



Practical application of Xen management API with Light Weight Language (JRuby)

November 20, 2008

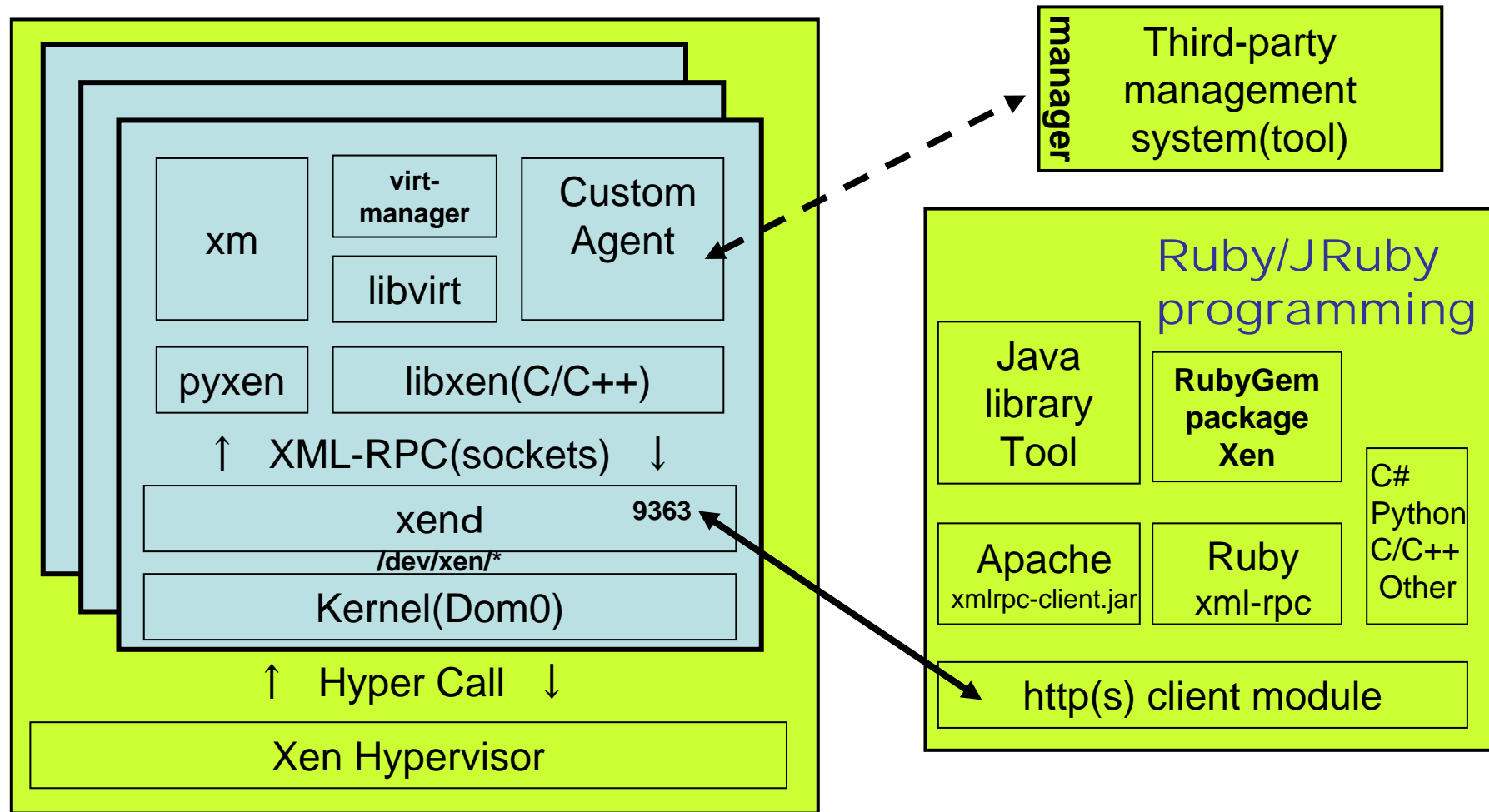
Shinetsu Isawa

ITOCHU Techno-Solutions Corporation (CTC)
Platform Products Promotion & Engineering Dept
IT Engineering Division
Tokyo, Japan

XenAPI 1.0 Overview

- **Standardize data model**
manipulate distributed persistent objects
- **Standardize XML-RPC**
Standard wire protocol for Remote API Calls
Language Bindings (with any language)
Sent over HTTP/S, Stateful session-based
- **legacy interfaces Not recommended**
xend-http-server(8000), xend-tcp-xmlrpc-server(8006)
- **Stable over the long term**
xen-api-server(9363) All new, all shiny Xen-API interface
released officially with Xen 3.1.0

XenServers & Management distributed vs centralized

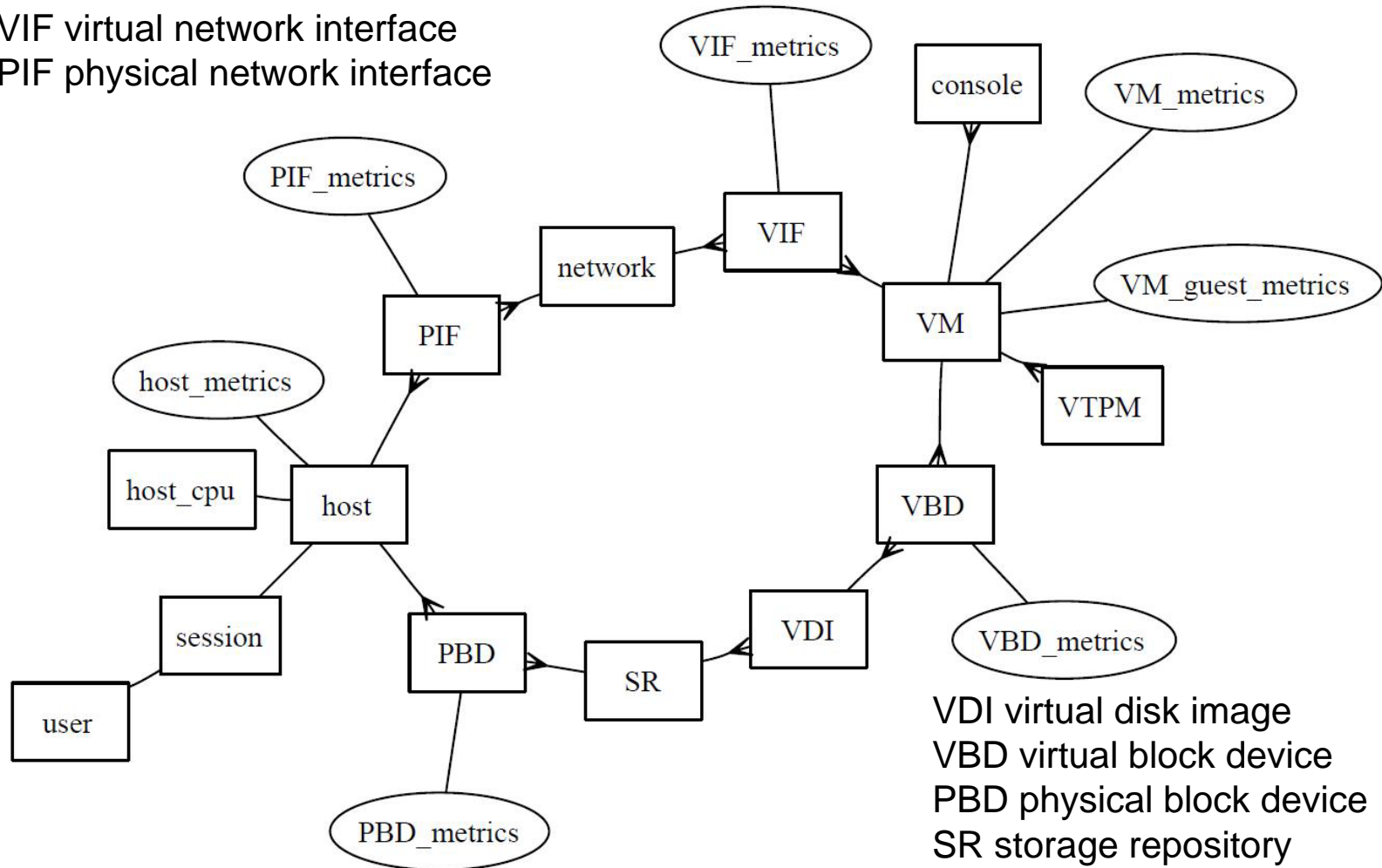


Class relationships

network

VIF virtual network interface

PIF physical network interface



Class methods(example)

- getter & setter

Properties

uuid,label,state,other_config(stringMap type),etc...

reference to related objects(ex: metrics)

MAC,MTU,qos,device_config,status_code

- For VM

start,(un)pause,(hard/clean)_shutdown,(hard/clean)_reboot,suspend
,resume,clone,etc...

- For metrics

memory/total/actual/free,VCPUs/number,last_updated,etc...

- Other

create,destory,plug,unplug,get_all,location

Programming

- **References vs UUID**

References : encoded as XML-RPC strings on the wire (opaque types)
server generate them, understood only by the server (valid in session)
UUID : permanent ID names for objects (OSF DCD format)
For translating between them : `get_by_uuid`, `get_uuid`

- **XML-RPC Library**

`session.login_with_password(string uname, string pwd)`
Set up input parameters, execute with them and get the result
The result may be complex data structure like `HashMap`
May need some manipulation to get the value from it

- **High-level Class Library**

Create host object (login to xen host) and get the handle
Get the reference of target object and call its method
(get from property, create, `find_by_name uuid...`)

Setup for play

- **Xen Server side configuration**
configure the ports, interfaces, and access controls
`/etc/xen/xend-config.sxp`
`(xen-api-server ((0.0.0.0:9363)))`
and then restart xend
- **Management side setup**
JDK setup (1.5 or later)
JRuby download from <http://dist.codehaus.org/jruby/>
`tar zxf jruby-bin-1.1.3.tar.gz -C /usr`
Add PATH (`usr/jruby-1.1.3/bin`)
`jruby -S gem install xen`

Programming sample

```
require "rubygems"  
gem 'xen'  
require 'xen'  
  
xh = Xen::Host.new('192.168.11.3', 9363)  
  
xh.vms.each do |vm|  
  puts "VM UUID: " + vm.uuid + " Name-label: " + vm.name unless vm.is_dom0?  
end
```

```
puts "Enter Guest OS name for reboot"  
vmin = gets.chomp  
xh.find_vm(vmin).clean_reboot!
```

Error handlings are omitted for simplify

Reference: [lib/ruby/gems/1.8/gems/xen-0.1.2.1/README](https://github.com/rubygems/gems/blob/master/xen-0.1.2.1/README)

“xm create” command and API

Edit RubyGem Package

lib/ruby/gems/1.8/gems/xen-0.1.2.1/lib/xen/vm.rb

:actions_after_shutdown => 'destroy'

:PV_kernel => '/boot/vmlinuz-2.6.18-xen'

:PV_args => 'root=/dev/sda1 ro 3'

Create Ruby main program

```
require "rubygems"
```

```
gem 'xen'
```

```
require 'xen'
```

```
h = Xen::Host.new('192.168.11.3', 9363 )
```

```
network =Xen::Network.find_by_name "eth0", h
```

```
vm = h.create_vm "tty", 128*1024*1024
```

```
vdi_sda1 = h.create_vdi "vdi1", "file:/xen/tty/rootfs"
```

```
vbd1 = h.create_vbd "sda1", vm, vdi_sda1
```

```
vif = h.create_vif vm, network, "00:30:48:88:81:07"
```

```
vm.start!
```

```
kernel = "/boot/vmlinuz-2.6.18-xen"
memory = 128
name = "tty"
vif = [ " ]
dhcp = "dhcp"
disk = ['file:/xen/tty/rootfs,sda1,w']
root = "/dev/sda1 ro"
```

The diagram consists of a light blue box on the right containing configuration parameters. Arrows point from this box to specific lines in the code on the left: one arrow points from 'kernel = "/boot/vmlinuz-2.6.18-xen"' to ':PV_kernel => "/boot/vmlinuz-2.6.18-xen"', another from 'memory = 128' to 'vm = h.create_vm "tty", 128*1024*1024', a third from 'root = "/dev/sda1 ro"' to ':PV_args => "root=/dev/sda1 ro 3"', and a fourth from 'disk = ["file:/xen/tty/rootfs,sda1,w"]' to 'vdi_sda1 = h.create_vdi "vdi1", "file:/xen/tty/rootfs"'. Additionally, an arrow points from the 'disk' line to 'vbd1 = h.create_vbd "sda1", vm, vdi_sda1'.

References

- **Xen management API project**
<http://wiki.xensource.com/xenwiki/XenApi>
- **Xen Summit April 2007**
http://www.xen.org/files/xensummit_4/XenSummit_API_Slides_2007-04-18_Ewan.pdf
- **English manual**
<http://wiki.xensource.com/xenwiki/XenApi?action=AttachFile&do=get&target=xenapi-1.0.6.pdf>
- **Japanese manual**
<http://gisawa.googlepages.com/XenAPIJ.pdf>