

NLVMUG– 16 maart 2017

VMUG[®] USERCON



VMUG **USERCON**



vROAPI

the Personal Information Processor
for datacenter automation survival



“Power is gained by

SHARING KNOWLEDGE

Not hoarding it”



Ruurd Keizer
@RuurdKeizer
devops.lol
vroapi.net

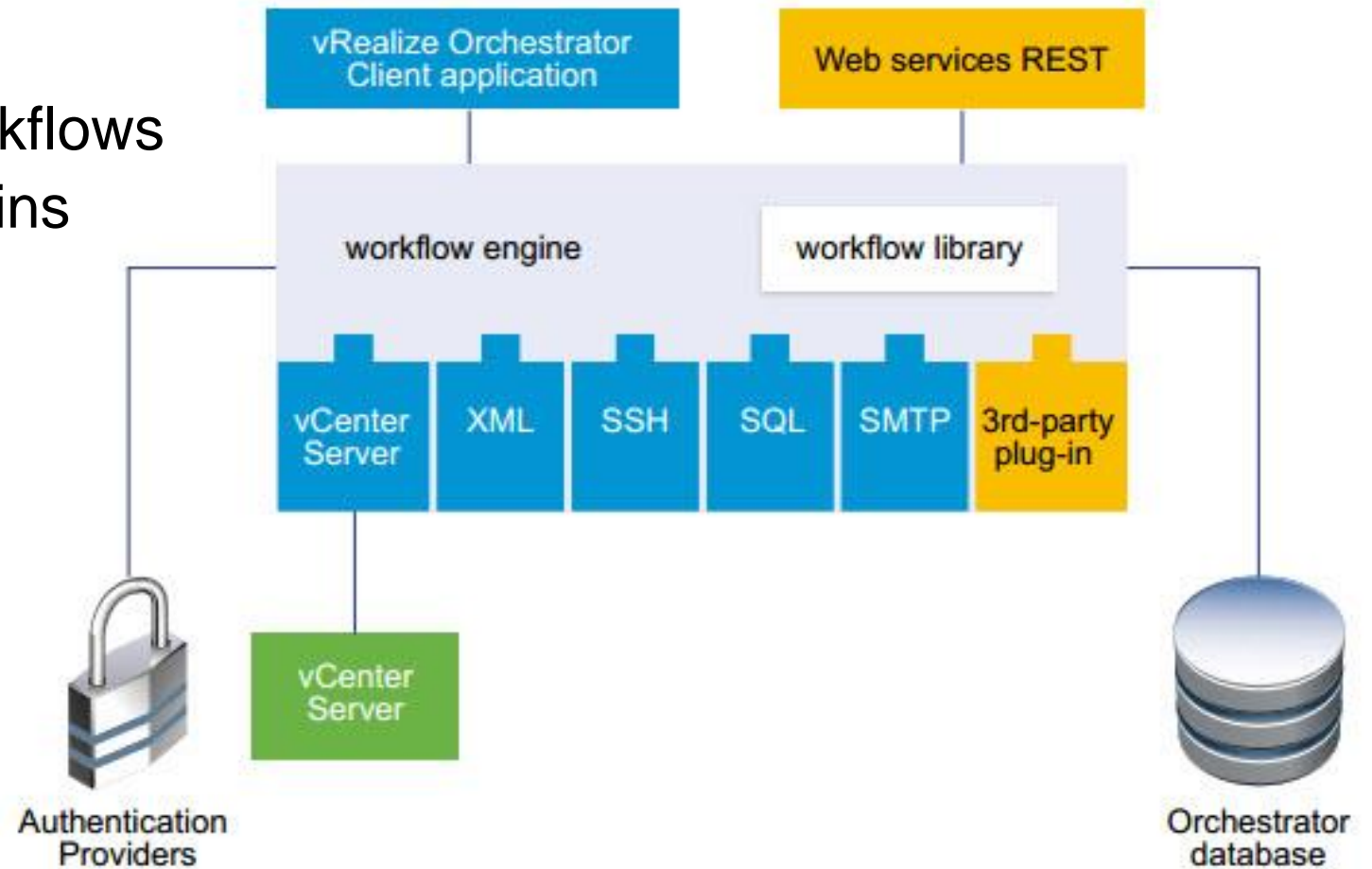
vRO ?



*“vRO is a **free** vCenter feature that helps automate VMware vCloud Suite, vSphere and other virtualization management system processes.”*

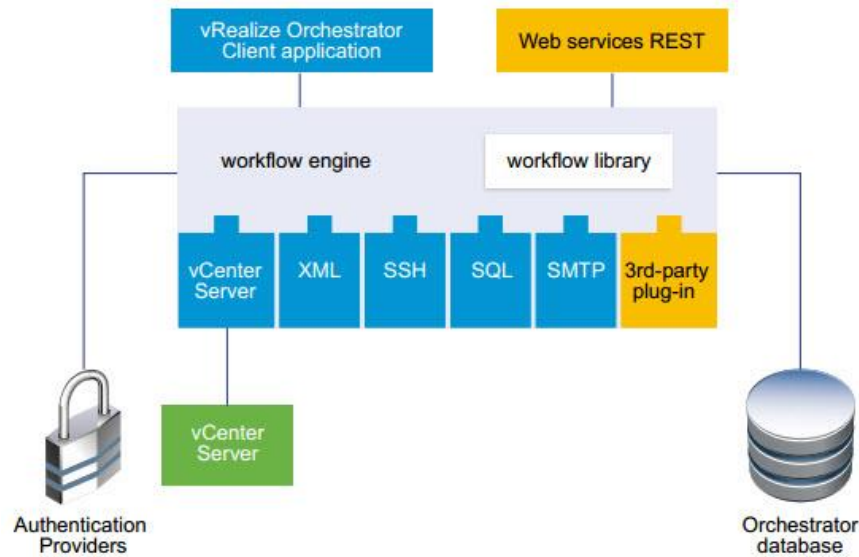
Architecture

- Workflow engine
 - Predefined & custom workflows
 - Extensibility through plugins



vCenter feature

Workflows can be enabled on vSphere resources

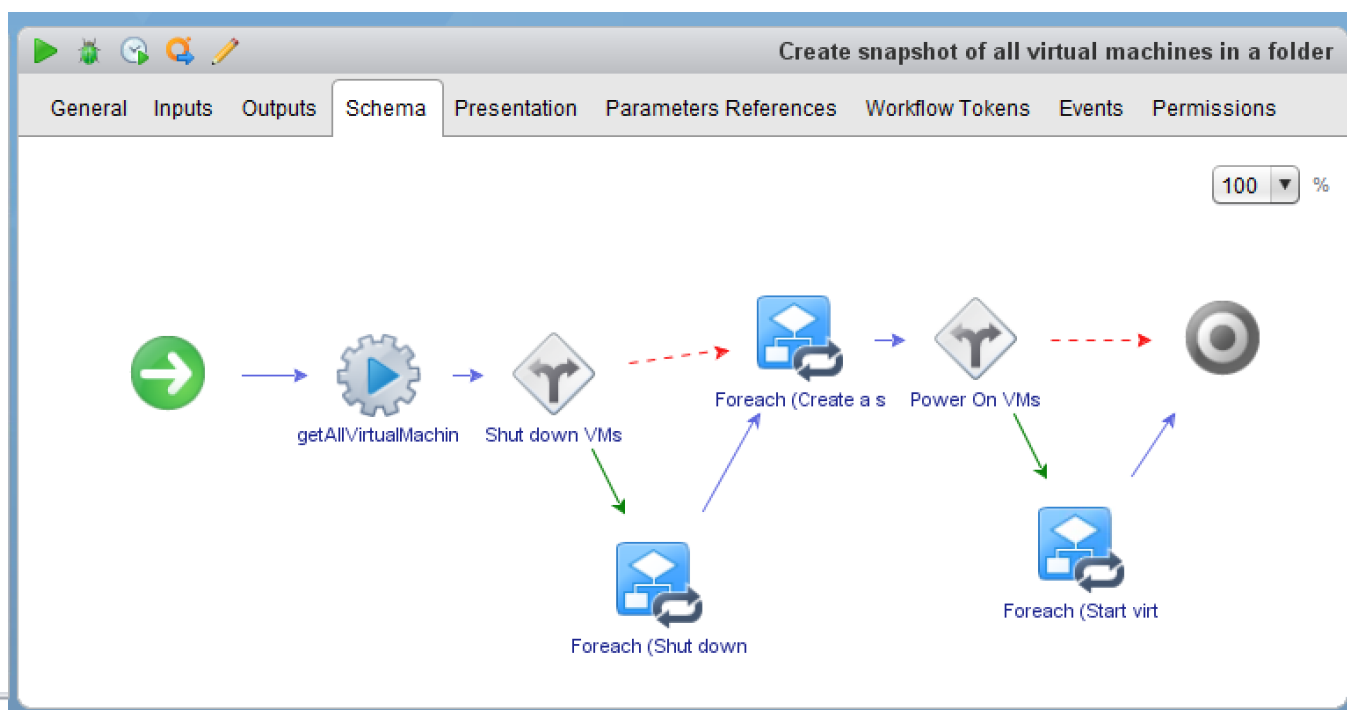


The screenshot shows the VMware vSphere Web Client interface. The 'Navigator' pane on the left shows the hierarchy: Home > vCenter.dr-lab-a.local > Lab - A > Cluster 1. The 'Actions' menu for Cluster 1 is open, listing various operations such as 'Add Host...', 'New Virtual Machine', 'New vApp', 'New Resource Pool...', 'Deploy OVF Template...', 'Storage', 'Host Profiles', 'Edit Default VM Compatibility...', 'Assign License...', 'Settings', 'Move To...', 'Rename...', 'Tags', 'Add Permission...', 'Alarms', 'Delete', and 'Update Manager'. A sub-menu for 'All vRealize Orchestrator plugin Actions' is also visible, listing actions like 'Add DRS virtual machine group to cluster', 'Add virtual machines to DRS group', 'Move virtual machines to another vCenter Ser...', and 'Remove virtual machines from DRS group'. The 'Recent Tasks' table at the bottom is empty.

Task Name	Target	Status	Initiator
-----------	--------	--------	-----------

Customization

- Simple use cases
 - Workflow editor (drag & drop)
- Advanced scenarios
 - Custom scripts (Javascript)



```
VC:VirtualMachine mapToVCVM(Properties vmProperties)
```

```
var type = 'VC:VirtualMachine';  
var vmMoRefId = vmProperties.get('EXTERNAL_REFERENCE_ID');  
var foundVms = System.getModule("com.vmware.vcac.asd").findVcObjectsByMoRefId(type, vmMoRefId);  
if (!foundVms || foundVms.length == 0) {  
    System.warn('No VMs with MoRefId ' + vmMoRefId + ' were found');  
    return null;  
} else {  
    if (foundVms.length > 1) {  
        System.warn('More than one VM with MoRefId ' + vmMoRefId + ' was found');  
    }  
    return foundVms[0];  
}
```

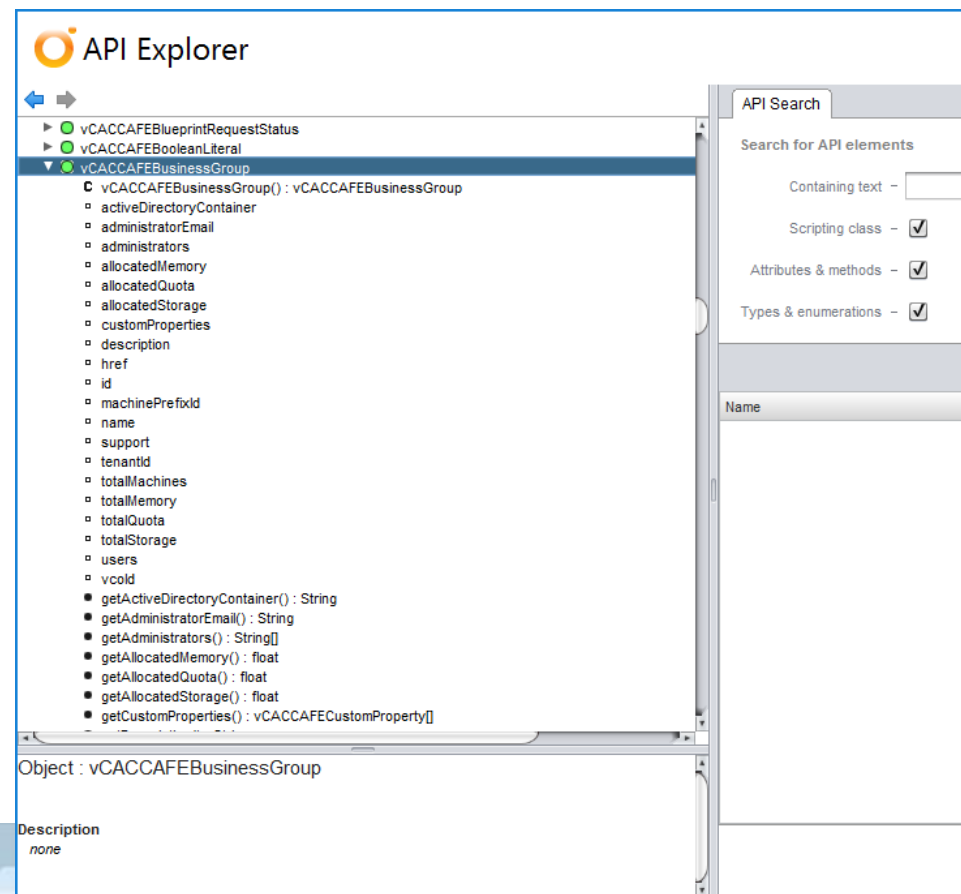

So far so good.

Let's start coding.

And you will find out...

The API explorer **SUCKS**

- Modal
 - you can not work on your code while the search is open
- No reverse type search
 - which methods give back this class?
- Not available outside vRO client
 - the client always has to be open
- Can only search in installed plugins
 - no version comparison



So we wrote our own

- Reverse type search!
 - which methods give back this class?
- Available in the browser!
 - always there
- Can search in all plugins and versions!
 - including version comparison
- Free!



vROAPI.NET

- Demo
 - Search
 - Explore
 - Immediate info on how to get the type



*“vRO
Pipboy”*

vROAPI.NET architecture

- .NET Core
 - Runs on Linux
- Responsive UI
 - Works on Mobile
- In memory database
 - FAST
- 40 MB RAM footprint



*“vRO
Pipboy”*

Conclusion

- vRO is an extremely powerful tool
- The editor and API explorer suck
- Use vROAPI.net

(still looking for sponsors)

Questions?
Thanks for Attending!

