Melodic
Big data cloud

Multi-cloud Execution-ware for Large-scale Optimised Data-Intensive Computing

H2020-ICT-2016-2017
Leadership in Enabling and Industrial Technologies; Information and Communication Technologies

Grant Agreement Number
731664

Duration
1 December 2016 -
30 November 2019

www.melodic.cloud

Deliverable reference
D4.2

Date
30 November 2019

Responsible partner
7bulls.com

Editor(s)
Łukasz Szymański

Author(s)
Łukasz Szymański, Paweł Skrzypek, Daniel Baur

Reviewers
Sebastian Schork, Ernst Gunnar Gran

Approved by
Ernst Gunnar Gran

ISBN number
N/A

Document URL:
https://www.melodic.cloud/deliverables/D4.2 Provider Agnostic Interface Mapper.pdf

Title

Provider agnostic interface mapper

Executive summary

This deliverable presents the additional development and activities following the work presented in deliverable D4.1 “Provider agnostic interface definition & mapping cycle” [3] for the Executionware component of Melodic. In particular, the pricing model for the node candidate feature is described, and changes related to fixing node candidate timeout issues are presented. Also, additional activities related to improving the maintenance and serviceability of the Executionware has been described.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 731664

Document	
Period Covered	M16-24
Deliverable No.	D4.2
Deliverable Title	Provider agnostic interface mapper
Editor(s)	Łukasz Szymański
Author(s)	Łukasz Szymański, Paweł Skrzypek, Daniel Baur
Reviewer(s)	Sebastian Schork, Ernst Gunnar Gran
Work Package No.	4
Work Package Title	Executionware
Lead Beneficiary	7bulls.com
Distribution	PU
Version	1.0
Draft/Final	Final
Total No. of Pages	8

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731664

Table of Contents

List of Figures

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731664

1 Introduction

The purpose of this deliverable is to describe additional tasks, features and activities related to the Executionware component of the Melodic project, i.e. the ones not covered in deliverable D4.1 "Provider agnostic interface definition & mapping cycle" [3]. As seen in the overview of the Melodic architecture in Figure 1, the Executionware fulfils four main tasks: (a) it provides a cloud-agnostic interface to access features of multiple cloud providers in an unified way; (b) it delivers resource management capable of allocating and managing resources from these providers; (c) it supplies a data processing layer, on top of resource management, able to execute the user's defined processing tasks; and finally, (d) it provides monitoring services gathering runtime information of the managed resources and deployed tasks.

Within the context of the Melodic project, the Executionware has two major points of interaction: (a) the Adapter component of the Upperware and (b) the API offered by the cloud providers. For the interaction with the Upperware, the Executionware provides a RESTful API giving the Upperware access to its resource management capabilities and monitoring services. On the cloud provider side, the Executionware implements a provider agnostic interface that is then mapped to the data format of the respective cloud provider allowing the Executionware to allocate and manage resources across multiple providers.

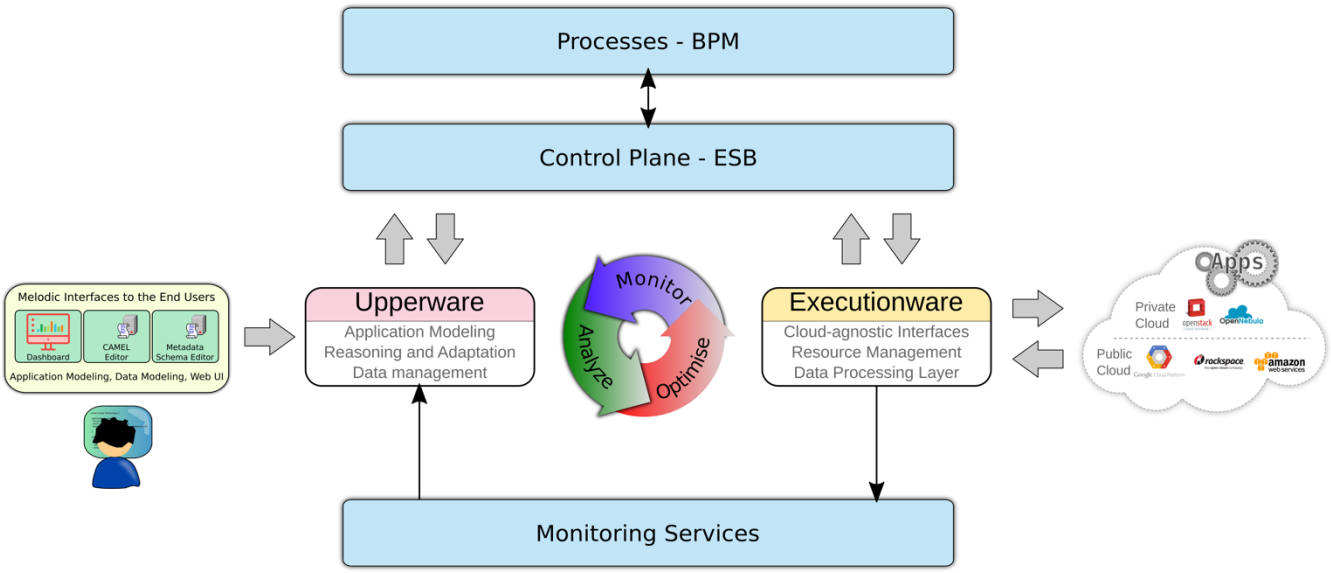

Figure 1: *Melodic Architecture*

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731664

2 Scope of the document

This document is intended for the general audience interested in what features were added to the Executionware by the Melodic project, in addition to the features described in D4.1 "Provider agnostic interface definition & mapping cycle" [3]. The work of this deliverable depends on the D4.1 "Provider agnostic interface definition & mapping cycle", D2.1 "System specification document" [2], and D2.2 "Architecture and Initial Feature Definitions" [1] deliverables. The related work for this deliverable is the same as the related work presented in deliverable D4.1 "Provider agnostic interface definition & mapping cycle".

3 Structure of the document

The rest of this document is structured as follows: Section 4 presents the new features added to the Executionware, after the features described in deliverable D4.1 "Provider agnostic interface definition & mapping cycle" [3]. Section 5 presents an issue discovered after the implementation of the changes described in D4.1 "Provider agnostic interface definition & mapping cycle" [3], and the method for solving the issue, while Section 6 describes the activities related to increasing the quality and maintenance level of the Executionware. Finally, Section 7 shortly summarizes and concludes the current status of the Executionware interfaces.

4 Additional features

The initial set of features of the Executionware implemented to support Melodic as described in D2.1 "System specification document" [2] and D2.2 "Architecture and initial feature definitions" [1], was presented in deliverable D4.1 "Provider agnostic interface definition & mapping cycle" [3]. This section presents additional features covering the pricing model, price fetching and applications to node candidates mapping and reasoning.

4.1 Pricing model

The cost discovery mechanism is based on "MultiCloudPricingService", Cloudiator's internal service class, which, as its name implies, is specially designed to collect price lists from multiple Cloud Service Providers (CSPs). This service uses a pricing supplier factory that creates specialized suppliers for each CSP that is being offered. Every pricing supplier needs to follow a specified interface to hook up into the discovery mechanism, which then provides the discovered price lists to the database. From there, pricing information can be used in other platform components through a pricing domain repository (e.g. attaching prices to node candidates). Details of the initial pricing model's implementation are described in deliverable D4.4 "Resource Management Framework" [4]. The pricing data model has been extended to cover the different pricing models of the leading CSPs Azure and GCP in addition to the already covered pricing

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731664

model of AWS. The current entity relationship diagram for that pricing model is presented on Figure 2. Additional attributes needed to properly handle pricing models for different CSPs have been added. The initial tests of that model show that it fully covers the pricing model for selected resource types.

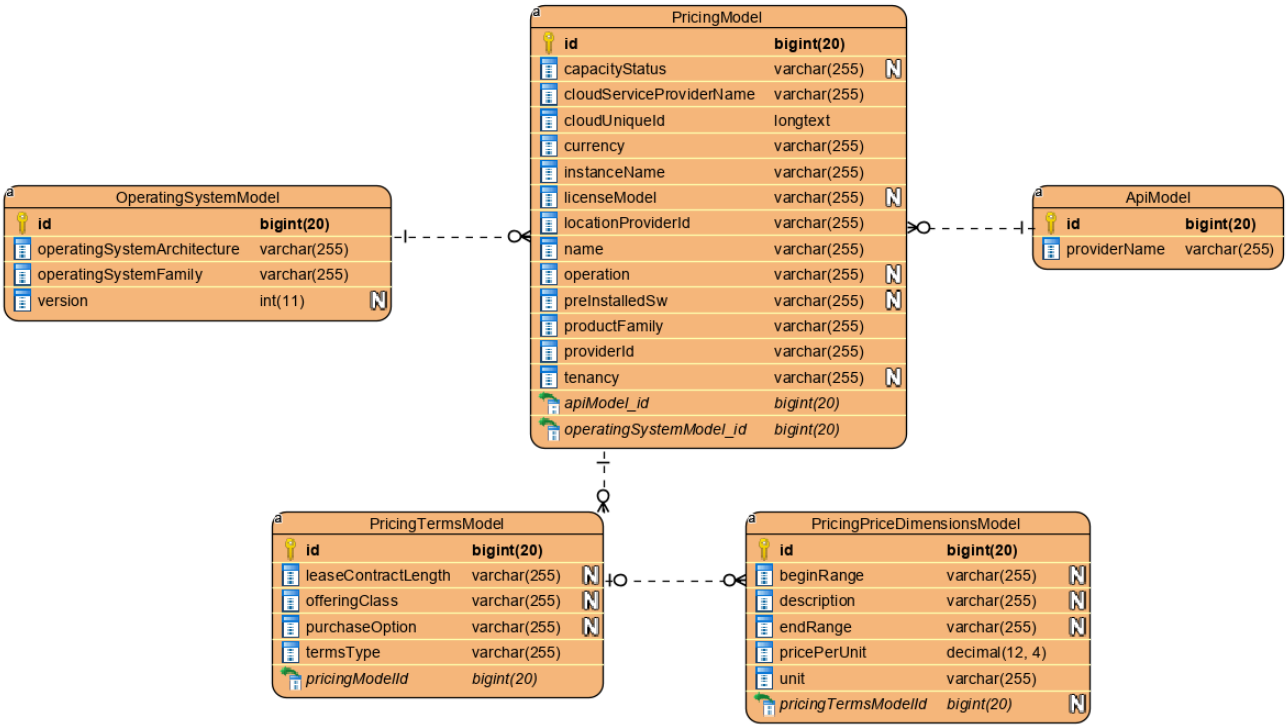

Figure 2: *Pricing Data Model*

4.2 Node candidates for grouping

The node candidates feature fetches a list of flavours of virtual machines from cloud computing providers. The initial version of MELODIC could deploy only one component per virtual machine. The purpose of the grouping feature of the MELODIC platform is to be able to deploy multiple components per one virtual machine. The Executionware node candidates feature has been verified and technically evaluated against the grouping feature. The tests of deployment of multiple components per virtual machine have also been successfully conducted. As a result, it has been proven that the deployment structure of Executionware presented in deliverable D4.1 “Provider agnostic interface definition & mapping cycle” [3] fully covers the requirements of the grouping feature.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 731664

5 Issues

In this section we present and issue discovered during the tests of release 2.5 of the Melodic platform related to the Executionware. The way of solving this issue is described in the following section.

5.1 Timeout fetching node candidates

During the tests of release 2.5 of the Melodic platform, an error was encountered related to a timeout during the fetching of node candidates (VM flavours). This timeout, connected to the MemCache[1] component of the Melodic platform, expired when the number of the node candidates reaches a significant number, like above 10 000 candidates.

A solution has been implemented by splitting fetched node candidates into different groups based on the size of the VM flavour, and then fetching one group at a time. As the cache is loaded infrequently, the time of loading and updating of the cache is not a critical issue. This solution will be further tested with release 3.0 of the Melodic platform.

6 Maintenance and development workflow

This section of the document describes maintenance and development workflow of the interface mapper.

6.1 Maintenance and serviceability

The Executionware is responsible for the communication with the Cloud Computing providers and deployed components. Due to that, proper error handling, retrying and logging is very important to increase maintainability and serviceability of the solution. These requirements are based on knowledge and experiences with developing high-quality IT systems for enterprises and SMEs. Thus, the Executionware has been extended with additional capabilities related to maintenance and serviceability. Proper error handling has been introduced. It allows to catch errors during multicloud communication and properly handle them. Also, a retry feature repeating a given operation has been introduced. Additionally, errors are logged using the same informative template. Altogether, the following requirements related to serviceability and maintainability have been implemented:

1. Error handling for external cloud computing operations.
2. Error logging in a unified way.
3. Retry mechanism for selected operations.

The above requirements allow to increase the TRL of the Executionware to level 6.

[1] https://www.php.net/manual/en/book.memcache.php

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731664

7 Current status of Interfaces and Conclusion

The interfaces exposed by the Executionware to other modules of Melodic (mainly the Upperware) have not been changed. The design prepared for Release 2.0 of Melodic as depicted in deliverable D4.1 "Provider agnostic interface definition & mapping cycle" [3] has proven to be flexible and generic enough to fulfil the requirements for deploying applications in an optimized way. Also, for newly added features such as the pricing fetching and grouping, the already designed interface and object structures has proved to be enough to properly handle the corresponding operations mechanisms.

All in all, in this deliverable we have presented the new features related to the pricing model and the node candidates grouping feature. Furthermore, we have presented a discovered issue with respect to a MemCache timeout and Executionware, and how the issue was solved. Additionally, we have presented some additional tasks and activities, which have been done to increase the stability and serviceability of the Executionware. The goal was to increase the TRL of the module to level 6.

8 Bibliography

[1] Yiannis Verginadis *et al.*, "D2.2 Architecture and initial feature definitions", the Melodic H2020 Project Deliverable D2.2, Feb. 2018.
[2] Yiannis Verginadis *et al.*, "D2.1 System specification document", the Melodic H2020 Project Deliverable D2.1, Jun. 2017.
[3] Daniel Baur *et al.*, "D4.1 Provider agnostic interface definition & mapping cycle", the Melodic H2020 Project Deliverable D4.1, Jun. 2018.
[4] Daniel Baur *et al.*, "D4.4 Resource Management Framework", the Melodic H2020 Project Deliverable D4.4, Nov. 2019

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731664