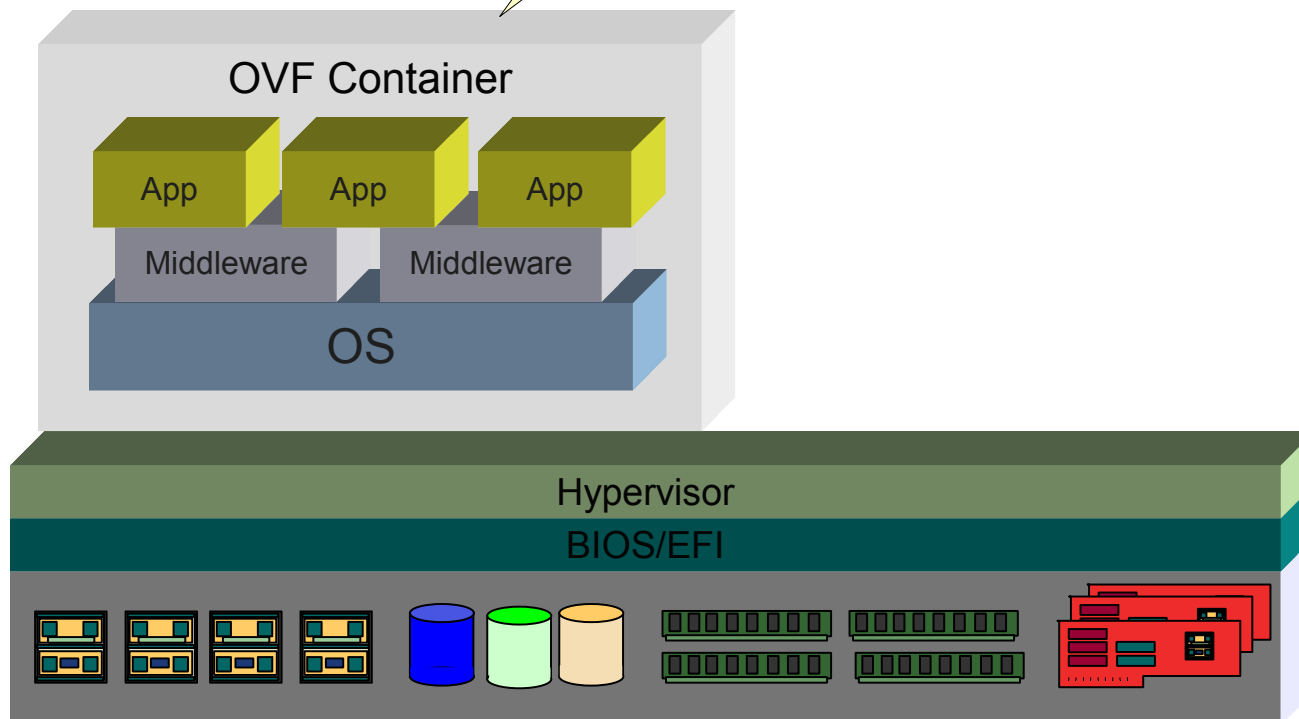


Open OVF Proposal

- **Open-source library and tools to support the OVF**
 - <http://xml.coverpages.org/ni2007-09-11-a.html>
- **IBM will be donating code and invites all to form an open community around OVF**
- **Eclipse Public License**
 - <http://www.eclipse.org/legal/eplfaq.php>
- **Contact:**
 - Mike D. Day <ncmike@us.ibm.com>

Open Virtual Appliance Format (OVF)

- DMTF Standard, Open Packaging format for Virtual Appliances
 - Meta-data, configuration, file systems, disk images, memory images
- XML-encoded
- Security and licensing features
- Enables cross-hypervisor Appliance portability
- Enables a neutral Appliance Ecosystem



Why open-ovf?

- **Leverage development resources by cooperating on common plumbing**
 - Finish the plumbing faster, keep more resources focused on the higher-value solution stack
- **Build out the cross-hypervisor components of OVF**
 - Prevent fragmentation of the OVF across the industry
 - Provide a reference architecture
- **Establish conventions in OVF usage that promote transportability of virtual machines**
- **Promote the usage of OVF and its adoption as a standard VM Appliance format**

OVF Characteristics

- **Tar file with following contents:**
 - .ovf descriptor
 - xml-encoded, basic schema, appliance meta-data
 - Contains URI references to other files
 - disk images, iso, cpio archives, etc.
 - Contains virtual hardware resource descriptions
 - Memory, network interfaces, storage devices, etc.
 - .mf manifest file
 - Digest of sha-1 hashes of referenced files
 - .cert certificate
 - Signature of the manifest

Missing from OVF: Linkage to an actual Hypervisor

"The OVF format supports portable, efficient, and secure distribution of virtual machine templates and virtual appliances. The file format is optimized for these goals. In contrast, the hypervisor requires files that are optimized for efficient execution" ("VMWare OVF Tool" paper). Also "OVFs must be installed before they can be run; a particular virtualization platform may run the VM from the OVF, but this is not required" ("OVF Whitepaper 0.9")

To be useful, OVF needs traction—the ability to install and run VAs--
with KVM, Xen, Linux Containers.

This is a goal of the Project.

Project Goals

- **Provide a complete support library for creating, using, and maintaining OVF appliances**
- **Establish as the de-facto standard toolkit for working with OVF**
- **Establish the interoperability properties of OVF**
 - Encourage the use of appliances across different hypervisors
 - Support automated guest enlightenment
- **Support efforts to advance the OVF specification within the DMTF, especially the environment section**
- **Enable widespread adoption of OVF while preventing fragmentation**

OVF Technical Characteristics

- **Good DTD and Schema Definition**
 - VM description
 - Packaging, signatures, references
 - Composability and Decomposition
 - Create an appliance by collecting (through reference) discrete components
 - Re-use components across multiple appliances
- **Lack of Platform Introspection and Cross-hypervisor Capability**
 - This is the missing “environment” section of the specification
 - Environment: The platform upon which the appliance is hosted
 - We need OVF to support cross-hypervisor operation of Appliances

Technical Focus Areas

■ Primitive support

- create, destroy, read, write OVF packages
- parse, navigate and modify document model of an OVF
 - Like any well-formed XML, an OVF can be parsed to an in-memory document model with branches and leaf objects
- compose from parts, decompose into parts, package parts into an OVF

■ Environment support

- Running appliance needs to situational awareness
 - What hypervisor is hosting me? What other appliances are here?
- XML-RPC server instance provides access to environmental information
- Rendezvous: Appliances need to find each other
- scripting host: provide a posix shell that appliances can use to automate their own deployment.

Environment Support, cont'd.

■ **Guest Enlightenment**

- Appliance kernel uses introspection to discover the underlying hypervisor and replaces its own kernel with a paravirtualized kernel
 - Also replaces paravirtualized drivers if necessary

■ **Workload Enlightenment**

- Appliance workload uses introspection to discover the underlying hypervisor
- Appliance also uses introspection to discover other appliance workloads running on the same server
- Appliance uses hypervisor management interfaces to tune its own workload
 - Scheduling, network configuration, shared memory transport, etc.

Implementation

■ Python

- Rapid development environment
- Good xml support
- Good regexp engine
- Linkage to other languages
 - Java, C
- Amenable to later optimization in C

■ XML-RPC interfaces

- Good linkage to SOA and Web Services