



December 3-6, 2007, Santa Clara Marriott, Santa Clara, CA

# Open Virtual Machine Format (OVF)

**René W. Schmidt**  
Principal Engineer, VMware Inc.



# Disclaimer



Clara Marriott, Santa Clara, CA

The DMTF was formed to lead the development, adoption and unification of management standards and initiatives for desktop, enterprise and internet environments

- The information in this presentation represents a snapshot of work in progress within the DMTF.
- This information is subject to change. The Standard Specifications remain the normative reference for all information.
- For additional information, see the Distributed Management Task Force (DMTF) Web site.  
<http://www.dmtf.org>.

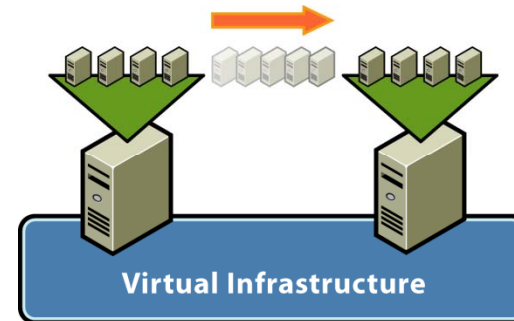
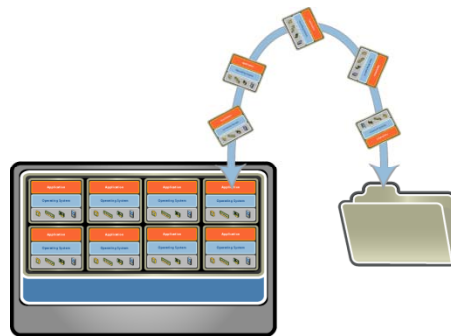


# Overview

- Virtual Machine Mobility
- Open Virtual Machine Format
- Workflow
- Portability
- DMTF
- Summary

# Virtual Machine Mobility

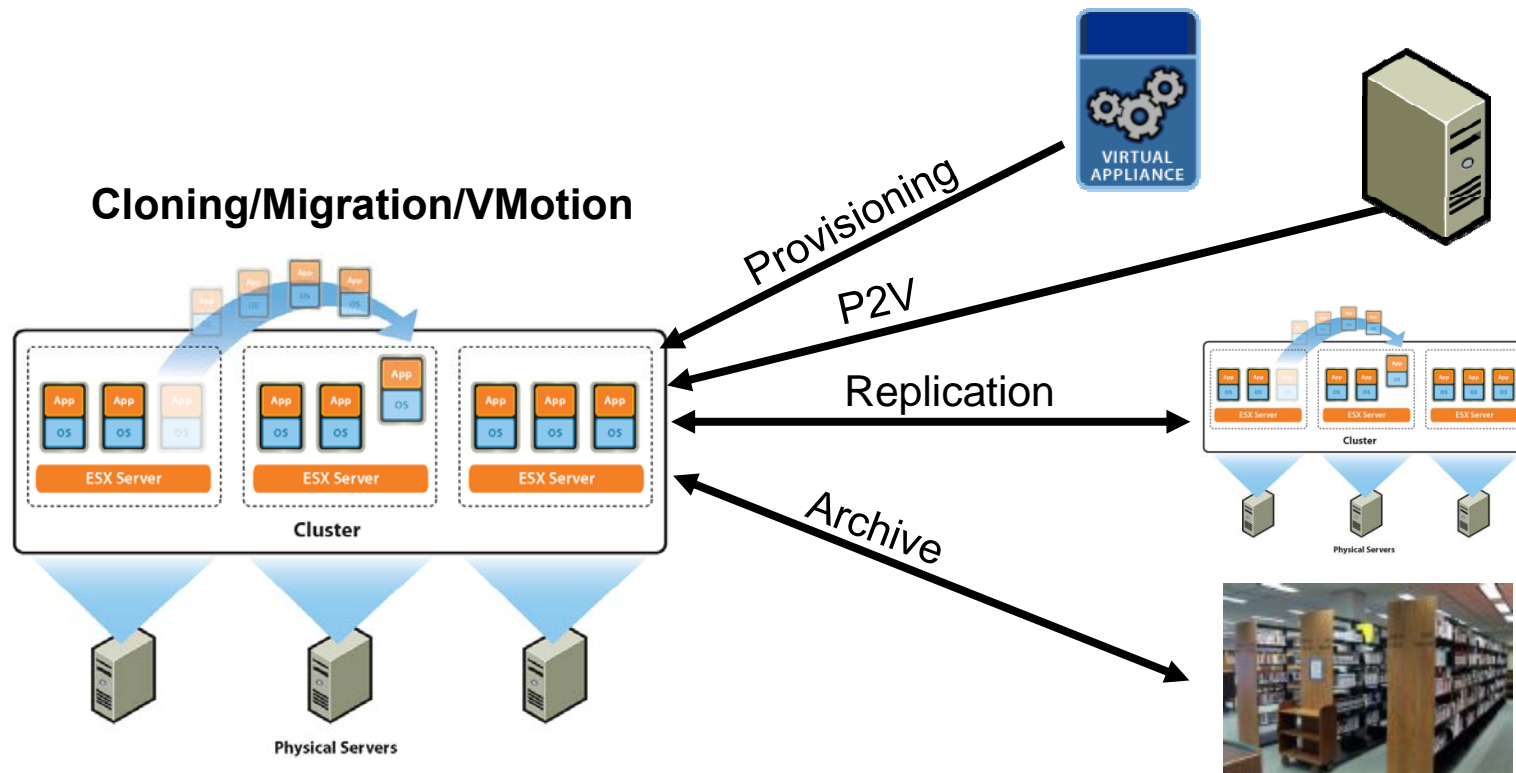
- A key property of virtualization is the encapsulation of the entire state of a virtual machine
  - A powered-off VM is completely described as a set of files
  - A powered-on VM can be VMotioned between hosts



- VMware has pioneered the use of virtual machine mobility

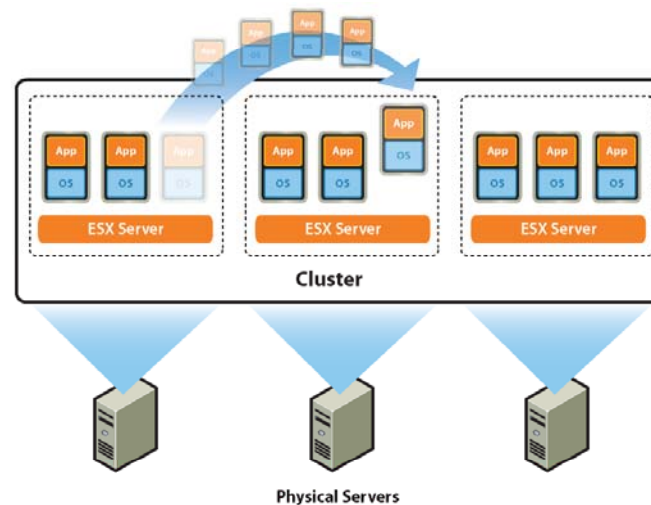
# Virtual Machine Mobility

- Encapsulation has enabled a broad range of new opportunities in Virtual Infrastructure



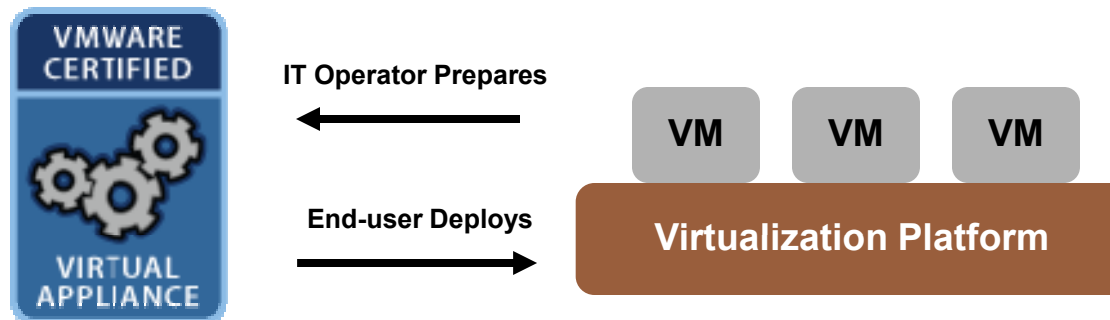
# Virtual Machine Mobility

- HA has increased application availability
- DRS has resulted in great improvements in resource utilization and IT productivity
- Virtual Machine mobility enables new processes such as Power Management to reduce overall IT costs



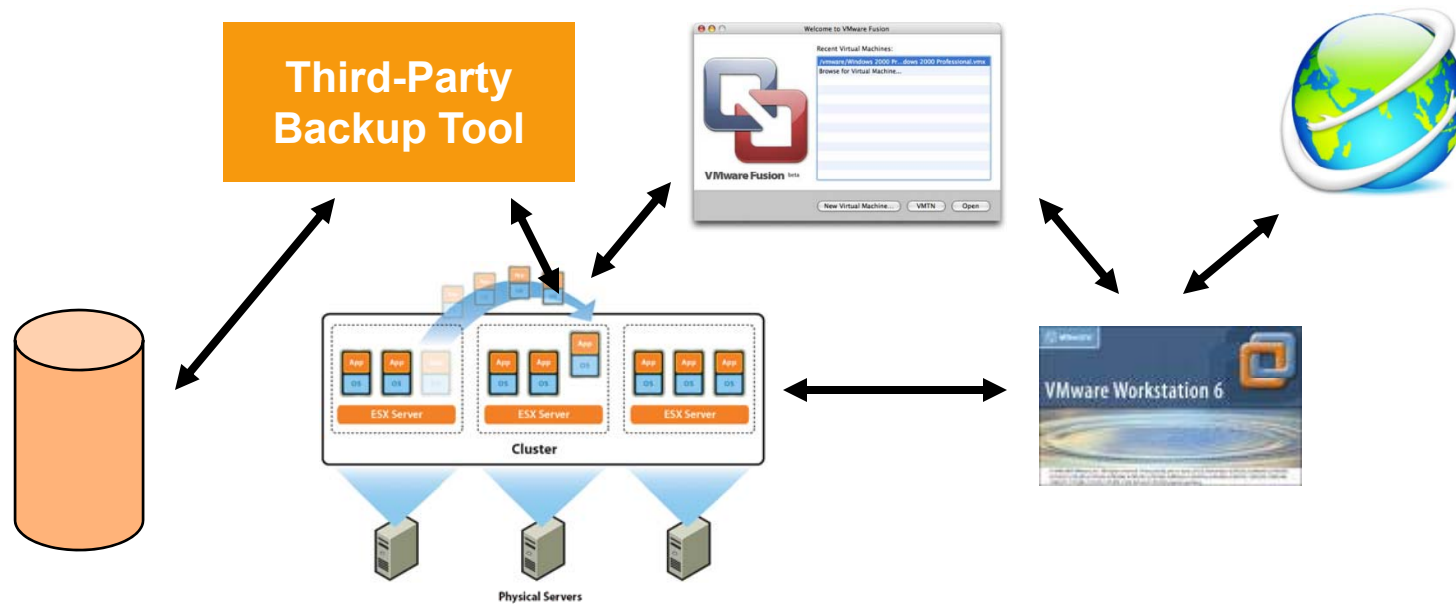
# Virtual Appliances

- Provisioning is a critical function required to deploy new applications to the virtualization infrastructure
- Virtual Appliances have made it easier to deploy new applications
- But challenges remain ...

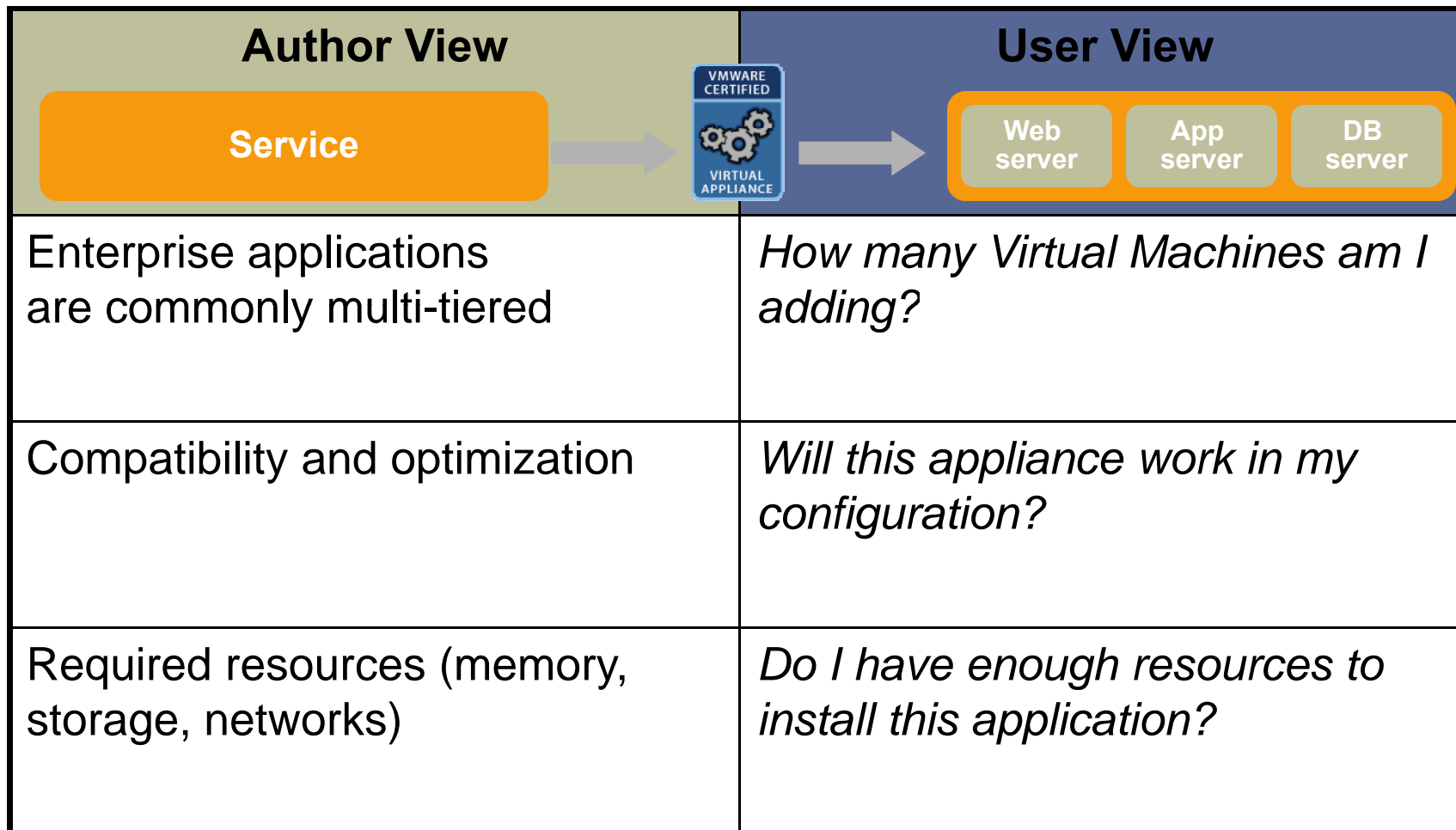


# Large Scale Virtual Machine Sharing

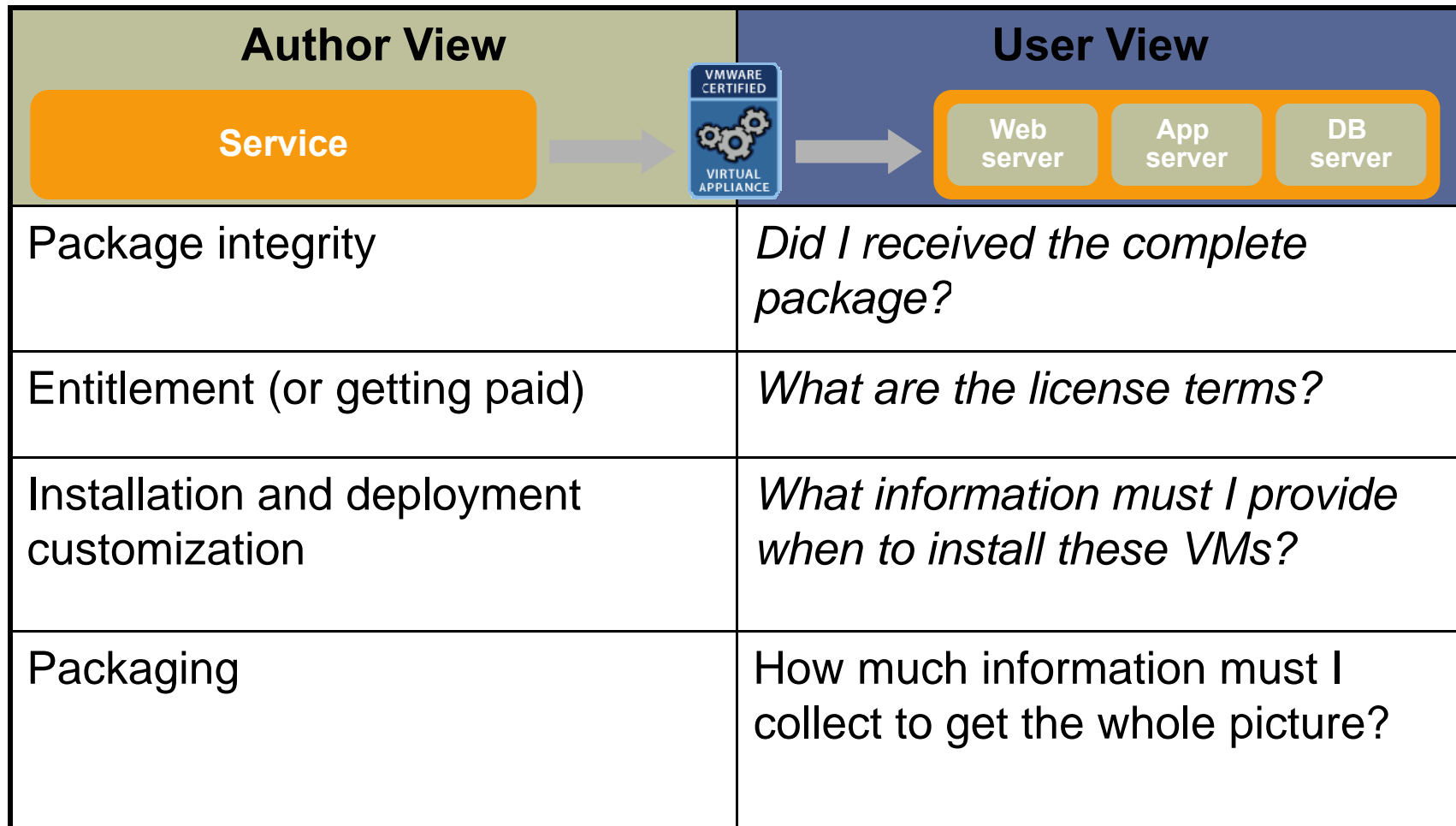
- How do different vendors exchange VMs?
- How do different products interoperate on meta-data?
- Stability of data?
- Suitability of format for distribution?



# Perspectives on a Virtual Appliance



# Perspectives on a Virtual Appliance





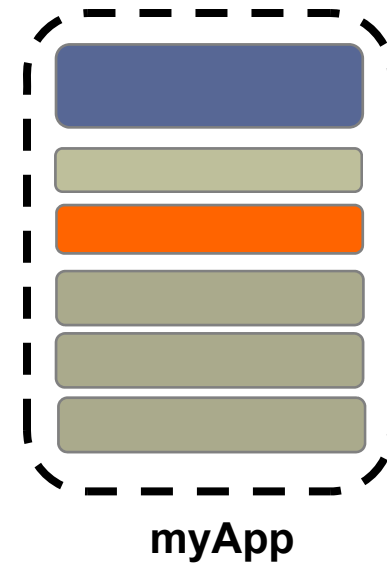
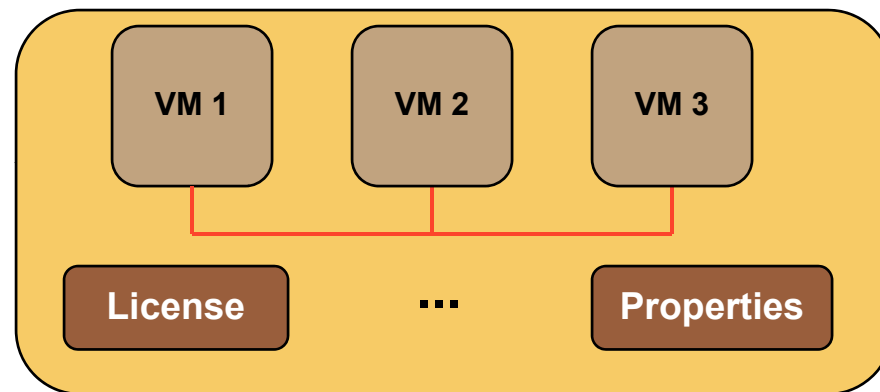
# Objective

- A standard for packaging and distribution of Virtual Machines
  - A package format that provides a complete description of a single VM or complex multi-VM environments
  - Optimized for distribution
  - Infrastructure to securely install, configure, and run Virtual Appliances
  - Extensible to facilitate value-added features
  - Vendor and platform independent
  - Based on open, industry standards

*Open Virtual Machine Format (OVF)*

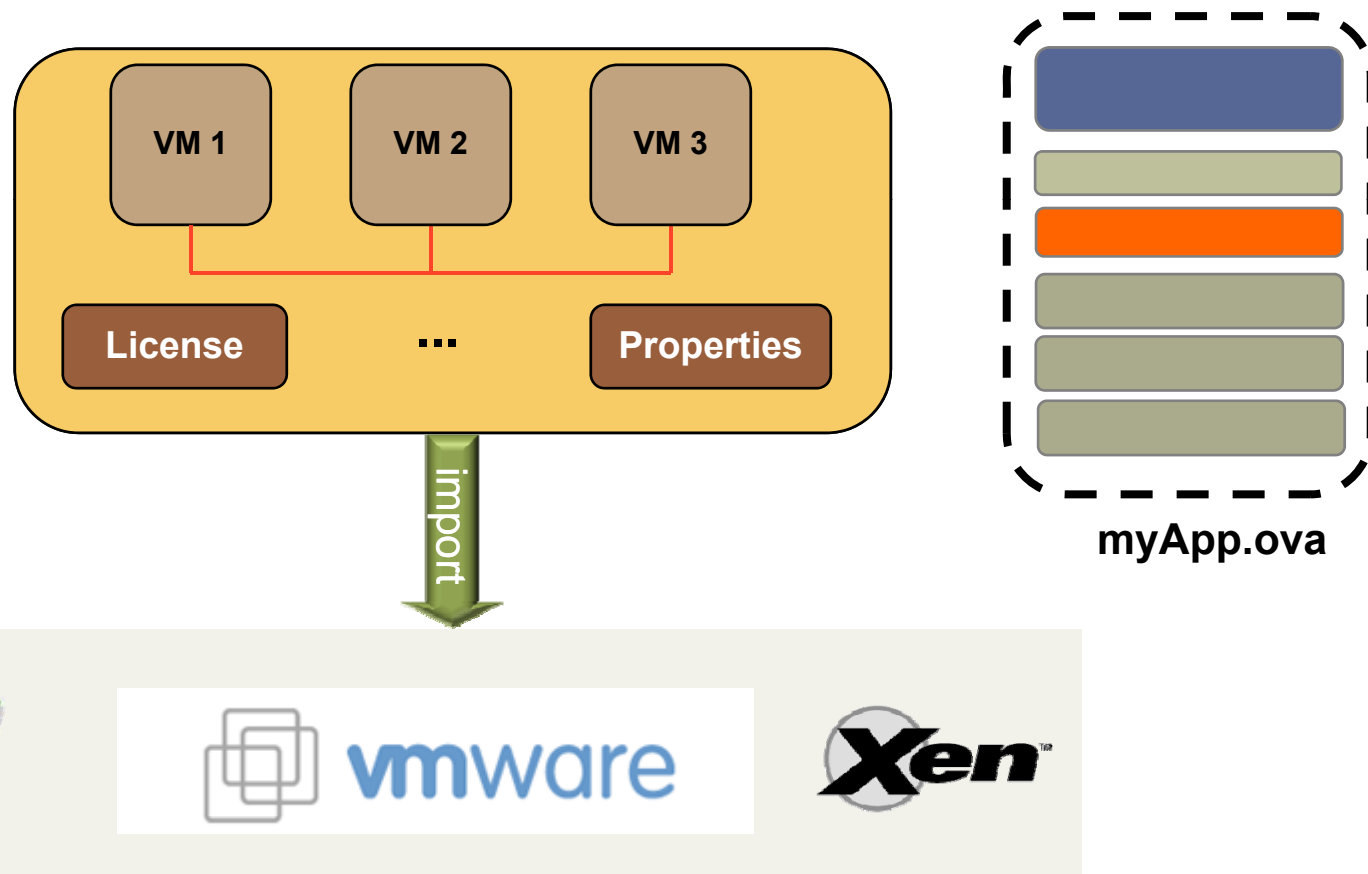
# Open Virtual Machine Format (OVF)

A distribution format for virtual appliances



# Open Virtual Machine Format (OVF)

A multi-vendor format

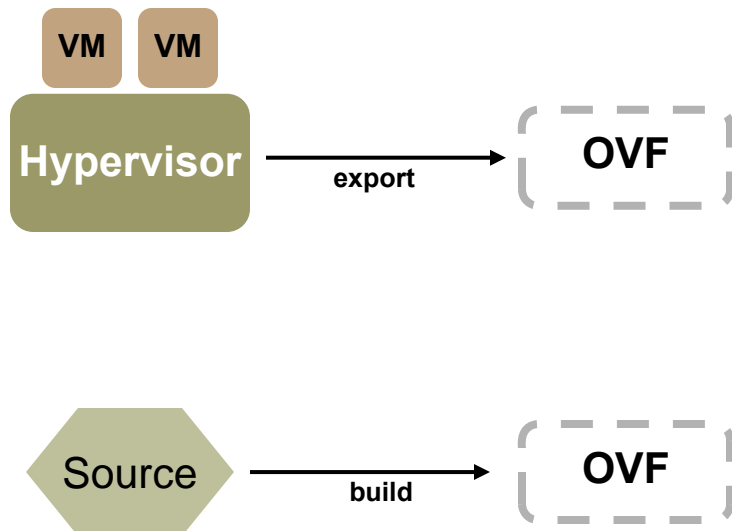


# OVF Workflow

## Create → Publish → Deploy

- **Author Role:** To create a complete software package that makes life simple for the deployer, and to publish the package
- **Deployer Role:** Deploy and customize the template for this particular instance

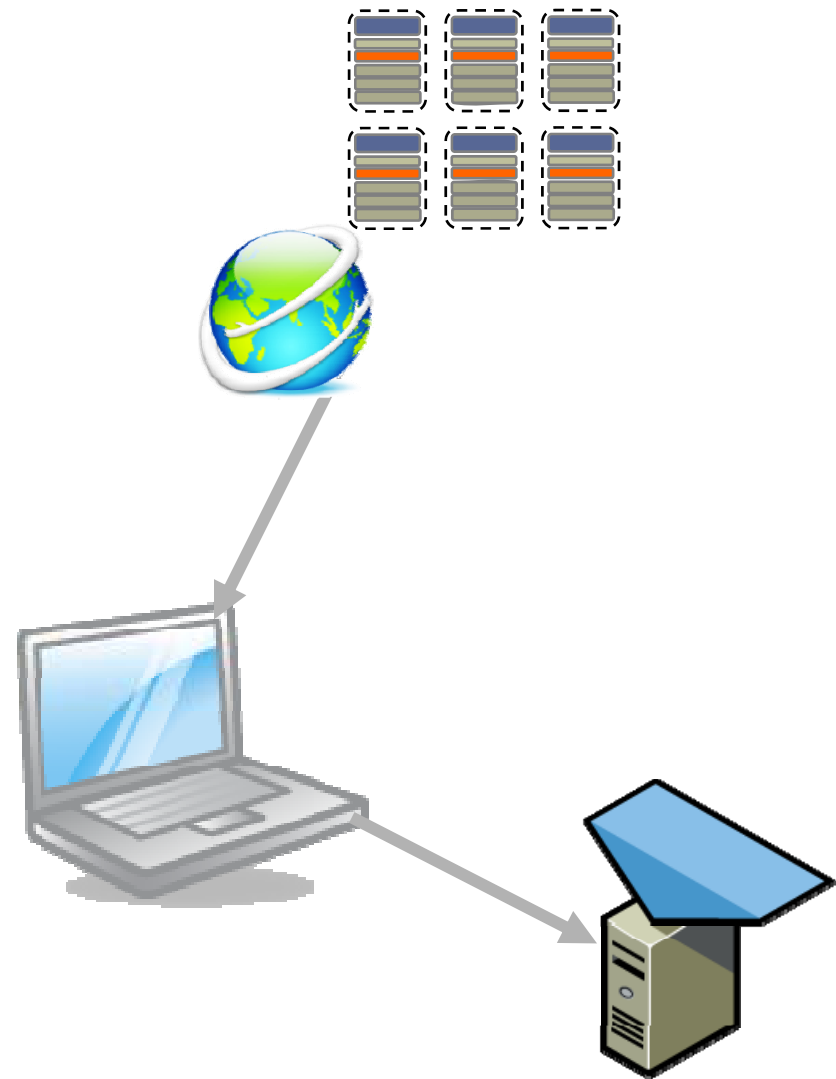
# OVF Authoring



- A virtual machine can be exported into OVF or built from source
- The author can describe “portable hardware” for maximum interoperability or optimum performance
- The virtual machine images can be compressed to facilitate distribution
- The author can sign the package to guarantee its integrity

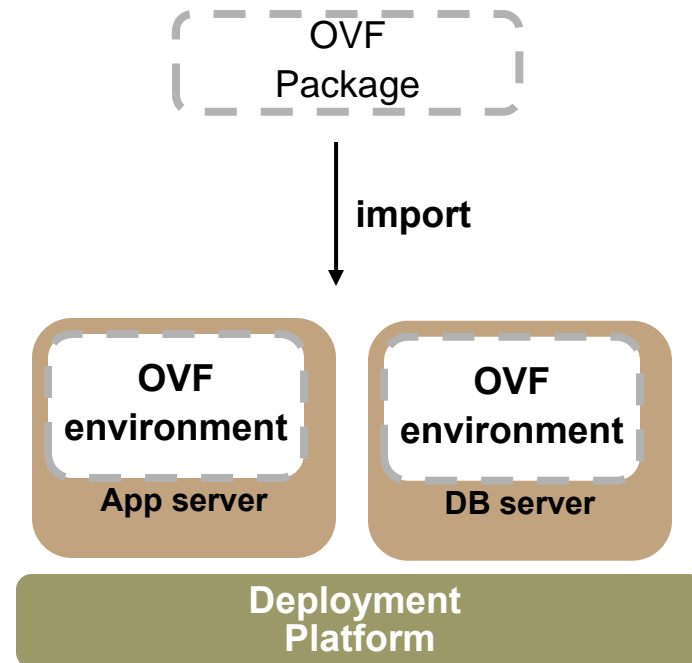
# OVF Distribution

- Users can preplan deployment
  - Descriptor file to validate resource and system requirements, license requirements
- Download format can be optimized for faster retrieval
- Manifest and certificates to validate file contents



# OVF Deployment

- Validation
  - Verify licensing, security, integrity
  - Resource requirements and placement
  - Application Properties (IP addresses and passwords)
- Conversion
  - Convert virtual disks to run-time format
- Installation
  - Provide runtime environment for application (customization and localization)
  - Provide installation feedback to user

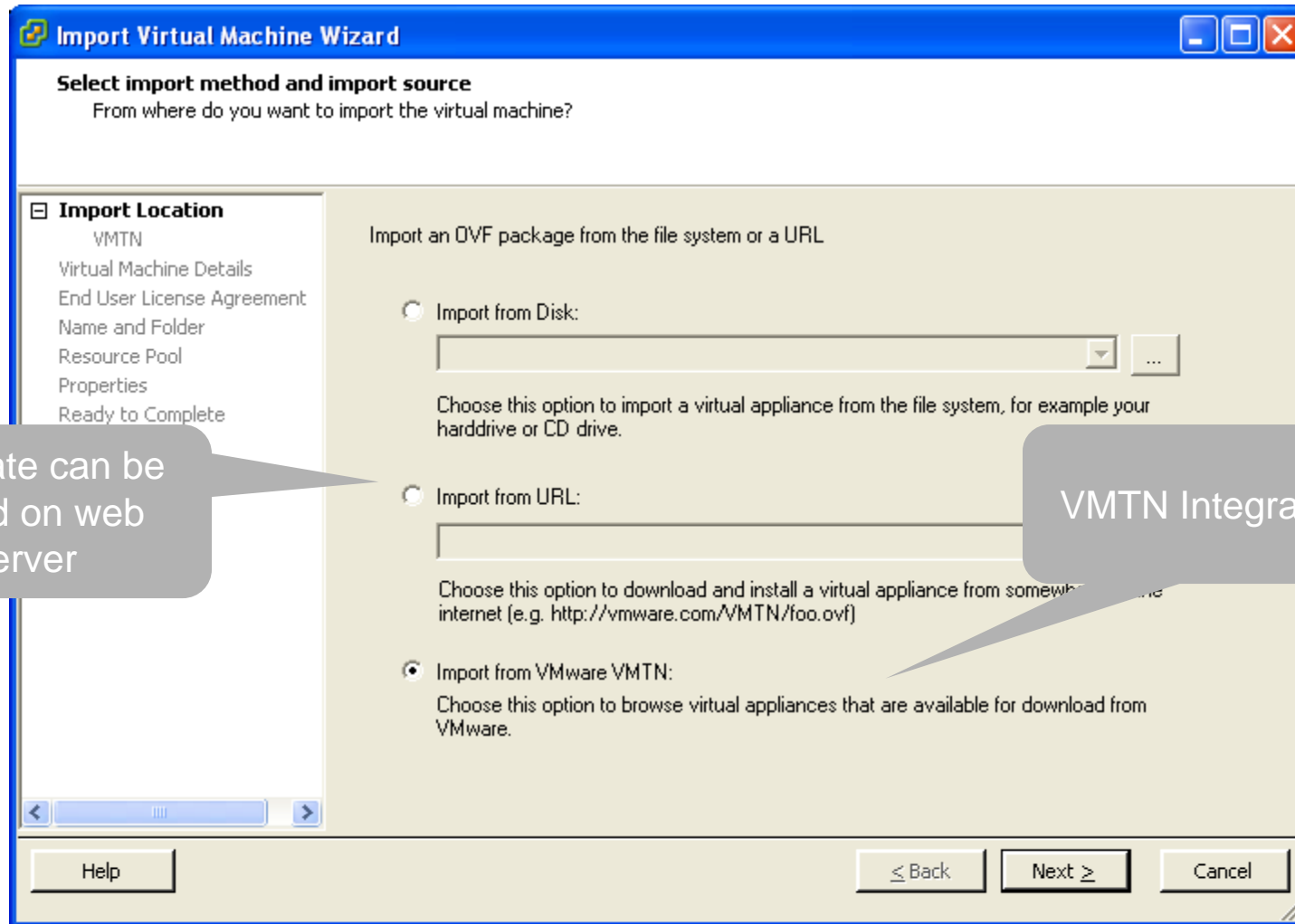




## Scenario: Import of Multi-Tier Service in VC

- Scenario
  - Installation of a template containing multiple VMs
- Key Points
  - Create multi-VM templates
  - Can be downloaded from a Web Server
  - Application-level customization is supported through properties
  - Instantiated as a service object in VC inventory

# Step 1: Select Source



**Import Virtual Machine Wizard**

**Select import method and import source**  
From where do you want to import the virtual machine?

**Import Location**

- VMTN
- Virtual Machine Details
- End User License Agreement
- Name and Folder
- Resource Pool
- Properties
- Ready to Complete

Import an OVF package from the file system or a URL

Import from Disk:

Choose this option to import a virtual appliance from the file system, for example your harddrive or CD drive.

Import from URL:

Choose this option to download and install a virtual appliance from somewhere on the internet (e.g. <http://vmware.com/VMTN/foo.ovf>)

Import from VMware VMTN:

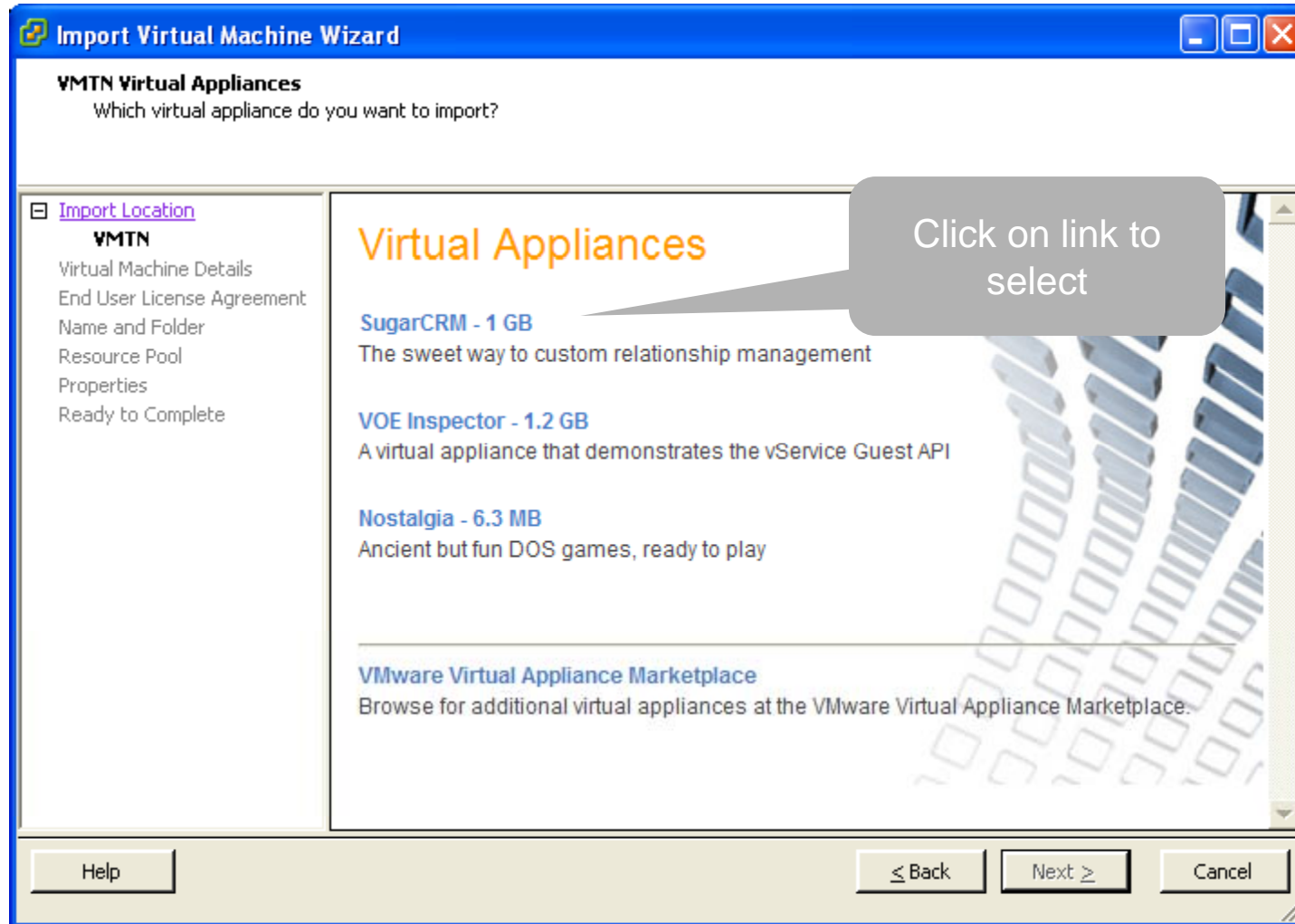
Choose this option to browse virtual appliances that are available for download from VMware.

Help    < Back    Next >    Cancel

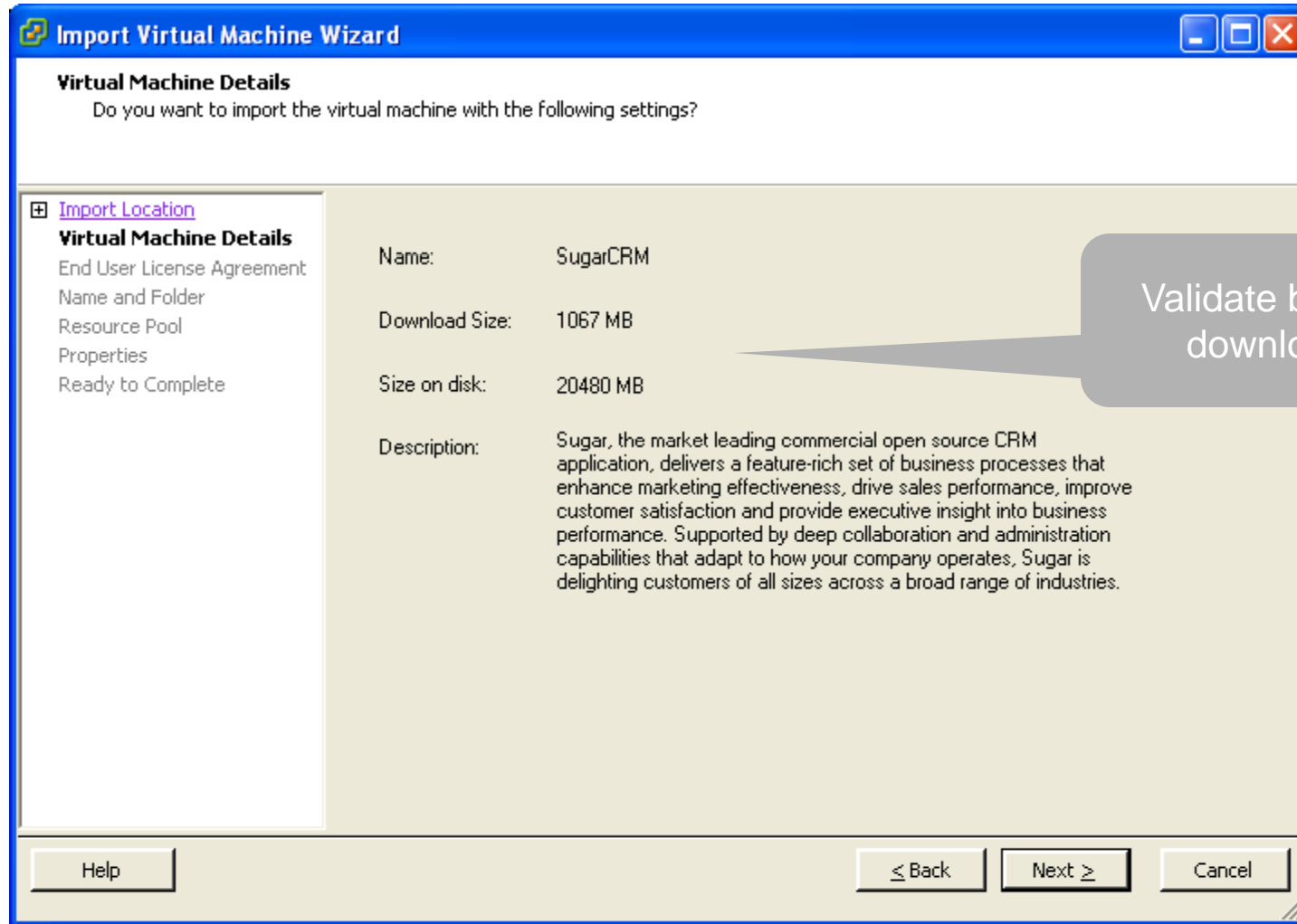
Template can be stored on web server

VMTN Integration

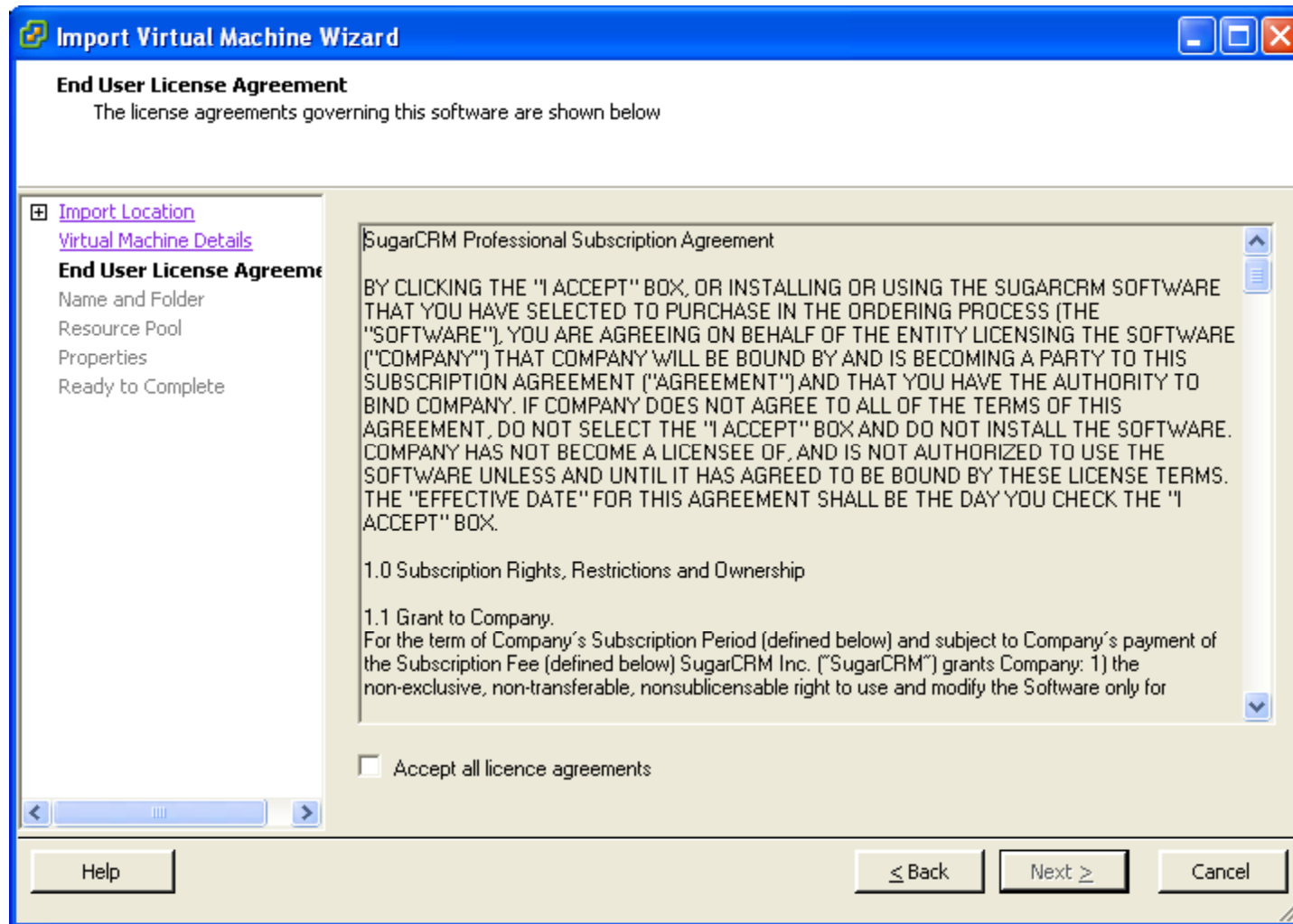
# Step 2: Select Appliance



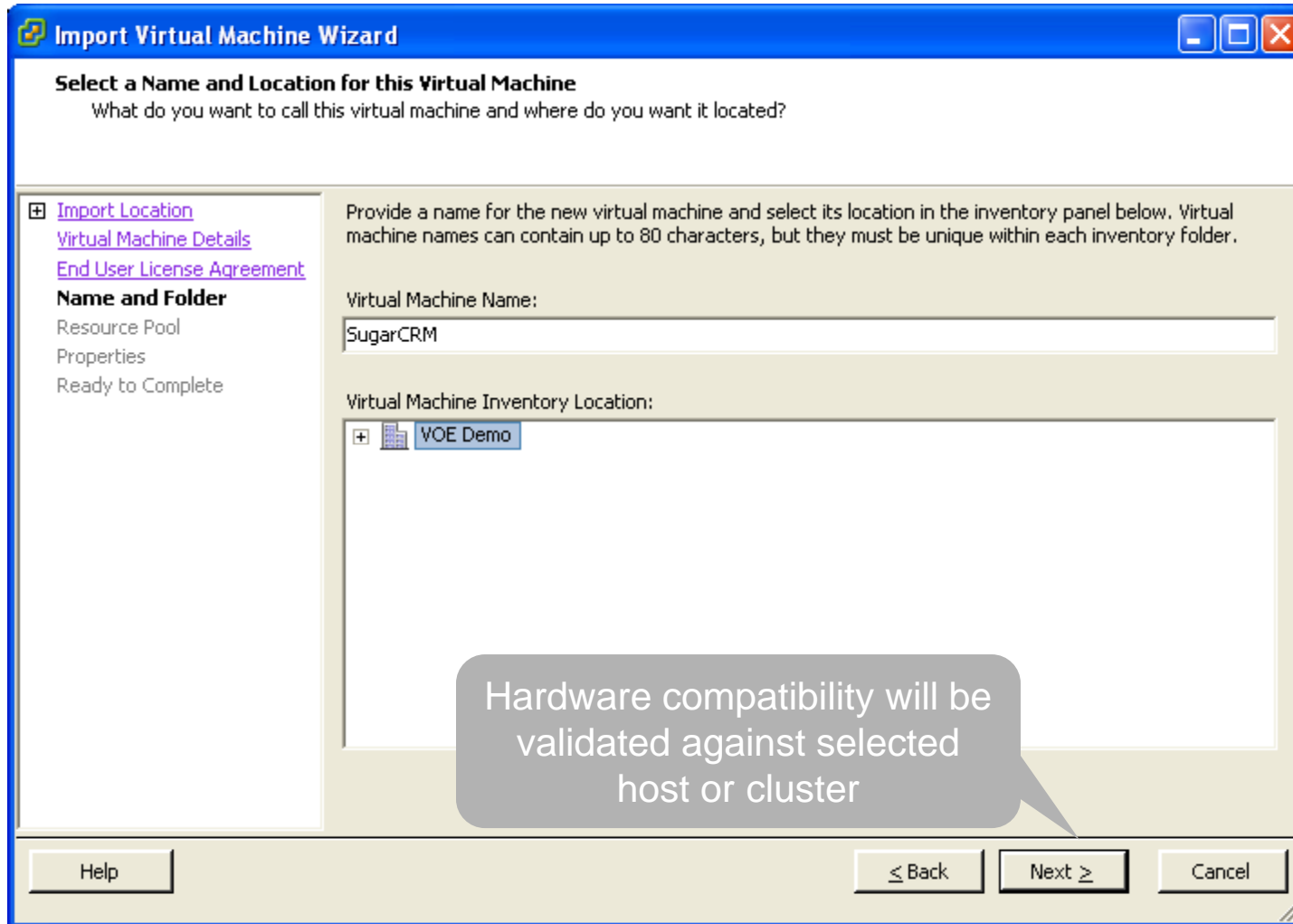
# Step 3: Summary Information is Shown



# Step 4: Accept License



# Step 5: Name and Location



**Import Virtual Machine Wizard**

**Select a Name and Location for this Virtual Machine**  
What do you want to call this virtual machine and where do you want it located?

**Import Location**  
[Virtual Machine Details](#)  
[End User License Agreement](#)  
**Name and Folder**  
Resource Pool  
Properties  
Ready to Complete

Provide a name for the new virtual machine and select its location in the inventory panel below. Virtual machine names can contain up to 80 characters, but they must be unique within each inventory folder.

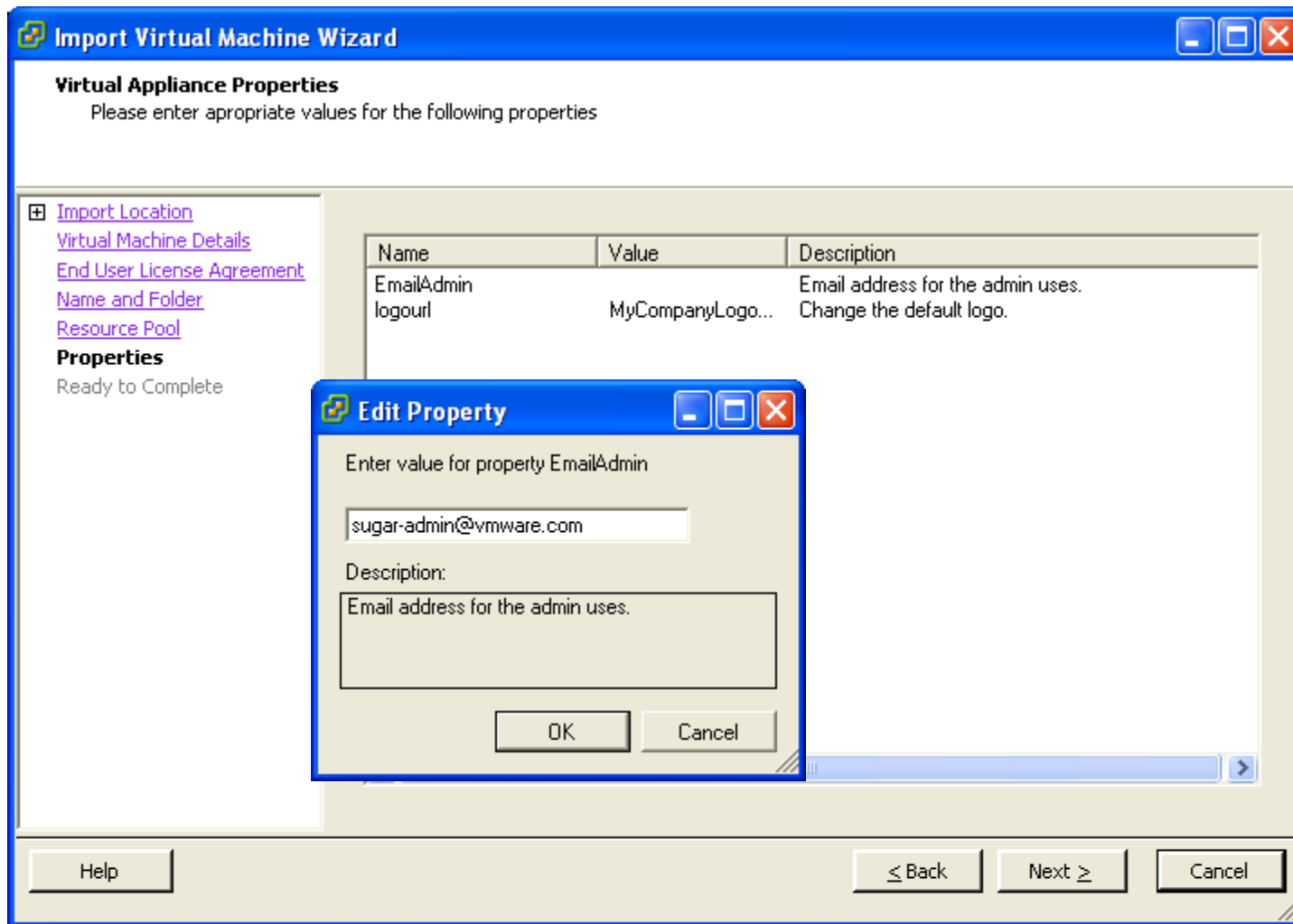
Virtual Machine Name:  
SugarCRM

Virtual Machine Inventory Location:  
+ VOE Demo

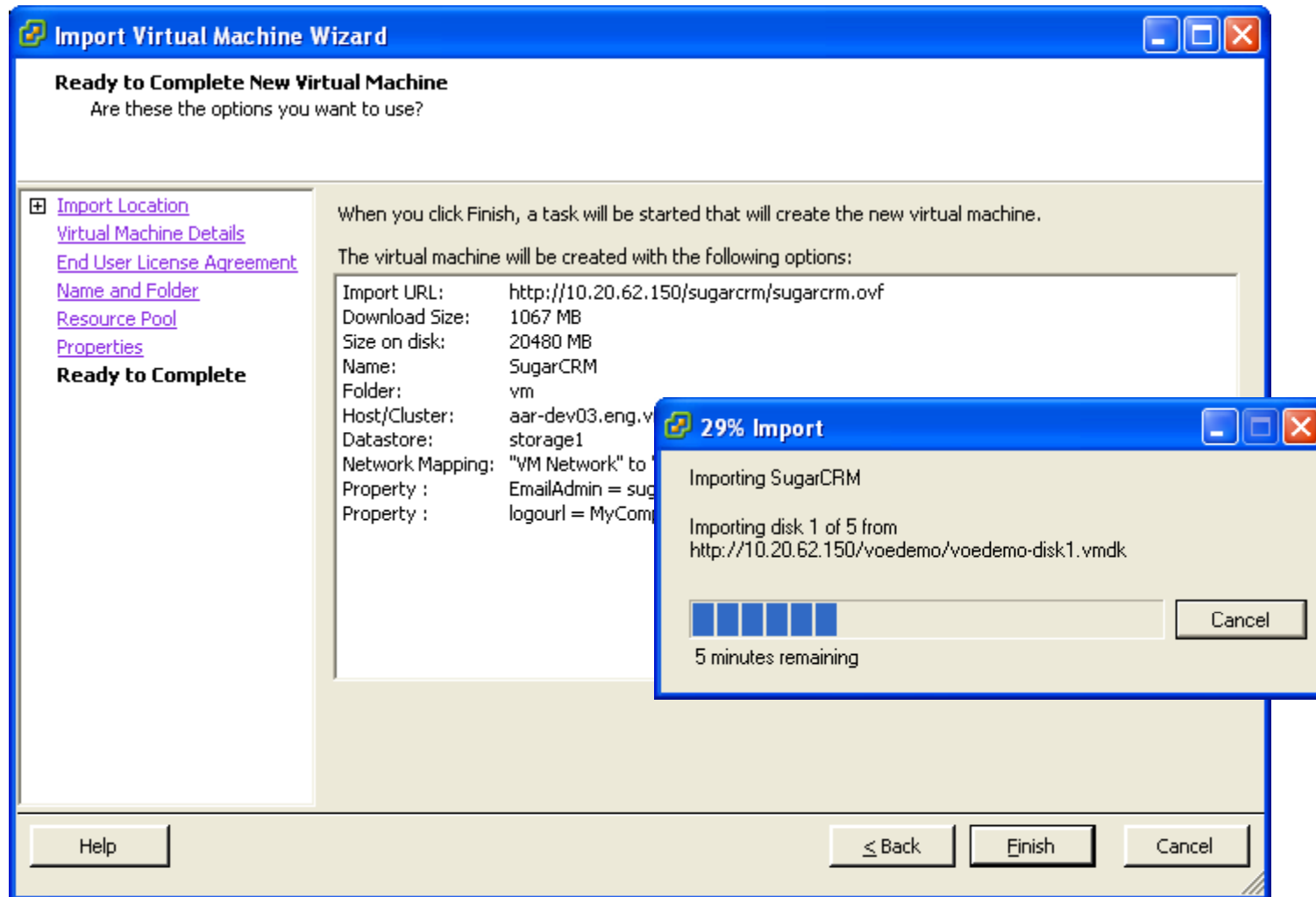
Hardware compatibility will be validated against selected host or cluster

Help    < Back    Next >    Cancel

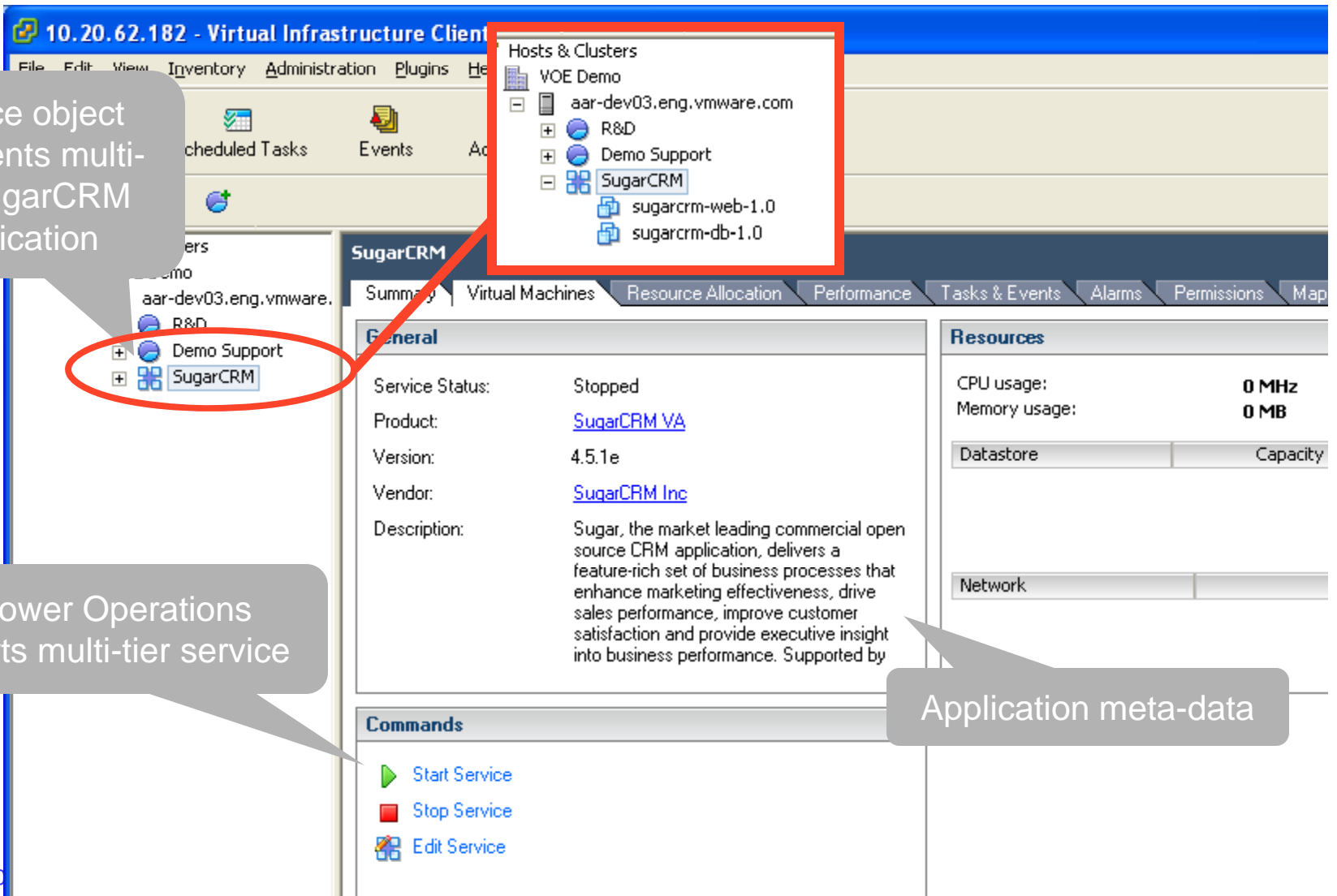
# Step 7: Application-level Customization



# Step 8: Download



# VC Inventory Object Created



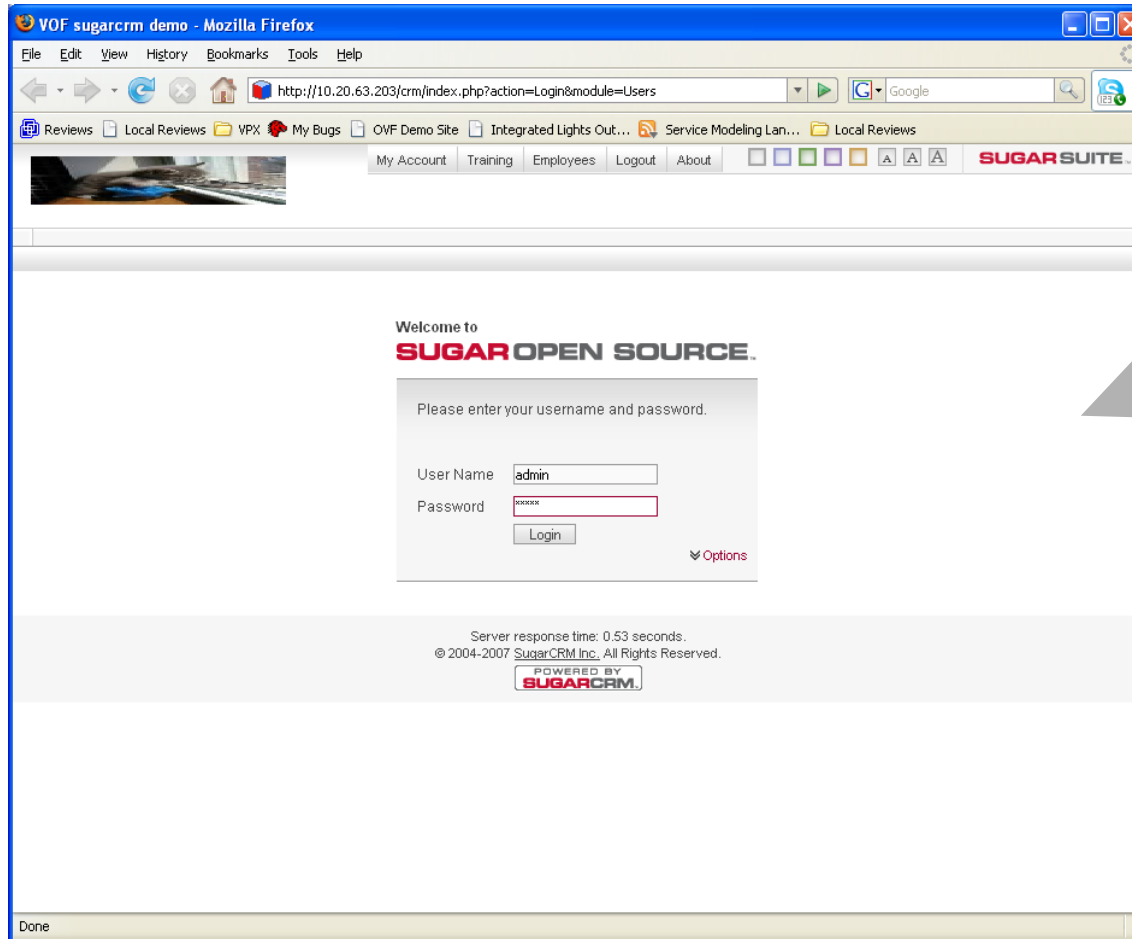
The screenshot shows the VMware vSphere Client interface. The main window displays the 'SugarCRM' service details for the host 'aar-dev03.eng.vmware.com'. The 'General' tab is active, showing service status as 'Stopped', product as 'SugarCRM VA', version as '4.5.1e', and vendor as 'SugarCRM Inc'. The 'Resources' tab shows CPU usage at 0 MHz and memory usage at 0 MB. The 'Commands' section includes 'Start Service', 'Stop Service', and 'Edit Service' options.

Annotations on the screenshot include:

- A red box highlights the 'Hosts & Clusters' tree view, showing the path: **VOE Demo** > **aar-dev03.eng.vmware.com** > **R&D** > **Demo Support** > **SugarCRM** > **sugarcrm-web-1.0** > **sugarcrm-db-1.0**.
- A red oval highlights the 'SugarCRM' service icon in the left-hand navigation pane.
- A grey callout bubble points to the 'SugarCRM' service icon with the text: "Service object represents multi-tier SugarCRM application".
- A grey callout bubble points to the 'Start Service' command with the text: "Power Operations Starts multi-tier service".
- A grey callout bubble points to the 'General' tab with the text: "Application meta-data".



# SugarCRM Login Screen

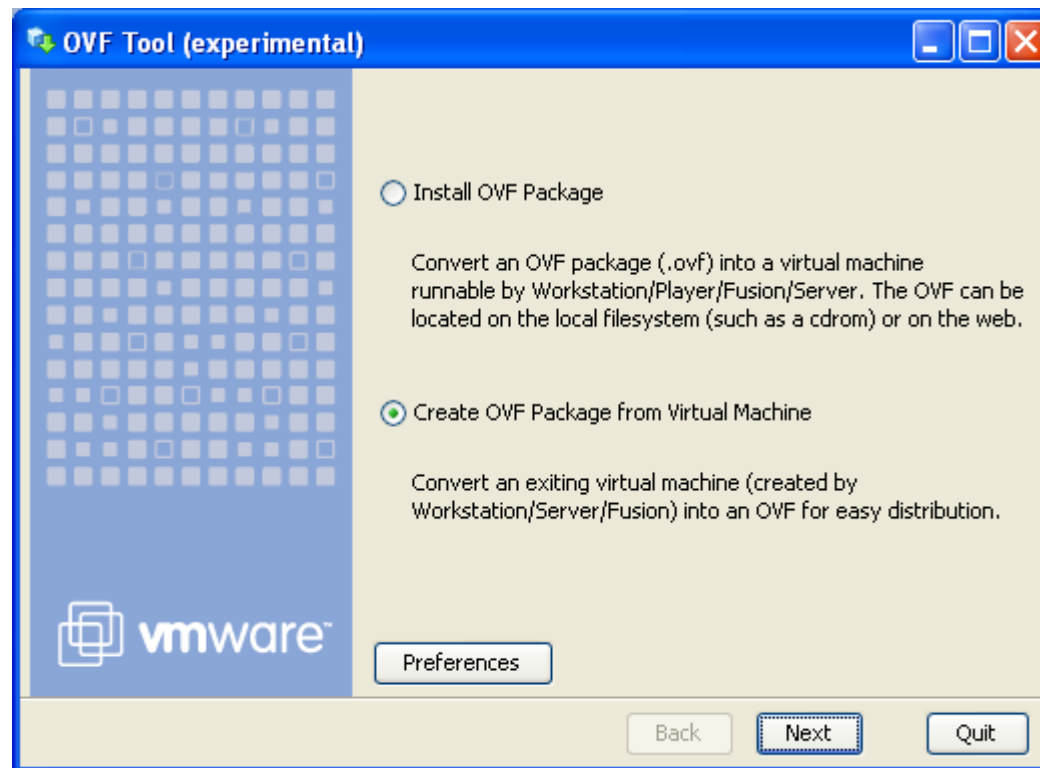


Web server and DB server starts in correct order.

Automatic network configuration.

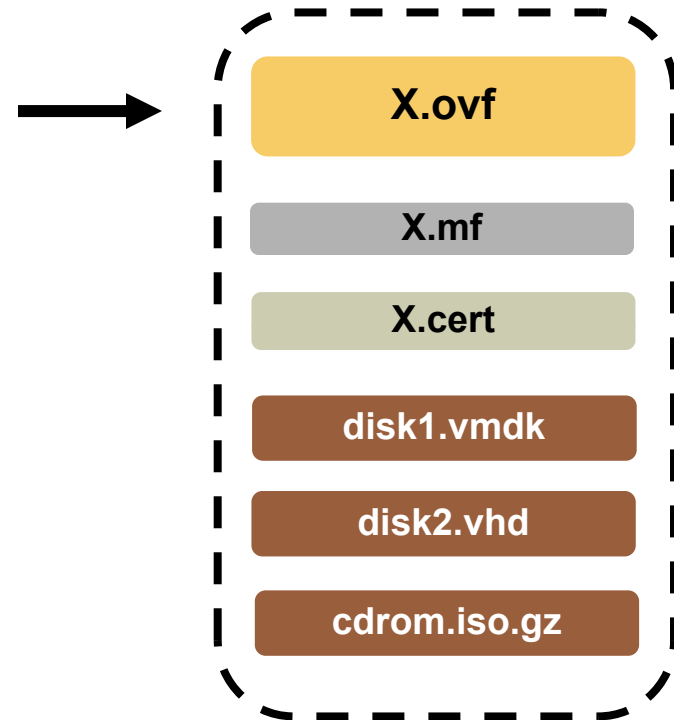
# OVF Tool

- Experimental tool converts from/to hosted formats



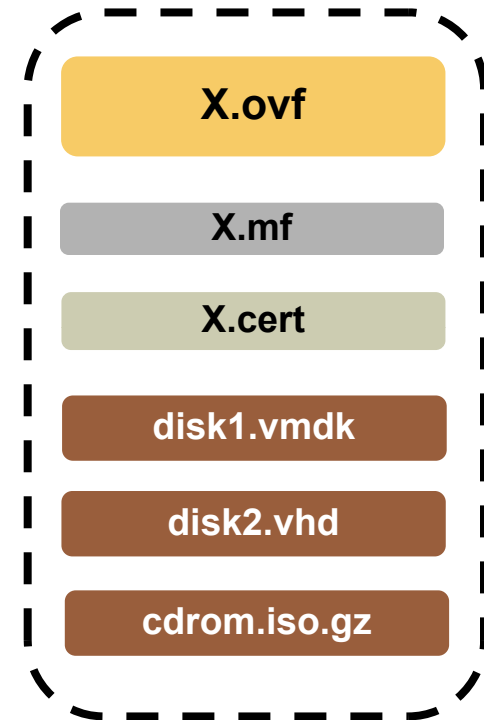
# OVF Format

- Descriptor File
  - Encodes product details into an XML-based, extensible format
  - Includes:
    - Details about each file in package
    - Description of virtual hardware
    - License information
  - Supports one or multiple virtual machines



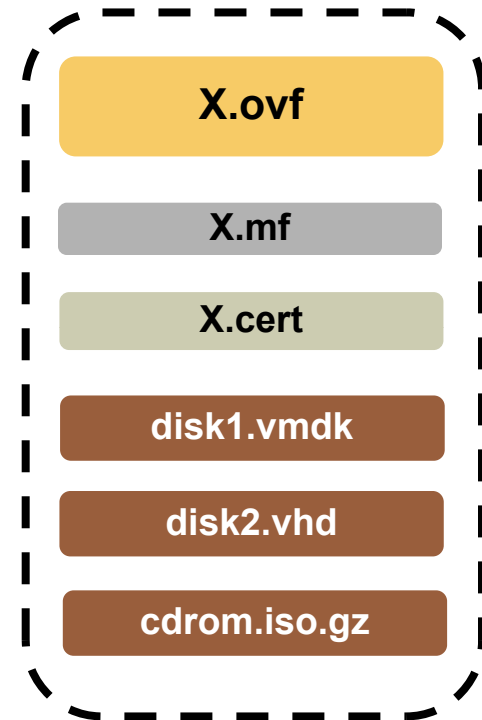
# OVF Format

- Manifest File (Optional)
  - Contains a list of files and their SHA-1 digest entries
  - Allows the user to enumerate and validate the virtual machine's files



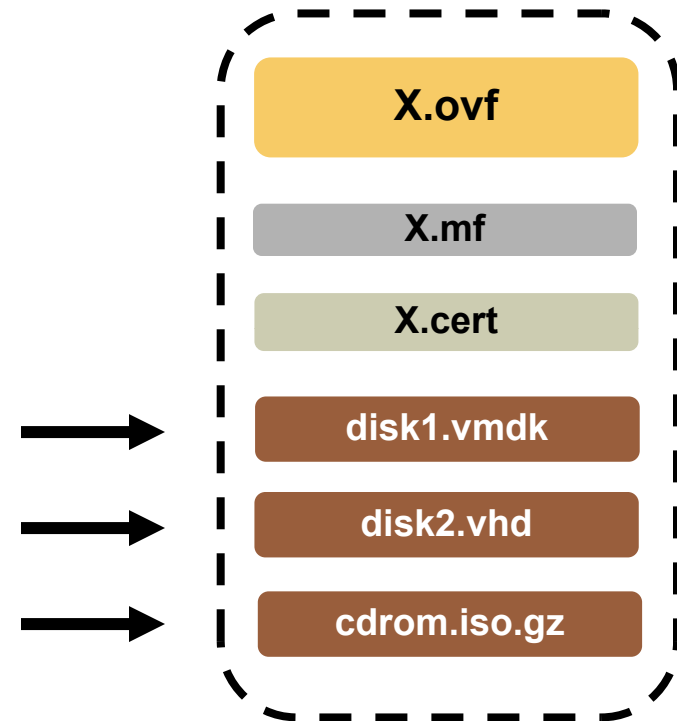
# OVF Format

- Certificate File (Optional)
  - Provides a digital certificate to protect the files



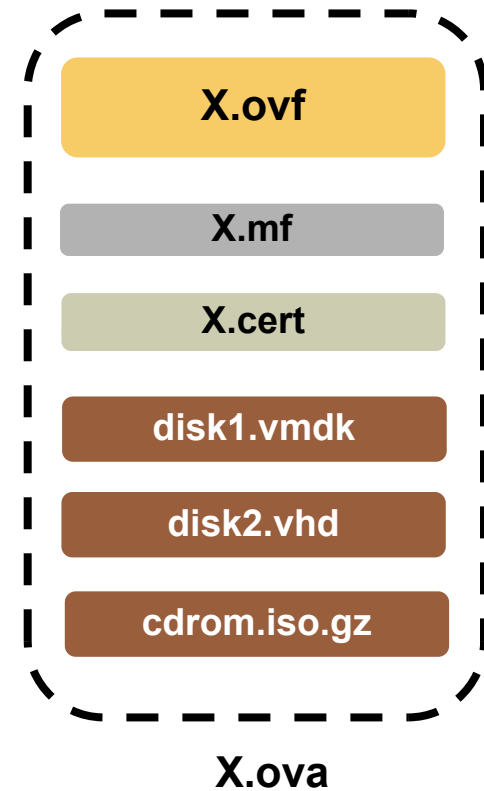
# OVF Format

- Virtual Disks
  - Compiles the list of Virtual Disks required for the Virtual Machine(s)
  - OVF supports any Virtual Disk format with a publicly-disclosed specification
- Supporting Files
  - ISO and other files
  - Can be compressed and chunked



# OVF Format

- Package File (optional)
  - Allows OVF files to be packaged as a single file
  - TAR used as packaging technology





# OVF XML Descriptor

- An XML document defined by an XML Schema
- Extensible
  - Organized as an `Envelope` of `Sections`. Each section specifies of particular kind of meta-data
  - 10 core sections defines (`VirtualHardwareSection`, etc.)
  - Extensible with new `Section` types
- Contents
  - `VirtualSystem`: A single virtual machine
  - `VirtualSystemCollection`: A collection of `VirtualSystem` and `VirtualSystemCollection` entities



# OVF XML Descriptor: Overview

```
<?xml version="1.0" encoding="UTF-8"?>
<ovf:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
    xmlns:ovf="http://...">
  <References>
    <File id="file1" href="vmdisk1.vmdk" size="1834334"/>
    ...
  </References>

  <Section xsi:type="..." required="true|false">
    <Info>Describes what this section is all about</Info>
    ...
  </Section>
  ...
  <Content xsi:type="...">
    ... sections ...
  </Content>
</ovf:Envelope>
```



# OVF XML Descriptor: Disks

- Lists all virtual disks in a package
- Virtual disks are described outside VMs
  - Allows multiple VMs to point to the same content
  - Allows base disks to be shared between virtual machines
  - Disk chains are supported

```
<Section xsi:type="ovf:DiskSection_Type">
  <Disk id="vmdisk1" fileRef="file1"
        capacity="8589934592" populatedSize="3549324972"
        format="http://www.vmware.com/vmdk.html#compressed"/>
  ... more disks ...
</Section>
```



# OVF XML Descriptor: VirtualHardware

OVF specifies "virtual hardware requirements"  
Based on CIM  
Multiple virtual hardware families are supported

```
<Section xsi:type="ovf:VirtualHardwareSection_Type">
  <Info>500Mb, 1 CPU, 1 disk, 1 nic virtual machine</Info>
  <System>
    <vssd:VirtualSystemType>vmx-04, xen3</vssd:VirtualSystemType> </System>
    <Item>
      <rasd:Caption>1 virtual CPU</rasd:Caption>
      <rasd:Description>Number of virtual CPUs</rasd:Description>
      <rasd:InstanceId>1</rasd:InstanceId>
      <rasd:ResourceType>3</rasd:ResourceType>
      <rasd:VirtualQuantity>1</rasd:VirtualQuantity> </Item>
    <Item>
      <rasd:Caption>256 MB of memory</rasd:Caption>
      <rasd:Description>Memory Size</rasd:Description>
      <rasd:InstanceId>2</rasd:InstanceId>
      <rasd:ResourceType>4</rasd:ResourceType>
      <rasd:AllocationUnits>MB</rasd:AllocationUnits>
      <rasd:VirtualQuantity>256</rasd:VirtualQuantity> </Item>
```



# OVF XML Descriptor: Properties

- Application-level Configuration

- Typed key/value pairs

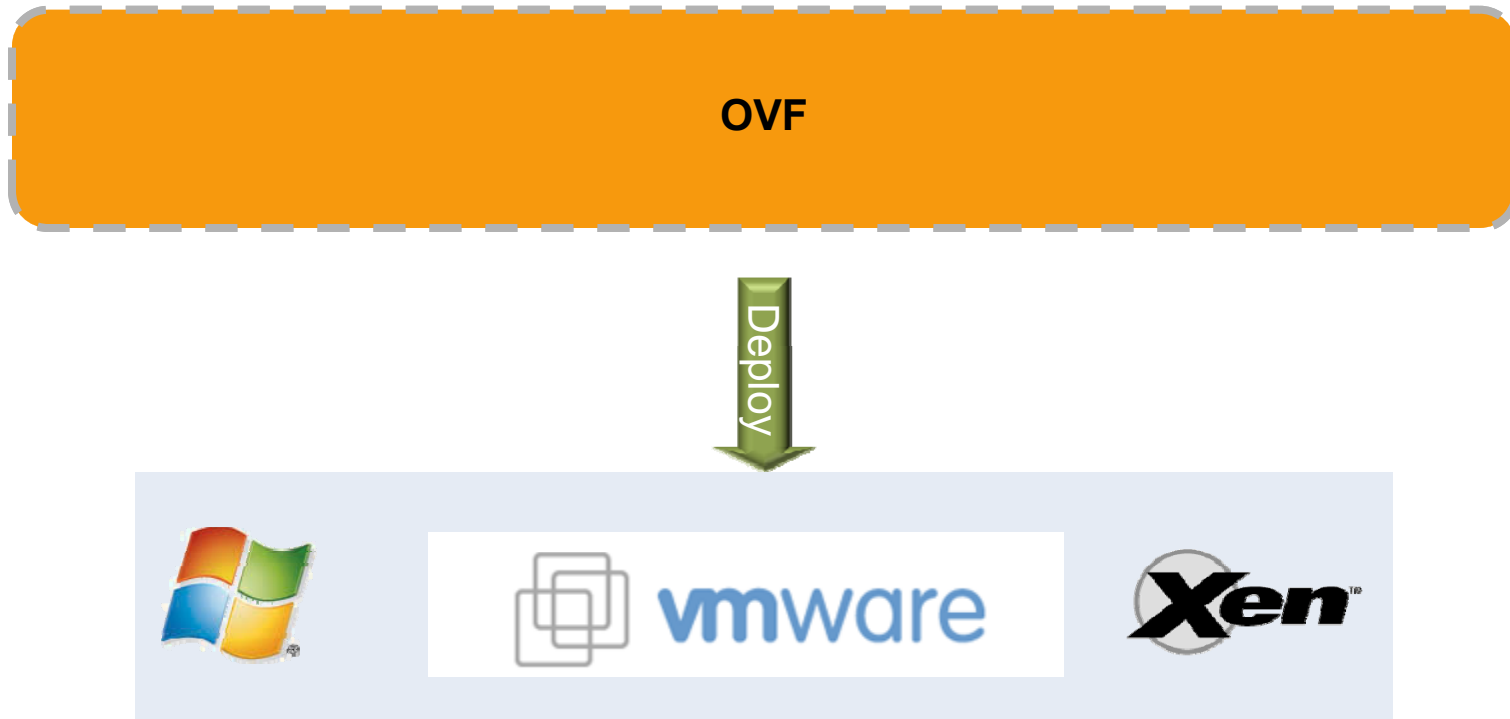
- Available inside the guest

- Can e.g. be used as size arguments for virtual disks

```
<Section xsi:type="ovf:PropertySection_Type">
  <Info>Defines the properties used by the service</Info>
  <Property key="admin.email" type="string">
    <Description>Email address of administrator</Description>
  </Property>
  <Property key="app.ip" type="string"
  defaultValue="192.168.0.10">
    <Description>The IP address of this
  appliance</Description>
  </Property>
</Section>
```

# Portability

- How to create an appliance that can run anywhere?





# Portability Categorization

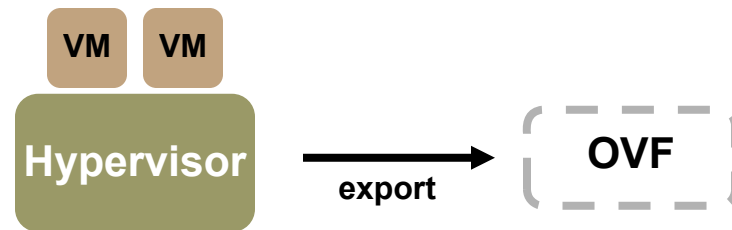
- Level 1
  - VM only runs on a particular product and CPU architecture
  - Typically due to suspended VM state or snapshots of powered-on VMs
- Level 2
  - Run on a specific family of virtual hardware
  - Sufficient for internal use in tightly managed organizations
  - Typically needed due to lack of driver support
- Level 3
  - Runs on multiple families of virtual hardware



# Portability

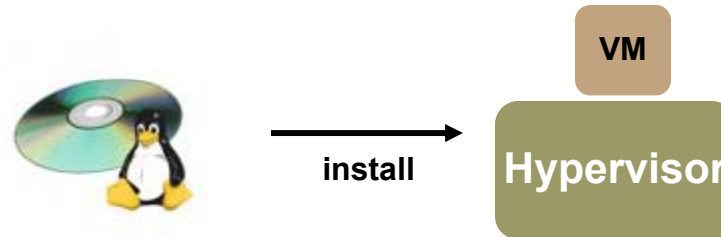
- OVF **supports** Level 1, 2, and 3 interoperability
  - The format does not automatically make virtual machines portable
- Author responsibility
  - Describe concisely in OVF descriptor the level of interoperability
    - Based on target audience
    - Based on guest software capabilities
  - OVF supports multiple virtual hardware profiles to facilitate this
- Importer responsibility
  - Validate virtual hardware descriptor against platform capability
  - Ensures robust user experience

# Example: Level 2 Compatibility



- Exporting a VM will generally yield Level 2 compatibility
  - Virtual Machine expects a specific virtual hardware profile
  - VirtualCenter templates and provisioning operations are Level 2

# Example: Level 3 Compatibility



- An OS installer is generally level 3
  - Broad support for (virtual) hardware
  - Initial boot is Level 3 - then degenerates into Level 2 portability



# How to Create a Level 3 OVF

- Why?
  - ISV interested in as big a touch point as possible
  - Requires guest software to be created in a way that it can adapt to the virtual hardware upon boot
- Several Solutions Possible
  - Software is "sys-prep'ed" to do hardware discovery upon boot
    - Operation can be performed once or every boot
  - Treat appliance as an "application-install"
    - Software in guest is installed on first boot
      - Similar to using a regular OS install approach
    - OVF supports a "post-install" boot option



# OVF and DMTF

- The OVF specification is accepted as a draft specification by the DMTF
- Developed by a subgroup of the SVPC group
- SVPC: System Virtualization Partitioning and Clustering DMTF Workgroup with participation from EMC (VMware), Microsoft, IBM, HP, Sun, Novell, XenSource, Hitachi, Intel and others
- OVF 1.0 specification is expected in the first half of 2008.



# OVF Goals

- **Complete description of Virtual Machine**  
Supports multiple disk formats and virtual hardware platforms.
- **Supports single VM and multi-VM configurations**  
Models multi-tier services consisting of multiple VMs, including composed services.
- **Optimized for distribution**  
Supports verification, signing, versioning, and handling of licensing terms.
- **Optimized for simple, automated user experience**  
Supports validation of entire package, provides descriptive information to user, and includes application level meta-data.



# OVF Goals

- **Vendor and platform independent**  
Does not rely on specific host platform, virtualization platform, or guest operating system properties.
- **Extensible**  
Allows vendor specific meta-data and provides an extensible hardware description model supporting future advances in virtualization technology.
- **Localizable**  
Supports user visible descriptions in multiple locales.
- **Open standard**  
A collaboration of key vendors and will be submitted to an accepted industry forum.



# Conclusion

- VMware's contributions of the "Open Virtual Machine Format" enables the industry to deliver vendor independent Virtual Appliances
- Broad industry collaborations, including Microsoft, Dell, HP, IBM, and XenSource/Citrix.
- VMware is fully committed to the OVF specification and intends to enable OVF support across its product lines.
- Draft OVF Specification available at [www.vmware.com/go/ovf](http://www.vmware.com/go/ovf)