vmware technologyexchange

DE-03

Introduction to vCloud API

Tichomir Tenev

vmware

Disclaimer

This session may contain product features that are currently under development.

This session/overview of the new technology represents no commitment from VMware to deliver these features in any generally available product.

Features are subject to change, and must not be included in contracts, purchase orders, or sales agreements of any kind.

Technical feasibility and market demand will affect final delivery.

Pricing and packaging for any new technologies or features discussed or presented have not been determined.

About the Speaker

Tichomir Tenev, Sr. Staff Engineer, vCloud

- Currently at VMware's Cloud Service Mgmt & Automation Group
- With VMware since 2001
  - Worked on Workstation and vCenter
  - Helped design the vSphere API and several internal APIs and frameworks.
- Prior to VMware, designed and implemented an energy trading system for a startup.
- Education
  - M.Eng. in Electrical Engineering & Computer Science from MIT
  - BS in Computer Science and Mathematics from MIT

Agenda

Salient Points

Context

Key concepts

Deep dive

vCloud API Salient Points

vmware technologyexchange

Standardizable

- Platform independent
  - Can be used from virtually any platform
  - Can be implemented on a variety of platforms
- Straightforward to use and implement
- Powerful and extensible
- Standards-based

Pure-virtual

- Resource semantics described independent of physical semantics
  - Physical infrastructure is hidden
  - Virtualization features are obscured
- Simpler semantics focused on end-use and not on virtualizing

vCloud API Features

Interface for:

- Providing and consuming resources in the Cloud
- Deploying and managing virtualized workloads in the Cloud
- Migrating virtualized workloads between Clouds.

REST-based

Multi-tenant

Self-service

Service provider automation

vCloud API and vSphere API

vSphere API –

“Under the Hood” API

- Used to create virtual resources
- Virtualization API
  - Exposes physical<->virtual mapping
- Targeted at sys-admins
- Product specific
  - Exposes vSphere’s capabilities
  - Tied to a specific implementation
- Rich and powerful
- VMware’s Cloud OS Platform API

vCloud API –

“Driver Seat” API

- Uses virtual resources
- Pure-virtual API
  - Hides underlying physical resources
- Targeted at cloud tenants
- Product agnostic
  - Standardizable
  - Variety of implementations
- Simple yet powerful
- Implementable on the vSphere API
- Does not replace the vSphere API

vCloud Ecosystem

Content Providers

- Developers, IT Admins, ISVs, Enthusiasts

Content

- Virtualized Software Solutions (vApps)
- Existing legacy solutions, or new apps written for the Cloud

Content Respositories

- Web Servers, Object Stores, Virtual Appliance Marketplace

Cloud Service Providers

Enterprises

vApp – Next generation VM concept

vmware technologyexchange

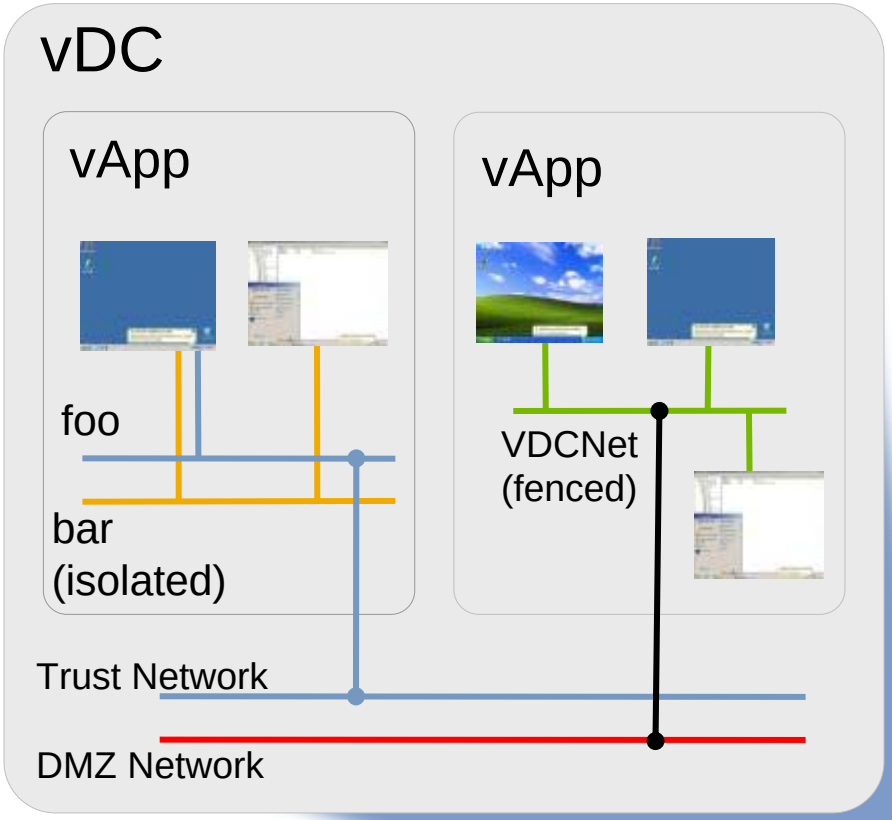
An uplifting of a virtualized workload

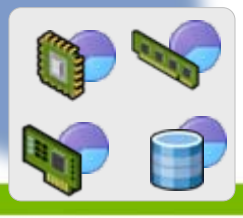
- VM = Virtualized Hardware Box
- vApp = Virtualized Software Solution
- Encapsulation, isolation and mobility higher up the stack

Properties:

- Comprised of one or more VMs
- Encapsulates requirements on the deployment environment
- Distributed as an OVF package

vDC – vApp Deployment Environment





Type of Commodity

- Compute, Storage and Network SLAs

Quantity

- MB of RAM, MHz of CPU, GB of Storage

Scope for Over-provisioning

Other features

- L2 Networks
- Persistent vApps

Deep Dive

Simple Example: Power On

Request

```
POST
https://vcloud.example.com/vapp/413/power/action/powerO
n
```

Simple Example: Power On

Request

```
POST
https://vcloud.example.com/vapp/413/power/action/powerO
n
```

Simple Example: Power On

vmware technologyexchange

Response

```
202 Accepted

<?xml version="1.0" encoding="UTF-8"?>
<Task href="https://vcloud.example.com/task/389"
    type="application/vnd.vmware.vcloud.task+xml"
    startTime="2009-7-31T09:30:47Z"
    status="running" ...>

    <Link rel="task:cancel”
      href="htt.../task/389/action/cancel"/>

    <Owner href="https://vcloud.example.com/vapp/413"
      type="application/vnd.vmware.vcloud.vapp+xml“
      name="My vApp"/>
</Task>
```

Simple Example: Look at a vApp

Request

```
GET https://vcloud.example.com/vapp/413

Content-type: vnd.vmware.cloud.vapp+xml
```

Simple Example: Look at a vApp

Request

```
GET https://vcloud.example.com/vapp/413

Content-type: vnd.vmware.cloud.vapp+xml
```

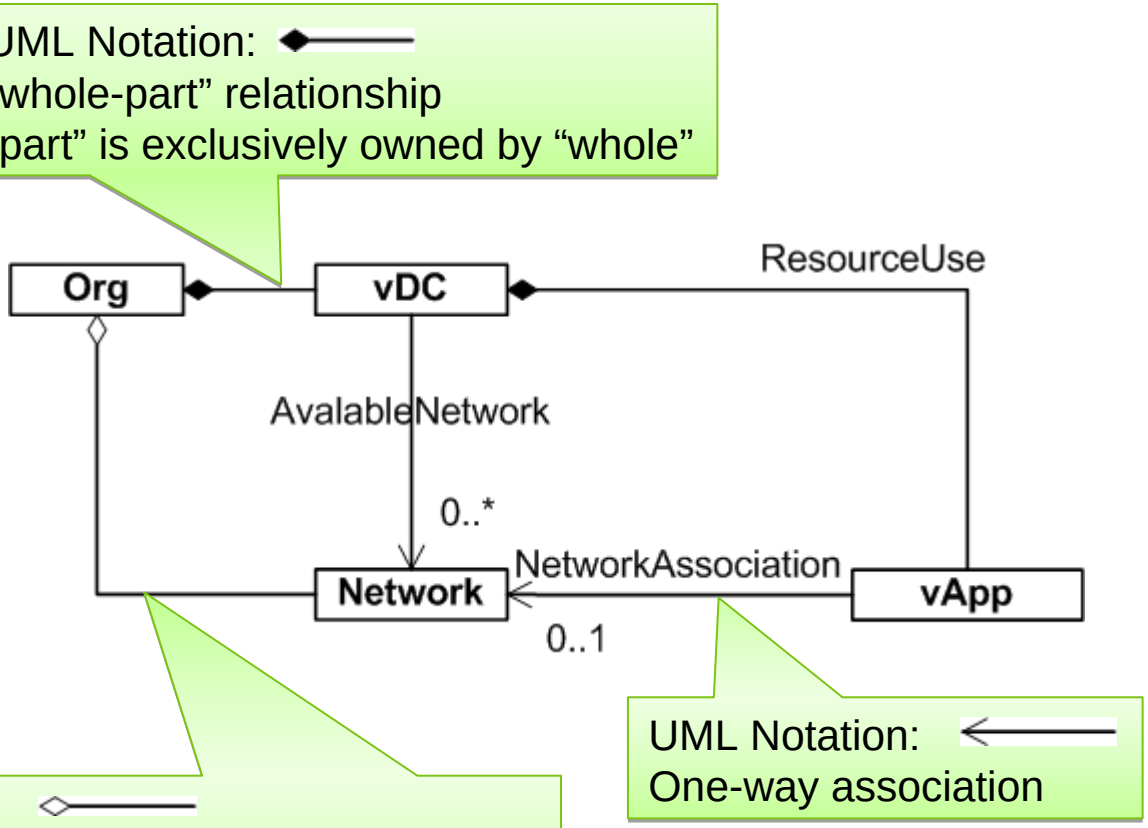
Simple Example: Look at a vApp

Response

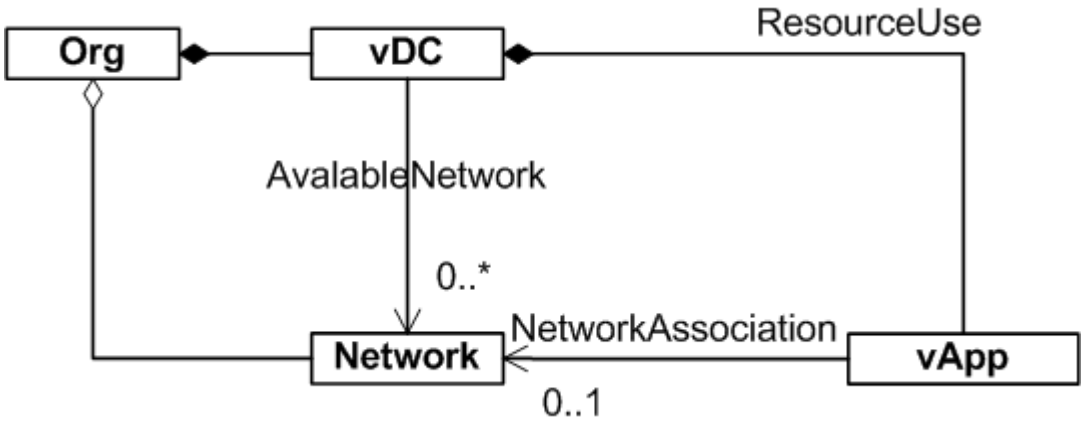
```
<VApp name="My vApp"
    status="1"
    href="https://vcloud.example.com/vapp/413" ...>

  <Link rel="up" href= "https://vcloud.example.com/vdc/128"/>
  <NetworkSection>...</NetworkSection>
  <ovf:OperatingSystemSection ...>
     <Link rel="edit" href="http..." ... />
     <Description>Microsoft Windows Server 2003</Description>
  </ovf:OperatingSystemSection>
  <ovf:VirtualHardwareSection ovf:transport="iso">
     <Link rel="edit" href="http..." ... />
     <Item>…</Item>
     …
  </ovf:VirtualHardwareSection>
</VApp>
```

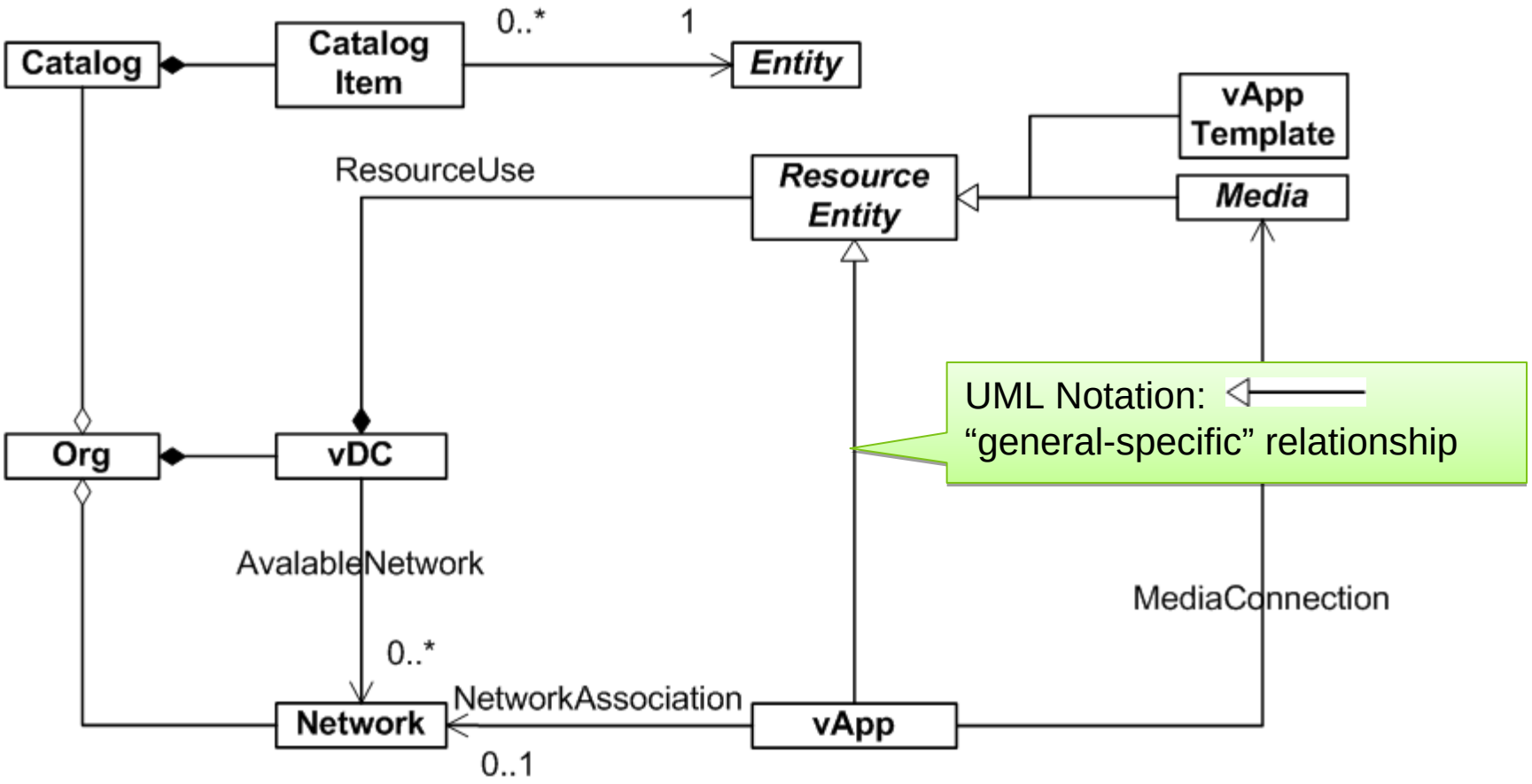
Entity Model



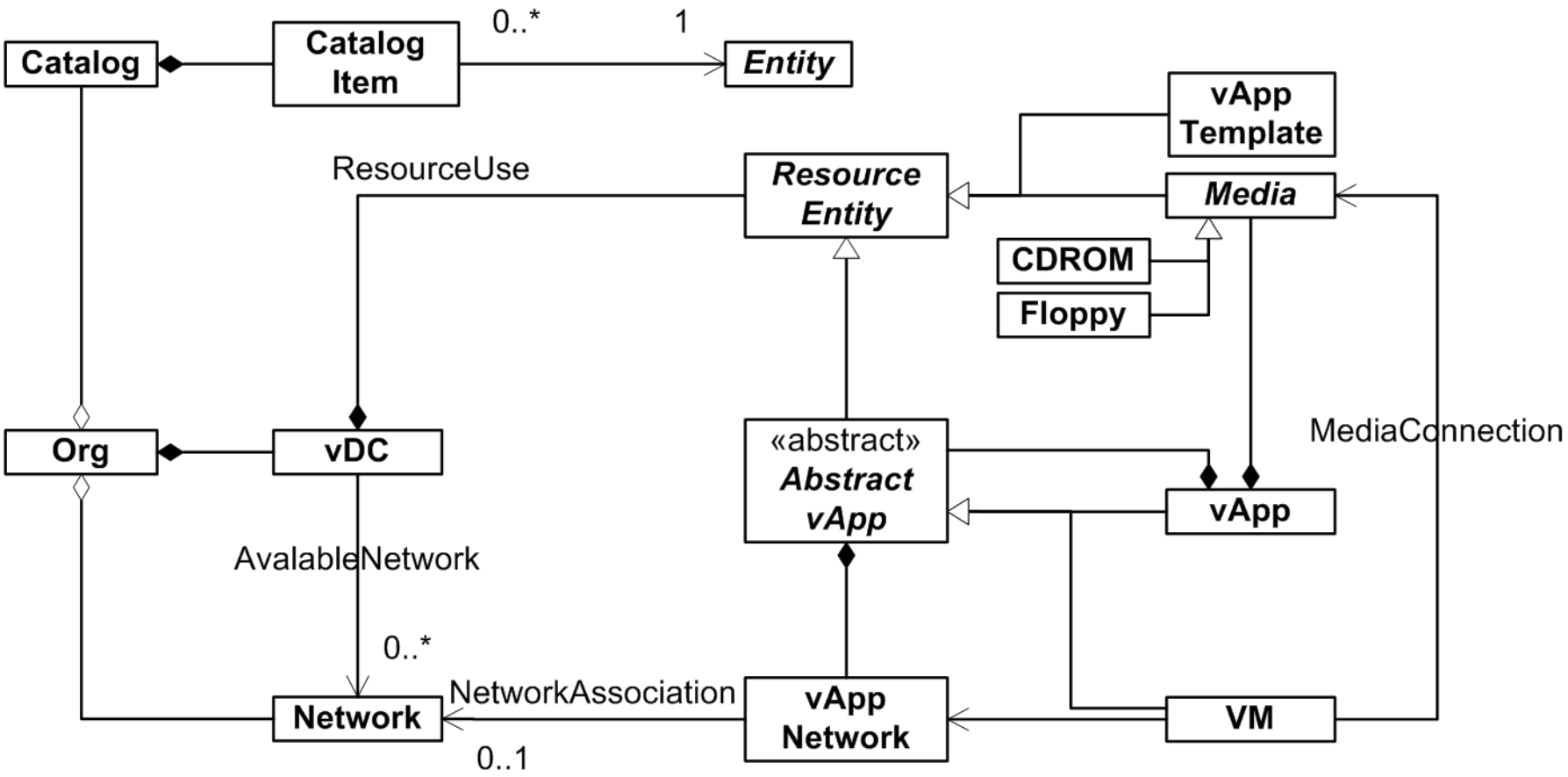
Entity Model



Entity Model



Entity Model



Example Workflow:
Instantiating a vApp Template

Finding a vApp Template

Request

```
GET https://vcloud.example.com/vdc/128
```

Finding a vApp Template

Response

```
<Vdc href="https://vcloud.example.com/vdc/128"
    name="Main Vdc" ...>

    <ResourceEntities>
       <ResourceEntity
          href="https://vcloud.example.com/vAppTemplate/111"
          type="application/vnd.vmware.vcloud.vAppTemplate+xml"
          name="Ubuntu Template with vsftpd"/>
       <ResourceEntity href="https://vcloud.example.com/media/112"
          type="application/vnd.vmware.vcloud.media+xml"
          name="Ubuntu Boot Floppy"/>
       <ResourceEntity href="https://vcloud.example.com/media/113"
          type="application/vnd.vmware.vcloud.media+xml"
          name="Ubuntu ISO Image"/>
    </ResourceEntities>
   ...
</Vdc>
```

Get Information about a vApp Template

vmware technologyexchange

Request

```
GET https://vcloud.example.com/vAppTemplate/111
```

Response

```
<VAppTemplate
href="https://vcloud.example.com/vAppTemplate/111"
   name="Ubuntu Template with vsftpd"
   status="1"
   ...>
   <Description>Description of Ubuntu Template with
vsftpd</Description>

</VAppTemplate>
```

Instantiating a vApp Template

Request

```
POST
https://vcloud.example.com/vdc/128/action/instantiateVAppTemplate

<InstantiateVAppTemplateParams
   name="Linux FTP server" ...>
   <VAppTemplate href="https://.../vAppTemplate/111" />
   <InstantiationParams ...>
      <NetworkConfigSection>
         <NetworkConfig name="My vApp Net">
            <Features>
               <vmw:FenceMode>allowInOut</vmw:FenceMode>
               <vmw:Dhcp>true<vmw:Dhcp>
            </Features>
            <NetworkAssociation href="https://.../network/14">
         </NetworkConfig>
      </NetworkConfigSection>
   </InstantiationParams>
</InstantiateVAppTemplateParams>
```

Instantiating a vApp Template

Response

```
<VApp name="Linux FTP server"
      status="0"
      ...>
    <Link rel="up" href="https://vcloud.example.com/vdc/128"
          type="application/vnd.vmware.vcloud.vdc+xml" />
    ...
</VApp>
```

Example Workflow:
Upload/Download of a vApp Template

Upload Example: Initiating the upload

Request

```
POST https://vcloud.example.com/vdc/128/action/uploadVAppTemplate

<UploadVAppTemplateParams name="Example Image" ...>
   <Description>Example VApp Template</Description>
   <ovf:Envelope>
      <ovf:References>
         <ovf:File ovf:href="disk0.vmdk"
            ovf:id="file1"
            ovf:size="1950489088"/>
        ...
      </ovf:References>
      ...
  </ovf:Envelope>
</UploadVAppTemplateParams>
```

Upload Example: Initiating the upload

Response

```
<VAppTemplate name="Example Image" status="0"
    href="https://vcloud.example.com/vAppTemplate/268"...>
    ...
    <Files>
       <File name="example.mf" bytesTransferred="0" ...>
          <Link rel="upload:default"
            href="https://../adfe4..55cbdef/example.mf"/>
       </File>
       <File name="example-disk0.vmdk" size="1950489088"
            bytesTransferred="0">
          <Link rel="upload:default"
            href="https://../adfe4..55cbdef/disk0.vmdk"/>
       </File>
       ...
    </Files>
</VAppTemplate>
```

Upload Example: Sending the data

Using simple PUT

```
PUT https://../adfe4..55cbdef/disk0.vmdk
Content-length: 1950489088

[disk0 data]
```

Using ranged PUT

```
PUT https://../adfe4..55cbdef/disk1.vmdk
Content-Range: 105639111-205639110/205639111
Content-length: 100000000

[disk0 data range]
```

Download Example: Finding the download URL

Request

```
GET https://vcloud.example.com/vAppTemplate/222
```

Response

```
<VAppTemplate name="Ubuntu Image with vsftpd"
   href="https://vcloud.example.com/vAppTemplate/112"
   status="1"...>
   ...
   <Link rel="data"
      href="https://vcloud.example.com/ad3...782a/example.ovf"
      type="application/ovf+xml" />
   <Link rel="data"
      href="https://vcloud.example.com/ad3...782a/example.ova"
      type="application/ova+xml" />
</VAppTemplate>
```

Download Example: Getting the data

Getting the OVF descriptor

```
GET https://vcloud.example.com/ad3...782a/example.ovf
```

Response

```
<Envelope ...>
  <ovf:References>
    <ovf:File ovf:href="disks/example.vmdk"
       ovf:id="file1" ovf:size="1950489088"/>
  </ovf:References>
  ...
</Envelope>
```

Getting the <References> files

```
GET https://vcloud.example.com/ad3...782a/disk/example.vmdk
```

Response

```
[example.vmdk data]
```

Complete list of operations

vmware technologyexchange

vApp Operations

POST <vapp-uri>/action/{deploy, undeploy}
POST <vapp-uri>/power/action/{powerOn, powerOff}
POST <vapp-uri>/power/action/{reset, suspend}
POST <vapp-uri>/power/action/{shutdown, reboot}
GET <vapp-uri>/screen
POST <vapp-uri>/screen/action/acquireTicket

vApp Configuration Operations

POST <vapp-parent-element-uri>
DELETE <vapp-element-uri>
PUT <vapp-element-uri>

Inventory Listing

GET <vapp-uri>
GET <vdc-uri>
GET <vAppTemplate-uri>
GET <media-uri>
GET <network-uri>

Catalog Management

GET <catalog-uri>
POST <catalog-uri>/catalogItems

Upload/Download/Provisioning Operations

POST <vdc-uri>/action/composeVApp
POST <vdc-uri>/action/instantiateVAppTemplate
POST <vdc-uri>/action/instantiateOvf
POST <vdc-uri>/action/annotate
POST <vdc-uri>/action/uploadVAppTemplate
POST <vdc-uri>/media
PUT <upload-uri>
GET <download-uri>
DELETE <resourceEntity-uri>

Task Management

GET <tasks-list-uri>
GET <task-uri>
POST <task-uri>/action/cancel

Service Provider Automation

Create/Delete/Update Org
Create/Delete/Update vDC for an Org
Create/Delete/Update an Org Network
Create/Delete/Update Org Catalogs
Create/Delete/Update Users, Groups, Roles

Q & A