

HORIZON2020 FRAMEWORK PROGRAMME

TOPIC EUK-03-2016

“Federated Cloud resource brokerage for mobile cloud services”

European
CommissionHorizon 2020
European Union funding
for Research & Innovation

D3.2

BASMATI Analysis and Modelling of Users and Applications: Software Prototype

Project acronym: BASMATI

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

Contract no.: 723131

Workpackage:	3	Users and Applications Modelling and Analysis
Editor:	Ioannis Violos	ICCS/NTUA
Author(s):	Ioannis Violos	ICCS/NTUA
	Konstantinos Tserpes	ICCS/NTUA
	Vinicius Monteiro de Lira	CNR
	Ganis Zulfa Santoso	ETRI
Authorized by	T. Varvarigou	ICCS
	Y. W. Jung	ETRI
Doc Ref:	3.2	
Reviewer	T. Varvarigou	ICCS
Dissemination Level	Public	

Document History

Version	Date	Changes	Author/Affiliation
v.1.0	18-4-2018	Review version ready	Ioannis Violos ICCS
v.1.0_KT	24/04/2018	Commented version	K. Tserpes/ ICCS

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 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.

Executive Summary

This deliverable is the accompanying report of the main demonstrator deliverable for the Knowledge Extractor (KE) component. The role of the KE is to provide predictions concerning the resource utilization and the user mobility. The Knowledge Extractor (KE) component is implemented as a service on an Openstack inspired cloud infrastructure, called Okeanos. KE can predict the number of users that will be gathered close to each Point of Interest (PoI) and the VM resources that each hosted application demands in the subsequent time intervals.



Table of Contents

Executive Summary	5
1. Introduction	1
1.1. Relationship to Other Deliverables	1
1.2. Scope of the KE component.....	1
2. Knowledge Extractor Demonstration	2
2.1. KE Location	2
2.2 KE - Sequence Diagram.....	3
2.3 KE Interface	4
2.4 KE Video Description	6
3. Prototype Installation Guide	6
3.1. Download code from GitLab	7
3.2. Import code and export war	7
4. Conclusions	7

1. Introduction

This deliverable demonstrates the KE functionality that implements the models and techniques that cover the tasks T3.1 User Mobility Understanding and Modelling, T3.2 Application Analysis and Modelling, and T3.3 Situational Knowledge Acquisition and Cloud-across Service Monitoring. The description of Machine Learning methodology that we followed is provided in the deliverable D3.1 Analysis and Modelling of Users and Applications: Design and Specification and and the section 2 of the deliverable D3.3 Situational Knowledge Acquisition and Inter-Cloud Service Monitoring

1.1. Relationship to Other Deliverables

This deliverable is largely linked with:

- D3.1 Analysis and Modelling of Users and Applications: Design and Specification, which presents the architecture and interfaces of the component that implements the KE
- D3.3 Situational Knowledge Acquisition and Inter-Cloud Service Monitoring, that refers to a linked component that related to infrastructure monitoring.

1.2. Scope of the KE component

This component is comprised of the following functionality:

- Knowledge base update - Application resources
- Knowledge base update - Users positions
- Provide - current status of Users positions
- Provide - current status of Application resources usage
- Predict - the users distribution close to POIs
- Predict - the resources demands of Applications as deployed to VMs
- Validation of a Deployment Plan

2. Knowledge Extractor Demonstration

2.1. KE Location

The KE component software, a video demonstrating its use and the service endpoint are located respectively at:

YouTube URL: <https://youtu.be/h3g88qe5YHk>

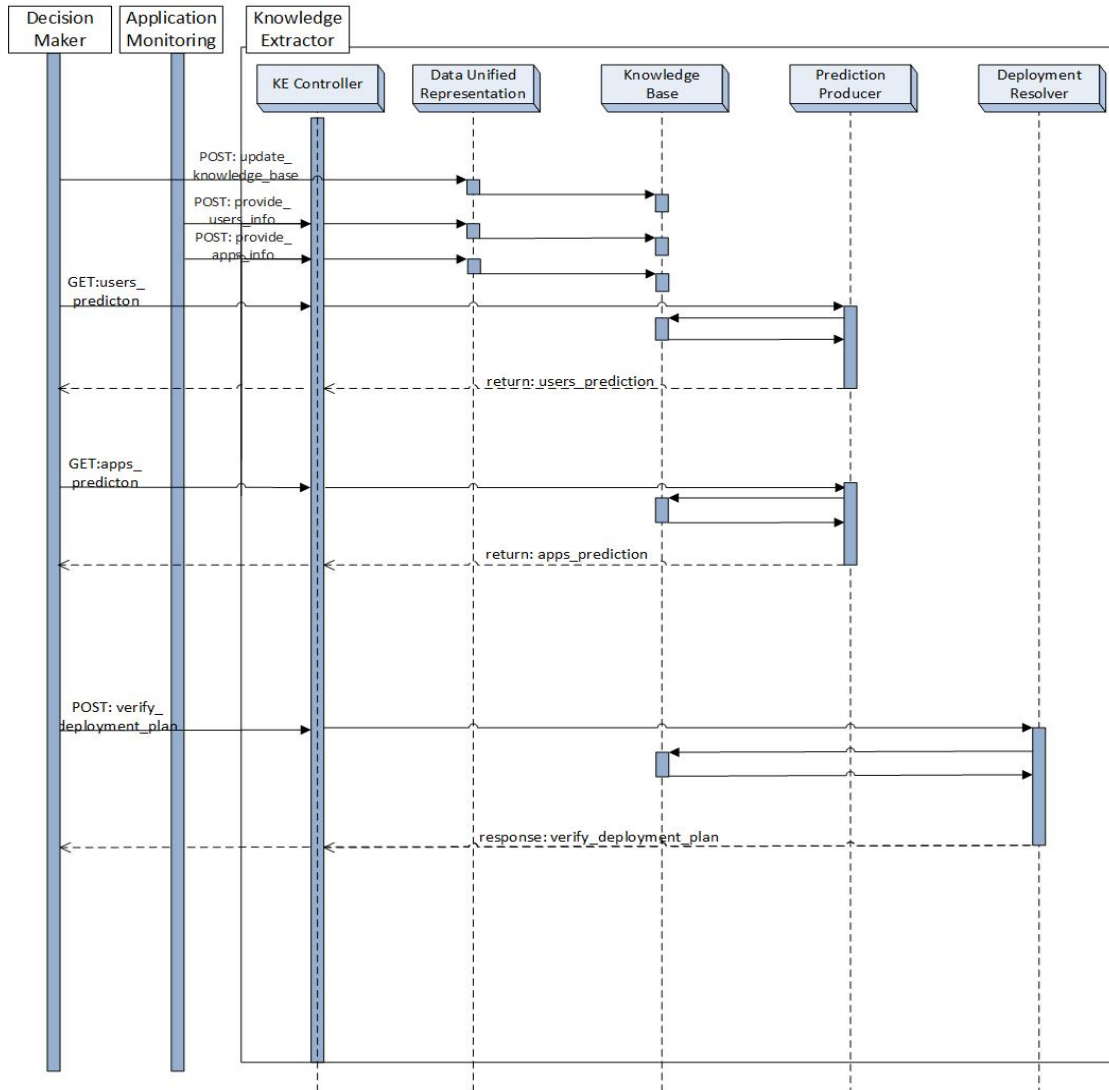
GitLab URL: http://basmati.amenesik.com/code/platform/modules/knowledge_extractor
http://basmati.amenesik.com/code/platform/modules/knowledge_extractor

Okeanos Cloud URL : <http://snf-754710.vm.okeanos.grnet.gr:8080> (Credentials needed)

Related deliverables URL: <https://www.dropbox.com/sh/65cpn4tdl2k2a7y/AADfyz-zxCfKJfYGb-GPxdiZa?dl=0> <https://www.dropbox.com/sh/65cpn4tdl2k2a7y/AADfyz-zxCfKJfYGb-GPxdiZa?dl=0>

2.2 KE - Sequence Diagram

In what follows we present the sequence diagram of the component.



2.3 KE Interface

Name	Resource	Type	Connection from	Description
Knowledge Base Update	http://snf-754710.vm.okeanos.grnet.gr:8080/knowledge_extractor2/rest/update_knowledge_base	POST Consumes(TEXT_PLAIN)	Application Monitoring	Interface between KE and AM that provides to the KE the training datasets that will be stored in the knowledge base.
Provide Users Positions & infrastructure info	http://snf-754710.vm.okeanos.grnet.gr:8080/knowledge_extractor2/rest/provide_users_info	POST Consumes(JSON)	Application Monitoring	Interface between KE and AM that provides to the KE the upcoming context users data.
Provide Application Resources info	http://snf-754710.vm.okeanos.grnet.gr:8080/knowledge_extractor2/rest/provide_apps_info	POST Consumes(JSON)	Application Monitoring	Interface between KE and AM that provides to the KE the upcoming deployed applications meta-data.

User Mobility Prediction	http://snf-754710.vm.okeanos.grnet.gr:8080/knowledge_extract_or2/rest/users_prediction	<p>GET Produces(JSON)</p>	<p>Decision Maker</p>	<p>A request from DM to KE to predict the mobility of the users.</p>
Application Resources Prediction	http://snf-754710.vm.okeanos.grnet.gr:8080/knowledge_extract_or2/rest/apps_prediction	<p>GET Produces(JSON)</p>	<p>Decision Maker</p>	<p>A request from DM to KE to predict the resource needs of the applications.</p>
Evaluation Deployment Document Request	http://snf-754710.vm.okeanos.grnet.gr:8080/knowledge_extract_or2/rest/verify_deployment_plan	<p>POST Consumes(JSON)</p>	<p>Decision Maker</p>	<p>Interface between KE and DM to evaluate a deployment plan.</p>

2.4 KE Video Description

The Youtube video provides information about the following aspects:

1. KE Eclipse Project
2. GitLab where the code hosted
3. Deployed on Okeanos to run as a service constantly 24/7/365
4. Postman' s Collection calls with headers - body. KE component is a back end component that consumes and provides JSON files. Its functionality demonstrated using an executable description of APIs.
5. we will present the functionality of KE using the Postman that will call the RESTfull operations.
 - a. Update Knowledge Base for supervised ML techniques
 - b. Provide users info
 - c. Provide application info
 - d. Predictions on applications
 - e. Prediction on users
 - f. Deployment resolver

It is important to mention that this demonstration covers the case in which the KE service is called for first time. In the first KE call, the following are taking place:

- The construction of supervised machine learning models
- The prediction of users and application based on the provided current status of them.

The first call needs more time for execution cause of the construction of supervised machine learning models. All the next calls have real time response.

3. Prototype Installation Guide

We also provide the source code and installation instruction for readers to try the software.

3.1. Download code from GitLab

To clone the source code: `git clone <URL>`.

To synchronize the source code: `git pull origin`

You should be in the right file paths and add an SSH key to GitLab account.

where URL: `git@basmati.amenesik.com:code/platform/modules/knowledge_extractor.git`

3.2. Import code and export war

import from eclipse the project as cloned from GitLab and export a war file. The war file can be deployed and run in any cloud provider that has Apache Tomcat installed.

4. Conclusions

We have designed and developed KE to provide predictions of the distribution of users gathered close to pois and the application demands that run on cloud VMs. The prediction techniques that developed are based on state of the art machine learning methods that are adapted in accordance with the Basmati needs. This report explains firstly how to download the source code and deploy the KE service to a VM and demonstrates the KE functionalities in combination with the corresponding uploaded video on youtube.