stage and finally automated models are generated in the final stages. This research will help to solve basic cloud computing problems like vendor lock in. This research can be improved in the second stage by using behavioral design patterns to code the applicationWolfgang (1994).

In this paperJamshidi et al. (2013) a review on cloud migration research topic is done and what are current limitations, tools available and open issues are discussed. This research shows that there is lack of tool support available to automate migration tasks into the cloud. This research also discuss the similarities and differences between SOA and Cloud migrations. It mainly focuses on the issues of migration of legacy applications into the cloud.

This researchFuhr et al. (2012) is about Model-Driven Software Migration which combines forward and reverse engineering techniques to migrate legacy software into service oriented architecture. In this paper they have developed a SOAMIG model to migrate legacy systems into service oriented architecture. There are four phases of SOAMIG model. They are Preparation, Conceptualization, Migration and Translation. I am going to use process model and tool support used in this research in my research as service oriented architecture and cloud basically have the same properties and migration of application is almost same in service oriented architecture as in cloud. The preparation phase includes the preparation of application by doing some partial renovation.In conceptualization phase the feasibility of the application is checked by firstly dividing the application into smaller components and then by migrating some of this components and checking whether it is possible to migrate the application or not. In the migration stage firstly the migration is setup and then the whole application is migrated. In the transition

phase the improvements in the migrated application is done by adding some code. The preparation phase can be improved by reverse engineering the application to extract the design patterns used by the application. This will help to know the reusable components of the application and thus will help in migrating the application. For conceptualisation and Migration phase seven disciplines are suggested. They are business modelling, legacy analysis, target architecture, Strategy Selection, Realization, Testcases and Cutover. A generic tool suite called SOAMIG tool suite was also suggested to help in Migration Analysis. Also extractor tools are suggested to extract the convert the legacy code into models. The next section discusses about the papers as explain how model driven approach can help to extract design patterns from the application. This paperJamshidi et al. (2015) discusses about the the cloud migration focusing on the architectural perspective of the application. In this paper appropriate architecture migration patterns are based on the objectives of the application to be migrated. The migration patterns that are re-architecture include Redployment pattern, Cloudification, Relocation, Multi-Cloud Refactoring, Multi-Cloud Rebinding, Replacement and Multi-Application Modernisation. This research will help me in migrating application that need architectural changes before migrating the applications to the cloud.

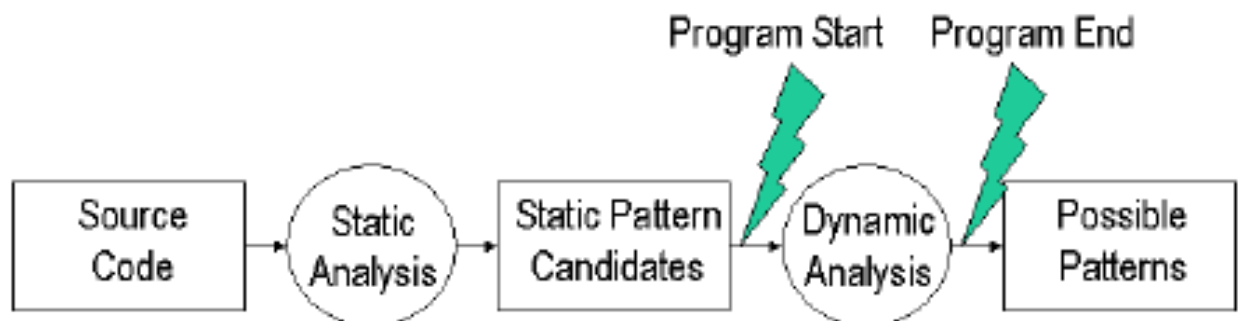
In this research design pattens in code is detected by the combination of static and dynamic approaches.

The next section consist of research papers about how database driven driven applications can be migrated into cloud using design patterns

## 2.2 Review of design pattern research papers

This paper Alnusair and Zhao (2009) describes the reverse engineering process to extract design patterns from application by using model driven approach. An ontology model called Web Ontology Model (OWL) is used along with Resource Description Framework (RDF), Semantic Web Rule Language(SWRL) and SPARQL Query language to extract the design patterns from the source code. This research was used to recover common design patterns like observer pattern and visitor pattern. I am going to use this tools and research to extract design patterns from the applicatons and mapped this design patterns with cloud design patternsLeymann et al. (2014).

In this researchHeuzeroth et al. (2003) design pattens in code is detected by the combination of static and dynamic approaches. This research is different from the above as it does not rely on coding. In this research code is divided into smaller components. Static analysis is done on this smaller components and dynamic analysis is done by executing the smaller codes. It means at first the static analysis is done and then this results can be further treated using dynamic research.



This paper Jamshidi et al. (2015) discusses about the the cloud migration focusing on the architectural perspective of the application. In this paper appropriate architecture migration patterns are based on the objectives of the application to be migrated. The migration patterns that are re-architecture include Redployment pattern, Cloudification, Relocation, Multi-Cloud Refactoring, Multi-Cloud Rebinding, Replacement and Multi-Application Modernisation. This research will help me in migrating application that need architectural changes before migrating the applications to the cloud.

The next section consist of research papers about how database driven driven applications can be migrated into cloud using design patterns.

## **2.3 Multi-tenancy Patterns: Review of Database Driven applications**

In this research Adewojo, Bass, Hui and Allison (2015) cloud computing deployment patterns are proposed to support the cloud migration process. A template is applied to shared component pattern. The research uses the cloud design patterns in comparison with object oriented design patterns. The contribution of this research is enhancement of shared component pattern which is a cloud deployment pattern. The shared component pattern is the first level of multi-tenancy pattern. The advantage of this application is that it minimises resource usage in application. The application developed using this research was implemented by using the following patterns: Model-View-Controller(MVC), Service Oriented Architecture(SOA), Representational State Transfer(REST) and cloud deployment patterns. The limitation of shared component pattern is that it compromises privacy, performance and security. The next paper is an extension of this paper in which tenant isolated pattern and dedicated pattern is discussed.

This paper Adewojo, Bass and Allison (2015) describes the multi-tenancy property of cloud computing. Here the tenant-isolated and dedicated component pattern is used to facilitate migration of database driven application to cloud. This patterns are implemented in data storage model of the application. The disadvantage of tenant isolated pattern is that it affects the performance of the application and also security is compromised. The disadvantage of dedicated component is that it is very costly but its advantage is that it provides the highest level of privacy.

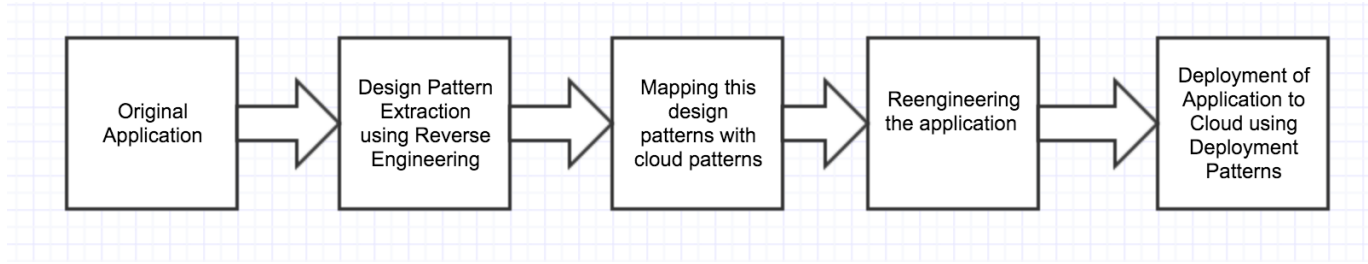
## **3 Research Method and Specification**

This section describes the research method that will be using to describe my research proposal.

The proposal is to recommend suitable patterns for software migration to cloud.

### **3.1 Basic Architecture**

The basic flow of the research is as follows. The first stage is to extract the design patterns used by the application. It is done by re-engineering the application source code. Another approach is to use static and dynamic analysis of the application to extract design patterns. The next stage is to mapped the design patterns extracted form source code with cloud patterns Erl et al. (2015). This will re-engineer the application with cloud properties; for example, scalability, flexibility and resource sharing. The final stage is to deploy this application to cloud using cloud deployment patterns.



### 3.2 Research Specification

The first stage is to extract design patterns from a given application. Some of the tools that extract the design patterns include Semantic Web Ontology (OWL), Resource Development Framework (RDF), Semantic Web rule language (SWRL) and SPARQL Query Language. In the stage of mapping cloud patterns with design patterns can be analysed by mapping the characteristics of both and getting a unified solution. This can be done by using semantic based models. By mapping the design patterns with cloud will help in re-engineering the application. By mapping the design patterns with cloud based patterns the application will be re-engineered and then it can be migrated to the cloud. By mapping the patterns the basic functionality of the application will not change hence there is no need of completely redesigning the application.

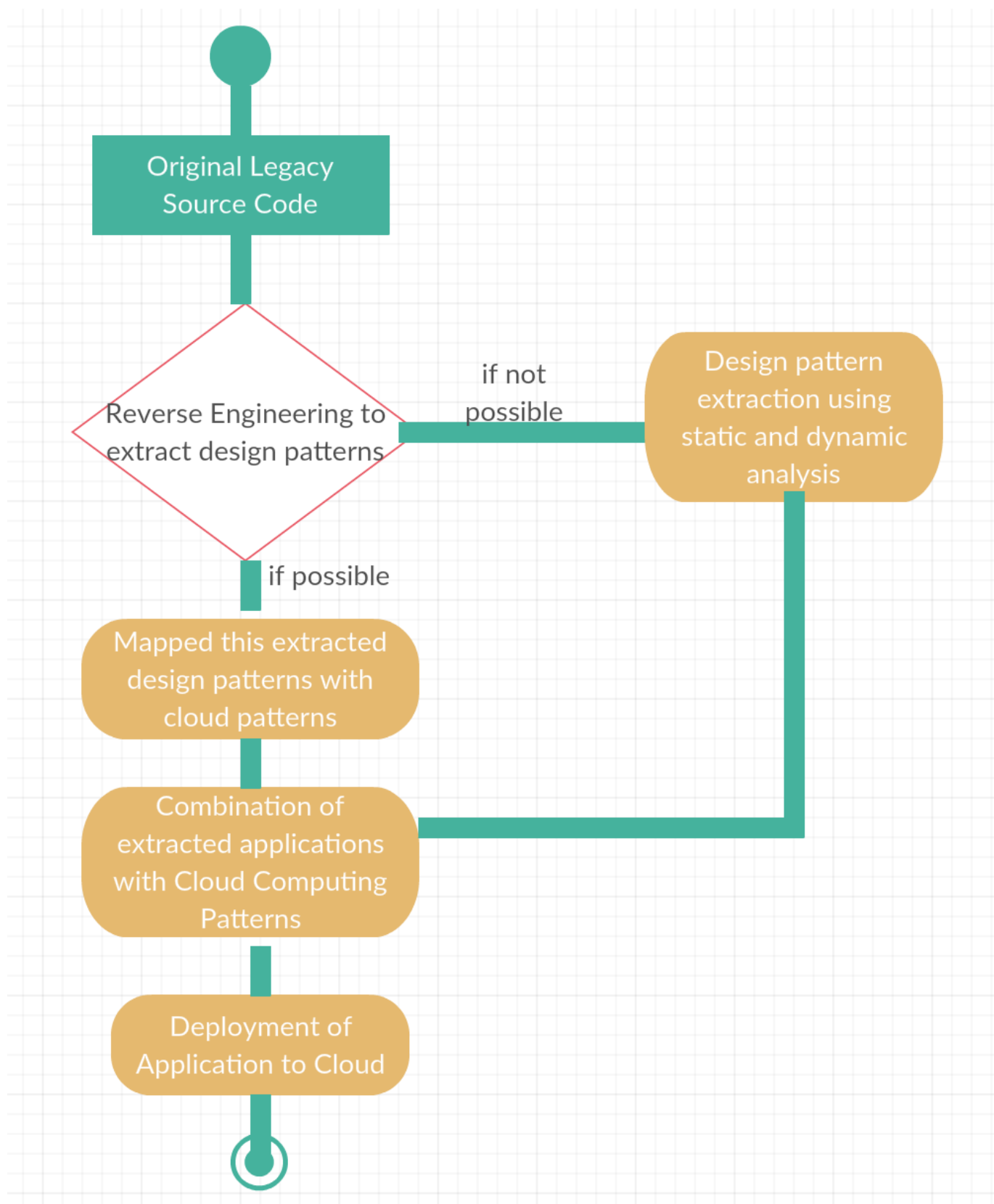
The application redesigning phase is where application is created using design patterns of both cloud and object oriented programming. Object Oriented design pattern include Creational Patterns, Structural Patterns and Behavioral Patterns. Creational Patterns include creation of new object. The applications that are developed in cloud are more virtual in nature.

The cloud deployment patterns that has to be chosen depends on the the type of application. If the application is database driven then multi-tenancy patterns can be used. If the application is service oriented then service facade pattern can be used.

### 3.3 Tools used

The application going to be created in Java language in my implementation but it can be created in any other language. The tools that I am going to use for re- designing in Java language are Java AWT Swing. Maven and Gradle will be used to create dynamic web projects. JMS will be used for message transferring. JUnit will be used to test the java applications. Hibernate will be use for object relational mapping and also for connecting to the databases. This whole application will be developed in Eclipse which is a open source integrated development environment.

The migration testing that will be used to check whether the application has been fully migrated to cloud include database migration testing which is used to check whether the database has been successfully migrated to cloud or not. Also technology migration testing is done to check whether the application has been successfully migrated to cloud. This includes the testing from on premise application to the new application that was created that migrated to the cloud

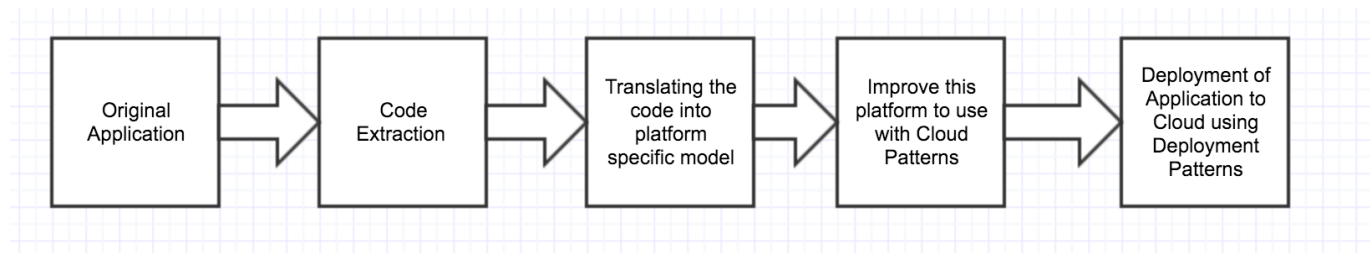




### 3.4 Activity Diagram

The below figure is the activity diagram of my research. The green colour is the object. The diamond is used for decision making. The orange colour describes the various activity.

### 3.5 Alternative Approach



The original application can be in any language like Java,.Net etc but the design patterns used is language independent. Hence it make sense to extract the design patterns rather than extracting source code as it will be a time consuming process to extract source code from different language. But in some cases if the application is too complex that retrieval of design patterns is not possible. In such an alternative approach is in place to translate code into platform specific models and then modernise this platform to be used with cloud computing patterns for software migration. This is an alternative approach in case the first approach does not work. Also this platform specific model will be a virtual pattern to automate all the tasks and it is easy to modernise the virtual pattern and use with cloud patterns as cloud supports virtualisation.

|   | Task Name  | Duration | Start    | Finish   |   |   |   |
|---|--|----------|----------|----------|---|---|---|
|   |  |          |          |          | S | M | T |
| 1 | Reverse Engineering to extract design patterns               | 10d      | 05/23/16 | 06/03/16 |   |   |   |
| 2 | Code Extraction tools using Reverse Engineering              | 10d      | 06/03/16 | 06/16/16 |   |   |   |
| 3 | Mapping of extracted design patterns to cloud patterns       | 10d      | 06/16/16 | 06/29/16 |   |   |   |
| 4 | combination of patterns to platform specific model           | 10d      | 07/29/16 | 08/11/16 |   |   |   |
| 5 | Deployment of Application to Cloud using Deployment Patterns | 10d      | 07/08/16 | 07/21/16 |   |   |   |
| 6 | Extraction of legacy code from application                   | 10d      | 07/21/16 | 08/03/16 |   |   |   |
| 7 | Testing the Application                                      | 7d       | 08/08/16 | 08/16/16 |   |   |   |
| 8 | Making the final dissertation                                | 11 d     | 08/16/16 | 08/30/16 |   |   |   |

## References

- Adewojo, A., Bass, J. and Allison, I. (2015). Enhanced cloud patterns: A case study of multi-tenancy patterns, *Information Society (i-Society), 2015 International Conference on*, IEEE, pp. 53–58.
- Adewojo, A., Bass, J., Hui, K. and Allison, I. (2015). Cloud deployment patterns: Migrating a database driven application to the cloud using design patterns, *Proceedings of the World Congress on Engineering and Computer Science*, Vol. 1.
- Alnusair, A. and Zhao, T. (2009). Towards a model-driven approach for reverse engineering design patterns, *Proceedings of the 2nd International Workshop on Transforming and Weaving Ontologies in MDE (TWOMDE 2009). Denver, Colorado, USA*, Vol. 531.
- Bergmayr, A. (n.d.). Cloud-oriented software migration- a model-driven approach, *LASER*.
- Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii, *Evolutionary Computation, IEEE Transactions on* **6**(2): 182–197.
- Erl, T., Cope, R. and Naserpour, A. (2015). *Cloud computing design patterns*, Prentice Hall Press.
- Fehling, C., Leymann, F., Ruehl, S. T., Rudek, M. and Verclas, S. (2013). Service migration patterns—decision support and best practices for the migration of existing service-based applications to cloud environments, *Service-Oriented Computing and Applications (SOCA), 2013 IEEE 6th International Conference on*, IEEE, pp. 9–16.
- Fleck, M., Troya, J., Langer, P. and Wimmer, M. (2014). Towards pattern-based optimization of cloud applications., *CloudMDE@ MoDELS*, Citeseer, pp. 16–25.
- Fuhr, A., Horn, T., Riediger, V. and Winter, A. (2013). Model-driven software migration into service-oriented architectures, *Computer Science-Research and Development* **28**(1): 65–84.
- Fuhr, A., Winter, A., Erdmenger, U., Horn, T., Kaiser, U., Riediger, V. and Teppe, W. (2012). Model-driven software migration: Process model, tool support, *Migrating Legacy Applications: Challenges in Service Oriented Architecture and Cloud Computing Environments: Challenges in Service Oriented Architecture and Cloud Computing Environments* p. 153.
- Guillén, J., Miranda, J., Murillo, J. M. and Canal, C. (2013). A uml profile for modeling multicloud applications, *Service-Oriented and Cloud Computing*, Springer, pp. 180–187.
- Heuzeroth, D., Holl, T., Högström, G. and Löwe, W. (2003). Automatic design pattern detection, *Program Comprehension, 2003. 11th IEEE International Workshop on*, IEEE, pp. 94–103.
- Jamshidi, P., Ahmad, A. and Pahl, C. (2013). Cloud migration research: a systematic review, *Cloud Computing, IEEE Transactions on* **1**(2): 142–157.

- Jamshidi, P., Pahl, C., Chinenyeze, S. and Liu, X. (2015). Cloud migration patterns: a multi-cloud service architecture perspective, *Service-Oriented Computing-ICSOC 2014 Workshops*, Springer, pp. 6–19.
- Leymann, C. F. F., Retter, R., Schupeck, W. and Arbitter, P. (2014). Cloud computing patterns.
- Mohagheghi, P. and Sæther, T. (2011). Software engineering challenges for migration to the service cloud paradigm: Ongoing work in the remics project, *Services (SERVICES), 2011 IEEE World Congress on*, IEEE, pp. 507–514.
- Wolfgang, P. (1994). *Design patterns for object-oriented software development*, Reading, Mass.: Addison-Wesley.