Softloan - A Federated Virtual Desktop (a cloud solution to scalability)

Author: Jacob-Steen Madsen (M.Sc.), Jacob Christiansen (M.Sc.), David Simonsen (M.Sc.)

Keywords: Federated services, Virtual Desktop, Library services, Cloud Computing, Virtualization.

Software as Library Service

This paper will address the possibility to setup a system so that complex software systems can be thought of as simple web resources which could be made available via library websites without the hazzle of installing and configuring your personal computer.

Background

Software vendors often give access to downgraded trial software packages in order to let end users test it. For the end user it is annoying not to have access to the full software package as the interesting features often are disabled. Having to install test software that 'pollutes' the local computing environment is a barrier that keeps potential buyers from trying the software.

If users could try the full software package in a limited time span, without having to install the software locally – then testing procedures would be easier. End users will be able to try the full end product and not refrain from installing - since there is nothing to install, only a web based service.

Software vendors would not have to worry about their software being copied and used as pirated software because the software wouldn't actually be installed in the computing environment of the end user.

A scenario like this would be a win-win situation for the end users, the software vendors, and it could potentially be part of positioning libraries in the public sphere.

In order to accomplish the vision there are quite a few issues to discuss some of which will be discussed in the following. In this paper, focus will be on the scenario. In order to show that it is not just a vision but also a practical solution – there's also a technical addendum.

The scenario

The following is a user story – a vision, told in laymen terms. This vision will be explained technically afterwards.

A man reads an article in a newspaper about a piece of software called "wobgobber".

"Wobgobber" targets issues he knows are problematic at work. From the article he get's the impression that "wobgobber" might be a useful at work.

In order to try "wobgobber" he opens his library's website. Within the librarys Online Public Access Catalog (OPAC) he searches for "wobgobber" and find that it is possible to make a reservation of "wobgobber", just as if it was a book.

He then logs on to the website - clicks on the reserve button. He's asked the normal questions about when he would like to access "wobgobber" and the loan is then scheduled.

Once confirmed on the web – and a receipt is send to his email account.

In the email or from the website he's directed to a website where he can access wobgobber within the timeslot he has mad a valid reservation of wobgobber. Within his browser he's granted access to a fully functioning desktop with “wobgobber” - the software he is now loaning.

He can now use “wobgobber” until the loan expires. He can even leave the desktop, shut the connection - and then later return to resume work - as can be done with any remote desktop connection. When the loan expires “wobgobber” will no longer be available.

The service is now closed and he’ll have to make another reservation if he hasn’t finished his work with “wobgobber” - if it is allowed.

Implications

Besides the technical aspects which is addressed in the technical addendum to this paper there are a few other things that should be addressed.

- License negotiation
- Identifying partners who can run the setup
- Identifying channels to expose the service to the public.
- ….

Technical Addenum

Technology maturation

Today “Cloud Computing”, Virtualization, “Software as a Service” are buzzwords. In reality it is all about being more efficient in the way we are using our computing resources. On top of this identity and access management is maturing in organizations which are now joining of identity federations.

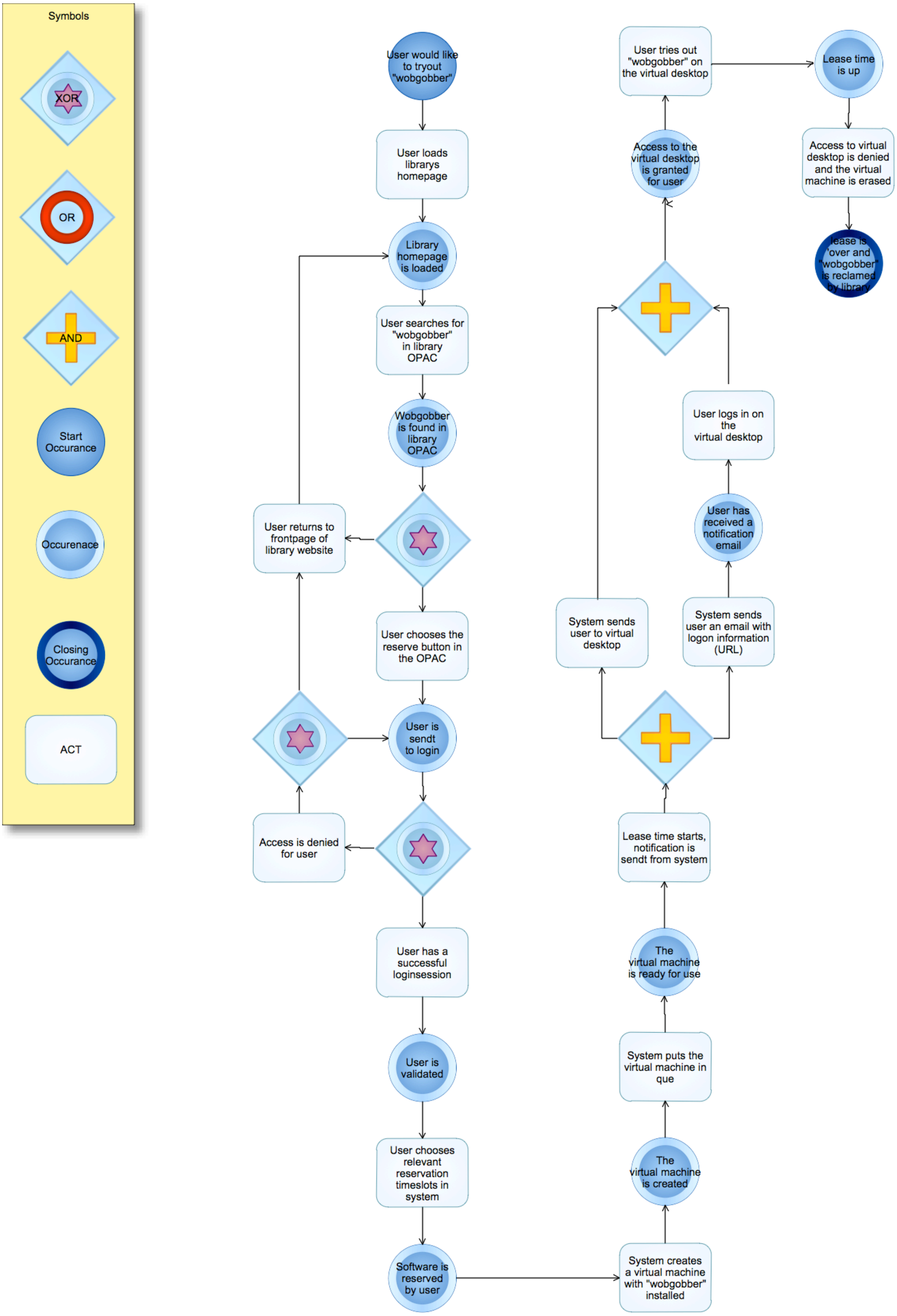
Because these technologies are maturing it is now possible to talk about software as a 'library resource' you can borrow as an end user.

Technical solution

To support the user story described above we need to identify the technical components. Below is list of software needed.

1) Open Public Access Catalog (OPAC), to present search results (available software) to users
2) Authentication and preferably authorization software.
    1. It would be preferable to open the service for federated access. Doing this will make the library able to offer services to the broader public. The economics in establishing a setup like the described will demand a broad user base, othervise it will not be cost efficient. Federations are a very easy way to ensure a big userbase.
3) Browser based remote desktop client that supports several protocols and that can run on multiple platforms and browsers.
4) Operating systems (Windows, Linux, OSX, etc.)
5) The software packages to be loaned out
6) Virtualization software
    1. XEN-source, vmware, hyperV etc.
    2. Could be a cloud service.
7) A directory to store information about the user in the session period. (And possible after)
8) Dynamic firewall configuration.

Userflow


Symbols
XOR
OR
AND
Start Occurance
Occurenace
Closing Occurance
ACT
User would like to tryout "wobgobber"
User loads librarys homepage
Library homepage is loaded
User searches for "wobgobber" in library OPAC
Wobgobber is found in library OPAC
User returns to frontpage of library website
User chooses the reserve button in the OPAC
User is sendt to login
Access is denied for user
User has a successful loginsession
User is validated
User chooses relevant reservation timeslots in system
Software is reserved by user
System creates a virtual machine with "wobgobber" installed
The virtual machine is created
System puts the virtual machine in que
The virtual machine is ready for use
Lease time starts, notification is sendt from system
System sends user to virtual desktop
System sends user an email with logon information (URL)
User has received a notification email
User logs in on the virtual desktop
Access to the virtual desktop is granted for user
User tries out "wobgobber" on the virtual desktop
Lease time is up
Access to virtual desktop is denied and the virtual machine is erased
lease is 'over and "wobgobber" is reclamed by library

Specific Software to be used:

The goal could be accomplished in a variety of ways. In this article the focus will be on open source products where feasible. This is done in the belief that until setups like the one described is mainstream, it will be more convenient to have access to source code. Everything described here will of course also be possible to accomplish with closed source software as well.

The OPAC

OPAC's[1] are part of the exisiting library infrastructure – so no reason to dwell on that. There is a task in exposing the software in the OPAC's but that has been done many time before.

The administrative software

For administration software like VCL, created at North Carolina State University[2], would do. The project is part of the Apache.org Foundation and for the time being it is at 'incubator' status[3]. The VCL software has an administration module which keeps track of reservations and loans of system images, when they are due for creation and when they should be reclaimed. The VCL software monitors and balances the utilization of the virtual machines so that the reservations are kept within what's possible to deliver. VCL is freeware and open source.

The Authentication and Authorization software

SimpleSAMLphp[4] as the AAI[5] module would be a natural choice – it is open source and market leading. It is crucial to separate Identity Management (IdM) from the service in order to focus o the core functionality of the service.

SimpleSAMLphp offers a variety of standard methods for identity providers to connect their users to the service. SimpleSAMLphp is freeware and open source.

Openldap + Samba

A directory service that can create permanent or temporary accounts for the users is also needed. This is done so that access to the virtual desktops can be personalized and be prepared for interactions between this service and others (file server, print services etc). Creation of user accounts or mapping an unknown identity to a directory gives the possibility to create other services when returning to other contexts – which in many cases would be wanted. The creation of accounts in the directory should be triggered from the login via the AAI software. The open source directory implementation that can be used cross platform is openldap[6] + samba[7]. The password for the account in Openldap should be set with an algorithm including a random function. In case of re-login from the same user, the password should be set to something new and cryptic so that it could only be used inside the session of the login. Openldap + Samba is both freeware and open source.

The Client

The company "Nomachine"[8] delivers a suitible server and web client. The nomachine server is setup on a linux-box and can be configured only to be reached via an ssl encrypted connection. The server acts as a proxy to the end-user. The nomachine is not freeware but open source.

[1] http://en.wikipedia.org/wiki/OPAC

[2] https://vcl.ncsu.edu

[3] http://incubator.apache.org/projects/vcl.html

[4] http://rnd.feide.no

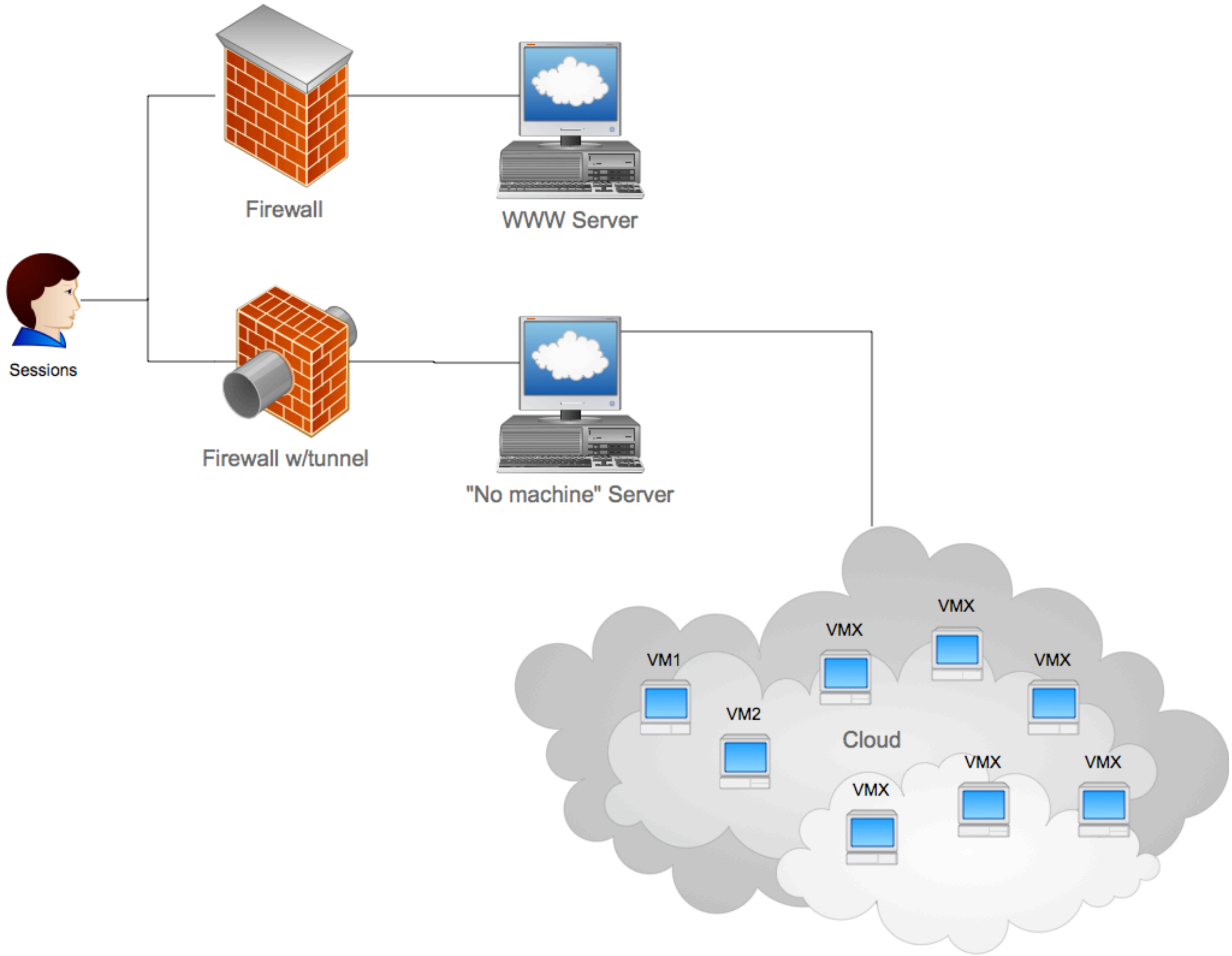
[5] Authentication Autorisation Infrastructure

[6] http://www.openldap.org

[7] http://www.samba.org/

[8] http://www.nomachine.com

The web client (an applet[9]) can connect to desktops using different protocols (rdp, vnc and x) through the server tunnel.



Based on what resource the user tries to reach the web client is configured accordingly.

A Dynamic Firewall

For security reasons there should only be access to the virtual machine from IP-addresses corresponding to the machine from which the end user authenticates him/herself. Thus a successful authentication procedure should trigger the firewall to open for connections between the users' computer and the Nomachine server (port 22).

Vmware, barebone metal

The VCL software supports both barebone setups and virtual machines. So if balanced correctly the setup can make use of hardware already installed for other purposes. For instance workstations can be used in off-hours for better utilization of existing clock-cycles.

If the setup should be even more flexible – cloud resources could be used instead of hosting the virtual machines locally.

Conclusion

The technology to create a setup as described is already there. Proofs of concepts has been made

[9] An applet is a small JAVA program that can run within the boundaries of a browser.

and they have shown satisfactory results. So what’s left are primarily policy issues and non technical work.