HORIZON2020 FRAMEWORK PROGRAMME

TOPIC EUK-03-2016

"Federated Cloud Resource Brokerage for Mobile Cloud Services"

D4.4
Multimodal Brokerage and Offloading: Software Prototype

Project acronym: BASMATI

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

Contract no.: 723131

Workpackage:	4 Multimodal Brokerage and Offloading: Software Prototype	
Editor:	Jörn ALTMANN	SNU
Author(s):	Ram GOVINDA ARYAL	SNU
	Emanuele CARLINI	CNR
Authorized by	Young Woo Jung K. Tserpes	ETRI ICCS/NTUA
Doc Ref:	D4.4	
Reviewer	K. Tserpes	ICCS/NTUA
Dissemination Level	Public	

BASMATI Glossary

Term/Acronym	Definition
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.
BUDaMaF	BASMATI Unified Data Management Framework
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
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
DASFEST	An 3-day long music festival taking place in Karlsruhe, Germany every July
Decision making	The decision making supports the runtime placement of the components of applications on resources, considering application requirements, user requirements, and the set of available resources identified.
DM	Decision Maker
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
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.
KE	Knowledge Extractor
Mobile cloud services	Online services offered by cloud resources to support mobile apps. The backend of the mobile apps.
Multi-Objective Optimization	It requires an algorithm that can handle more than one objective function. The objective function can consider different economic values (e.g., time cost, energy cost, and monetary cost) besides different technical values (e.g., execution time, delay). It is used to find an optimal solution.

MVD	Mobile Virtual Desktop
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 service. It may be related to some aspects of the QoS and QoP, but can also take into account other metrics.
RB	Resource Broker
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.

Executive Summary

This is an accompanying report to the demonstrator of the decision maker software. It informs how to execute the decision maker. The technical details such as the architecture, logic, interfaces and relations to other components for the decision maker are presented in the BASMATI deliverable D4.3 “Multimodal Brokerage and Offloading: Design and Specification”.

Table of Contents

1 Introduction

1.1 Purpose of this Deliverable

This deliverable presents the software of the decision maker, which has been specified in deliverable D4.3 “Multimodal Brokerage and Offloading: Design and Specification”. In particular, the software comprises the implementation of the:

1) Algorithm aimed at identifying the optimal set of resources to assign to the components of the application, using
2) Techniques for multi-objective, optimal service placement that makes adaptive offloading decisions at runtime using techno-economic information.

1.2 Relationship to other Deliverables

As this deliverable, D4.4, provides the software prototype of the decision maker for deciding on the optimal placement of application components onto federated clouds, it follows the specification given in deliverable D4.3.

This deliverable, D4.4, is also related to the deliverable D4.5 and D4.6, as the decision maker (deliverable D4.3) uses the resources proposed by the resource broker (design information is available in deliverable D4.5 and implementation information is given in deliverable D4.6).

2 Implementation of the Decision Maker

2.1 Requirements

The Decision Maker has been developed and tested under Linux Ubuntu 16.04. However, other Unix-like operation systems, as well as Windows OSs, can be suitable to execute the component.

The Decision Maker is a web service that exposes several rest APIs and responds to request accordingly.

The Decision Maker is written in python, and therefore requires the following python libraries to run:

- flask==0.10
- flask-restful==0.2.12
- flask-cors==3.0.2
- requests==2.13.0
- numpy>=1.13.1

The source code of the Decision Maker can be downloaded from the Basmati Git Lab repository, located at the following URI:

```
basmati.amenesik.com/code/platform/modules/decision_maker.git
```

2.2 Installation

The Decision Maker and the Resource Broker work together to fulfill the Basmati requirement of identifying the best fitting resources to an application. Although the Decision Maker and the Resource Broker complement each other, the Decision Maker can also run as a stand-alone component.

2.2.1 Decision Maker

To execute the Decision Maker, a python process needs to be launched, which can be performed in the following way:

```
> python src/server_api.py
```

The Decision Maker will start and be listening at the address: http://[localhost]:8034/. It is possible to change the network port by modifying the file server_api.py, where the port is indicated as a variable.

2.2.2 Resource Broker

The installation and deployment instructions for the Resource Broker can be found in deliverable D4.6.

If the Resource Broker should not be considered by the Decision Maker, it can be opt out through a variable in server_api.py.

3 Conclusion

This deliverable gives a brief overview about how to execute the Decision Maker, as a stand-alone component or together with the Resource Broker. Further details can be found in deliverable D4.3, deliverable D4.5, and deliverable D4.6.