



Project Acronym: **OPTIMIS**
Project Title: **Optimized Infrastructure Services**
Project Number: **257115**
Instrument: **Integrated Project**
Thematic Priority: **ICT-2009.1.2 – Internet of Services, Software and Virtualisation**

Virtual Machine Contextualizer User Guide

Activity 3: Service Deployment

WP 3.1: Service Deployment Optimization

Due Date:	M34	
Submission Date:	31/03/2013	
Start Date of Project:	01/06/2010	
Duration of Project:	36 months	
Organisation Responsible for the Deliverable:	University of Leeds	
Version:	1.0	
Status	Final for submission	
Author(s):	Django Armstrong	ULEEDS
Reviewer(s)		

Project co-funded by the European Commission within the Seventh Framework Programme

Dissemination Level

PU	Public	X
PP	Restricted to other programme participants (including the Commission)	
RE	Restricted to a group specified by the consortium (including the Commission)	
CO	Confidential, only for members of the consortium (including the Commission)	



Version History

Version	Date	Comments, Changes, Status	Authors, contributors, reviewers
1.0	2012-04-13	Final for submission	Django Armstrong, Django Armstrong
1.1	2013-03-15	Final for submission	Django Armstrong, Django Armstrong



Table of Contents

1	INTRODUCTION	6
1.1	GLOSSARY OF ACRONYMS.....	6
2	VIRTUAL MACHINE CONTEXTUALIZER USER GUIDE	7
2.1	RELEASE INFORMATION	7
2.2	INTRODUCTION.....	7
2.3	FUNCTIONALITIES.....	7
2.3.1	<i>Functionality 1 – Deployment time contextualization on of Virtual Machines.....</i>	<i>7</i>
2.3.2	<i>Functionality 2 – Runtime Contextualization Mechanism</i>	<i>12</i>
2.3.3	<i>Functionality 3 – Runtime Recontextualization Mechanism.....</i>	<i>13</i>
2.4	KNOWN LIMITATIONS	13
2.4.1	<i>Limitation 1.....</i>	<i>13</i>
2.5	GETTING STARTED.....	13
2.5.1	<i>Contributors.....</i>	<i>13</i>
3	REFERENCES	14

Index of Figures

No table of figures entries found.

Index of Tables

No table of figures entries found.



1 Introduction

This document includes the user guide for the software component Virtual Machine Contextualizer.

1.1 Glossary of Acronyms

Acronym	Definition
D	Deliverable
WP	Work Package
VMC	Virtual Machine Contextualizer

2 Virtual Machine Contextualizer User Guide

2.1 Release information

Component Name	Release Number	Release Date
Virtual Machine Contextualizer	3.0	2013-03-31

2.2 Introduction

The Virtual Machine Contextualizer (VMC) gathers and aggregates context data from multiple sources for the purpose of configuring a Virtual Machine (VM) with a cluster of virtual resources. The VMC has the following features:

- 1) Ability to insert context data into a VM
- 2) Prepare VM for receipt of context data.

2.3 Functionalities

2.3.1 Functionality 1 – Deployment time contextualization on of Virtual Machines

2.3.1.1 Internal Architecture

The VMC uses the Service Manifest from the SD/DO to set the context of one or more virtual machine at runtime. This involves the VMC assembling data from several sources. A local file system based image repository is used to access a VM image. If the VM image is of the wrong type for a selected IP hypervisor it will be converted. In addition, the image is mounted and modified to include initialization scripts that interact with the guest operating system at boot time, preparing it to receive context in a reusable fashion. After the image has been prepared, it and its associated ISO CD images containing contextualization data are stored back in the image repository for later propagation to an IP.

The lightweight initialization scripts embedded in a VM image bootstrap to additional scripts held within an ISO CD ROM image mounted within the guest operating system. At run time the bootstrapped scripts access contextualization data stored within the ISO CD ROM image, giving a VM an identity.

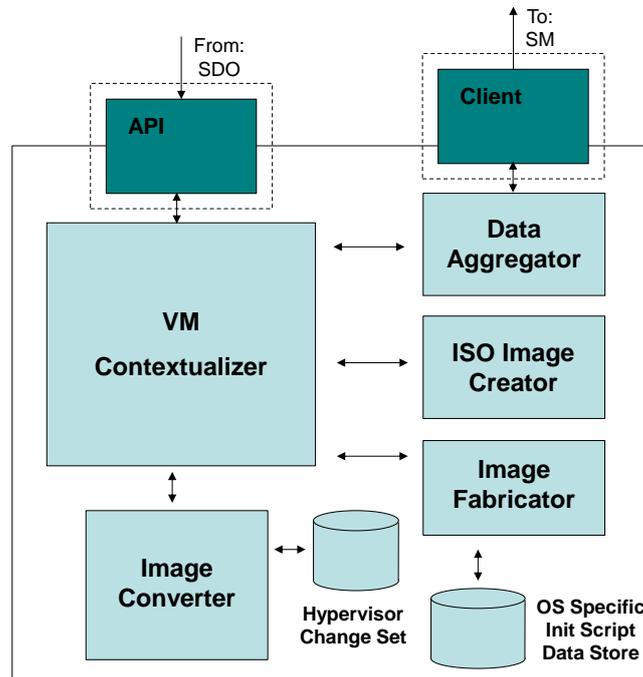


Figure 1: VM Contextualizer Component Level Diagram

The inclusion of the ISO image as a mechanism to store contextualization information provides a facility to separate the contextualization data from the VM image. This removes the need to create multiple unique VM images for each VM that is required to be contextualized, while also improving the security of the contextualization process as PKC certificates and keys are not stored in the VM image itself, rather the ISO image.

2.3.1.1.1 Sub-component – ISO Image Creator

The ISO Image Creator sub-component manipulates a set of template ISO images. In addition, this sub-component is responsible for the cloning of ISO images and the embedding of context data from the Data Aggregator. The ISO Image Creator has access to a single set of context data processing scripts that are agnostic of a VM’s operating system, embedded in the ISO image and responsible for reading context data from the ISO image at run time. An internal interface provides access to contextualization scripts embedded in an OPTIMIS enabled VM image.

2.3.1.1.2 Sub-component – Image Fabricator

The Image Fabricator manipulates VM images, passed to it by the VM Contextualizer, using Linux system tools. The sub-component installs OPTIMIS VM level components. In addition, software dependencies of a service can be embedded into the image, if the VM Contextualizer supports them. If this is not the case it is assumed that the Service Developer will have already setup the context of the service’s software dependencies within the VM image provided to the VM Contextualizer.

A set of prefabricated operating system specific contextualization scripts, from a data store, are embedded for each operating system with foreseen use at the IP level. The contextualization initialization scripts provide two main functions. The first is to set the context of OPTIMIS components that facilitate running a service in addition to the configuration of the Operating System. This involves setting the context of:

- The OPTIMIS Data Management tools at the VM level and any associated software dependencies, which assist in mounting block stores in a VM at runtime
- The OPTIMIS VPN management component, to enable the VM to communicate with other VMs on a VPN.
- The Monitoring Probes, enabling monitoring data of non-functional service KPIs present in a SLA as requirements to be sent to the SP.
- The License Manager, to enable automated registration of license tokens with software dependencies at the point of VM execution (Offline) or during run time (Online).
- The VM's NIC attached to the IP's Infrastructure when DHCP is unavailable, configured with an appropriate Internet Protocol address by decoding the MAC address of the NIC set by the IP. This contextualization mechanism requires no input from the SDO. The MAC address is encoded by the VM Manager so that the first four groups of two hexadecimal digits define the default gateway address once converted to decimal format. The last two groups of hexadecimal digits define the last two segments of the VM's address on a Class A or B network. Combining the two, the VM address can be obtained by the contextualization scripts. The following is a worked example:
 - MAC address: **0C-A8-01-01-02-0A**
 - Gateway address: **192.168.1.1**
 - VM address: **192.168.2.10**

The second function of the contextualization scripts is to provide context to the software dependencies of a service. Inclusion of these scripts is optional and is dependent on what software OPTIMIS supports.

2.3.1.1.3 Sub-component – Image Converter

The Image Converter transforms images from one format to another for the purpose of supporting interoperability between IPs using dissimilar hypervisor technology. The Image converter makes use of “qemu-img” to convert between image standards. Depending on the hypervisor and operating system, conversion can require the changes made by the Image Fabricator to be reverted and reapplied. The following Hypervisors are envisaged to be supported:

- Xen
- KVM,
- VMware
- VirtualBox

In addition the subsequent image formats will be supported:

- Raw
- QCow2
- VDI
- VHD
- VMKD

Besides support for multiple image formats the Image Converter will support Copy on Write and fully encrypted images where applicable.

2.3.1.1.4 Sub-component – Data Aggregator

The Data Aggregator is responsible for the gathering of configuration data from multiple sources. The component parses the Service Manifest for the following contextualization requirements:

- License Management: License server end-point on a per-service basis and/or SM embedded license tokens on a per VM instance (or per VM-type?) basis.
- Programming Model: SSH keys needed and software dependencies on a per VM-type basis.
- Data Management: DM server end-point for mounting VM level block storage on a per-service basis.
- Security Management: Specification of the VPN network topology associated with VM-type.
- Monitoring: Probe end-point for reporting KPIs to.

IP specific attributes (like the DM endpoint) are sent to the SP during the last phase of SLA negotiation. In addition to parsing the Service Manifest, the Data Aggregator calls the Security Manger to generate keys and certificates on-the-fly.



2.3.1.2 Sequence diagram

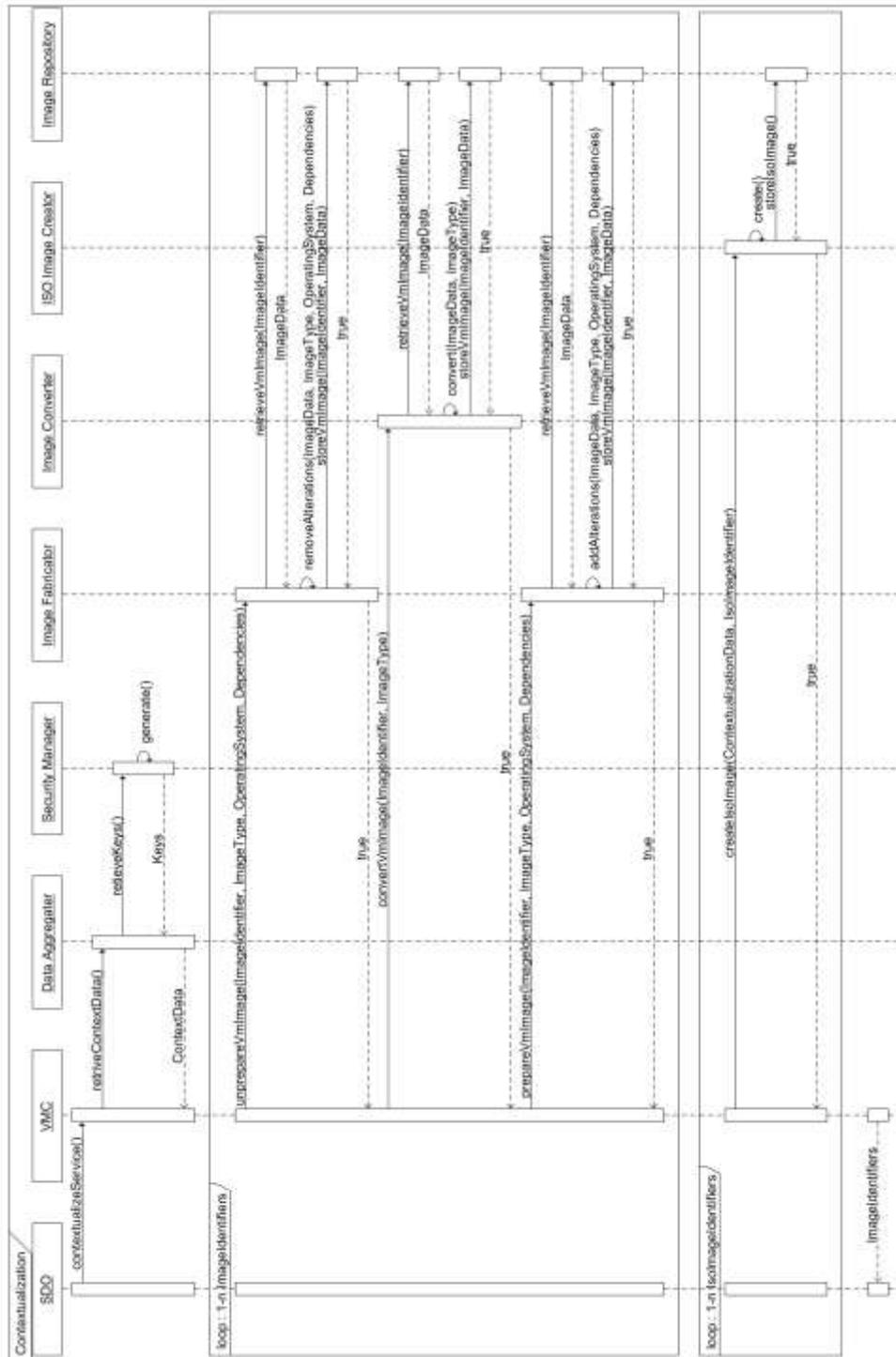


Figure 3: Sequence Diagram of interaction between OPTIMIS components and the VM Contextualizer

Figure 3 describes the interactions of the VM Contextualizer with its sub components and other components in the OPTIMIS Toolkit at the SP level. From the sequence diagram, it can be seen the SDO initializes the contextualization of a service by supplying the Service Manifest to the VMC. The VMC then calls the Data Aggregator to parse the service manifest and call the Security Manager to generate keys. The contextualization data is then used by the VMC to call the Image Fabricator to prepare a VM image for receipt of this data at runtime.

When conversion of a VM image is required the Image Converter is called and if the image has been prepared previously by the VMC, the Image Fabricator is invoked twice, before and after conversion to un-prepare and then prepare again the image for receipt of contextualization data. Calls to the Image Fabricator and the Image Converter access the local file system based Image Repository. After all VM images have been prepared and/or converted the VMC invokes the ISO Image Creator to create ISO CD image files containing contextualization data on a per instance basis. The resulting files are then stored in the Image Repository. Finally the VMC returns a list of Image Identifiers represented as URIs (both VM and ISO) to the SDO that correspond to the contextualised service.



Figure 4: Sequence Diagram of VMC progress callback to the SDO

To enable the SDO to monitor the progress of the VMC and present this to an End-User via a GUI, the VMC provides a single callback (Figure 4) which returns the phase of contextualization and the percentage completion of the phase. The phases returned in order are:

- Parsing Manifest
- Gathering Contextualization Data
- Decomposing VM Image(s)*
- Converting VM Image(s)*
- Fabricating VM Image(s)*
- Generating ISO Images

Phases marked with an asterisk are considered optional.

2.3.2 Functionality 2 – Runtime Contextualization Mechanism

Figure 2 illustrates the contextualization process of a VM at the beginning of its execution. During the boot sequence of a VM the contextualization tools mount an ISO CD image that contains contextualization data and scripts to processes this data into a useable form.

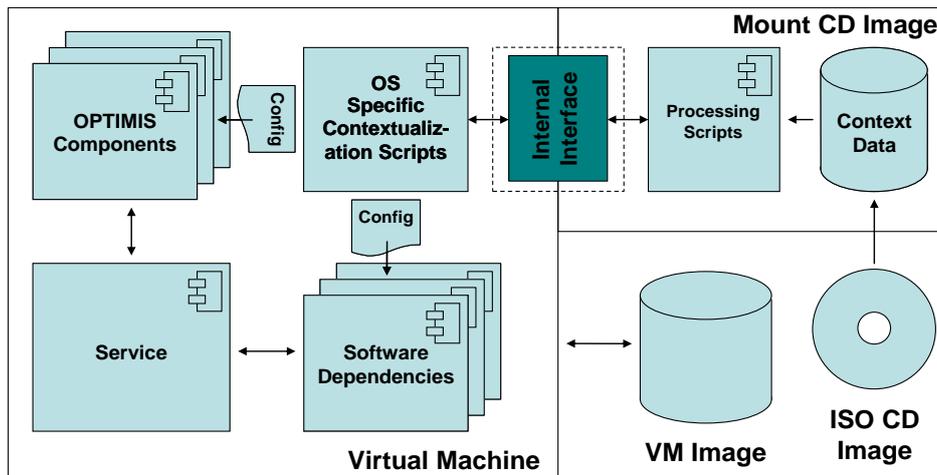


Figure 2: Interaction between VM image and ISO Image at run time

These scripts communicate through a known interface to the OS specific contextualisation scripts embedded in the VM image. These OS specific initialization scripts manipulate configuration files of associated OPTIMIS components and software dependencies that support the End User's service(s), setting their identity (context) within a component cluster. If the need arises the scripts can remain in a daemon like mode if a component or software dependencies requires continuous update to its context.

2.3.3 Functionality 3 – Runtime Recontextualization Mechanism

Experimental Recontextualization of a VM is provided in this release as a proof of concept. On migration of a VM from one host to another, Recontextualization is triggered via a standalone version of the VMC running at the host level using the Libvirt API as a layer of abstraction to change the ISO cdrom device. For further details on this mechanism please refer directly to the source code of `VmcApiRecontextTest.java` and `VirtualMachineRecontextualizer.java`.

2.4 Known limitations

These limitations are known to exist in this release of the software:

2.4.1 Limitation 1

The preparation of Virtual Machine images is currently performed outside the scope of this component by the OPTIMIS ICS component.

2.5 Getting Started

This component can be used as a java library or standalone.

User guide can be found at

https://bscw.scai.fraunhofer.de/bscw/bscw.cgi/283640/DeploymentOptimizer_User_Guide.doc.

2.5.1 Contributors

Django Armstrong

3 References

- [1] QEMU – A generic and open source machine emulator and virtualizer, Website - <http://wiki.qemu.org>
- [2] Kpartx – A tool for creating device maps from partition tables, Website - <http://linux.die.net/man/8/kpartx>
- [3] Website – A tool for creating a hybrid ISO9660/JOLIET/HFS filesystem with optional Rock Ridge attributes, - <http://linux.die.net/man/8/mkisofs>