

COMPETITIVENESS AND INNOVATION FRAMEWORK PROGRAMME

CIP-ICT-PSP-2013-7



SERVICE DISTRIBUTION NETWORK AND TOOLS FOR INTEROPERABLE PROGRAMMABLE, AND UNIFIED PUBLIC CLOUD SERVICES

Deliverable D4.2a

Migration, Adaptation, Localization and Governance Tools

Workpackage	WP4 – Framework Implementation, Integration and Test
Editor(s):	Giannis Ledakis, Panagiotis Gouvas, Juri Hudolejev, Ilja Livenson
Responsible Partner:	NICPB & SILO
Quality Reviewers	Fotis Karayannis (URNS), Géry Ducatel (BT), Joshua Daniel (BT), Christos Kanellopoulos (URNS)
Status-Version:	V1.0
Date:	23/01/2015
EC Distribution:	Public
Abstract:	This deliverable will reflect the outcomes of tasks T4.1 and T4.3. It is a documentation part of the integrated prototype for migration, adaptation, localization and governance services of the STRATEGIC framework.

Document Revision History

Version	Date	Modifications Introduced	
		Modification Reason	Modified by
v1.0	23/01/2015	First version	NICPB & SILO

Contents

EXECUTIVE SUMMARY	7
1 INTRODUCTION	8
1.1 SCOPE AND PURPOSE OF THE DOCUMENT.....	8
1.2 TARGET AUDIENCES	8
1.3 STRUCTURE OF THE DOCUMENT.....	8
2 CLOUD ENABLEMENT TOOLS	9
2.1 MARKETPLACE-BASED ENABLEMENT.....	9
2.1.1 Workload base settings.....	9
2.1.2 Workload configuration parameters.....	11
2.1.3 Workload deployment scripts.....	11
3 INFRASTRUCTURE ONBOARDING	12
4 MIGRATION TOOLS	16
4.1 INVENTORY-BASED APPROACH	16
4.2 BINARY IMAGE RE-CONTEXTUALISATION.....	17
4.3 DATA MIGRATION	19
5 ADAPTATION AND LOCALIZATION TOOLS.....	20
5.1 ADAPTATION TOOLS.....	21
5.2 LOCALIZATION TOOLS.....	23
6 GOVERNANCE TOOLS.....	25
7 CONCLUSIONS	28

List of Figures

FIGURE 1 – STANDARDIZED CLOUD INTERFACE COMPONENTS	12
FIGURE 2: GRAPHICAL USER INTERFACE OF TARGET CLOUD INFRASTRUCTURES MANAGEMENT..	15
FIGURE 3: ADDING CLOUD INFRASTRUCTURE THOUGH GRAPHICAL INTERFACE	15
FIGURE 4: CREATING CLOUD PROFILE THOUGH GRAPHICAL INTERFACE	15
FIGURE 5: START SERVER MIGRATION, SELECT A TARGET (PRIVATE CLOUD).....	17
FIGURE 6: IDENTIFYING THE SERVER TO MIGRATE	18
FIGURE 7: CLOUD PROFILE IMAGES	18
FIGURE 8 – LOCALIZATION & ADAPTATION TOOLKIT COMPONENTS	20
FIGURE 9 – APPLICATION MANAGEMENT COMPONENTS	20
FIGURE 10: CREATING PARAMETERS WITH METADATA CONSTRUCTION TOOLKIT	21
FIGURE 11: SETTING PARAMETER VALUES THOUGH THE METADATA TOOLKIT	22
FIGURE 12: INTERACTION BETWEEN IP REGISTRY AND SERVICE STORE	24
FIGURE 13: FILTERED RESULTS BASED ON LOCAL NEEDS	24
FIGURE 14 – GOVERNANCE COMPONENT COMPONENTS.....	25
FIGURE 15 – BASIC ENTITIES GOVERNED BY SERVICE STORE, AS SHOWN IN GRAPHICAL INTERFACE.....	26
FIGURE 16 – MANAGING A SERVER THROUGH GRAPHICAL INTERFACE.....	27

List of Tables

TABLE 1: DEFINITIONS, ACRONYMS AND ABBREVIATIONS	6
TABLE 2: MOST IMPORTANT FUNCTIONALITIES FOR SERVICE STORE.....	13
TABLE 3: LIST OF SUPPORTED INFRASTRUCTURES	14
TABLE 4: IP REGISTRY CALLS RELATED TO SERVICE STORE.....	23
TABLE 5: SUPPORTED CORE FUNCTIONALITIES OF GOVERNANCE TOOLKIT	26

Definitions, Acronyms and Abbreviations

Acronym	Title
CMS	Content Management System
ECP	European Cloud Partnership
ENISA	European Union Agency for Network and Information Security
ICT	Information and Communications Technology
ISV	Independent Software Vendor
IP	Infrastructure Provider
IPR	Intellectual Property Rights
LGFL	London Grid for learning
PSN	Public Services Network
SIG	Special Interest Group
SME	Small-Medium Enterprise

Table 1: Definitions, Acronyms and Abbreviations

Executive Summary

The main target of this deliverable is to describe the application migration, adaptation, localization and governance tools offered through the STRATEGIC Service Store. The work is based on the specification defined in D3.2 and is complementary to that document.

Each class of tools is described in a separate section and provides technical guidance for achieving a particular goal.

Cloud enablement tools are intended for porting existing distributed applications to STRATEGIC Service Store and making them re-deployable on multiple IaaS offerings. A set of tools helps with definition of the base settings of the application, definition of the configuration parameters as well as deployment scripts.

To reduce vendor lock-in and allow service providers to offer hosting services to STRATEGIC applications, infrastructure on-boarding procedure and tools are described. The provided migration tools and procedures are intended for moving a deployed application from one IaaS provider to another. Based on the type of application, one of three proposed migration options can be chosen. The first method – Inventory-based – requires definition of a configuration of the “golden image” that is then recreated on a target platform. Binary image re-contextualisation method allows to transfer the whole binary image to the target cloud. Data migration is a method of application migration through data volume migration. This method is still in development and would require support from the underlying IaaS stack.

Application adaptation and localization is done through possibility to define and configure settings of the STRATEGIC Service Store offering. Adaptation is a process that for increasing flexibility of deployment of the application. Application packager can allow tuning certain aspects of the application so that the technical administrator of the municipality or other pilot can then use for localization of the application.

Governance tools are meant for managing the lifecycle of the deployed application - modifying its run state and de-provisioning an application.

The deliverable serves as a reference for the WP5 activities (pilot use case operations) by defining the STRATEGIC toolkit for typical operations based on the STRATEGIC Service Store. While this toolkit is not final, it covers basic requirements of the pilots for the whole lifecycle of the applications. The second iteration of this deliverable will include extended and refined tools based on the experience and feedback from the WP5 activities.

1 Introduction

1.1 Scope and purpose of the document

The main goal of the STRATEGIC project is to facilitate organisations and notably public bodies to leverage the benefits of public cloud services, through boosting three complementary adoption directions: (a) the porting of existing on-line services to the Cloud, (b) the adaptation and localisation of existing services, which have been successfully deployed by other organisations and (c) the composition of new public cloud services on the basis of available legacy services.

This deliverable – D4.2a – is an outcome of the task – T4.1. The focus of the task was on the provision of a range of tools that would enable cloud developers and ISVs to migrate distributed services to the cloud and adapt and localize them on deployment. In addition, methods for integration with multiple IaaS systems were to be analysed.

The task was relying on the brokerage services and the STRATEGIC Service Store developed in Task 4.2.

The purpose of the document is description of the devised tools and processes for the goals of the task 4.1.

1.2 Target audiences

The target audience for this document is the technical partners of this project, the technical administrators of the Pilots and the independent software vendors (ISV) who wish to resell applications through the STRATEGIC Service Store.

1.3 Structure of the document

This document is split into four main chapters in addition to Chapter 1 and they comprise of:

- The Chapter 2, "Cloud enablement tools", covers tools that are used for moving a distributed application to a cloud;
- The Chapter 3, "Infrastructure onboarding", covers process of extending STRATEGIC infrastructure with new IaaS providers.
- The Chapter 4, "Migration tools", covers tools and process for migration of applications from one IaaS provider to another.
- The Chapter 5, "Adaptation and localization tools", covers STRATEGIC Service Store's tools for modifying application on deployment.
- The Chapter 6, "Governance tools", covers management of the already provisioned services.

2 Cloud enablement tools

Cloud-enabled STRATEGIC applications are defined as packages in a STRATEGIC Service Store (aka marketplace) with metadata conforming to the STRATEGIC metadata defined in D3.1 "Specification of Cloud-Enablement and Migration Solutions and Services". A referenced document contained an example of composition of a new package for a hypothetical use case using visual tools provided by a marketplace. The method and a supporting STRATEGIC Service Store constitute the main method for cloud enabling of the STRATEGIC pilot applications.

Below we list a standard flow for cloud enablement and links to the STRATEGIC metadata model fields.

For more detailed description of all the steps, D3.1 should be consulted. A user role responsible for cloud-enabling the application is referred to as "packager". The marketplace is logically split into graphical UI (frontend) and server-side implementing business-logic (backend). These terms are used below to indicate direction of data flow.

2.1 Marketplace-based enablement

The first two steps are new workload initialization and configuration of the basic info (sections 5.3.1 and 5.3.2 of D3.1). In the console, the packager navigates to Marketplace > Packaging and enters application properties: name and runtime. Once submitted, this data is used in the backend to create a new workload object with the following metadata parameters set:

- Name: application provided by packager in 'AppName' input
- ImageURL: selected from pre-defined set based on 'In language' dropdown value selected by user, or empty if selected value has no image URL associated

After the workload object is created in the backend, the packager can update its metadata properties (section 5.3.3 of D3.1).

2.1.1 Workload base settings

The following workload properties can be set or changed on 'General Settings' page in frontend:

- Name: application name; populated from workload object's property 'Name'; cannot be empty
- Version: application version
- License: application license identifier -- usually an acronym (examples: Apache, MIT, Commercial)
- Category: application category; can be selected from pre-defined set of categories – see section 5.3.3 of D3.1 for details
- ImageURL ('Icon' field): application icon URL; if workload object's property 'ImageURL' is set, image is displayed from that URL; new image can be uploaded using file upload control; uploaded image (if any) is stored on

server; workload object's property 'ImageUrl' is populated with URL generated for uploaded image (if any)

- Information.ApplicationHomePageURL ('Homepage' field): application homepage
- Information.Summary ('Summary' field): application summary (short description)
- Information.Description ('Description' field): application description (longer and more detailed)
- Information.SupportInfo ('Support' field): application support information (documentation, support channels, bug tracker URL etc.)
- Information.GettingStartGuidelines ('How to Get Started' field): getting started instructions (installation and configuration manual, first steps, best practices etc.); text is entered in WYSIWYG widget and is automatically rendered to HTML
- SupportedOS.OS.* ('Supported OS' field): list of operating systems the application can run on; application can support more than one operating system, so multiple checkboxes can be selected (including zero)
- DeployMethod ('Deploy Method' field): application deployment method; single value can be selected from pre-defined set of deployment methods.
- Accessibility: visibility of application (private or public); single value can be selected from pre-defined set of visibility methods -- see section 5.3.3 of D3.1 for details

'Name' field must be set to non-empty string. All other fields may be empty.

Every selected 'Supported OS' checkbox provides these values to backend:

- SupportedOS.OS.Name
- SupportedOS.OS.Version
- SupportedOS.OS.Architecture

For example, once 'ubuntu 12.04' checkbox is selected, the following request is sent to backend (pseudo-code):

```
workload.SupportedOS.add({
  OS: {
    Name: 'Ubuntu'
    Version: '12.04'
    Architecture: 'x86_64'
  }
})
```

2.1.2 Workload configuration parameters

Workload configuration parameters can be set or changed on 'Parameters' page in the frontend. Initially workload has no configuration parameters, so none is displayed. The packager can add configuration parameter by clicking '+ Config Item' button. As a result, a new section is added on the page, where the packager can provide the following configuration options:

- ConfigurationParameters.Configuration.Name ('Display Name' field): configuration parameter name
- ConfigurationParameters.Configuration.Type ('Type' field): configuration parameter type; single value can be selected from pre-defined set of parameter types -- see section 5.3.4 for details
- ConfigurationParameters.Configuration.Required ('Required' checkbox): indicates if this configuration parameter is required (yes if checked) for application deployment
- ConfigurationParameters.Configuration.DefaultValue ('Default Value' field): default value for this configuration parameter name; should be used in deployment if user does not provide the configuration value explicitly
- ConfigurationParameters.Configuration.Description ('Description' field): configuration parameter description
- ConfigurationParameters.Configuration.ReadOnly ('Read only' checkbox): indicates if this configuration parameter is writable (yes if unchecked) by user deploying the application.

All fields are optional.

2.1.3 Workload deployment scripts

Workload deployment scripts can be added or edited on the 'Package Scripts' pages in the frontend. Initially the workload does not include any deployment scripts. The packager can provide a single script for any of the application lifecycle phases:

- Installation ('install' page)
- Start ('startup' page)
- Restart ('refresh' page)
- Removal ('uninstall' page)

The following workload metadata parameters can be set or updated on the 'Package Scripts' pages:

- DeploymentDescriptor.DeploymentScript.Name ('Name' field): script name
- DeploymentDescriptor.DeploymentScript.Language ('Script language' field): script language: 'shell' for UNIX-like servers and 'powershell' for Windows servers
- DeploymentDescriptor.DeploymentScript.Content ('Script' field): actual script

All fields must be set to non-empty strings.

3 Infrastructure onboarding

Interoperability in STRATEGIC refers to the ability of STRATEGIC framework to support different IaaS infrastructures. Infrastructure onboarding is the process of adding new IaaS to the ecosystem of STRATEGIC.

As specified in deliverable D2.3 STRATEGIC Framework Architecture and Technical Specifications the interoperability is based on the usage of Standardized Cloud Interface. The components that compose Standardized Cloud Interface are displayed in **Figure 1**.

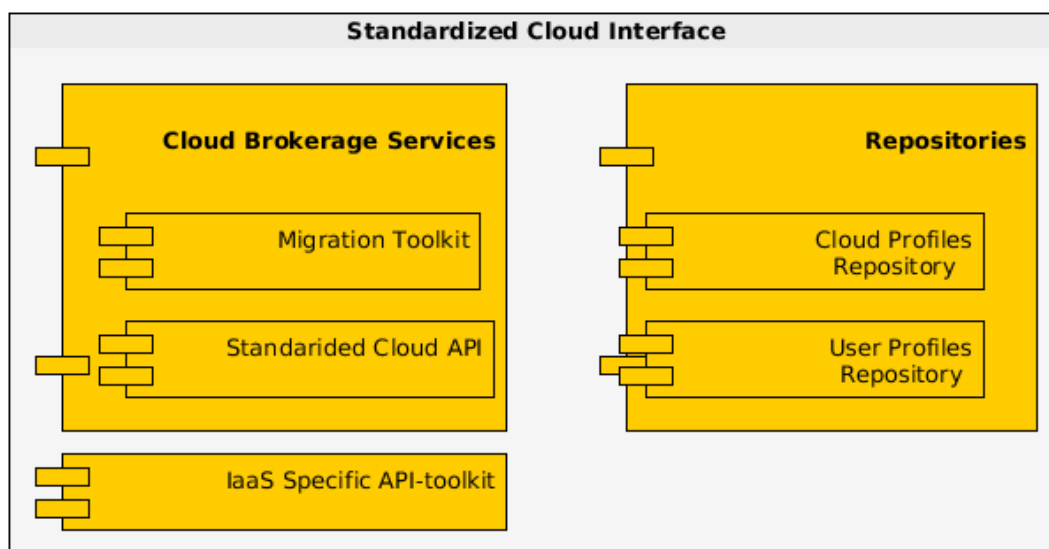


Figure 1 – Standardized Cloud Interface components

In order to support IaaS based infrastructures, STRATEGIC Service Store expects that each IaaS exposes its functionality via a set of RESTful APIs. The Service Store supports specific functionalities, but due to the diversity implied from the underlying infrastructure, not all functionalities are supported from all IaaS offerings. These functionalities are encapsulated in the Standardized Cloud API – a STRATEGIC Service Store implementation specific abstraction of operations that can be performed on supported clouds. In the following Table 2 the most important functionalities of Standardized Cloud API that are needed in order to support an IaaS in Service Store are identified.

Functionality	Description
Is Server Stoppable	The ability to stop a server
Support Server Operations	Ability to perform extended lifecycle operations on a server
Support Import Machine Image	The ability to support the process of importing machine images
Support Volume Operations	The ability to support CRUD operations on volumes

Functionality	Description
Support SSH Keys	The ability to support SSH keys injection
Support Account Provision	The ability to support different accounts and the provisioning of them
Support Load Balancers	The ability to support load balancers and their configuration
Support Port Range and Static Public IP	The ability to support configuration for the services

Table 2: Most important functionalities for Service Store

STRATEGIC Service Store can benefit from an existing ecosystem of supported cloud infrastructures. The technology behind the STRATEGIC platform currently supports the platforms presented in Table 3, integration of other platforms that uses similar APIs is possible.

IaaS	Description
OpenStack	OpenStack is an open-source software cloud computing software platform, released under the terms of the Apache License. It can be used as an infrastructure as a service (IaaS) solution that controls large pools of compute, storage, and networking resources throughout a datacenter, managed through a dashboard or via the OpenStack API.
BT Cloud Compute	BT Cloud Compute is an IaaS Service provided by BT.
CloudStack	Apache CloudStack is open source software designed to deploy and manage large networks of virtual machines, as a highly available, highly scalable Infrastructure as a Service (IaaS) cloud computing platform.
CloudPlatform	Citrix CloudPlatform is the commercially supported version of the Apache CloudStack orchestration platform from Citrix Systems.
Eucalyptus	Eucalyptus is an open-source computer software for building Amazon Web Services (AWS)-compatible private and hybrid cloud computing environments.
vCloud	vCloud is a cloud computing initiative from VMware which will allow customers to migrate work on demand from their "internal cloud" of cooperating VMware hypervisors to a remote cloud of VMware hypervisors.

IaaS	Description
Amazon EC2	Amazon Elastic Compute Cloud (EC2) is a central part of Amazon.com's cloud computing platform, Amazon Web Services (AWS). EC2 allows users to rent virtual computers on which to run their own computer applications.
Windows Azure	Microsoft Azure is a cloud computing platform and infrastructure, created by Microsoft, for building, deploying and managing applications and services through a global network of Microsoft-managed datacenters.
HP Cloud	HP Cloud is the set of cloud solutions available from Hewlett-Packard (HP) that offers public cloud, private cloud, hybrid cloud, managed private cloud, and other cloud service.
Rackspace	Rackspace Cloud offers four hosting products: Cloud Servers for on-demand computing power; Cloud Sites for robust web hosting; Cloud Load Balancers for easy, on-demand load balancing and high availability; and Cloud Files for elastic online file storage and CDN.

Table 3: List of supported infrastructures

In order to use an IaaS based infrastructure by STRATEGIC Service Store, the IaaS must become available to STRATEGIC Service Store. For this, network configuration is required in order to allow connections between the IaaS and STRATEGIC Service Store. Also for the on-boarding of an IaaS on STRATEGIC Service Store, the STRATEGIC administrator is able to control the whole process and have the final decision about the approval of the offered IaaS. The needed graphical user interface for the management of Cloud Infrastructure is called Cloud Data Center and is depicted in Figure 2.

Cloud Data Center










Enabled	Cloud Provider	Category	
<input checked="" type="checkbox"/>	 rackspace HOSTING	Public	Manage
<input checked="" type="checkbox"/>	 VMware vCloud vCloud vCenter vSphere	Public	Manage
<input checked="" type="checkbox"/>	 amazon web services™	Public	Manage
<input checked="" type="checkbox"/>	 hp Cloud	Public	Manage
<input checked="" type="checkbox"/>	 Windows Azure	Public	Manage
<input checked="" type="checkbox"/>	 CloudPlatform powered by Apache CloudStack	Private	Manage
<input checked="" type="checkbox"/>	 AppStack Managed	Public	Manage
<input checked="" type="checkbox"/>	 mock HOSTING	Public	Manage
<input checked="" type="checkbox"/>	 openstack CLOUD SOFTWARE	Private	Manage

Figure 2: Graphical user interface of target cloud infrastructures management

In order to add a new Cloud Infrastructure a specific form is available and is displayed in Figure 3.

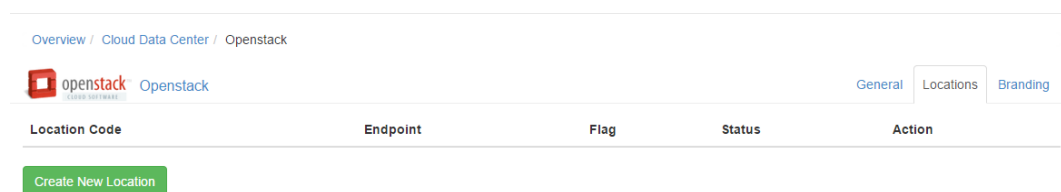


Figure 3: Adding Cloud Infrastructure through graphical interface

For the IaaS that are already available in STRATEGIC Service Store, each user has to create a specific Cloud Profile. Cloud Profiles represent the information stored for each user and for each specific IaaS in order to achieve IaaS level management through STRATEGIC. These profiles are stored in the Cloud Profiles repository. If the cloud target is public, the keys have to be provided and the cloud target becomes immediately available. If the cloud target is private, and the Cloud Profile has been setup by the Strategic Administrator, then the process is identical, the API key and secret key has to be provided as displayed in Figure 4.

Figure 4: Creating Cloud Profile through graphical interface

4 Migration tools

Given the high risk of being caught in a vendor lock-in situation, it is important to consider Cloud Migration from the beginning of a cloud project. Regardless of which IaaS migration solution is considered, it would be prudent to maintain a strong documentation discipline whilst building cloud based services.

The migration from an IaaS provider to another is prone to create engineering challenges. There can be multiple approaches available, the best one may depend on what target is being considered, and what platform is currently used for hosting (e.g. OpenStack, CloudStack, AWS...), amongst other contextual elements. However, due to underlying platform incompatibilities (different hypervisors, meta data wrapping, template formats) from different independent software vendors (Amazon, Citrix, VMWare, Microsoft, etc.) it is not practical to write a step by step guide. Migration from one IaaS to another, whether it is from public to private, or the reverse, and regardless of technologies involved can create integration issues, which require IT expertise.

This document discusses two approaches. The first approach is a duplicated redeployment underpinned by rigorous documentation of the software stack installation process, and a deep understanding of the interoperability design and architecture. Whilst this can be supported by software tools in some occasions, it is more a stringent documentation process which is required. The end goal is to be the ability to replicate the expected environment in a different context (i.e. different IaaS providers).

The second approach consists of an attempt to create a digital copy of the working solution, and to re-deploy into a new context. However, this can result in difficult engineering challenges such as re-contextualisation of VMs (Virtual Machines) for the new environment (hypervisor, bespoke functionalities, etc.) compatibility issues of images supported by the environment, loss of interoperability between Virtual Machines, network configuration issues due to de-contextualisation, etc. Evidently, as cloud deployments get bigger, challenges to support this approach also grow. The marketplace does have one tool to support this approach, which will be described in this document, however, accessibility to a single tool for such an impactful migration might not be enough.

4.1 Inventory-Based Approach

The inventory consists of keeping a track record of the installation history. Note, the Service Store does not provide tools to support the inventory-based approach; this section should be considered as an advisory and “best practice” approach. This process requires the creation of a “gold image” that normally represents the base installation. If the base installation is deployed on multiple-servers, a “gold image” of each server will be required. The configuration of the “gold image” can be saved as a Puppet¹, or a Chef² script assuming that the installation process is paired with one of these tools from day 1. In cases the software installation is not managed through those installation tools, a software inventory solution can be implemented. This software monitors software installation and creates an installation scripts by reverse engineering software

¹ Puppet: <http://puppetlabs.com/>

² Chef: <https://www.chef.io/>

deployment. In other words, inventory management tools scan to trace software installations, and attempts to create a reusable software stack installation set of scripts. Such tools include aptitude³ (Ubuntu), YUM⁴ (CentOS, RedHat), Synaptic⁵ (all Linux), Etckeeper⁶ (all Linux). Detailed usage of these tools or a comparison, is considered off topic for this document.

The migration process entails re-creating Virtual Machines in the new target Cloud Service Provider. The Virtual Machine may need to run a specific Operating System. If this is not available, the deployment scripts may prove incompatible. If the deployment is a private cloud, the correct templates might be loaded to the Service Store by a Service Store administrator. Migrating from a private cloud to a public cloud does not offer this flexibility.

When migrating to a private cloud provider (e.g. a corporate private cloud) the STRATEGIC Service Store can be configured to deploy to different “life cycle” environments. The life cycles are named Development, Staging, and Production, which correspond to a typical software life cycle. During an IaaS change (to a private) it is useful to create a workflow, or more, in the development life cycle. If the migration needs re-configuring, it can be done safely, and documented for future reference.

4.2 Binary Image Re-contextualisation

Binary images can be imported and added to the list of VM images available for a location. This allows to imported binary images from live VMs. The VM can only be imported for a private cloud deployment and currently, only a Linux environment can be processed.

To start the migration select “tools” from the top menu, and then “server migration”, see Figure 5.

Figure 5: Start server migration, select a target (private cloud)

The second step consists simply of pointing to the server that should be saved. Ideally, this server should be in its “golden image” state. It should be up to date with software and only the necessary amount of data and configuration to start over the deployment if needed without having to undo any integration or remove any data. As shown on Figure 6, the migration tool needs to know which OS type (between Windows, and Linux – at the time of writing only Linux OS types are supported). The host name or IP address, a port to SSH into the server (22 being default), and some root level credentials.

³ <https://help.ubuntu.com/10.04/serverguide/aptitude.html>

⁴ <http://yum.baseurl.org/>

⁵ <http://www.nongnu.org/synaptic/>

⁶ <http://etckeeper.branchable.com/>

Schedule Server Migration

1 Choose Cloud Provider > 2 Connect to Source Server > 3 Review & Start Migrate! >

* Name
Name of the template

OS Type
Operation system of the template

Source Server

* Hostname/IP

* SSH Port

* User

* Password

[Last Step »](#) [Next Step »](#)

Figure 6: Identifying the server to migrate

Once the migration is complete, the image will be available for reuse in the Cloud Profile menu which can be seen on Figure 7.

Cloud Profile - Alpha 3

Settings Regions Flavors Images Limits Network

All Locations

Name	Number of Servers	OS Type	Username	Password	Size (GB)	Location	Zone
Appstack Win2008R2 Xen	0	windows 2008R2	Administrator	(not defined)	80	Alpha3	Xen Compute Zone
Appstack Redhat 6.3 Xen_3.12_1	3	redhat 6.3	root	(not defined)	20	Alpha3	Xen Compute Zone
Appstack Redhat 6.6 VMwarezone	0	redhat 6.6	root	(not defined)	20	Alpha3	VMware Compute Zone
Appstack Ubuntu 12 Xen_3.12	0	ubuntu 12.04	root	(not defined)	20	Alpha3	Xen Compute Zone
Appstack Centos 6.3 VMWARE_3.12	0	centos 6.3	root	(not defined)	20	Alpha3	VMware Compute Zone
Appstack RedHat 6.6 VMWARE_1	0	redhat 6.6	root	(not defined)	20	Alpha3	VMware Compute Zone
Appstack Centos 6.3_Xen 3.123	0	centos 6.3	root	(not defined)	20	Alpha3	Xen Compute Zone
Appstack Centos 6.3 Xen_3.12_1	0	centos 6.3	root	(not defined)	20	Alpha3	Xen Compute Zone
Appstack Win2008R2 VMWARE	0	windows 2008R2	Administrator	(not defined)	50	Alpha3	VMware Compute Zone

Figure 7: Cloud Profile Images

4.3 Data Migration

Data migration refers to importing data volumes, or migrating storage. Currently the STRATEGIC Service Store is not providing migration tools, which support these processes. The migration of a Database might be addressed using one of the two solutions described above knowing that in either case the re-contextualisation might produce network configuration challenges with support software (e.g. firewall, load balancers, etc...).

Typically data migration will require a data export and import. If a data volume is exported, it would be sensible to consider integration at a lower level software such as OpenStack, or CloudStack.

5 Adaptation and localization tools

The STRATEGIC framework offers the adaptation and localization of existing public services by providing solutions and tools that allow the adaptation of the application for specific needs based on local and international requirements shaped by legal constraints related to security and privacy.

STRATEGIC Service Store provides the means for configuring and adapting applications and services with the Localization and Adaptation Toolkit and also with the Application Management Component. These components of STRATEGIC Service Store have been described in details in deliverable D2.3 [1] and are depicted in figures Figure 8 and Figure 9 respectively.

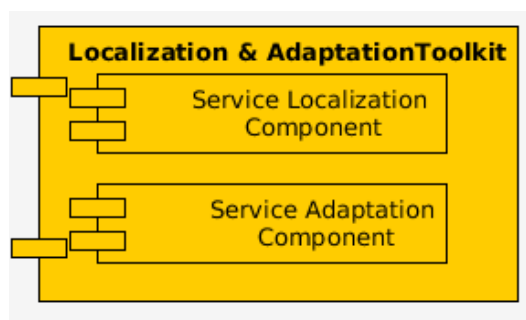


Figure 8 – Localization & Adaptation Toolkit components

Localization and Adaptation Toolkit is responsible for the providing the needed services that allow the localization and adaptation of applications published in the marketplace and it consists of the Service Adaptation Component that is responsible for the adaptation and Service Localization Component that is responsible for the localization.

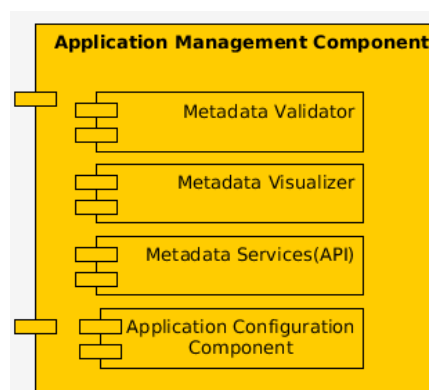


Figure 9 – Application Management components

Also, Application Management components are responsible for the management of application in terms of adaptation. This is possible with the metadata visualizer and validator that are responsible for the displaying, editing and validation of application metadata and the application Configuration Component that allows the configuration of application from Customer Administrator.

5.1 Adaptation tools

Based on Service Store, STRATEGIC offers the ability to instantiate the applications offered and also the ability to adapt them, both before and after deployment.

From a perspective of a developer that publishes applications, adaptation refers to the definition of appropriate metadata that application can use. These metadata are based on the STRATEGIC Workload Metadata Model that has been described in deliverable D3.1 Specification of Cloud-Enablement and Migration Solutions and Services [2].

STRATEGIC Service Store offers the graphical interface for the creation of these adaptation parameters. An example of this interface is shown in Figure 10.

The screenshot displays the 'Metadata construction toolkit' interface. At the top, there is a navigation bar with icons for Dashboard, Getting Started, Launch, VPCs, Workloads, Apps, Tools, Reports, and Settings. The main form includes the following fields and options:

- Approval State:** approved
- Name:** Public Portal App
- Accessibility:** private
- Category:** Samples
- App Cost(Hourly):** 0.0
- Icon:** A file selection field showing 'Avatar.png' and 'STRATEGIC_logo_cloud.png'.
- Summary:** This is a sample application published by STRATEGIC project
- Description:** This is a sample application published by STRATEGIC project
- Website Url:** http://mynewapphomepage.com
- License:** N/A
- How To Get Started:** A rich text editor with a toolbar containing 'Normal text', 'Bold', 'Italic', 'Underline', and other formatting options.
- Support:** A text area for providing support information.

At the bottom of the form, there are 'Update' and 'Back' buttons, and a 'Hide advanced settings' link.

Figure 10: Creating parameters with Metadata construction toolkit

This graphical interface consists of an input form for the parameters with the following attributes per each parameter:

- **Display Name:** How the parameter is displayed to users
- **Type:** The formatted field type to display to the user:
 - Number
 - Text
 - Server-select
 - Checkbox
 - Password

- Single-select
- **Default Value:** The pre-defined value for this parameter, allowing the user to simply accept this value
- **Display Group:** What group the input parameter should be displayed within
- **Description:** A text description of this input parameter
- **Required:** Indicates if the value is optional or required
- **Read-Only:** Indicates if the value is updateable or read-only to the user.

From the point of view of an Administrator that wants to deploy an application, the goal is to be able to adapt and then insatiate an application that exists on the Service Store. Adaptation parameters of an application should be offering the ability to change and affect the overall installation of the application.

STRATEGIC Service Store offers the needed graphical interface for configuration of these adaptation parameters. An example of this interface is shown in Figure 11 and it is in fact an input form that is generated based on the input that the developer who published the application gave, as shown in Figure 10.

Figure 11: Setting parameter values through the metadata toolkit

5.2 Localization tools

Aiming to provide country specific adaptation on Service Store, the Infrastructure Provider (IP) registry tool, which has been released within the OPTIMIS toolkit [3], is being integrated with the Service Store. The Service Store gains the ability to filter the target providers to guarantee that the infrastructures provided to public bodies complies with their legal constraints and their key data protection considerations.

STRATEGIC within the service metadata tries to model the parameters that describe the constraints and restrictions that public bodies have in order to take into consideration their data protection requirement as well as their localization needs. The system considers these restrictions before taking the decision of deployment over a target infrastructure provider as verification of compliance based on the providers regional information.

The IP registry is a standalone web service that may run over an application server like tomcat, it can be accessed through a REST API to filter the target provider that do not fulfil the localization requirements specified at service construction time.

The following calls have been identified as related IP Registry, but as the source code is available further expansion is possible.

Operation	Input	Output	Description
registerIP	Provider information	Response code	Allows to register a provider.
getAllIP	None	List of providers	Allows to get all IPs in the registry.
getdetails	Provider	Provider details	Allows to get information per provider
getIP	Service constraints	List of providers	Allows to filter the target providers that fulfil the constraints.

Table 4: IP Registry calls related to Service Store

The following Figure 12 provides a high-level overview of the interaction between the registry and the Service Store:

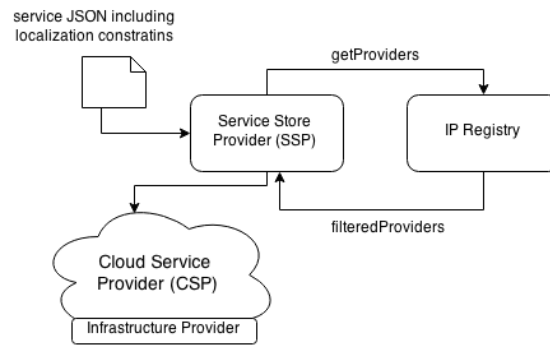


Figure 12: Interaction between IP Registry and Service Store

For the users, STRATEGIC Service Store offers the needed graphical interface for configuration of the legal constraints and also provides filtered results on the available infrastructure that can be used, based on the legal constraints defined. An example of this interface is shown in Figure 13. Please note, that IP Registry integration is still on-going at the moment of writing of this document and might not be available in the public version.

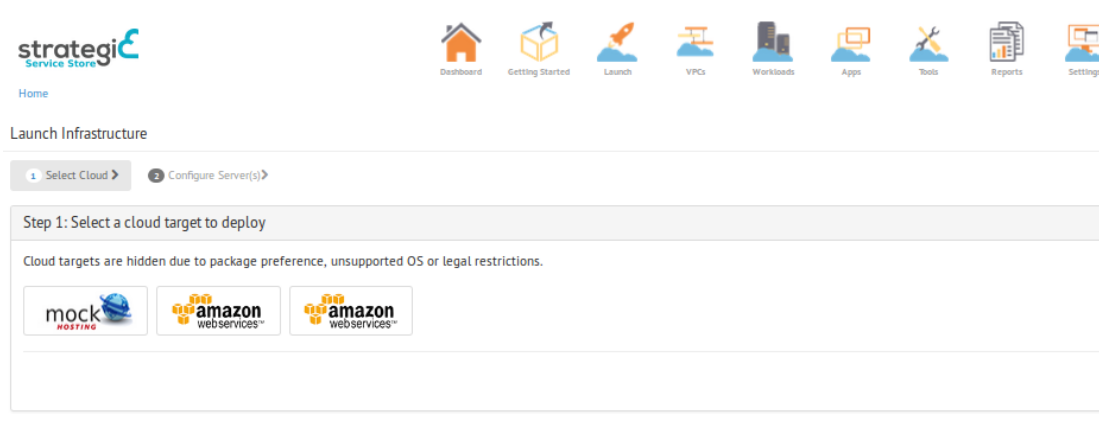


Figure 13: Filtered results based on local needs

6 Governance tools

STRATEGIC offers governance tools that allow the governance of deployed Services and Workloads through the Service Store. As defined in deliverable D2.3 [1] Governance Component is responsible for managing the state of the deployed applications on the infrastructure. It provides functionality to “activate” a deployed application, change its running state and “deactivate” the deployed application. Also Governance Component provides the functionality of configuration of settings and policies of deployed applications. The components of Governance Component as defined in deliverable D2.3 are depicted in Figure 14.

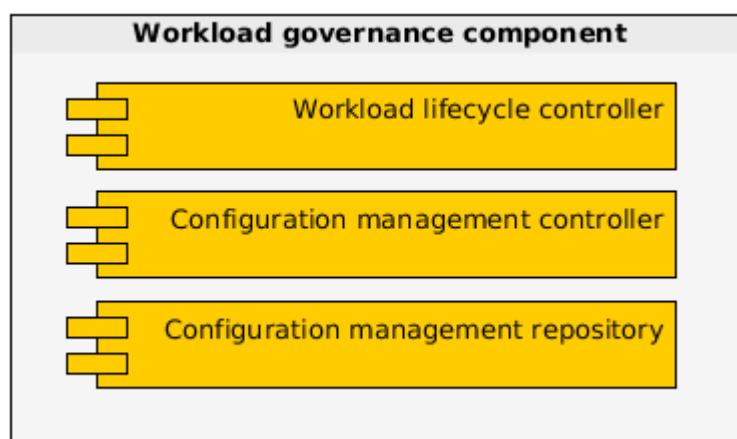


Figure 14 – Governance Component components

By using BT Service Store as a base platform, STRATEGIC Service Store offers the tools and functionalities needed for modifying the state of the application after provisioning, and introduces the following entities of that can be managed:

- **Workload**
A workload represents a collection and logical grouping of virtual servers, load balancers, security groups, applications that run on a specific cloud provider and location.
- **Tier grouping**
Tiers represent a logical grouping of application services running within a Workload.
- **Server**
Servers represent a compute resource running within a workload.
- **Application**
Applications that are available in Service Store and have been deployed to a virtual server in a Workload.
- **Load Balancer**
Load Balancers that allow loading balance servers within a workload.

Among them the basic entities are the Workload, the Server and the Application, and their status is displayed in the overview page of Service Store (Figure 15).

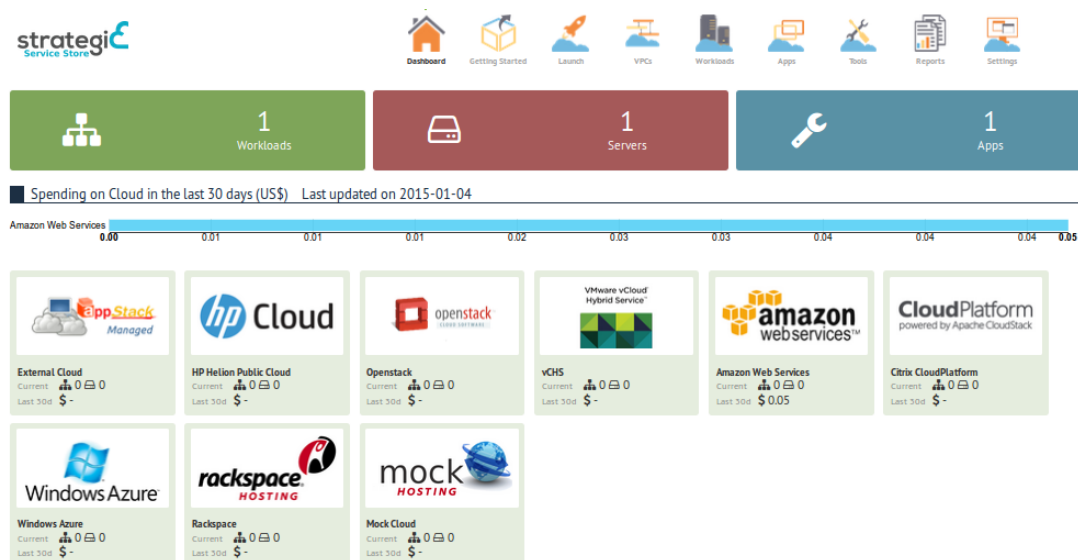


Figure 15 – Basic entities governed by Service Store, as shown in graphical interface

The functionalities that are supported and manage the state of a workload are described in Table 5.

Functionality	Description
Duplicate an entire workload	Duplicate an entire workload
Delete a workload	Delete a workload
Manage – configure Tiers	Managing a group of application services that run within a Workload
Add new servers	Adding new servers on a specific workload
Regroup VMs for management purposes	Regrouping of VMs is possible for management purposes
Start / Stop an entire VM	Modification of the state of a Virtual Machine
Start / Stop Application on its own	Modification of the state of the application
Manage - Reconfigure Application settings	Management of settings of a deployed application
Port-forwarding Management	Management of port forwarding rules
Add load balancing rules	Management of load balancing of the workload

Table 5: Supported Core Functionalities of Governance Toolkit

All the governance functionalities are offered through graphical user interface. The form of managing a server is depicted in Figure 16.

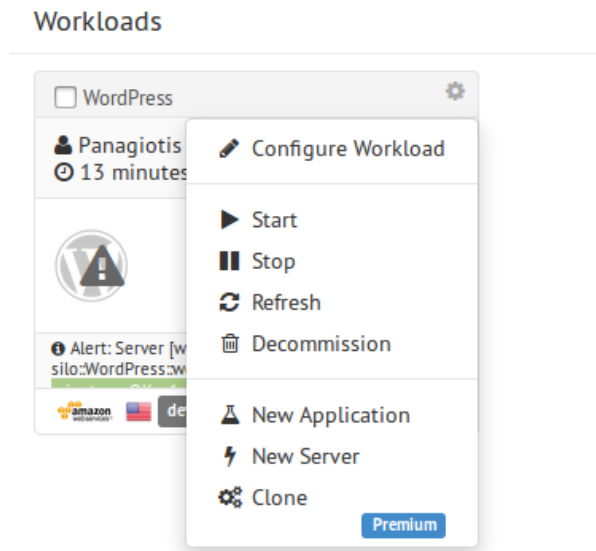


Figure 16 – Managing a server through graphical interface

It is possible to delegate governance within an organisation by using access management tools offered by STRATEGIC Service Store. This is useful for avoiding using a shared account and connected risks.

7 Conclusions

This document contains description of tools and methods for performing typical operations of the target users of STRATEGIC.

The tools are split into 4 sections:

- Cloud-enablement - for moving applications to the cloud;
- Migration – for moving cloudified application across clouds;
- Adaptation and localization – for providing customization options for the application template;
- Governance – for managing provisioned application;

In addition, a section on the tool support for enabling inclusion of a new IaaS system was provided. This allows removing the vendor lock-in on the IaaS level and ability to include local service providers that can potentially be adhering to geographical or legal requirements of the applications.

These sets of tools comprise a STRATEGIC toolkit that can be used by WP5 for performing activities related to pilot operations. While this toolkit is not final, it covers basic requirements of the pilots for the whole lifecycle of the applications.

The second iteration of this deliverable will include extended and refined tools based on the experience and feedback from the WP5 activities.

References

[1] STRATEGIC Deliverable D2.3 Strategic Framework Architecture Technical Specification, 2014

[2] STRATEGIC Deliverable D3.1 Specification of Cloud-Enablement and Migration Solutions and Services, 2014

[3] Optimis D7.2.1.3 Cloud Legal Guidelines, 2012, <http://www.optimis-project.eu/sites/default/files/content-files/document/d7213-cloud-legal-guidelines.pdf>