|  |
| --- |
| AMENESIK |
| BASMATI FEDERATION |
| Business Federation Architecture |
|  |
| **Jamie Marshall** |
| **5/18/2017** |

|  |
| --- |
| This document describes the architecture and operation of the components of the Basmati Business Federation Logic of the Amenesik Cloud Engine |

Table of Contents

[Introduction 2](#_Toc482906640)

[Amenesik Cloud Engine Components and Architecture 2](#_Toc482906641)

[Provider Service 3](#_Toc482906642)

[Customer Service Composition 3](#_Toc482906643)

[Over Booking 4](#_Toc482906644)

[Federation Relationships 4](#_Toc482906645)

[Federation Placement 5](#_Toc482906646)

[Cloud Federation Scenario 6](#_Toc482906647)

[Cloud Federation Architecture 7](#_Toc482906648)

[Customer Invoicing 7](#_Toc482906649)

[Cloud Federation Cost and Revenue Sharing 8](#_Toc482906650)

[Basmati Business Federation Demo 8](#_Toc482906651)

[Introduction 8](#_Toc482906652)

[Workflow 9](#_Toc482906653)

[Platform Design 9](#_Toc482906654)

[Federation Member 1 9](#_Toc482906655)

[Federation Member 2 9](#_Toc482906656)

[Federation Member 3 9](#_Toc482906657)

[Platform Production 9](#_Toc482906658)

[Login Accounts 10](#_Toc482906659)

[Subscription Accounts 10](#_Toc482906660)

[Federation Member 1 10](#_Toc482906661)

[Federation Member 2 10](#_Toc482906662)

[Federation Member 3 10](#_Toc482906663)

[Federation Relationships 10](#_Toc482906664)

[Federation Member 1 10](#_Toc482906665)

[Federation Member 2 11](#_Toc482906666)

[Federation Member 3 11](#_Toc482906667)

[Application Descriptions 11](#_Toc482906668)

[Application Deployment 11](#_Toc482906669)

[Conclusions 12](#_Toc482906670)

# Introduction

This document describes the architecture and components of the Basmati Business Federation Component extension to the Amenesik Cloud Engine. This component allows ACE platforms, each managed by independent commercial operators, to be inter-connected for business collaboration in the form of an automated cloud federation allowing resource and revenue sharing between the federation members.

Flexibility of operation is the overall guiding principle. There is no central federating entity. Each commercial platform operator may participate in commercial relationships with other partner members of the federation, as required to satisfy clearly identified requirements of their business.

# Amenesik Cloud Engine Components and Architecture

Before describing the details of the federation mechanisms, it is first necessary to explain the details of service management and composition within the Amenesik Cloud Engine.

SLA MANAGER

Resource and Service Description

Service Life Cycle Management

Resource Deployment

Service and Resource Placement

Multi Cloud Abstraction Layer

Resource Monitoring

The preceding diagram shows the major architectural components of the Amenesik Cloud Engine software that is to be used by each of the cloud federation members.

* The **resource and service description** component is responsible for the management of technical descriptions of applications and their operational configurations.
* The **SLA manager** is responsible for the management of service composition descriptions and the details of life cycle requirements.
* The **service and resource placement** plays the primary role in customer service composition from the corresponding customer service level agreements.
* The **central service life cycle management** component is the main platform orchestrator responsible for the animation and management of customer’s service instances.
* The **resource deployment** component contains the generic provisioning layer that dispatches placement and provisioning requests to the underlying cloud provider interfaces.
* The **multi-cloud abstraction layer** encapsulates the collection of cloud provisioning interfaces allowing deployment on all major public and cloud providers.
* The **resource monitoring** component provides monitoring feed-back from deployed resources to the service life cycle management component allowing real time management of scalability and fail over conditions as they occur.

This provides a cloud abstraction layer encapsulating the collection of provisioning characteristics of a member of a federation.

# Provider Service

When an Amenesik Cloud Engine is first started, and brought online, the cloud provider management component will load the resource quota descriptions from the provider facing service level agreements of each of its cloud provider interfaces.

The service level agreement associated with each cloud provider interface will describe an offer of resources or services in terms of their available quantity, granularity, geo-localization and price.

Each cloud provider interface has an associated account allowing cost management of resources provisioned through the corresponding cloud provider interface.

Different cloud provider types may describe their offer of resources and service in different ways.

Infrastructure as a Service (IaaS) provider interfaces describe their offer of service in terms of virtual machines, compute cores, RAM, storage size and network addresses.

Platform as a Service provider interfaces may describe their offer of service in similar terms, or, as applications, environments and storage and network facilities.

# Customer Service Composition

Cloud and Application services are offered by Amenesik Cloud Engine Operators to their Customers.

Customer facing service level agreements are used to describe offers of service by platform operators to their Customers.

Each customer is required to have an account with their platform operator to be able to gain access to the platform and be an initiating party in service level agreements describing their required provision of service by the operator.

These agreements comprise technical descriptions, commercial conditions governing their deployment and the operational conditions or guarantees that are to be respected by both parties during the resulting service life cycle.

Customer service level agreements are processed by the Amenesik Cloud Engine to create an instance of service to be made available to that customer. During this operation, known as service composition or instantiation, the cloud resources, required to meet the needs of service delivery, are identified, located and reserved through the placement engine of the Amenesik Cloud Engine.

The placement engine is responsible for the selection of suitable cloud resources from the amongst the resource quota descriptions governed by the service level agreements of the various cloud provider interfaces of the ACE Operator.

Resources that are selected in this way, during the placement operation, are marked as reserved, consequently reducing the resulting quantity available through the corresponding provider interface.

When all the resources required have been selected and reserved the resulting service instance is deemed ready and is returned for use by the customer.

The customer may then start the service and make use of the applications of which it is composed.

When a service instance is started, by a customer or by an operator, the reserved resources are deployed, installed and configured through the selected cloud provider interfaces and the corresponding resource quota are marked as consumed.

When a service is stopped, by a customer or by an operator, the resources deployed through the cloud provider interfaces are terminated and the corresponding resource quota are returned to the reserved state.

When an instance of service is no longer required by a customer it will be dismantled and the resources that were selected and reserved, through the providers’ interfaces, will be released to be made available for use in subsequent service composition placement operations for other instances and other customers.

# Over Booking

During the placement operation of service composition, resource quota availability information will be inspected to determine the suitability for selection for contribution of a cloud provider interface.

When no further suitable resources are available, through any of the cloud provider interfaces of an operator platform, then the service composition operation will fail. Overbooking may eventually be employed to alleviate this condition whereby the quantities of resources reserved during service composition, are allowed to exceed, by a certain percentage, the quantities expressed to be available in the original offer of service. This situation relies on precise operational knowledge, that has been acquired over time, concerning the average percentage difference between the quantity of reserved and undeployed resources and the quantity of resources that have been deployed and actively consumed.

# Federation Relationships

A federation relationship is directional and is established between two parties.

The first party, the initiator of the relationship, presents their offer of service, in the form of a service level agreement, to the second party, the responder, which may accept or refuse the offer.

A federation relationship management entry will be created in the responder’s database, for accepted offers, describing the initiator’s service proposition, in terms of cloud providers and their corresponding resource quota.

The publication service endpoint address, of the initiator, will be made available to the responder allowing direct and automated access to the description, placement and provisioning services of the initiator’s platform by the responder’s platform.

# Federation Placement

When federation relationships have been established between commercial operators, the effective pool of combined resources is increased accordingly.

When customer service composition fails due to insufficient local resources, and when permitted by the customer’s service level agreement placement algorithm, then federated provider quota, exposed through federation relationships, may be investigated to satisfy service composition needs.

Placement requests will be forwarded, via the publication service of the federated platform, to the remote placement service where resource selection and matching will be performed in the same manner as on the local platform.

Successful federated placement operations will return a pointer to the resulting placement instance to the originator containing the reference to their selected cloud service provider solution.

This placement pointer will be stored in the primary placement instance contributing to the customer service composition operation and subsequent placement state changes, where resources are consumed, reserved and eventually released, will also be forwarded to the remote platform to ensure consistent resource management and book keeping.

The selected remote cloud service provider, returned as the solution via the remote placement, will be used for all subsequent resource allocation and deployment operations as if it had been selected locally.

All service life cycle operations and actions (start, stop delete etc.) will be forwarded to the remote platform, through the remote placement and provider instance pointers, as and when they occur in complete transparency.

The perfect symmetry of the OCCI based model, on which the Amenesik Cloud Engine is based, allows this to be possible.

# Cloud Federation Scenario

The following diagram depicts a simple cloud federation scenario showing the relationships between three commercial platform operators, referred to here as Federation Member 1, Federation Member 2 and Federation Member 3.



Federation Member 1

Federation Member 2

Federation Member N

Customer-facing SLA

SLA facing SLA

Operator facing SLA

SLA facing SLA



Federation Member 1 is shown to have cloud provisioning interfaces and active subscription accounts with quota for the deployment of resources on an unnamed OpenStack provider and on Amazon AWS.

Federation Member 2 is shown to have cloud provisioning interfaces and active subscription accounts with quota for the deployment of resources on Amazon AWS, Microsoft Windows Azure and Google Compute Engine.

Federation Member 3 is shown to have cloud provisioning interfaces and active subscription accounts with quota for the deployment of resources on Google Compute Engine and IBM SoftLayer.

Bidirectional federation relationships are shown to have been established between Federation Member 1 and Federation Member 2.

Bidirectional federation relationships are also shown to have been established between Federation Member 2 and Federation Member 3.

Consequently, Federation Member 2 is in a relationship with both Federation Member 1 and Federation Member 3. This does not however mean that there is a visibility between Federation Member 1 and Federation Member 2

Each of these individual federation relationships is described and governed by an associated Service level agreement exposing the corresponding offer of service provisioning.

Service composition will be performed by each of the Federation Members for the instances of service to be delivered to their own customers as described by the corresponding customer facing service level agreements.

Federation Member 1 and Federation Member 2 are in the position to be able to delegate placement and selection of resource provisioning interfaces to the Federation Member 2 which, due to the stated bidirectional nature of their federation relationships, may further delegate responsibility to either Federation Member 1 or Federation Member 2.

The resulting federation situation is of great advantage to all three federation members allowing cloud service and resource deployment to be performed into public clouds that they would otherwise be unable to reach.

In addition, the total effective resource provisioning “fire power” of the resulting federation is doubled, concerning the provisioning quota available on Amazon AWS.

# Cloud Federation Architecture

The following diagram depicts the overall cloud federation architecture as described in the scenario above and shows the way in which operator’s platforms are daisy chained together through the placement engines and service level agreement management components of the individual cloud management platforms.

SLA MANAGER

Resource and Service Description

Service Life Cycle Management

Resource Deployment

Service and Resource Placement

Multi Cloud Abstraction Layer

Resource Monitoring

SLA MANAGER

Resource and Service Description

Service Life Cycle Management

Resource Deployment

Service and Resource Placement

Multi Cloud Abstraction Layer

Resource Monitoring

SLA MANAGER

Resource and Service Description

Service Life Cycle Management

Resource Deployment

Service and Resource Placement

Multi Cloud Abstraction Layer

Resource Monitoring

# Customer Invoicing

Platform operators are responsible the payment of the costs of all resources provisioned through the cloud provider subscription accounts associated with their available cloud provider interfaces.

Start and Stop transactions are created when resource provisioning and resource termination requests are received and handled, in response to customer service life cycle events, through cloud provisioning interfaces of a platform.

Transactions are identified by, and attached to, the customer account, the initiator of the service level agreement corresponding to the relevant service instance.

Transaction collation is performed periodically, by platform operators, for the processing of transactions associated with customer accounts and results in the preparation and presentation of their customers’ invoices.

The transaction collation process is responsible for the application of customer specific commercial conditions as may be expressed by the pricing elements of the terms of their corresponding service level agreements.

Subsequent invoice payment, by customers and operators alike, and the associated commercial accounting is outside of the scope of the Amenesik Cloud Engine.

# Cloud Federation Cost and Revenue Sharing

Cost and Revenue sharing between the platform operator members of a federation requires further extensions to the accounts, provider quota, placement, transaction and price management mechanisms currently in place in the Amenesik Cloud Engine.

To be able to track transactions and costs associated with resource deployment operations performed by and for remote federation members, is will be necessary to extend the transaction creation operation to allow the management of automated linkage of transactions to the relevant provider accounts of the related federation partners.

This will require that a provider account transaction collation process be developed, for the processing of the transactions associated with federation partner’s accounts, resulting in the preparation and presentation of the corresponding invoices, as described by specific commercial conditions, expressed in the terms of their corresponding service level agreements.

The detailed specification, and subsequent development, of this cost and revenue sharing mechanism for use with the automated cloud federation deployment described above, will be the subject of the subsequent work performed in year two of the BASMATI project.

# Basmati Business Federation Demo

## Introduction

This section of this document presents the Basmati Business Federation Demo and describes the steps required for its realisation. The demonstration shows the use of the Amenesik Cloud Engine with extensions developed during the first period of period of the Basmati project. These extensions include Docker container based provisioning through the Amazon AWS Elastic Container Service, SCALR IAAS cloud provisioning, the Application State Controller based on BEAM /TOSCA Application description documents and Automated Business Federation Support for Cloud Service provisioning.

The demonstration comprises three individual Amenesik Cloud Engine platforms, each specifically designed for the purposes of the demo and deployed three different regions of the globe on Amazon EC2. Each platform has been designed to offer a subset of the standard public cloud offering to demonstrate the way in which Cloud Service Federation can bring together different technology specialists in a fully automated and transparent manner.

Each of the ACE platforms will enter into Federation relationships with other platforms making their particular provisioning specialities available to the others.

The Basmati Use Case Application description documents will be introduced to each platform and will be used to create application state control instances irrespective of the ability of the providers own provisioning.

The applications will be started and the cloud resources will be deployed across the federation using the mutualised resources of the federation partners.

## Workflow

The following steps were accomplished in order to realise this demonstration.

## Platform Design

Three different versions of the Amenesik Cloud Engine were prepared each with its own specific cloud provisioning interfaces and their associated cloud subscription account.

### Federation Member 1

Deployed on Amazon EC2 in Dublin, Ireland with Amazon EC2, RDS, ECS and ElasticBeanStalk PaaS provisioning and OpenStack provisioning at OVH in Strasbourg and Gravelines, France.

This platform will use the domain name **basmati-eu.amenesik.com.**

### Federation Member 2

Deployed on Amazon EC2 in Frankfurt, Germany with Amazon EC2, RDS, ECS and ElasticBeanStalk PaaS provisioning, Microsoft Windows Azure and Google Compute Engine allowing worldwide coverage.

This platform will use the domain name **basmati-de.amenesik.com.**

### Federation Member 3

Deployed on Amazon EC2 in Seoul, South Korea, with Google Compute Engine, IBM Softlayer and SCALR provisioning accounts and interfaces allowing worldwide coverage and also bare metal deployment.

This platform will use the domain name **basmati-kr.amenesik.com.**

## Platform Production

The platform descriptions were prepared using the Amenesik Cloud Engine design tool allowing generation of the code, binaries, installation, configuration and deployable packages. The resulting ACE technical manifests were introduced into the master ACE deployment platform for the preparation of the three corresponding service level agreements for the description and control of their deployment. Service instances were created from the agreements then the instances were started.

Three virtual machines were started, one in each of the defined regions and the corresponding ACE platform packages were installed and configured.

## Login Accounts

The following steps require connection to the user interface dashboard of each of the three platforms, using a standard web browser and the following endpoint URLs

[http://basmati-eu.amenesik.com](http://basmati-eu.amenesim.com)

[http://basmati-de.amenesik.com](http://basmati-eu.amenesim.com)

[http://basmati-kr.amenesik.com](http://basmati-eu.amenesim.com)

For the creation of a user login account (user=basmati, password=any password,email=any email) followed by login as the newly created/user.

## Subscription Accounts

It is now necessary to connect to the Remote Command Server of each of the deployed platforms using a standard web browser and each of the following URLs

[https://basmati-eu.amenesik.com](http://basmati-eu.amenesim.com):8186/parser

[https://basmati-de.amenesik.com](http://basmati-eu.amenesim.com):8186/parser

[https://basmati-kr.amenesik.com](http://basmati-eu.amenesim.com):8186/parser

The relevant subscription documents are to be selected for each platform and parsed into the corresponding system

### Federation Member 1

* aws-subscriptions.xml
* ovh-subscriptions.xml

### Federation Member 2

* aws-subscriptions.xml
* waz-subscriptions.xml
* gce-subscriptions.xml

### Federation Member 3

* ibm-subscriptions.xml
* scalr-subscriptions.xml
* gce-subscriptions.xml

## Federation Relationships

The fully connected, bidirectional, federation relationships between the three platforms can now be created through the **Federation** tab of the ACE dashboard of each platform instance.

### Federation Member 1

This platform will be connected by a federation relationship to the publication services of member 2 and member 3 through the URLs:

[https://basmati-de.amenesik.com](http://basmati-eu.amenesim.com):8086/publication/

<https://basmati-kr.amenesik.com>:8086/publication/

### Federation Member 2

This platform will be connected by federation relationships to the publication services of member 1 and member 3 through the URLs:

[https://basmati-eu.amenesik.com](http://basmati-eu.amenesim.com):8086/publication/

[https://basmati-kr.amenesik.com](http://basmati-eu.amenesim.com):8086/publication/

### Federation Member 3

This platform will be connected by a federation relationship to the publication services of member 1 and member 2 through the URLs:

[https://basmati-eu.amenesik.com](http://basmati-eu.amenesim.com):8086/publication/

[https://basmati-de.amenesik.com](http://basmati-eu.amenesim.com):8086/publication/

## Application Descriptions

The two Basmati use case applications are described in BEAM / TOSCA format and are available for download via the following links:

[http://www.amenesik.com/tosca/tosca-trip-builder-v10a03-world.xml](http://www.amenesik.com/tosca/tosca-trip-builder-v10a-world.xml)

[http://www.amenesik.com/tosca/tosca-yellow-map-v10a02-world.xml](http://www.amenesik.com/tosca/tosca-trip-builder-v10a-world.xml)

## Application Deployment

These documents will be used by the CORDSCRIPT / OCCISCRIPT programs allowing creation of the associated application state controller instances.

* csp-create-trip-builder.txt
* csp-create-das-fest.txt

Which are to be launched through the remote command server interface of the required platform instance. The processing of the BEAM documents will generate the corresponding technical manifests and deployment agreements for the account indicated by the parameter provided when the script is launched. The resulting application controller will build application states and service instances corresponding to the deployment agreements.

The application control instance is then ready to respond to the following life cycle action invocation requests:

* Start
* Change
* Revert
* Save
* Stop

## Conclusions

The successful deployment of the BASMATI Use Case applications, exploiting cloud provisioning subscriptions and interfaces of remote federation member platforms, from the Application Controller of a single federation member, has been accomplished and demonstrates the benefits of the ACE Cloud Services Abstraction Layer. This allows specialisation of the roles of the cloud service provider operators as members of a commercial federation.