© All Rights Reserved

HORIZON2020 FRAMEWORK PROGRAMME

TOPIC EUK-03-2016

"Federated Cloud resource brokerage for mobile cloud services"

D4.6

Brokerage Platform Dynamic Structure: Software Prototype

Project acronym: BASMATI

Project full title: *Cloud Brokerage Across Borders for Mobile Users and Applications*

Contract no.: 723131

Workpackage:	4	Dynamic Brokerage and Federation: Software Prototype	
Editor:	Ganis Z. Santoso		ETRI
Author(s):	Ganis Z. Santoso		ETRI
	Jorn Altman		SNU
	Ram Govinda Aryal		SNU
	Emanuele Carlini		CNR
	Vinicius Monteiro de Lira		CNR
Authorized by	Young Woo Jung Konstantinos Tserpes		ETRI ICCS
Doc Ref:	4.6		
Reviewer	K. Tserpes		ICCS
Dissemination Level	Public		

© All Rights Reserved

Document History

Version	Date	Changes	Author/Affiliation
V0.1	28-03-2018	Created ToC	Ganis/ETRI
V0.2	18-04-2018	Add chapter 1, 2, 3, and 4	All partners
V0.3	19-04-2018	Revised subchapter 3.3.1	Emanuele/CNR
V0.4	20-04-2018	Add credential for Git repository access	Ganis/ETRI
V0.5	20-04-2018	Draft Version	All partners
V1	20-04-2018	Final Version	All partners

© All Rights Reserved

BASMATI Glossary

Term/Acronym	Definition
Mobile cloud services	Online services offered by cloud resources to support mobile apps. The backend of the mobile apps.
CP	Cloud Provider. The actor that provides the cloud infrastructure/resources, such as VMs
CSP	Cloud Service Provider. The actor that provides cloud services on top of a rent infrastructure from a CP
Cloudlet	Limited capacity infrastructures with virtualization capabilities, often used to support a limited amount of users or perform a limited set of operations on behalf of the central cloud infrastructure that hosts the complete application
Edge resources	Resources aimed to operate specialized functionality, located at the "edge" of the network infrastructure, thus, closer to the end users. Examples are (clusters of) RaspberryPis or cloudlets
BUDaMaF	BASMATI Unified Data Management Framework
KE	Knowledge Extractor
DM	Decision Maker
RB	Resource Broker
MVD	Mobile Virtual Desktop
DASFEST	An 3-day long music festival taking place in Karlsruhe, Germany every July
ACE	Amenesik Cloud Engine. The cloud service deployment tool through which actual federation is achieved
BEAM	BASMATI Enhanced Application Model. An extension of the TOSCA specification
ASP	Application Service Provider. A Federation user that rents resource services in order to provide an Application services to End-users
Brokering	The matchmaking support provided by BASMATI platform to decide about the best cloud resources to exploit for the execution of the back-end of BASMATI applications. This activity regards the placement of the services or data on computational resources and storages belonging to the cloud data centre and the cloudlets within the federation.
End user	A user who benefits the various application and infrastructure services provided by the Cloud. Within BASMATI, the most typical example is exploiting the Cloud federation via a mobile device (possibly a laptop) using specialized apps or a web browser.
Offloading	The ability of BASMATI platform supporting the runtime placement of the components composing the front-end of BASMATI applications on edge resources available nearby the end user. This activity takes place both when edge and mobiles exchange one each other their own workload or when such devices transfer some workload to the clouds or cloudlets. In BASMATI we often distinguish Front-end offloading, related to the mobile part of application, from Back-end offloading, concerning the server side of applications. The latter roughly translates to the known concept of Cloudbursting.
QoE	Quality of experience. It is a measure of a customer's experiences with a

© All Rights Reserved

	service. It may be related to some aspects of the QoS and QoP, but can also take into account other metrics.
Service handover	Service handover refers to the activity of transferring an active service between two computational resources (e.g. Cloudlets) with minimal or no disruption on the availability of the service. Ideally, service handover is transparent with respect to the user.
Situational Awareness	The ability of the BASMATI platform to recognise the “situation” characterising the actual combined status of users, applications and resources, aimed at achieving an effective and efficient management of applications and resources.

© All Rights Reserved

Executive Summary

The purpose of this report is to demonstrate the implemented algorithms in the components of brokerage platform. The algorithms are capable of producing outputs which are related to current usage and situation in the federations.

© All Rights Reserved

Table of Contents

© All Rights Reserved

1 Introduction

As a smart platform, BASMATI should be able to alter its behaviors based on the current status and the projected situations. In order to achieve that we are developing and implementing algorithms in our brokerage platform.

The demonstration should be able to meet the objectives of the brokerage platform in terms of scalability, reliability, and performance. Scalability assures that the decision maker and the resource broker still run with rising velocity (i.e., a linear deterioration of performance) if the number of resources and the number of factors raises. In a real world scenario, we confront daily appearing failures. In many applications, an error produces a re-start of the system or starting a system clone on other resources.

From the viewpoint of the decision maker and the resource broker, we examine specific errors that could happen in the context of a distributed service-oriented architecture. The performance of the decision maker and the resource broker depend on the methods being applied for making the resource selection and the data studied. Additionally, as methods from artificial intelligence are implemented, a compensation between the accuracy of the results and the duration of the calculation of the results can be studied.

1.1 Relationship to Other Deliverables

This deliverable presents the demonstration of software that is defined in the deliverable D4.5. D4.6 is also related to D4.4 report as the prototype in D4.4 is going to complement the prototype in this report.

The result of this report is going to be used to deploy resources. In that regard, the deployment plan from this prototype is valuables for deliverable D4.3 and D5.4.

1.2 Scope of the prototype

The prototype wasn't intended to be a complete demonstration of what BASMATI is capable of. However, the demonstration will show that BASMATI's brokerage platform's capability in creating a deployment plan which is based on the current status and usage in the federation. The processes after the creation of deployment plan is not going to be demonstrated here.

The components which are included in this demonstration are the **Decision Maker** and the **Resource Broker**. Both components will be explained in details on how to install and run these components. During the demonstration, other components such as the Application Repository and the SLA Manager can be seen. But the explanation on how to install those components are not going to be discussed here but in other deliverable reports.

© All Rights Reserved

2 Prototype Demonstration

The prototype is demonstrated through a video. The purpose of presenting the demonstration in a video format is to provide an accessible way to show the BASMATI project to readers.

2.1 Location

The video of the demonstration for this report was uploaded on Youtube. The URL for the video is https://www.youtube.com/watch?v=jN74yJOnh_U. It's highly intentional to upload the video to Youtube to provide better dissemination about BASMATI project to public since Youtube is a popular platform for videos and movies.

2.2 Description

As mentioned, the video is demonstrating the dynamicity of deployment plan as an output of brokerage platform. Since the User Interface (UI) is not integrated yet to the platform, we provide the demonstration with API calls in the console/terminal.

The console in the video is divided into four sections. In the left and top section, it's a console for the caller of the brokerage platform. In the top and right section, it's a console for decision maker. In the left and bottom section, it's a console for resource broker. The last section is the console for us to generate an SLA violation manually. We simulate the occurrence of a violation manually because it's a simulated one not real based on actual situation. The violation detection is not in the scope of this project.

In the video, the BASMATIZE action is generating a deployment plan. This deployment plan is based on Application Topology inside AR. Initially there's no violation detected therefore the plan is recommending two regions. At the latter part, there's a simulated SLA violation to indicate there's a region down. After this violation, the deployment plan only shows the region which is not down as a demonstration in dynamicity of the brokerage platform.

3 Prototype Installation Guide

To run the prototype, the source codes of the components are necessary. In this section we are going to provides instructions on how to get the code, install, and run each component in the brokerage platform.

3.1 Authentications

The source codes of the components are located in a private Git repository which can be accessed from anywhere in the world. However, to access it you need a credential which is:

username: guest

password: basmati_review7012

© All Rights Reserved

The GitLab repository url is:

git@basmati.amenesik.com/code/platform/modules/resource_broker.git

3.2 Requirements

Due to the availability of Docker[1], it's relatively easy to deploy the components since the required programs are automatically installed with our dockerfile and docker-compose. In theory it's possible to run the brokerage platform in Linux Operating System but we have only tried it in Ubuntu[2] 16. Another requirement is to have Git version control program as it is a program which will download the whole source code. So the requirements are:

1. Linux OS
2. Docker and Docker Compose
3. Git

These are the minimum requirements to deploy locally. The deployment to other platforms is not in the scope of this report.

3.3 Installation and Deployment Guide

3.3.1 Decision Maker

The installation and deployment guide for decision maker can be found in D4.4 report.

3.3.2 Resource Broker

These are the required steps to install and deploy resource broker:

```
# git clone http://guest@basmati.amenesik.com/
code/platform/modules/resource_broker.git ~/resource_broker

# cd ~/resource_broker

# docker-compose build

# docker-compose up -d

# ddocker-compose exec web ./manage.py migrate

# docker-compose exec web ./manage.py createsuperuser
```

Depends on your settings, there might be changes that you need to do at these files:

- web/config/settings.py
- nginx/sites-enabled/django_project

© All Rights Reserved

4 Conclusions

In this deliverable we have presented a stable version of the working prototype of the brokerage platform in terms of the generation of dynamic structure of deployment plan. The plan is adapted to the current situation of the federation and applications. The interface to and outside of the platform has been following the standard so it's easy for other components to integrate. In future work, the full integration to other components such as Application Controller, Cloud Provider Management, and Federation Monitoring in BASMATI platform should be completed.

5 References

[1] Merkel, Dirk. "Docker: lightweight linux containers for consistent development and deployment." Linux Journal 2014.239 (2014): 2.

[2] Raggi, Emilio, Keir Thomas, and Sander Van Vugt. Beginning Ubuntu Linux. Apress, 2010.