# Modelling with Logics – CSF100-106

Francesca Andreozzi – 853175

## Fanatical Support for Azure, Design Document

by Rackspace for GoCompare.

### Introduction

Azure Cloud System is a cloud system managed by Rackspace. The specification document aims at facilitating the migration from in house Disaster Recovery system to an off-premise solution. The reason is to maximize the system performance in case of failure and integrate the infostructure from in house to the cloud. Whilst the on-premise solution was faster, the Azure Managed Cloud solution is more cost effective. Nonetheless, the Azure Cloud, whilst being slower, aims to be more durable and effective, as less DR (disaster recoveries) are turned on at the same time.

The specification document explains and defines elements of the cloud system and exceptions, such as unexpected behaviour of the elements of the system. It defines the parameters which allow custom solutions to be explored, and solutions to implement the speed of the service.

### Natural Statements

1. “Blob storage stores file data. A blob can contain any type of text or binary data, such as a document, media file, or application installer. Every data blob is organized into an object called a container within each storage account.”

This statement defines the Blob Storage. The Blob storage is used to store temporary files for the application. All text and binary data can be stored in the Blob storage. The storage account is the parent structure of the container, which in turn the parent structure of the Blob storage.

1. “A storage account can contain any number of containers (but it must have at least one), and a container can contain any number of blobs.”

This statement defines the basic characteristics of a Blob storage. It must contain at least one container. A container can contain any number of Blobs. There are no minimal requirements to describe a container.

1. “Blob storage offers two types of blobs: block blobs and page blobs. Block blobs are optimized for streaming and storing cloud objects, and they are a good choice for storing documents, media files, and backups. Page blobs are optimized for representing IaaS disks and supporting random writes. An Azure virtual machine IaaS disk is a virtual hard disk that is stored as a page blob.”

This statement defines the two fundamental types of Blob storage. If a Blob storage is used to stream and store cloud objects, it is called Block Blob. If a Blob storage is used to represent IaaS disks and can support random writes, it is called Page Block.

1. “Rackspace support engineers will connect to the bastion server from known and controlled networks in various Rackspace data centres using the remote desktop protocol (RDP) or SSH. Once access has been established, support engineers will access your environment using RDP and/or WinRM and/or SSH from the bastion host.”

There are three different ways in which the Rackspace engineers can access the user’s environment from the bastion host: RDP, WinRM, SSH. Once they access the bastion server, the engineers can access the environment from any of them but not more than one at a time.

1. “If the infrastructure, performance and redundancy of certain private cloud services cannot be feasibly replicated due to change or technical constraints, then a robust hybrid solution to be explored.”

This statement describes the minimal requirement for the implementation of a robust hybrid solution. In case infostructure, performance and redundancy cannot be replicated, a hybrid solution will be explored. If, and only if the infostructure or performance or the redundancy cannot be feasibly replicated, a hybrid solution will be explored.

1. “Rackspace recommend that the default cryptographic settings are checked and confirmed suitable by the security team and if additional cryptographic requirements are required a custom IPsec/IKE policy be defined.”

This statement describes the basic cryptography settings to be used. The default settings must be checked and confirmed by the security team. If additional requirements are required a custom policy needs to be defined.

1. “AD DS is used to authenticate user, computer, application, or other identities that are included in a security domain. It can be hosted on-premises, but if your application is hosted partly on-premises and partly in Azure, it may be more efficient to replicate this functionality in Azure. This can reduce the latency caused by sending authentication and local authorization requests from the cloud back to AD DS running on-premises.”

This statement describes how to reduce the latency caused by sending the authentication and local authorization requests from the cloud back to AD DS. If the application AD DS is hosted partly on-premise and partly in Azure, and this functionality is replicated in Azure, the latency can be reduced.

### Logic

1. For statement number 1, we need to use predicate logic. This allows us to create an instance of the variable x where x can be quantified. The use of propositional logic could lead to misunderstanding as it would not allow us to specify the original condition of the variable x, but would be valid in every instance of x. In this instance a Blob can be composed of all instances of x where x is text data or binary data.
2. For statement number 2, we need predicate logic. For all the instances of storage, we need to assert the existence of one instance of storage, such as the storage account contains a container and there exists an instance of b where b is a blob and a container contains a blob. Therefore, we need to quantify the variables in order to solve the expression.
3. For statement number 3, we need to use predicate logic. The level of complexity requires the variables to be declared with quantifiers. For all storage, there exists at least one instance of container, such as storage contains the container, and there exists at least one instance of b such as b is blob, and blob is contained by the container.
4. For statement number 4, we need to use propositional logic. The statement can only be true or false when we apply the logic to the declared variables. These variables do not need to be quantified. This statement requires an exclusive OR () because only one condition can be true, but not more than one.
5. For statement number 5, we need to use propositional logic. The statement can only be true or false when we apply the logic to the cases stated. The properties of each variable are contained within the declaration of said variables. The variables don’t need any quantifiers, as the sole existence of said variable as true or false has an impact on the result of the expression.
6. For statement number 6, we need to use propositional logic. The statement can only be true or false if each of the variables are true or false. It does not require a quantifier as this is contained within the variable itself.
7. For statement number 7, we need to use propositional logic. This statement can be simplified with an if statement. If and only if the functionality can be replicated and the AD DS is partially hosted on-premise and partially off-premise, the latency can be reduced. There is no need to quantify the instances where the variable will be true or false, as the only way to reduce the latency is to satisfy F (functionality can be replicated) and P (partitioned host).

### Conversion

1. T(x) = “x is text data” Bin(x) = “x is binary data”

B(x) = “x is blob”

1. B(x) = “x is block” P(x) = “x is page”
2. Let C (a, b) denote that a contains b.

Let b = blob c = container s = storage account

1. B = bastion access E = access to environment RDP = RDP WinRM = WinRM SSH = SSH
2. S = hybrid solution F = feasibly replicated

I = infostructure P = performance R = redundancy

1. D = default settings Ch = checked Co = confirmed P = custom policy is defined
2. P = partitioned host F = functionality is replicated in Azure L = latency is reduced

### Conclusion

This exercise has shown the importance of using logics in the first steps of software development. Logics allow us to simplify natural language to an atomic form where the variables (for what concerns propositional logic) or the instances of such variables (for predicate logic) considered can only be true or false. This can be compared to writing pseudo code.

The specification document chosen gives an example of how a complex cloud system works. The use of logics to describe such mechanisms can be used to define the minimal requirements for the system to be functional.

However, this type of exercise can be only useful when in an academic environment, as in the work environment other tools would be available and the conversion of statements into logics would be considered redundant. Tools such as TSF, or any platform where software developers share and assign projects, provide a way to dissect the project requirements, and apply pseudo code.

Nevertheless, the use of logics at times lacks a full translation of what the project owner requested, and it also lacks in the development of creative solutions to problems. In these terms the unconditioned use of logics at early stages of the development of a software could be limiting in terms of evaluating possible alternative solutions to problems, at the risk of oversimplifying the requirements.