OPEN DATA CENTER ALLIANCE℠ USAGE: VIRTUAL MACHINE (VM) INTEROPERABILITY IN A HYBRID CLOUD ENVIRONMENT REV. 1.2

TABLE OF CONTENTS

Contributors 2

Legal Notice 3

Executive Summary 4

Purpose 5

Management 6

Taxonomy 7

Interoperability 8
- Usage Scenario Mapping 8

Usage Model Diagram 9

Usage Model Details 9

Usage Scenarios 10
- Usage Scenario 1 - Check Interoperability 10
- Usage Scenario 2 - Move or Copy between Two Cloud Environments 11
- Usage Scenario 3 - Leverage Common Operation/Interoperability 11

Life Cycle Model 12
- VM Life Cycle Model 12
- Interoperability Life Cycle Model 12
- Burst or Migration Scenario 13

Usage Requirements 15

RFP Requirements 16

MoSCoW Requirements 16

Summary of Required Industry Actions 18

CONTRIBUTORS

Alan Clarke, SUSE
Arivou Tandabany, Telstra
Avi Shvartz, Bank Leumi
Axel Knut Bethkenhagen, BMW
Ben MP Li, Deutsche Bank
Bernd Henning, Fujitsu
Catherine Spence, Intel
Eric Kristoff, ODCA Infra Workgroup
Erik Rudin, Science Logic
Erick Wipprecht, Disney Corporation
Fred Oliveira, Terremark
Geoff Poskitt, Fujitsu
Hans van de Koppel, Capgemini
Mick Symonds, Atos
Mustan Bharmal, T-Systems
Peter Pruijssers, Atos
Ravi A. Giri, Intel
Ryan Skipp, Deutsche Telekom
Stephanie Woolson, Lockheed Martin
Vince Lubsey, Virtustream

LEGAL NOTICE

OPEN DATA CENTER ALLIANCE℠ USAGE: VIRTUAL MACHINE (VM) INTEROPERABILITY IN A HYBRID CLOUD ENVIRONMENT REV. 1.2

EXECUTIVE SUMMARY

Ideally, cloud subscribers would like to select any cloud provider based on cost and performance/capabilities. They also want to link, based on need, private clouds made up of dedicated services with public clouds that consist of shared services. In order to make this feasible for the cloud consumer, the various hypervisor platforms and virtual machines (VMs) involved will need to be interoperable and portable.

Furthermore, consistent management interfaces would help considerably in enabling Interoperability among cloud environments. A number of standards and approaches can be considered to enable interoperation, which could include a standard format for machine packaging, catalog translation between clouds, and broker based translation leveraging proprietary agents, amongst others.

Managing systems, networks, and storage is already a complex endeavor. The addition of virtual resources—which are the foundation of cloud services—adds yet another layer of complexity. Challenges increase even further as workloads or virtual machines (VMs) cross the boundaries between data centers. These complexities do not come cheap. Between 2010 and 2014, Bain & Company estimates that IT organizations will spend up to $2 trillion in deployment and operations unless management practices can be automated and simplified[1].

The Open Data Center Alliance℠ (ODCA) recognizes the need for interoperable cloud environments and management solutions that incorporate standard mechanisms to create consistencies across all VMs. Interoperability is viewed from two perspectives of interconnectability and portability, as follows:

- Interconnectability—the parallel process in which two coexisting environments communicate and interact.
- Portability—the serial process of moving a system from one cloud environment to another.

This Usage Model recommends actions and processes to spur development of interoperable solutions aimed at lowering management complexity and costs, especially in heterogeneous, multi-vendor environments.

This update to the VM Interoperability model addresses a number of important additional dimensions including extending the portability concept, extending the life cycle model with States and Conditions, and increased alignment with external work such as that of the Open Virtualization Format (OVF) specification.[2]

This document serves a variety of audiences. Business decision makers looking for specific solutions and enterprise IT groups involved in planning, operations, and procurement will find this document useful. Solution providers and technology vendors will benefit from its content to better understand customer needs and tailor service and product offerings. Standards organizations will find the information helpful in defining standards that are open and relevant to end users. Regulators and auditors can use this document to review and monitor solutions and services, especially in focused areas supporting the growth of cloud computing.

[1] See www.opendatacenteralliance.org/phocadownload/odca-vision.pdf

[2] See the DMTF web site: www.dmtf.org/standards/ovf

PURPOSE

Well-defined VM Interoperability significantly reduces the risks and effort when working with different hypervisors. It greatly simplifies the complexity of handling multiple cloud platforms and minimizes the issues of managing workloads that are hosted on internal cloud platforms or across several different public and private cloud platform offerings. Actions could be clearly defined in terms of prerequisites, implied sub-activities, and the possible interoperability outcomes.

The business drivers for VM Interoperability are as follows:

- Application Migration: Ability to move applications from one discrete set of VMs from one cloud to another, available from the same or different cloud provider with minimal effort (for reasons of cost efficiency, functionality, service levels, etc.).
- Extend Private Cloud: Add additional computing resources seamlessly to an on-premise, enterprise private cloud environment; easily manage and move workloads among on-premise and public cloud provider environments (for reasons of short demand fulfilment, defined term projects, community functionality and sharing, etc.).
- Business Continuity: Migrate or replicate applications among providers to address outages, security breaches, or other disruptions. This is intended to encompass both disaster recovery and disaster avoidance.

As the feature sets of the available hypervisors vary, there should be a common command set or APIs (including functions such as Create, Start, Stop and Suspend, amongst others), which all hypervisors have to provide. Additionally, the support of differentiating features has to be ensured and constantly reviewed to determine if any of these features have become standard practice and, therefore, are required to be part of the common command set.

Consistent management is necessary for any virtualized environment. Ensuring interoperability of hypervisors will allow all vendors to easily develop interoperable management solutions that lower management complexity and cost, especially in a heterogeneous, multi-vendor and multi-cloud environment. For example, by supporting certain virtualization management standards, such as those being proposed by the Distributed Management Task Force (DMTF), VMs and their deployments can be managed in the same manner, independent of vendors. The entire virtualized environment can then be managed from a single management console.

This Usage Model extends the Compute Infrastructure as a Service (ClaaS) Master Usage Model which serves as the framework for ClaaS to be evaluated, acquired, and disposed of by enterprises in a way that reflects the ODCA member firms' vision of a robust and vibrant market by the end of 2014.

This Usage Model also assumes the compliance of involved hypervisor technology at both source and target locations with the DMTF Open Virtualization Format (OVF) specification as a basis. Machine packaging (according to the OVF specification) is one element of interoperability–negotiation of configuration between source and target locations (which may not be perfectly similar) is also necessary. Therefore, close examination of the actual catalogs of service elements must be considered, between source and target locations, to determine if there are any differences which may cause the OVF-based package to stumble. This enables identification of potential differences in standard configurations between the cloud platforms, which must be considered for dispensation. The Cloud Infrastructure Management Interface (CIMI) specification supports detailed catalog definition to the necessary level for the IaaS portion of cloud services. Additionally, there are a number of Cloud Provisioning standards to consider: for example, SPML (Service Provision Markup Language), SCIM (System for Cross Domain Identity Management–from OASIS); and then the integration of other tooling layers which handle configuration management: for example, CHEF or Puppet. For additional related information, see the Open Data Center Alliance[SM] Usage: Carbon Footprint Values Rev. 1.1[3].

[3] See www.opendatacenteralliance.org/ourwork/usagemodels

MANAGEMENT

VMs, and the hypervisors they are deployed on, need to be managed in a consistent way, amongst providers.

To be successful this means that VMs have to be "manageable," and the (cloud) management platforms need to be able to manage them, inter-operate them, across the different hypervisors and different clouds. For example, the portability of an OVF can be categorized into the following three levels:[4]

- Level 1: Only runs on a particular virtualization product and/or CPU architecture and/or virtual hardware selection. It is logically equivalent to a suspend in the source environment and a resume in the target environment. A live migration is possible at Level 1. However, Level 1 carries a number of operational restrictions, such as the preservation of IP addresses, limiting the applicability to virtual machines running in the same subnet and hypervisor.
- Level 2: Runs on a specific family of virtual hardware. Migration under Level 2 is equivalent to a shut-down in the source environment followed by a reboot in the target environment. Movement across different hypervisors is possible.
- Level 3: Runs on multiple families of virtual hardware. This is the most general framework for VM migration offering the greatest flexibility, essentially allowing a machine to be rebuilt to suit the target environment. This assumes advanced methodologies, such as integrated development and operations (DevOps) not in common practice today.

Aspects that need to be considered here when **creating VMs or standards for packaging** are:

1. Risk reduction (i.e., by means of certification and integrity mechanisms)
2. Efficiency
3. Operator error reduction
4. Portability/transport
5. Versioning
6. Security
7. Networking
8. Script type compatibility (for example, from VMware's vFabric Application Director)
9. Descriptions to be used by Cloud Service Catalogs and Brokers
10. Properties to be used by Service Management
11. Properties that indicate how they can be bundled
12. General extensibility
13. Localizable (local and geo-political compliance)/Internationalization
14. Vendor and platform independency
15. Configuration and modification
16. Vendor lock-in prevention
17. Virtual hardware aspects: What storage adapter is used etc.? Is this understood by the hypervisor attempting the installation?
18. Readiness for migration, including state definitions
19. Responsibility transfer point

[4] from the DMTF Feedback OVF, and referring to OVF WP DSP 2017_1.0.0: www.dmtf.org/standards/ovf

Practical implications for interoperating **management platforms** are necessary for the list below, and the capabilities for handling the aspects in the previous list of Aspects.

1. Handling of standard formats like the OVF packaging format for software appliances
2. Inspection and enrichment of the standardized formats
3. Provenance determination
4. Handling of migration implications
5. Resource reservation – ability to request and reserve resources
6. Cost model handling
7. Licensing mechanism for transferring licenses between providers and clouds
8. Responsibility management

TAXONOMY[5]

Actor	Description
Cloud Subscriber	A person or organization that has been authenticated to a cloud and maintains a business relationship with a cloud.
Cloud ISV	Independent Software Vendor, selling and supporting software like CMP software (Cloud Management Platform).
Source Cloud Provider	An organization providing cloud and network services and supplying services to cloud subscribers. A (public) cloud provider provides services over the Internet. The source cloud provider currently provides production-level services to the cloud consumer for this paper.
Target Cloud Provider	An organization providing cloud and network services and supplying services to cloud subscribers. A (public) cloud provider provides services over the Internet. The target cloud provider will provide additional or new production-level services to the cloud consumer for this paper, taking over services and/or adding capacity to extend or take over services from the source cloud provider.
Cloud Compliance Agency	An accredited entity that is responsible for ensuring compliance to cloud security standards. A Cloud Compliance Agency may also be a third party trusted by the cloud subscriber. They could then determine and monitor the security state of the cloud provider and respond to the cloud subscriber when requested.
Cloudbursting	Gartner defines **Cloudbursting** as "the use of an alternative set of public or private cloud-based services as a way to augment and handle peaks in IT system requirements at startup or during runtime. Cloudbursting can span between on-premises IT systems and services and the cloud, across multiple cloud providers or across multiple resource pools of a single provider. It can also be enabled across multiple internal data centers, across multiple external data centers, or between internal and external data centers."

[5] See www.opendatacenteralliance.org/index2.php?option=com_productsearch&view=lightbox&proid=14

INTEROPERABILITY

Interoperability perspectives are the basic categories that define the scope into which the business usages fall. Interoperability perspectives follow:

- Interconnectability —the parallel process in which two coexisting environments communicate and interact.
- Portability—the serial process of moving a system from one cloud environment to another.

Both of these demand provisions in the layers beneath the target layer to ensure success: for example, whether to interconnect or move an application, consideration has to be given to any PaaS or middleware layer, plus the operating system and possibly hypervisor. Network and storage conditions will also require consideration.

Interconnectability will require that a complete transportation path be present, and sufficiently open, between both environments, i.e., sufficient bandwidth, mutual addressability, shareable storage, and routes through firewalls and/or DMZs.

Portability may require such connecting facilities during any act of movement itself, but essentially the requirement is simply that a sufficiently similar environment exists at each end. The target environment is itself likely to include the ability to connect to any other applications, from the target environment just as was prevalent in the originating (source) environment.

Thus, although it may appear that these are two discrete meanings, in any practical deployment it will be seen that there are relationships and interdependencies between them. For instance, these days applications rarely run as complete "islands," but have connections with other applications, for instance throughout a logistics chain. During and after the movement from one location to another (portability), these connections must continue to function (interconnectability) such that the business process is not excessively disrupted.

Usage Scenario Mapping

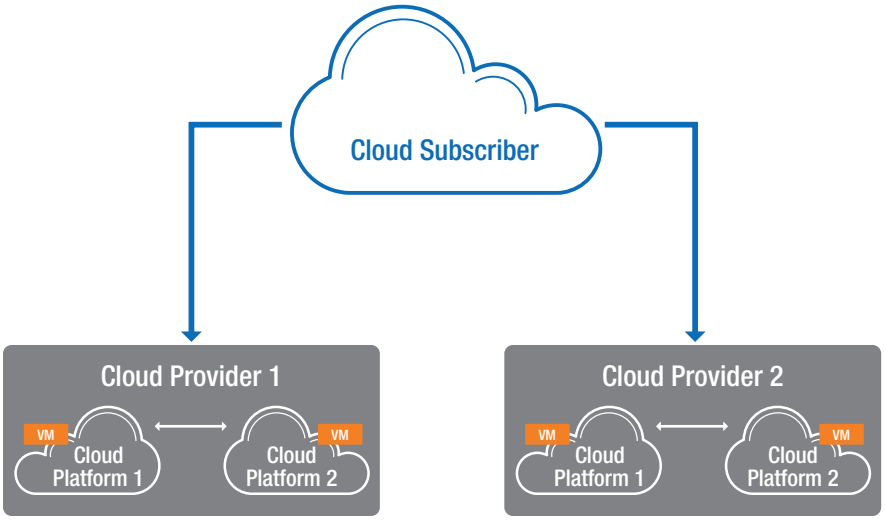
For each of the purposes identified in the "Purpose" section of this document, usage scenarios can be identified, aligning to the PaaS Interoperability perspectives. One or more usage scenarios may apply to the purpose within a perspective.

The described usage scenarios are:

- Usage Scenario 1: Check Interoperability
- Usage Scenario 2: Move or Copy between Two Cloud Environments
- Usage Scenario 3: Leverage Common Operation/Interoperability

Business Driver	Interconnectability	Portability
Application Migration	Usage Scenario 1: Check Interoperability	Usage Scenario 2: Move or Copy between Two Cloud Environments Usage Scenario 3: Leverage Common Operation/Interoperability
Extend Private Cloud	Usage Scenario 1: Check Interoperability	Usage Scenario 2: Move or Copy between Two Cloud Environments Usage Scenario 3: Leverage Common Operation/Interoperability
Business Continuity	Usage Scenario 1: Check Interoperability	Usage Scenario 2: Move or Copy between Two Cloud Environments Usage Scenario 3: Leverage Common Operation/Interoperability

USAGE MODEL DIAGRAM



USAGE MODEL DETAILS

Running a complex workload in a Hybrid Cloud environment is a significant undertaking which should become far simpler over time with improving standards (and their adoption as technologies develop), allowing for open solutions that minimize complexity and lower cost. This Usage Model (see diagram above) focuses specifically on running simple Compute IaaS workloads of a few VMs across multiple hypervisors in a Hybrid Cloud model. Future iterations and Usage Models will delve into the specifics around complex workloads including detailed networking and storage components, and the expansion of policy-based automation to control the status and usage of cloud providers programmatically. For the sake of this Usage Model, it should be clear that the cloud provider in many situations will be the Enterprise IT team facilitating and providing services to their software developers and users, as cloud subscribers.

USAGE SCENARIOS

Usage Scenario 1 - Check Interoperability

Determine if a move/migration of a VM of a cloud subscriber from one cloud provider to another, or to a different hypervisor within the same cloud provider, is possible, and the potential limitations for the move, such as, "Can the move/migration be live?" An understanding of all of the requirements for moving between hypervisors and cloud providers should be clearly detailed.

Assumptions:

1. The cloud provider implements the Open Data Center Alliance Service Catalog Usage Model.
2. Technical details of the current workload are known:
   - Number of VMs
   - Amount and type of memory
   - Amount and type of CPU core
   - Amount and type of network interface card (NIC)
   - Amount and type of disk
   - Amount of I/O required
   - Type and vendor of hypervisor
   - Backup services
   - Firewall policies and rules
   - Load balancing services
   - Logical network topology including AD, LDAP, DNS, DHCP elements
   - Administrator credentials
3. The cloud subscriber has access to all details about the Service Level Agreement (SLA), Operating Level Agreement (OLA), and any specific controls:
   - Availability
   - Specific functions such as remote copy, disaster recovery
   - Security root of trust
   - Consistency of Input/Output management (I/O controls)
   - Security and compliance (from compliance monitoring)
   - Carbon measurement
   - Geo hosting requirements
   - Licensing model and associated requirements
4. The cloud provider is asked programmatically, through a user interface, if they can fulfill the requirements defined by the stated technical and operational specifications.
5. The target cloud provider is less than 20 km to the current cloud provider so as not to warrant a long-distance workload migration.

Success Scenario 1:

The cloud provider can answer "Yes" to the question of whether the clouds interoperate. The cloud subscriber is able to (automatically) decide to move the workload.

Failure Condition 1:

1. The cloud provider does not understand the question, is not able to answer it, or is unable to provide any details pertaining to the question.
2. Interoperability is not a given.

Failure Condition 2:

The cloud provider is not able to fulfill some requirements but can define the differences. The cloud subscriber can make a choice on whether migration is possible. If the migration is possible, then the Interoperability can be created by following the goals discussed in "Usage Scenario 3 - Leverage Common Operation/Interoperability."

Usage Scenario 2 - Move or Copy between Two Cloud Environments

1. After successful completion of Usage 1 (Check Interoperability), the cloud subscriber will be able to reserve the required resources and complete the copy of or move an existing VM from cloud platform 1 (hypervisor A) to cloud platform 2 (hypervisor B) or from cloud provider 1 to cloud provider 2.
2. If performing a copy, the cloud subscriber will be able to bring VM B into a running state and start taking active workload and/or connections.
3. If performing a move, the cloud subscriber will be able to bring VM B into the running state to start taking active workloads and/or connections, and then shut down VM A.

Assumptions:

Usage Scenario 1 (Check Interoperability) has completed a successful return, showing that the source and destination VMs and/or cloud platforms are interoperable and therefore capable of handling a copy or move function.

Success Scenario 1:

The copy is completed successfully between cloud platform 1 and cloud platform 2, and both VMs are accepting active connections based on defined network configurations.

Success Scenario 2:

The move is completed successfully, VM A on cloud platform 1 is shut down successfully, and VM B on cloud platform 2 is able to start up and accept active connections.

Failure Condition 1:

The copy or move did not complete successfully, and therefore VM B is unable to enter the running state. The cloud provider will return an error code indicating the reason for failure. The cloud subscriber can then take corrective actions for subsequent retry of the operation.

Failure Condition 2:

The copy completed successfully; however, VM B is unable to enter the running state, due to various failure conditions. The Return Error code is expected, to allow for retry or failure event notice. VM A is not shut down.

Failure Condition 3:

The move is completed successfully, but VM A is unable to shut down. The Return Error code is expected, and usage scenarios should not continue with trying to bring VM B into an active running state until the issue is corrected.

Usage Scenario 3 - Leverage Common Operation/Interoperability

Ensure that all operational activities can be conducted with the same syntax across both VM A and VM B, or between two cloud providers. Operational activities could include: start VM, stop VM, increase CPU of VM, decrease memory of VM, snapshot of VM, and other runtime functions. These should be possible using either identical syntax or identical methods. For this usage, we give the example of increased CPUs on the VM; however, we expect success and failure conditions to be met for all runtime operational management functions of the VM.

Assumptions:

1. Usage 1 (Check Interoperability) has completed a successful return.
2. Usage 2 (Move or Copy) has been utilized where there are VMs running on either multiple cloud provider platforms or multiple hypervisors within a single cloud platform.

Success Scenario 1:

The command for increased CPU of VM can be called with the same syntax and/or API to multiple cloud provider platforms or to multiple hypervisor solutions. The command is called as each platform completes the request and provides a return message.

Failure Condition 1:

The command is submitted; however, cloud platform 2 is not able to understand the cloud subscriber's request for increased CPU of VM and returns the syntax error.

Failure Condition 2:

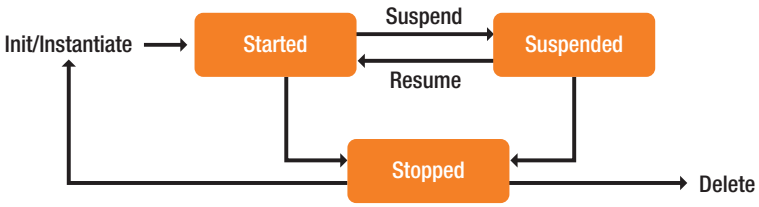
The command is submitted; however, cloud platform 2 does not take similar operational requirements into account, which can then impact the running operation of VM. For example, a previous requirement of "no reboot when CPU of VM has been increased" is not supported on cloud platform 2 and therefore an unexpected reboot occurs.

LIFE CYCLE MODEL

This section contains the VM life cycle model and the VM Interoperability life cycle model including transition states and conditions.

VM Life Cycle Model

For service deployment a VM has to be instantiated first, meaning that all necessary resources exist, and are able to run it. A unique ID is generated for further reference. A state for a VM implies a transitory position that a VM can be in, go to, or return from any other state. Hence, a started VM can be suspended or stopped, as depicted in the figure below.[6]



Prerequisites to start a VM include guaranteed resources for a defined timeframe (reservation period) and a defined set of parameters describing the VM (for example, number of cores, GB memory, OS flavor and version, network address and mask, necessary licenses) using the OVF. Starting and resuming a VM does not necessarily take place within the same infrastructure and/or cloud provider.

VM Status and Descriptions

VM Status	Description
Started	VM is running and usable by cloud subscriber
Stopped	VM is stopped. All configuration information is stored for possible restart.
Suspended	VM is suspended (suspend to disk [Advanced Configuration and Power Interface ACPI S4]).[6] If applicable, state of the VM is saved for resuming in the exact state (for example, check pointing). A suspended VM releases compute, memory, and network resources (but not storage) on the host machine/infrastructure platform.

Interoperability Life Cycle Model

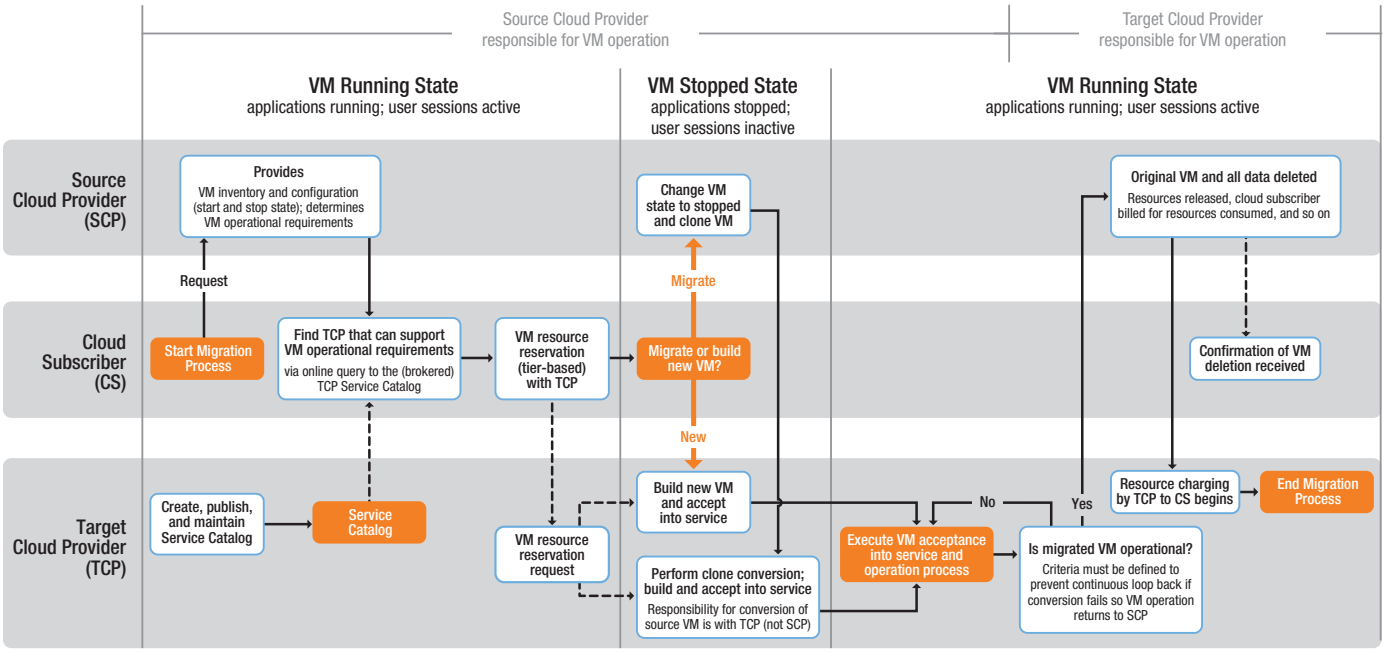
This section extends the VM life cycle model with the life cycle for VM Interoperability (see figure on the next page). The intent of this section is to provide a deeper technical description of the VM Interoperability in accordance with the usage scenarios defined earlier in this paper.

The standard flow description for VM Interoperability steps are described next:

1. VM is well defined by the cloud subscriber (IT role). A cloud subscriber may choose from a cloud subscriber's internal catalog of available VMs.
2. Cloud provider catalog is checked against desired VM configuration. (Usage Scenario 1)
   – Positive response (configuration and resources available) – reservation of resources at cloud provider
   – Negative response – check of next cloud provider catalog
3. Beginning of VM life cycle (Usage Scenario 2) at cloud provider. VM gets initialized and instantiated at cloud provider using the reserved resources. VM might be started by now. If cloud subscriber decides not to use this specific VM, delete process with release of all associated resources is started. A running VM might be suspended or stopped. Suspended VM might be started at previously saved suspended state (if possible) or stopped. Stopped VM may be reinitialized for restart or deleted. Deletion means release of all associated resources for that specific VM.

[6] The states of a VM in this life cycle model correspond to the Start/Stop cloud services used in the Standard Units of Measure (SUoM) documents where the actors (cloud subscriber and cloud provider) communicate with the purpose of creating and handling one or multiple VMs with the goal of a defined service for the cloud subscriber.

VM Migration to New Cloud Provider



Burst or Migration Scenario

In the case of full utilization or increasing utilization resulting in diminishing capacity of one cloud platform and/or the cloud subscriber's decision to use a different cloud provider, communication between cloud platforms and/or cloud providers and cloud subscriber on VMs is necessary.

1. Check availability of desired VM configuration at target cloud provider (see VM Migration to New Cloud Provider graphic above).
2. A positive answer is given (configuration and resources available) reserve resources at target cloud provider. Negative answer – try next provider catalog or stay at source cloud provider.
3. Suspend VM at source cloud provider.
4. Initialize VM at target cloud provider.
5. Start VM at target cloud provider.
6. In case of a successful start, VM (resources and configuration) at source cloud provider is fully shut down and freed up.

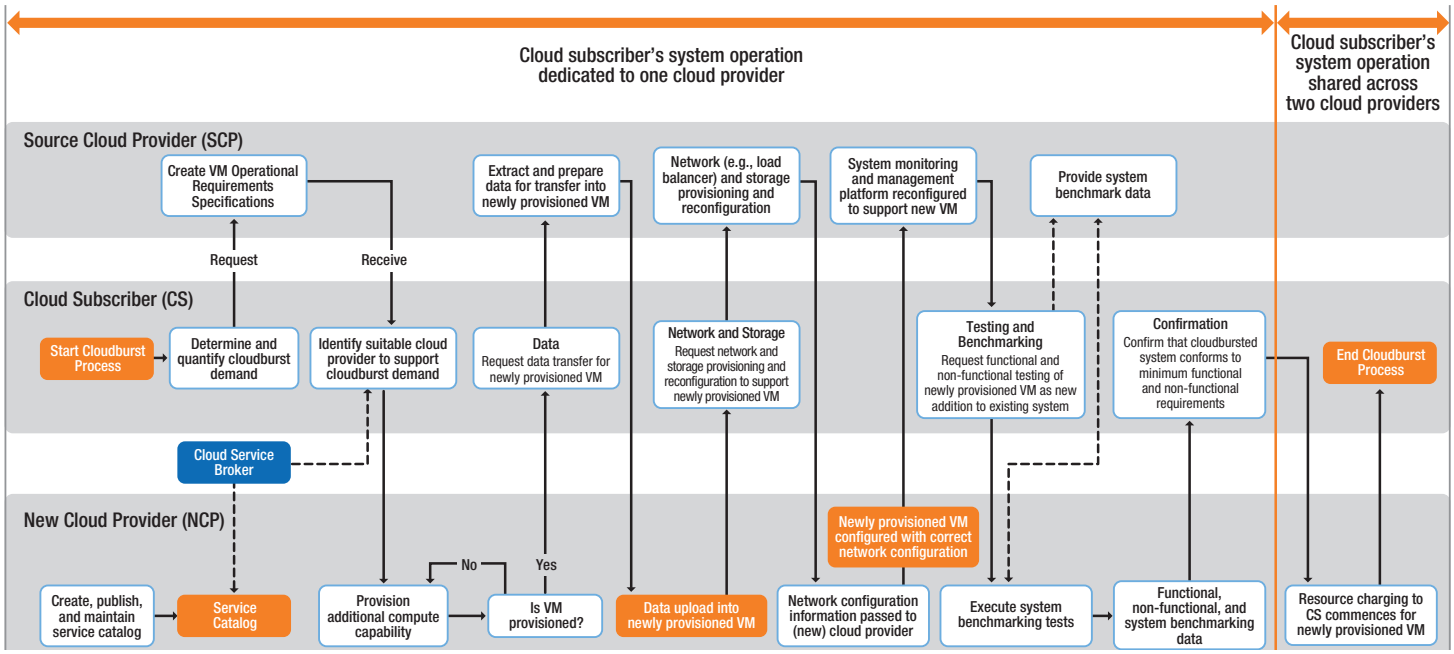
Back-out option – in case of not succeeding to start VM at destined target cloud provider, a init/start at source cloud provider of VM may be initiated.

Scenario is valid for different cloud providers (source and target) and different cloud platforms of the same cloud provider.

Standardized communication, for example, for availability check, acknowledgement, denial, and successful or unsuccessful start of VM is necessary.

This is explained graphically in the diagram on the next page, with emphasis on the transition of responsibility point, shown by the orange arrows and line. The following process flow demonstrates the use case involving two cloud providers who, ultimately, at the end of the process, are jointly responsible for the ongoing provision of the cloud subscriber's system (of virtual computing resources). Please note that key elements within this process require careful consideration and are not discussed here, such as the licensing implications, application architecture prerequisites, and so on.

VM Cloudburst Between Cloud Providers



Notes and assumptions:

- The cloud provider has put in place a guaranteed resource assignment and allowable burst allocation model to support the cloud subscriber's VMs.
- The allowable burst allocation model is assumed to represent 100% of the guaranteed resource assignment. The allowable burst allocation is a cost driver. Amount of % might be correlated to Bronze up to Platinum or depend on total amount of ordered resources. For example, if 28 GB of RAM represents the guaranteed resource assignment for a VM, then the allowable burst allocation is an additional 28 GB, for a total of 56 GB of RAM. There is a different process in place to revise the allowable burst allocation to exceed the stated 100% of guaranteed resource reservation.
- It is assumed that this resource burst process will not exceed the allowable burst allocation setting for a VM.
- The maximum allowable CPU resources for a VM is pre-determined by the hypervisor layer employed by the cloud provider. Hence, this represents a hard limit that cannot be exceeded or over-allocated. For example, for the Microsoft Hyper-V system, this is 4 vCPUs per VM, while for VMware's vSphere it is from 8 vCPUs per VM to 32 vCPUs per VM (for Enterprise Plus version of vSphere 5).
- The maximum allowable RAM resources for a VM are also pre-determined by the hypervisor layer employed by the cloud provider. Hence, this represents a hard limit that cannot be exceeded or over-allocated. For example, for the Microsoft Hyper-V system, this is 64 GB of RAM per VM[7], while for VMware's vSphere it is from 24 GB RAM per VM to 1 TB of RAM per VM (for Enterprise Plus version of vSphere 5)[8].
- There is a "pay-as-you-go" model in place where the cloud subscriber is billed for the resources their VMs are allocated (whether the VM actually uses these resources or not).
- The cloud subscriber is directly responsible for all software license management activities that arise as a result of any changes to the CPU-based licensing model for software running on a VM.
- Cloudbursting implies "more of the same"; rather than "adding something new that was not there before." For example, more CPU; more RAM; more storage; more bandwidth. Hence, "add new network link" is not deemed to be "cloudbursting."
- Scope of this use case is for cloudbursting at IaaS level – not at the PaaS or SaaS levels.
- Application that runs on cloud is a distributed application designed to support cloudbursting; it runs across multiple nodes (such as an Apache Hadoop-style system) and hence can increase computing resources via the provisioning of additional discrete VMs to the "cluster"; rather than adding more computing power (CPU/RAM) to a single VM.
- Database tier cannot be cloudbursted due to additional complexities associated with database partitioning across multiple nodes to maintain database integrity and consistency.

[7] See technet.microsoft.com/en-us/library/ee405267(WS.10).aspx

[8] See www.vmware.com/files/pdf/VMware-vSphere-Competitive-Reviewers-guide-WP-EN.pdf

VM Interoperability Process

Sender	Message Type	Receiver
Cloud subscriber	Configuration request	Cloud provider (target)
Cloud provider (target)	Configuration confirmation/denial	Cloud subscriber
Cloud subscriber	Reservation	Cloud provider (target)
Cloud provider (target)	Reservation confirmation/denial	Cloud subscriber
Cloud subscriber	Initialization	Cloud provider (target)
Cloud provider (target)	Initialization confirmation/denial	Cloud subscriber
Cloud subscriber	Start command	Cloud provider (target)
Cloud provider (target)	Start confirmation/denial	Cloud subscriber
Cloud subscriber	Suspend command	Cloud provider (source)
Cloud provider (source)	Suspend confirmation/denial	Cloud subscriber
Cloud subscriber	stop command	Cloud provider (source)
Cloud provider (source)	Stop confirmation/denial	Cloud subscriber
Cloud subscriber	Delete command	Cloud provider (source)
Cloud provider (source)	Delete confirmation/denial	Cloud subscriber

USAGE REQUIREMENTS

The features in the table below are derived from the usage scenarios and life cycle models in the previous sections, and are aligned with the ODCA Standard Units of Measure for IaaS[9]. For this Usage Model, it is not intended that all of the features in a given column must be supported as a group. In practice, a given cloud service provider solution will combine different service levels for different elements. For example, all Bronze Performance feature requirements must first be met before combining features from any other performance level. For instance, Gold Security features can be combined with Bronze.

Service Levels

	Bronze	Silver	Gold	Platinum
Application	Basic	Enterprise equivalent	Critical market or business sector equivalent	Military or safety-critical equivalent
Security	As per ODCA Provider Assurance Usage Model[10]			
Formal Documentation	Online documentation for all service interfaces, GUIs, and command lines			
Web Services Interface Standard	Programmatic web services in cloud provider's choice of standard	Programmatic web services in cloud provider's choice of standard	Programmatic web services in recognized industry standard, consistent with ODCA concepts	Programmatic web services in recognized industry standard, consistent with ODCA concepts
Packaging Standard	Packaging support export and input in provider's choice of standard	Packaging includes import and export in recognized industry standard, consistent with ODCA concepts	Packaging includes import and export in recognized industry standard, consistent with ODCA concepts	Packaging includes import and export in recognized industry standard, consistent with ODCA concepts
Interop Scope	Single VM	VM collections	VM collections	VM collections with policy-based automation
Packaging Attributes	Localizable	Localizable; SLA and QoS; Platform compatibility	Performance enhancing (virtual queues, single root IT virtualization)	Memory management (memory over commits, hardware-assisted memory virtualization), VM-specific drivers
Configurable Rules	None	Firewalls, load-balancers, functional and defined	Security Root of Trust; consistent I/O controls	Carbon measurements; Geo hosting requirements
Discovery	Online services catalog	Online services catalog with published web services	Online services catalog with published web services	Online services catalog with published web services
Management	Discovery and VM tracking	Diagnostics and correlation between physical and virtual resources	Metrics for physical host service-level validation; live migration	Simultaneous live migration of multiple VMs

[9] See www.opendatacenteralliance.org/index2.php?option=com_productsearch&view=lightbox&proid=14
[10] See www.opendatacenteralliance.org/docs/ODCA_ProviderAssurance_Rev. 1.1_Final.pdf

RFP REQUIREMENTS

Following are requirements that the Alliance recommends should be included in requests for proposal to cloud providers to foster proposed solutions support VM Interoperability and consistency among management solutions.

The RFP questions are distilled from the parameters and models identified in the Usage Model, and represent a conceptual evaluation base, rather than a specific technology analysis. The questions are intended to provide a departure to enable a potential cloud subscriber to ensure that they are aware of all of the relevant dimensions which they should consider in their evaluation of a proposed cloud service. These conceptual areas may lead to the potential cloud subscriber extending them with specifics pertinent to their own organizations' requirements, as relevant.

The Conceptual RFP questions for the VM Interoperability Usage Model are as follows:

- **ODCA Principle Requirement** – Solution is open, works on multiple virtual and non-virtual infrastructure platforms, and is standards-based. Describe how the solution meets this principle and any limitations it has in meeting these goals.
- **ODCA VM Interoperability Usage Model 1.1** – Solution should be able to check if a cloud provider or a hypervisor can accept SLA requirements stated by subscriber through an API.
- **ODCA VM Interoperability Usage Model 1.1** – Solution should be able to move/migrate VMs between hypervisors in a fully automated fashion through API and GUI controls.
- **ODCA VM Interoperability Usage Model 1.1** – Solution should have the ability to control VM life cycle management across multiple hypervisors and cloud providers (for example, start/stop/grow/shrink/destroy/snapshot).
- **ODCA VM Interoperability Usage Model 1.1** – Solution should have the ability to discover attributes of VMs and monitor their status regardless of hypervisor and cloud provider.

For an online assistant to help you detail your RFP requirements, go to www.opendatacenteralliance.org/ourwork/proposalengineassistant.

MOSCOW REQUIREMENTS

Another view of the requirements that are important to VM Interoperability can be seen in the **MoSCoW** prioritization table on the next page, which enables stakeholders to understand the importance of each requirement in relation to one another, and enable evaluation of responses to the RFP questions.

As documented in Wikipedia[11], the MoSCoW categories are as follows:

- **MUST:** Describes a requirement that must be satisfied in the final solution for the solution to be considered a success.
- **SHOULD:** Represents a high-priority item that should be included in the solution if it is possible. This is often a critical requirement, but one which can be satisfied in other ways if strictly necessary.
- **COULD:** Describes a requirement which is considered desirable but not necessary. This will be included if time and resources permit.
- **WON'T:** Represents a requirement that stakeholders have agreed will not be implemented in a given release, but may be considered for the future.

As a principle, all requirements are expected to be multi-vendor and open. Key requirements need to be met across vendors and hypervisors.

These requirements are determined by a combination of requirements set to hypervisor vendors and cloud service providers.

[11] See en.wikipedia.org/wiki/MoSCoW_Method

MoSCoW Prioritization Requirements

Requirements for Phase	Requirement Title	Description	MoSCoW
All Phases	Standardized	At a minimum, all hypervisor vendors need to support the DMTF's Open Virtualization Format (OVF).	Must
	Open	Publically available from the public domain.	Must
	Secure	Security attributes (security zone restrictions, compliance requirements).	Must
	Portable	For portability, cloud providers must also support the import and export of VM packages per OVF.	Must
	Efficient	A standard recommended process must exist describing how to migrate VMs.	Must
	Extensible	The current standards and services must demonstrate the potential and capability to be extended as technology functionality evolves (usually through use of open documented standards).	Must
	Internationalized	Localizable attributes (monetary symbols, language specification).	
	Identified	Feature attributes to determine the stakeholders related to a VM at all times (ownership, lease).	Should
Packaging (Hypervisor Vendors)	SLA	Cloud service-level attributes beyond a single VM.	Must
	SLA QoS	SLA and Quality of Service (QoS) attributes.	Must
	Performance	Performance-enhancing attributes (virtual queues, single root I/O virtualization).	Must
	Platform Compatibility	Platform attributes (instruction sets and hardware-assisted features are needed).	Should
	Memory	Memory management attributes, including: • Memory over commits – allowed/not allowed • Hardware-assisted memory virtualization	Must
	VM-Specific Drivers	List of specific drivers and versions which form a part of the underpinning VM.	Should
	Configurable Rules	Such as: • Firewalls, load-balancers for VMs, VM startup rules • Security root of trust • Consistency of I/O management (I/O controls) • Security and compliance (for compliance monitoring) • Carbon measurement • Geo hosting requirements	Should
Deployment	Preparation	Identification of VM states and user transaction queuing/rerouting or load transfer schedule.	Must
	Distribution	Define point of transfer of transactions and load/service to responsibility of alternate cloud provider.	Must
	Deletion (Clean Up)	Perform VM clean up and deletion at source site (if VM is transferred), according to defined criteria (memory scrub, storage, backups, etc.).	Must
	Licenses	Identify relevant products and licenses and license models at source site, which must be replaced at target site, and their sub-factors.	Should

Requirements for Phase	Requirement Title	Description	MoSCoW
Management (Cloud Service Providers using CMPs)		Cloud providers must support standard and consistent ways of discovering, configuring, managing, and monitoring virtual systems in the cloud.	Must
	Life Cycle Management	Discovery and Inventory: Provide consistent mechanisms for discovering VMs and their attributes (CPU, memory, NICs, storage, VM vendor). Life Cycle Management: Consistent control and management of operational life cycle of VMs across hypervisors. All hypervisors should provide mechanisms to create, modify, enable, disable, destroy, suspend, snapshot, and monitor/query changes on a virtual computer system.	Must
	Sprawl	Give easy access to usage and ownership data for sprawl control.	Should
	Golden Clone	Feature identification of clone status to manage golden clone in production which can be continually refreshed. Feature attributes for monitoring.	Should
	Monitoring	Monitoring: Detection and tracking of VMs. SLA, availability, performance monitoring, and usage statistics.	Should
	Diagnostics	Diagnostics: Consistent set of attributes and functions to provide correlation between virtual and physical resources.	Should
	Collection	Workload-based Logical Collections: Workload-specific collections of VMs, networking, and storage components that can be managed as a whole, with policy-based automation.	Could
	Migration	Live Migration of VMs: Support non-disruptive scheduled maintenance and dynamic workload balancing across different hypervisors/clouds. This includes supporting: • Extended migration and flex migration, ensuring migration success across different hardware-assisted virtualization server platform generations • Simultaneous live migrations of multiple VMs • Consistent methods for initiating live migration • Consistent metric for physical host service-level validation (for example, does the host have adequate resources to meet a VM's service-level requirement?) • Prioritization on concurrent live migration jobs	Should
Retirement	Analyze	Analyze impact of retirement by checking stakeholders and integration/dependencies.	Must
	Stop	Announce to stakeholders that VM will be retired. Stop a VM and remove it from production.	Must
	Decommission	Delete a system or service and perform the service termination activities (data and memory deletion and scrubbing, resources returned to resource pool, etc.).	Should
	Remove from Catalog	Terminate a service and begin migration of remaining running systems to alternate available options.	Could

SUMMARY OF REQUIRED INDUSTRY ACTIONS

In the interest of giving guidance on how to create and deploy solutions that are open, multi-vendor, and interoperable, the ODCA has identified specific areas where the Alliance believes there should be open specifications, formal or de facto standards, or common IP-free implementations. Where the ODCA has a specific recommendation on the specification, standard or open implementation, it is called out in this Usage Model. In other cases, we intend to work with the industry to evaluate and recommend specifications in future releases of this document.

The following industry actions are required to refine this Usage Model:

1. The ODCA needs to engage with DMTF to align this Usage Model with the DMTF's Virtualization Management (VMAN) specification efforts.
2. The solution providers need to propose solutions that are aligned with the VMAN specifications for implementing this Usage Model.