



Cloud Computing Migration Tool

Jianbo Zheng / Dr. Weichang Du

UNIVERSITY OF NEW BRUNSWICK, FACULTY OF COMPUTER SCIENCE

Introduction

Software as a Service (SaaS) is more and more popular as a new software delivery pattern. For conventional application, the software runs on the purchaser's local computer and is used in an exclusive way. While for SaaS application, the software runs on the service provider's server, and is shared by all the tenants with other server side resources. Due to resources sharing in server side, SaaS applications have advantages like high flexibility, elastic scalability, lower cost and easy management. It is the trend to migrate conventional application to SaaS application.

Compared with a whole new development, reusing the existing software components is a costless and riskless way in the migration.

The objective of our research is to implement a semi-automatic tool to migrate the conventional application to Cloud. We aim to make the transition of conventional application to SaaS application easily and also have a high quality of security, reliability and performance.

Multi-Tenancy

Multi-Tenancy is a reference to the mode of operation of software where multiple independent instances of one or multiple applications operate in a shared environment. The instances (tenants) are logically isolated, but physically integrated. Multi-Tenancy is considered as one of the essential feature of cloud computing.

Migration Tool

From conventional application to SaaS application, as one application instance will be shared by multiple tenants, isolation and access control are the major concern. The following several key aspects need to be considered and created in my migration:

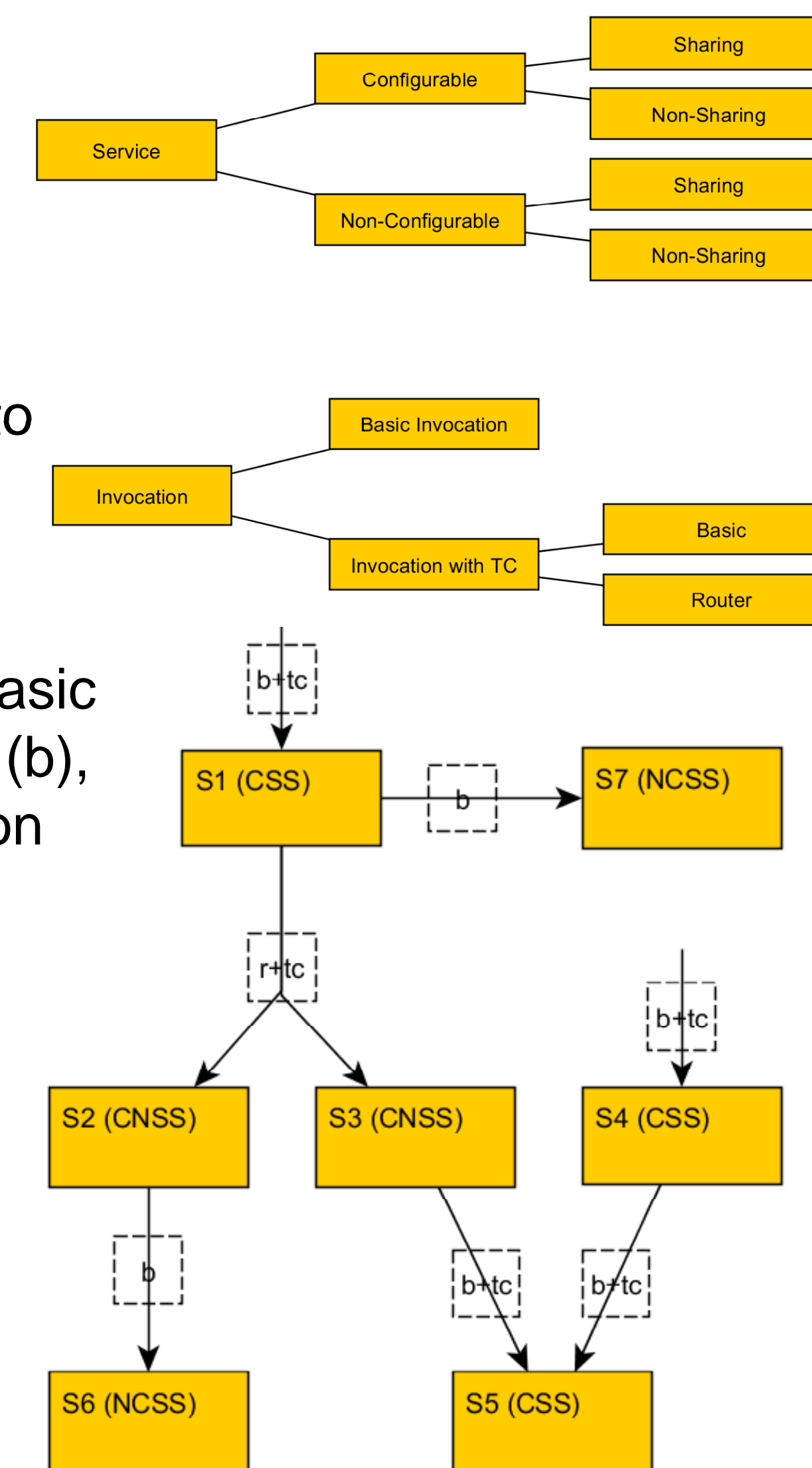
- Tenant management
It is a new subsystem and is used for maintaining the tenant information. And it can be generated by configuration file.
- Identification and authentication
This part takes the access control and combines the tenant verification with the existing user identification.
- Data isolation
In tools, we use separated DB model. By setting up a separated DB for each tenant, we can get the highest level of data isolation and the maximum code reusing.
- Service proxy
Based on the basic types of service and patterns, we will classify the services and invocations and use proxy layer to add tenant context and redirect the service request.

Patterns of services invocation in MTA

There are four basic types of service in MTA: Configurable and Sharing Service (CSS), Configurable and Non-Sharing Service (CNSS), Non-Configurable and Sharing Service (NCSS), and Non-Configurable and Non-Sharing Service (NCNSS).

Here the configurable means the service needs to get the tenant context before it runs, and the sharing means the service will be shared by tenants or not.

After combining the services, we can get three basic patterns in service invocations, Basic Invocation (b), Basic Invocation with TC (b+tc), Router Invocation with TC (r+tc).



		Different patterns of service invocation		
caller	callee	CSS	CNSS	NCSS
CSS	CSS	b+tc	b+tc/r+tc	b
CNSS	CNSS	b+tc	b/b+tc	b
NCSS	NCSS	-	-	b

This is a semi-automatic tool for application developer. Using this tool, application developer can easily convert a conventional application to a multi-tenancy application and deploy it to cloud.

To convert the conventional application, application developer need to create the configuration files from the template configuration file first.

Take the target application and configuration file as input, use the tools to create the gray parts in right diagram and combine them running in a Multi-Tenancy way.

