The Design of
The Virtual Labs Hosting Environment

(A Few Thoughts)

VLEAD - IIIT, Hyderabad

Internal Review Meeting

12-November-2013

The Design Intent

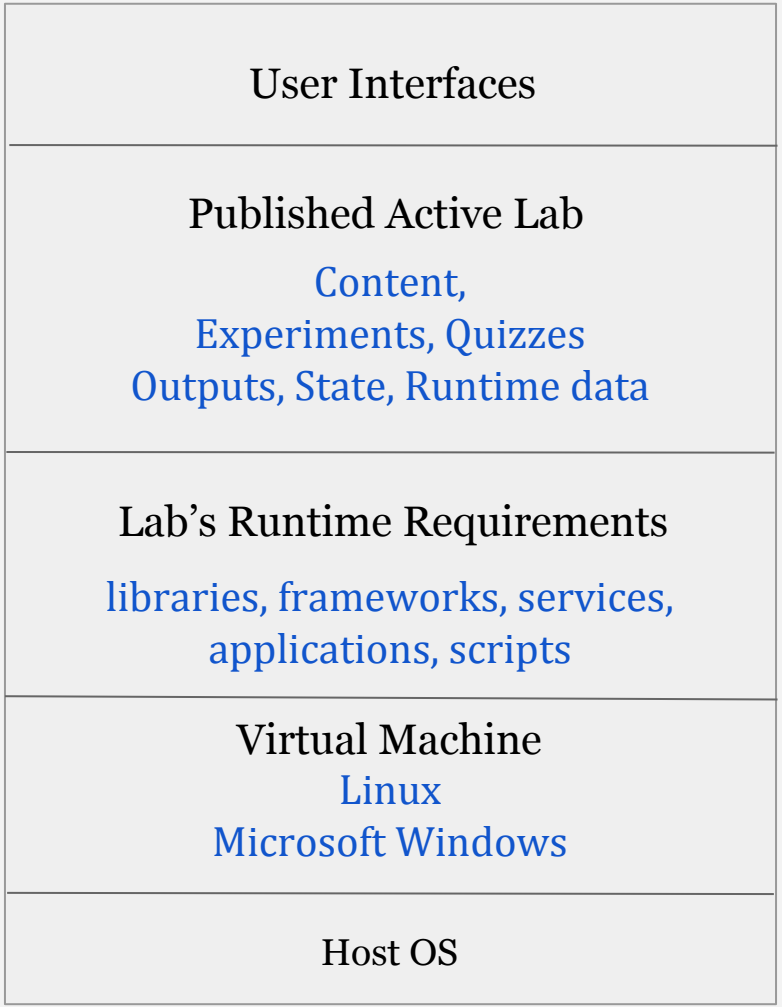
- Enable scalability
- Improve availability
- Improve security

Additional Services

- Lab monitoring
- Lab backup
- VM snapshots
- Stats

Scalability and Availability

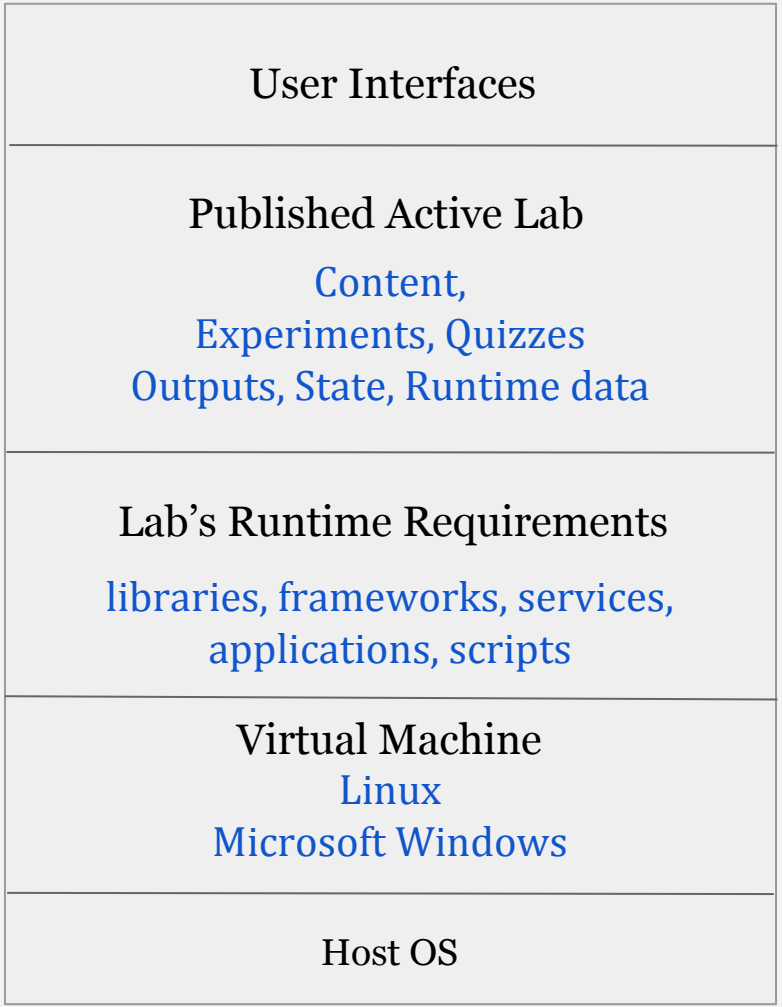
- Enable varieties of Lab-VM configurations
  - One lab per VM
  - One lab in multiple VMs
  - Many labs in one VM
  - Many labs in many VMs



A lab is accessed using one or more URLs. These URLs may point to different instances of a lab hosted in different ways for scalability and availability reasons,

It is possible that the users interacting with a lab generate data that is stored as a part of the lab state. This data can be backed up anytime.

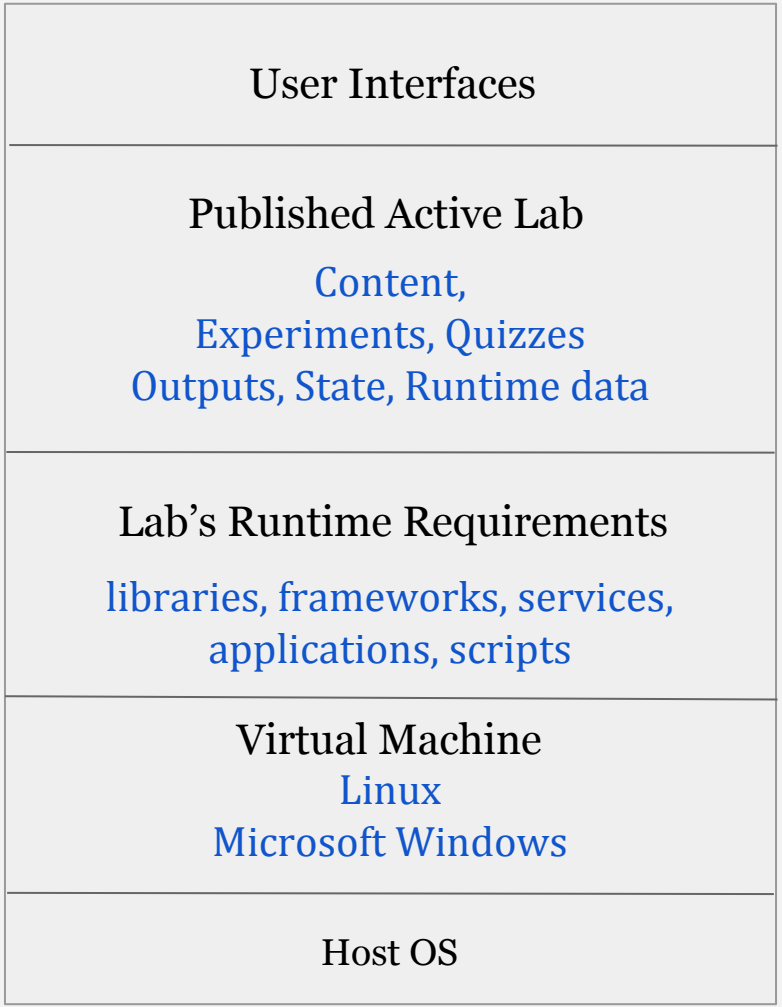
The RAM and Disk requirements are specified when a VM is created. These *may be* reconfigured even after publishing the lab



LDAP Integration

WebProxy, access policies, auditing

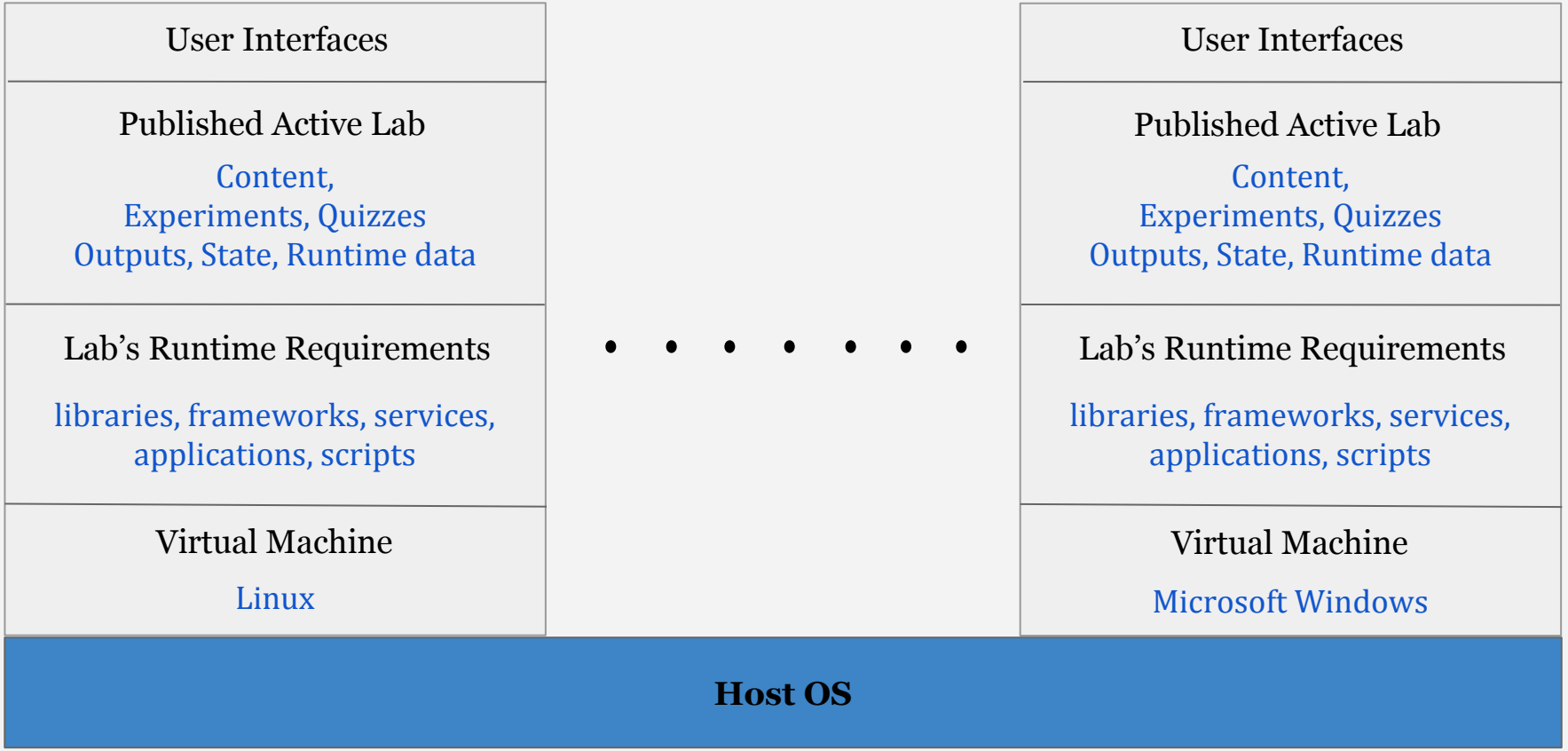
Firewall, network analyzer, encryption, SSH, gateways

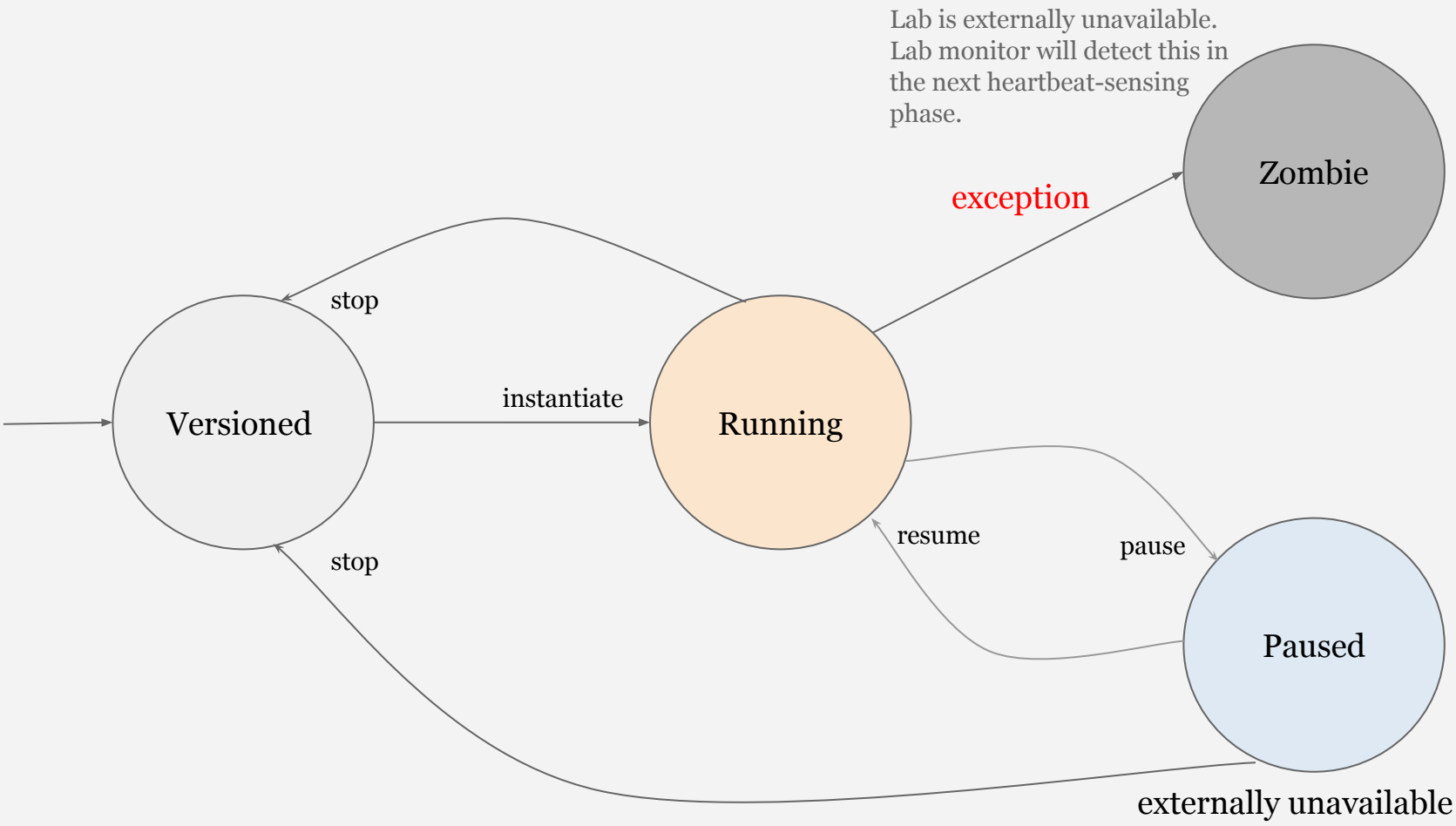


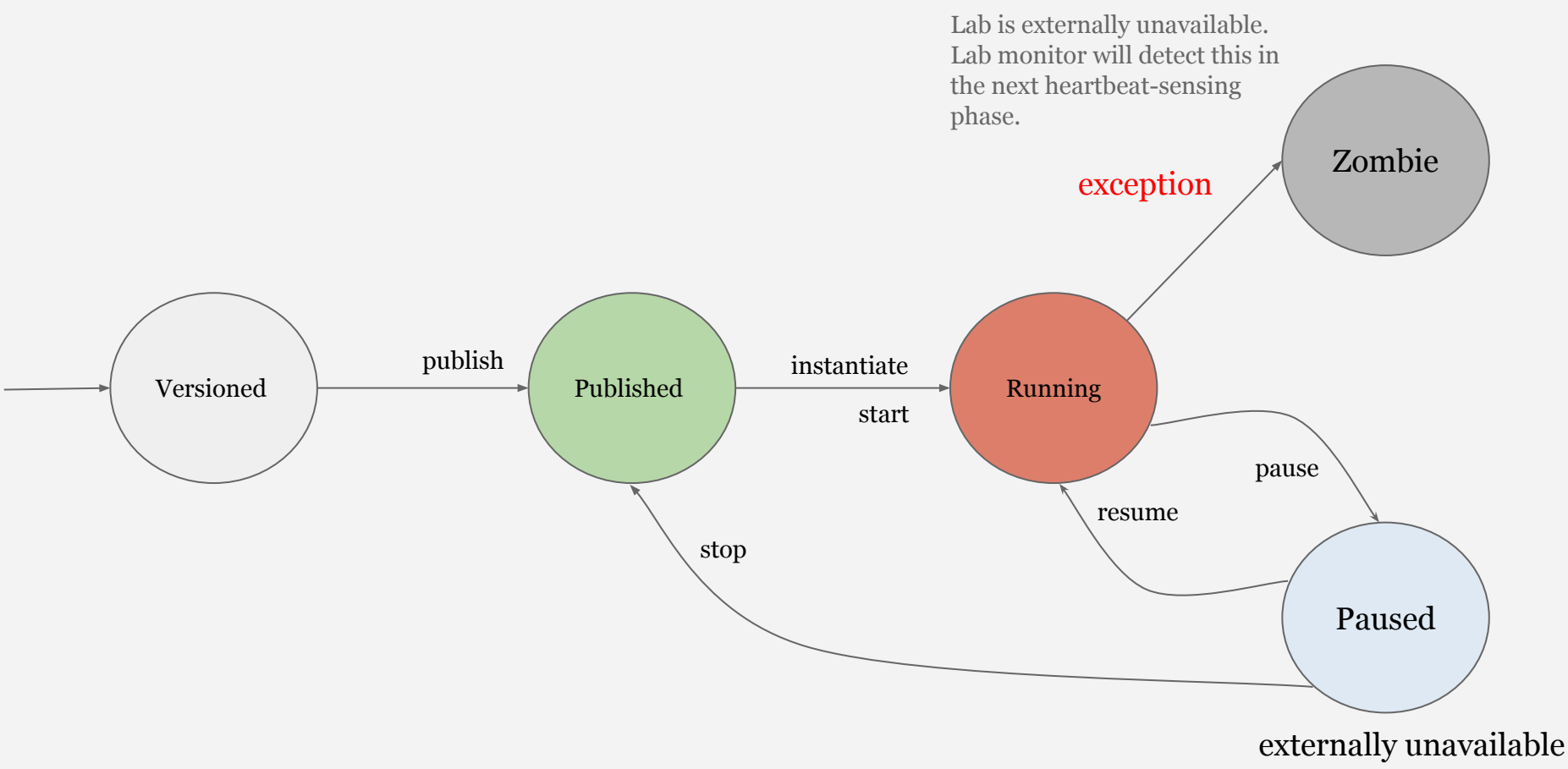
Multiple lab instances

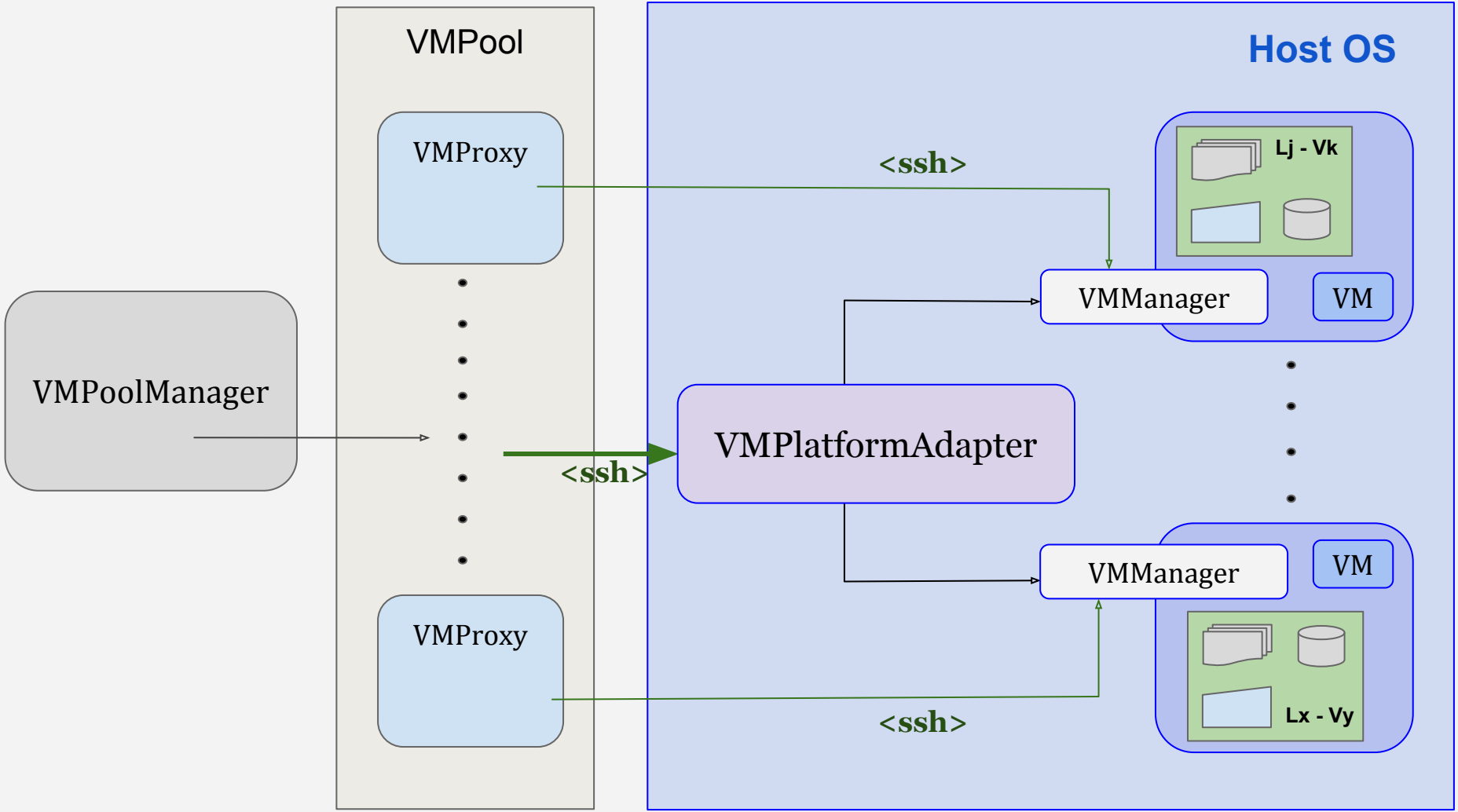
Multiple VMs per Lab

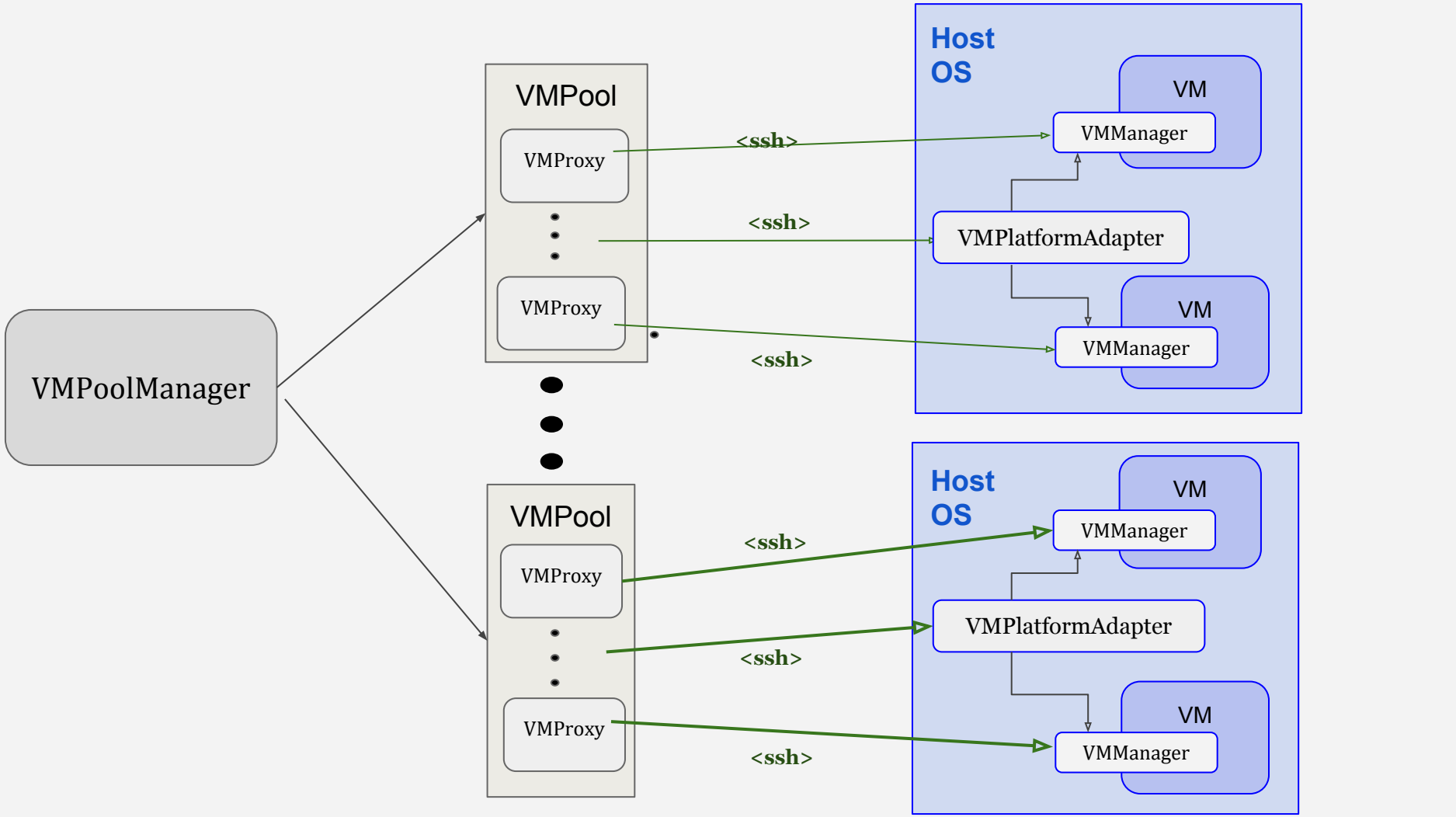
Load balancer, RAID, power backup, battery and cache

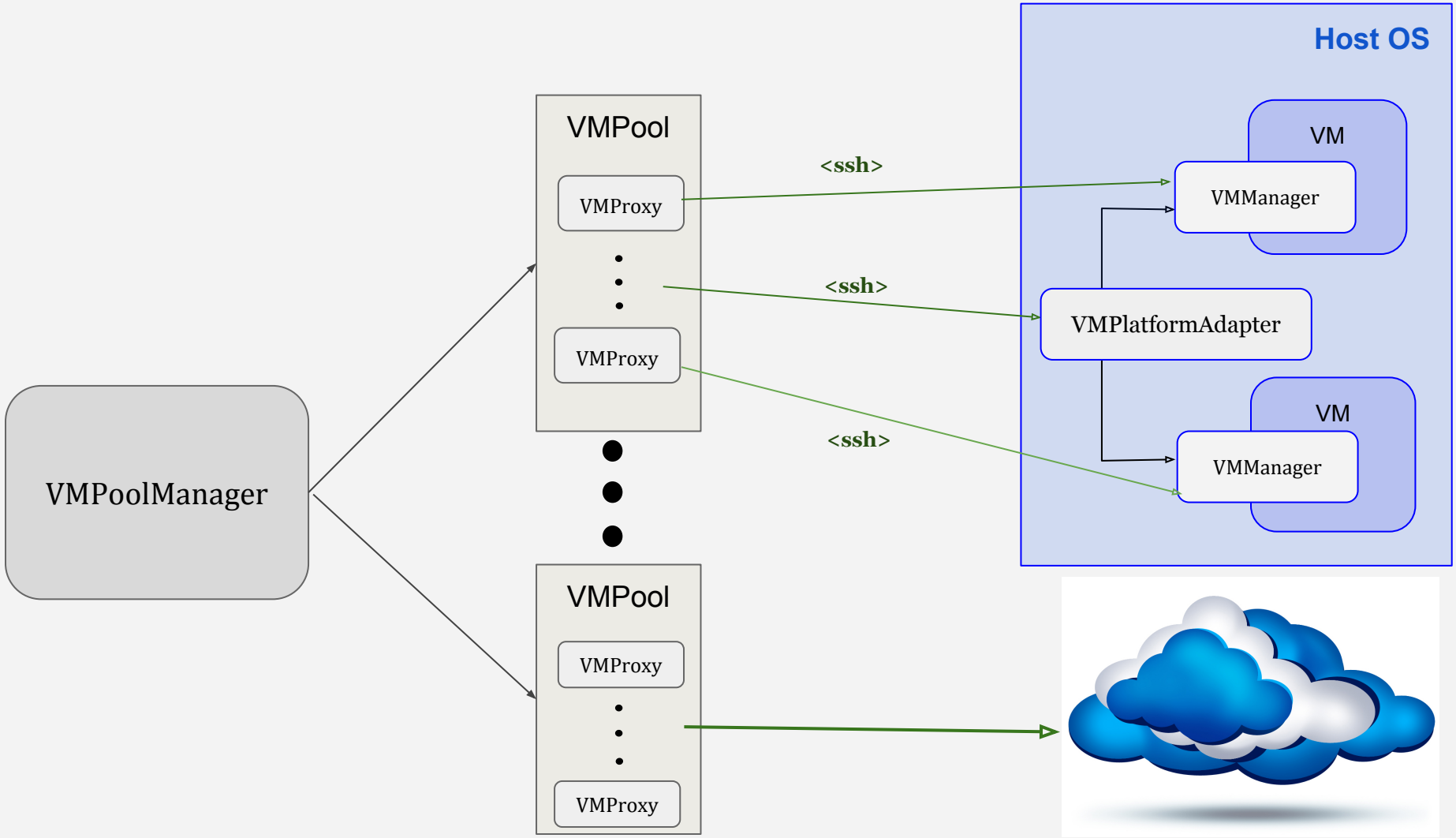

User Interfaces
Published Active Lab
Content,
Experiments, Quizzes
Outputs, State, Runtime data
Lab's Runtime Requirements
libraries, frameworks, services,
applications, scripts
Virtual Machine
Linux
User Interfaces
Published Active Lab
Content,
Experiments, Quizzes
Outputs, State, Runtime data
Lab's Runtime Requirements
libraries, frameworks, services,
applications, scripts
Virtual Machine
Microsoft Windows
Host OS


Lab is externally unavailable. Lab monitor will detect this in the next heartbeat-sensing phase.
Zombie
exception
stop
Versioned
instantiate
Running
resume
pause
stop
Paused
externally unavailable


Lab is externally unavailable. Lab monitor will detect this in the next heartbeat-sensing phase.
exception
Zombie
Versioned
publish
Published
instantiate
start
Running
pause
resume
stop
Paused
externally unavailable


VMPool
Host OS
VMProxy
<ssh>
Lj - Vk
VMManager
VM
VMPoolManager
VMPlatformAdapter
<ssh>
VMManager
VM
VMProxy
<ssh>
Lx - Vy


VMPoolManager
VMPool
VMProxy
VMProxy
VMPool
VMProxy
VMProxy
<ssh>
<ssh>
<ssh>
<ssh>
<ssh>
<ssh>
Host OS
VM
VMManager
VMPlatformAdapter
VM
VMManager
Host OS
VM
VMManager
VMPlatformAdapter
VM
VMManager


Host OS
VMPool
VMProxy
VMProxy
VMPool
VMProxy
VMProxy
VMPoolManager
<ssh>
<ssh>
<ssh>
VM
VMManager
VMPlatformAdapter
VM
VMManager

The Development Plan

- Four phases of successive refinements
  - One lab per VM (Linux VMs)
  - One lab in many VMs (Linux & Windows VMs)
  - Many labs in many VMs
  - Many labs in many VMs across Clouds

The First Iteration

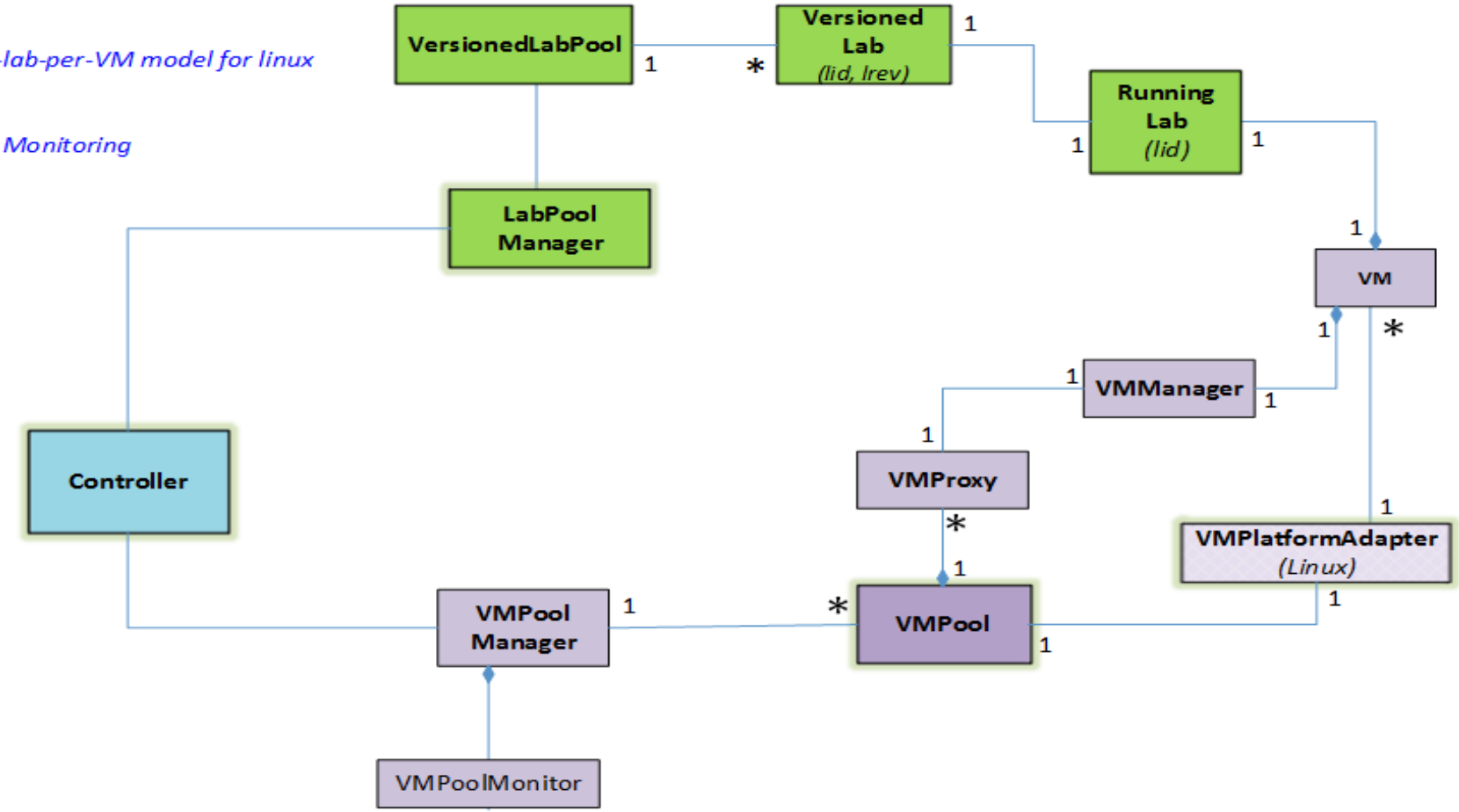
- Automated Lab instantiation
  (driven by LabSpecs)
- One lab per Linux-VM model
- VM monitoring
  (status, heart beats, allocated resources)

Iteration 1

Automate Lab Instantiation (driven by LabSpecs)

Realize One-lab-per-VM model for linux labs

Support VM Monitoring



The Second Iteration

- Multiple VMs per lab

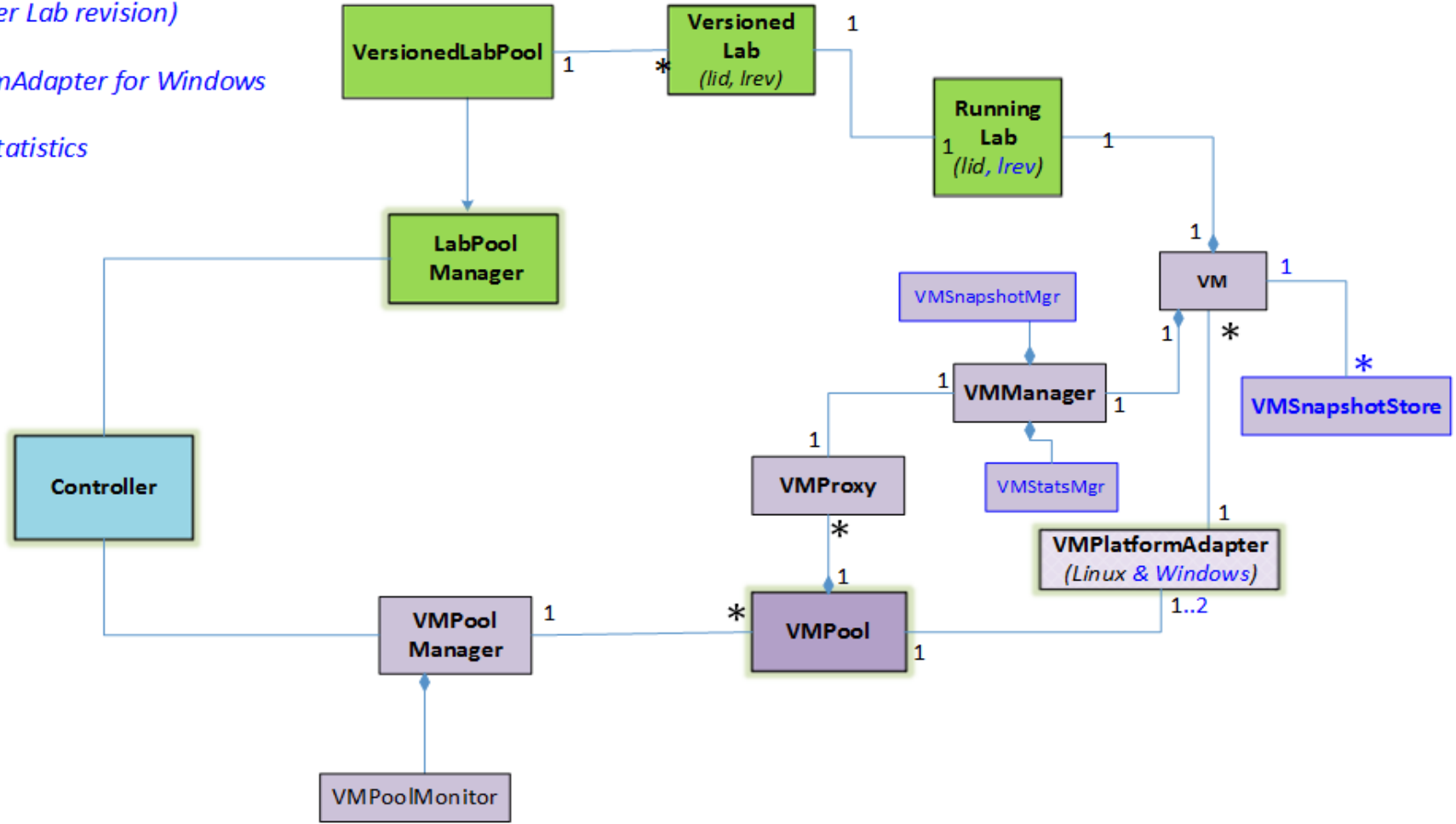
  (One-VM per Lab revision)
- *VMPlatformAdapter* for MS Windows
- VM snapshots (lab revision backup)
- Scheduling Backups
- Basic VM Statistics

Iteration 2

Realize multiple VMs per lab
(One-VM per Lab revision)

VMPlatformAdapter for Windows

Basic VM Statistics



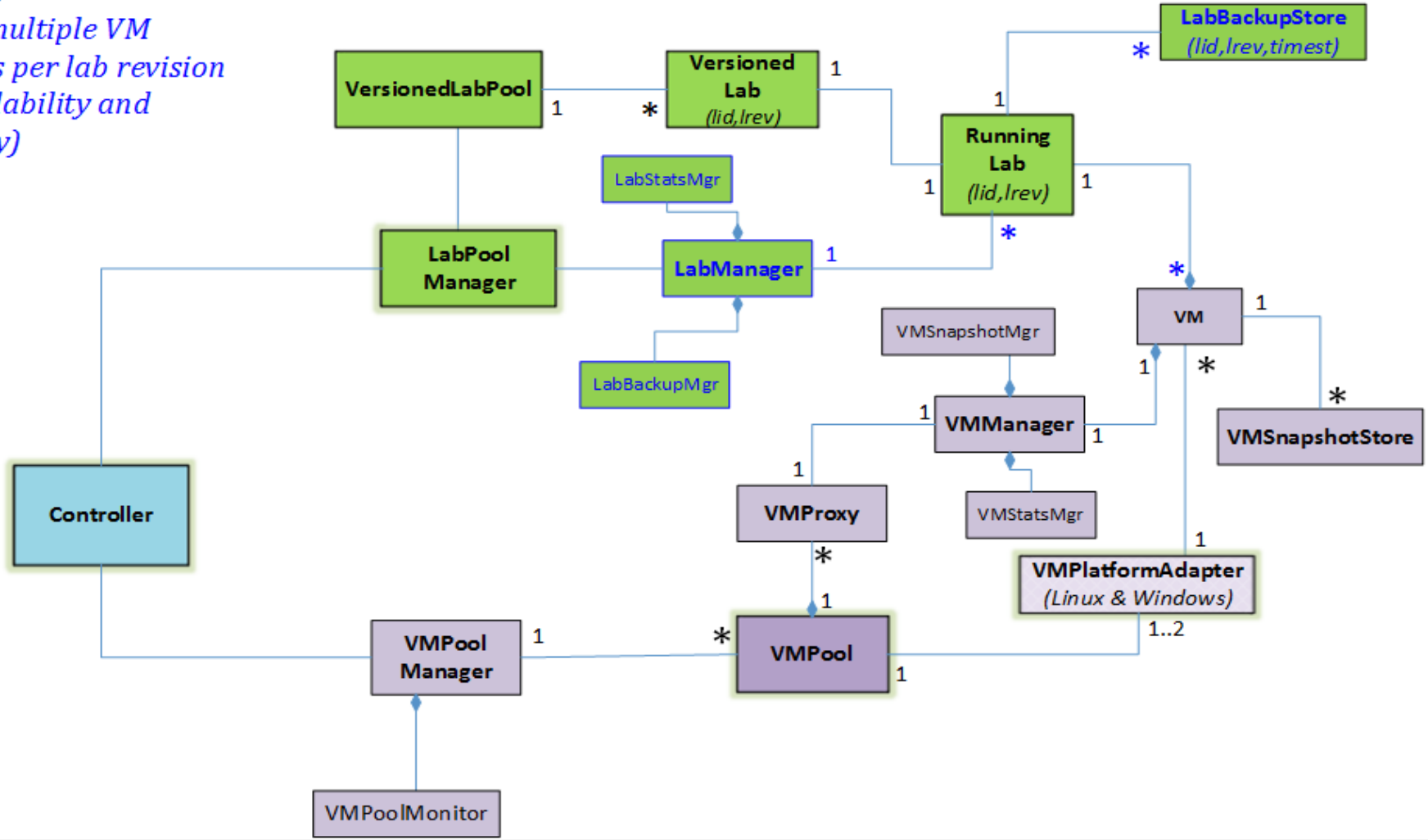
The Third Iteration

- Multiple VM instances per lab revision
- Implement Backup at Lab level

  backup and restore lab's data
- Lab statistics

Iteration 3

Realize multiple VM Instances per lab revision (for availability and reliability)



The Fourth Iteration

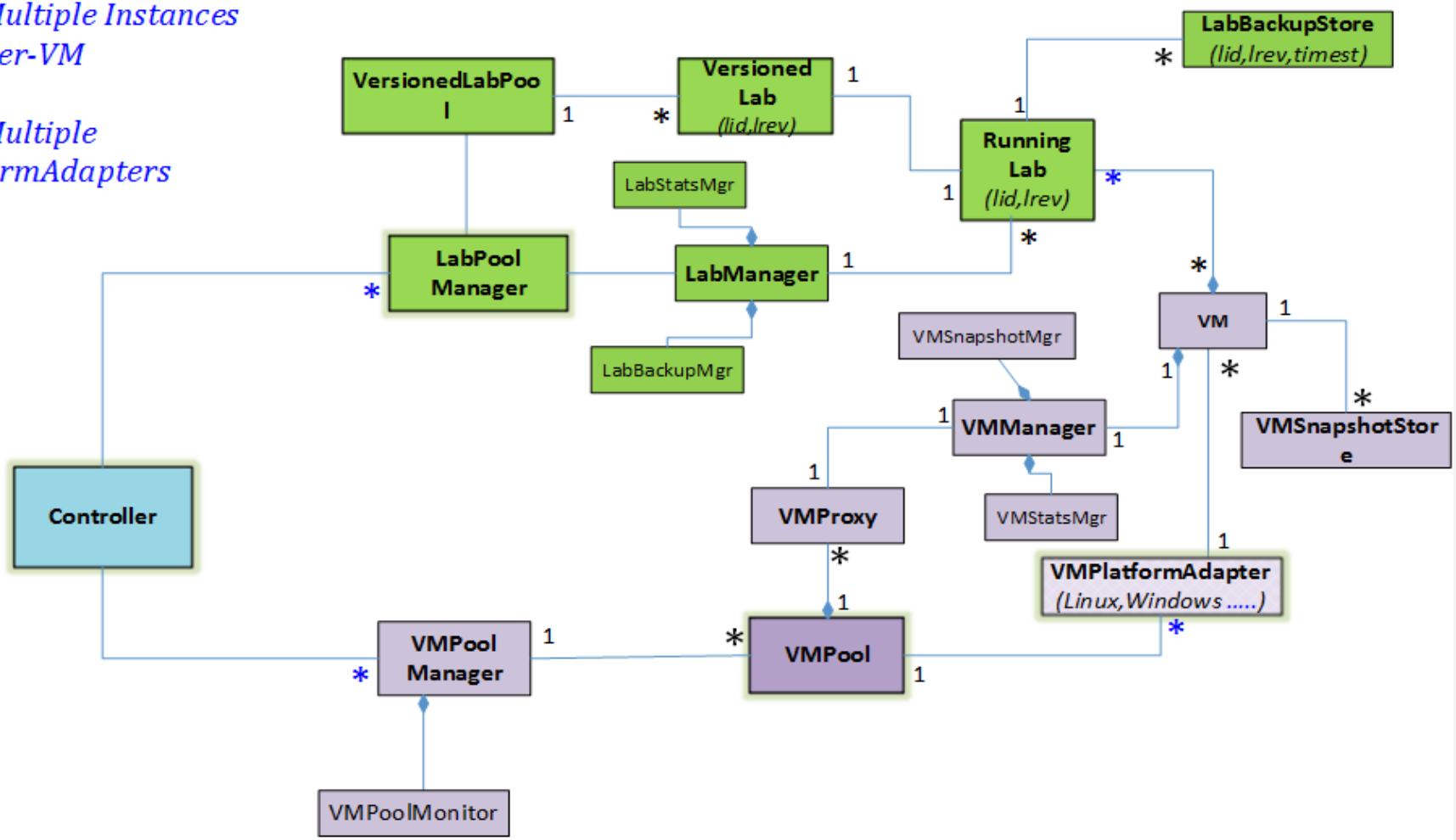
- Realize Multiple Instances of Labs-per-VM
- Multiple *VMPlatformAdapters* (geographically separated data centers and different clouds)
- Additional features

  (Load balancing, access policies, auditing policies)

Iteration 4

Realize Multiple Instances of Labs-per-VM

Realize Multiple VMPlatformAdapters



Next Step for the First Iteration

One team of three developers to create *LabPoolManager* and *LabPool* components

One team of three developers to work on *VMPoolManager*, *VMPool* and *VMManager* components