Journal of
Virtual Worlds Research
jvwresearch.org ISSN: 1941-8477

The Metaverse Assembled

April 2013
Volume 6, No. 1

Volume 6, Number 1
The Metaverse Assembled
April 2013

Journal of Virtual Worlds Research
jvwresearch.org ISSN: 1941-8477

Volume 6, Number 1
The Metaverse Assembled
April 2013

Architecting Scalable Academic Virtual World Grids: A Case Utilizing OpenSimulator

Charles J. Lesko
East Carolina University, USA

Yolanda A. Hollingsworth
Middlesex Community College, USA

Abstract

To better understand the technological requirements of academic institutions looking to implement an OpenSimulator virtual world grid, an observational study was performed to better understand the solution requirements. The purpose of this presentation is to provide an analysis of the parameters and considerations utilized to architect a scalable, open-source virtual world grid for use in various academic delivery scenarios. This specific case focuses on the detail leading up to deployment of the solution, and includes a discussion regarding solution selection and incorporation of various virtualization technologies to maximize institutional hardware resources based on established functional need. The computing resources utilized for this case were allocated via a virtualized infrastructure. Discussion and results include presentation of a proposed layered model outlining the solution elements and their relationships as well as various approaches to structuring and organizing in-world content and activity.

1. Introduction

In many ways, we are in the infant stages of a significant dimensional shift in how we present and then interact with information across the Web, and more specifically within academic online delivery settings. Over the past decade, various virtual world solutions have sought to immerse its participants utilizing a variety of technological approaches, software applications and computing resource. These technologies have now evolved to a point where academics are embedding them within their online and blended course deliveries looking to provide more interactive, collaborative and immersive three-dimensional (3D) presentations for their students (Dalgarno, Lee, Carlson, Gregory, & Tynan, 2011); (Martin, Diaz, Sancristobal, Gil, Castro, & Peire, 2011); (Margaryan, Littlejohn, & Vojt, 2011). Although the promise of these new technologies has been shown to provide a unique platform to facilitate interaction, socialization, and learning (Baldi & Lopes, 2012) virtual worlds are still infant technologies that require further understanding. A critical link to that understanding is the ability to architect and deploy virtual world solutions across the broad spectrum of the academic arena that is continually challenged with limited technical resources, both human and material.

To baseline our efforts here, it is important to scope the solution that is being architected and presented here. Although many have attempted, there is no single accepted definition for what a virtual world is functionally; however, some key descriptors are typically mentioned when describing a virtual world solution (Heudin, 2000); (Bartle, 2004); (Bell, 2008); (Jensen, Phillips, & Strand, 2011). For the purposes of this case study the following description of an academically focused virtual world will be utilized: a computer-based, three dimensional, immersive environment in which end-users, via their avatar proxies, are able to create and manipulate virtual objects and spaces. Additionally, these virtual worlds provide end-users with the ability to interact and collaborate with multiple users; they also include facilities that enable users to communicate both synchronously and asynchronously and interact with an established learning management solution, all from within the virtual environment. In short, a virtual world provides students with the means, through an avatar, to establish a persistent identity and participate in virtual collaborations within a predefined virtual world space.

The historical development of virtual world environments had its beginnings aligned with that of computer gaming and then later with online social networking sites (Noor & Lobeck, 2009). Unfortunately most of the virtual world solutions today are being designed independently of each other utilizing a wide variety of architectures and protocols. Educational institutions have unique challenges including: what solutions to embrace; what technical requirements to levy on their students; and how to appropriately secure and maintain these virtual world solutions in the face of multiple pedagogical methods and need (Warburton, 2009); (Thompson, 2011); (Wasko, Teigland, Leidner, & Jarvenpaa, 2011). To provide the required levels of technical functionality, current virtual world solutions are typically architected as various stacks of services with each virtual world not only simulating its own three dimensional (3-D) virtual space but also providing its own identity services, content hosting, digital rights management, instant messaging, virtual economies, social networking elements such as groups, and other services (Capalini, 2009); (Blair, 2011); (Trescak, Esteva, & Rodriguez, 2011); (Malaby, 2009).

To meet third millennial educational institutions essential learning capabilities of flexibility, inclusiveness, collaboration, authenticity, and relevance most are in the process of evaluating various online technologies like virtual worlds (Felix, 2005); (Rapanotti & Hall, 2011); (Li, D'Souza, & Du, 2011). Based on this understanding of academic need, the following functional requirements were established as essential capabilities for an open-source, scalable, academic, virtual world grid solution for this case study. These solution requirements were further subdivided in four categories: Accessibility, Capability, Scalability, and Security outlined in Table 1.

Table 1: Solution functional requirements by category

	FUNCTIONAL REQUIREMENT	CATEGORIES
1	Integrate with a learning management solution.	Accessibility
2	Access various online social networking toolsets external to the virtual world solution.	Accessibility
3	Support multiple avatar's synchronously within a single grid.	Capability
4	Provide online collaborative capability to include synchronous text and voice chat.	Capability
5	Enable end-users to create, manipulate, interact with 3D virtual objects and places, including mesh objects.	Capability
6	Provide scalability of virtual grid space.	Scalablity
7	Provide reusability and cloning capability of virtual creations and grid spaces.	Scalablity
8	Maintain a secure, persistent virtual environment rendered in 3D.	Security
9	Control access into and out of virtual environment via secure authentication and controlled user account assignment	Security
10	Group avatar accounts and permissions.	Security

The intent of this case is to establish virtual world grid spaces to begin prototyping virtual learning environments for academic employment supported by a solution that meets the functional requirements outlined in Table 1.

2. Purpose of This Case

The purpose of this case presentation is to provide an analysis of the parameters and considerations utilized to architect a scalable, open-source, virtual world grid for use in various academic delivery scenarios. This specific case focuses on the detail leading up to deployment of the solution and includes discussion regarding solution selection and incorporation of virtualization technologies to maximize institutional hardware resources based on established functional need.

Of the many challenges facing institutions looking to implement virtual world technologies, four key categories highlight the clear complexity of this challenge: perceptual, technical, operational, and pedagogical. Perceptual challenges are typically caused by the misconception that virtual worlds are merely gaming platforms and are not designed or appropriate for educational delivery. Aside from bandwidth, processing and security concerns, technical challenges typically focus on not only the lack of tools for facilitating collaborative interactions between end-users in real-time but also the lack of interoperability between the different virtual world platforms possessing significant challenges. Current operational challenges include the relatively steep learning curves for academic institutions to maintain virtual world solutions such as the system and information storage solution sustainment and the ever present legal restrictions surrounding student information security. Finally, there are the pedagogical challenges that relate to the educational value and assessment of the technology, as well as the intellectual property and ownership issues associated with virtual world solution usage (Kelton, 2008);(Winkler, 2013); (Hogan, 2010). Of the four challenge categories, the focus of this case was on the technical aspects of the solution. Although they are viable challenges in and of themselves, the issues of perception, operation, and pedagogy are not addressed here.

3. Case Background and OpenSimulator Selection

The case institution (a doctoral-level university) began exploring immersive environments and their educational value in the spring of 2007 by leasing one quarter (1/4) of a sim (short for simulator) of virtual world space from Linden Labs Second Life domain. A sim is a virtual world 3D space that in Second Life typically equates to an area 256 meters square (so in this case the initial lease equated to a virtual space of 128 meters square (Rymaszewski, Au, Wallace, Winters, Ondrejka, & Batstone-Cummingham, 2007). Since 2007, the university, which now maintains nine full Second Life sim's, has offered courses at the undergraduate and graduate levels and has offered high school advanced placement courses as well. Additionally, short class sessions, workshops, meetings, and conferences have also been conducted from the universities leased Second Life virtual spaces.

In the fall of 2011, the university began evaluating the open source, multi-platform solution OpenSimulator (also referred to as OpenSim) as a possible collaborative media option for academic delivery. OpenSimulator is a platform for operating a virtual world environment supporting multiple independent virtual regions connecting to a single centralized grid; OpenSimulator can also be used to create a private grid that remains accessible only within a finite network infrastructure. The OpenSimulator is supported in both Windows and Linux-based operating environments and it supports both MySQL and MSSQL database technologies. Primary coding for OpenSimulator is developed using C# with a .Net framework (Allison & Miller, 2012); (Fishwick, 2009).

The core working space in OpenSimulator is the 'region' which is similar to a sim in Second Life. A region is what the end-user sees when they log into OpenSimulator; it is the visible virtual working space where the avatars interact. The region is a square piece of virtual landscape that can be further developed to contain such topologies as deserts, mountains, roads, buildings, classroom spaces, vast oceans, and other virtual space. A collection of multiple regions then forms what is typically referred to as a grid. Fundamentally, a grid provides organizational structure to the many regions by managing the relative position of each region within the virtual world. The grid also manages such services as permissions, inventory and user access (OpenSimulator.org - Services, 2012); (Allison & Miller, 2012).

The current deployment of OpenSimulator utilizes a server shell called ROBUST which runs a collection of virtual world management services. There are currently (14) services that are managed collectively by the ROBUST server that include: Assets Service, Authentication Service, Authorization Service, Avatar Service, FreeSwitch Service, Friends Service, Gatekeeper Service, Grid Service, Grid User Service, Inventory Service, Login Service, Presence Service, User Accounts Service, and the User Agents Service (OpenSimulator.org - HyperGrid, 2012); (Fishwick, 2009). Based on the functional criteria outlined earlier in Table 1, it was determined that OpenSimulator solution coupled with additional open source support applications not only met most of the established criteria but also provided a platform similar in design to Second Life, thus reducing some of the in-world learning curve required by both students and grid managers. Overall similarities in end-user viewers, in-world content design, and account management made it a viable option to move forward with.

4. Virtual World Architectural Solution

At its basic level, today's virtual world solutions include four key components: (1) a simulated environment or virtual world solution; (2) an end-user client or viewer; (3) a collection of collaborative resources available from within the virtual world environment; and (4) a network infrastructure that encapsulates and supports the virtual world solution. For computing and network resources this case utilized a virtualized infrastructure. A virtualized infrastructure allows users to transform hardware resources into a more flexible software-based resource and (Ausweb, 2012) more specifically, provides for the ability to compile and then redistribute multiple hardware resources such as processors, memory, storage, and network controllers to create one or more fully functional virtual machines. These virtual machines or VM's can support their own operating system and applications – thus reproducing the same capabilities of one or more singular physical computing platforms (Dawson & Saeed, 2012).

In developing a scalable virtual world infrastructure, virtualization provides us with the ability to both isolate and encapsulate application activity. Since VM's can share the physical resources of one or more computers, they are essentially isolated from each other just as if they were separate physical platforms (Shi, Jin, Pan, Huang, Yu, & Jiang, 2012). A VM is essentially a software container that packages or encapsulates a complete set of virtual hardware resources inside a software solution. The concept of encapsulation makes VM's incredibly scalable and much easier to manage (Narten, Sridharan, Dutt, Black, & Kreeger, 2012).

The physical resources required for this virtual world solution (including computing, networking and storage) are all consolidated and managed under a single virtualized computing umbrella; computing capacity is then subdivided into multiple VM's as required for the solution. In describing the architecture for a scalable virtual world grid a layered model is being presented. Case analysis in this instance provides for a solution architectural model in four distinct layers: a client layer; a robust layer; a collaborative layer; and a regional layer (See Figure 1).

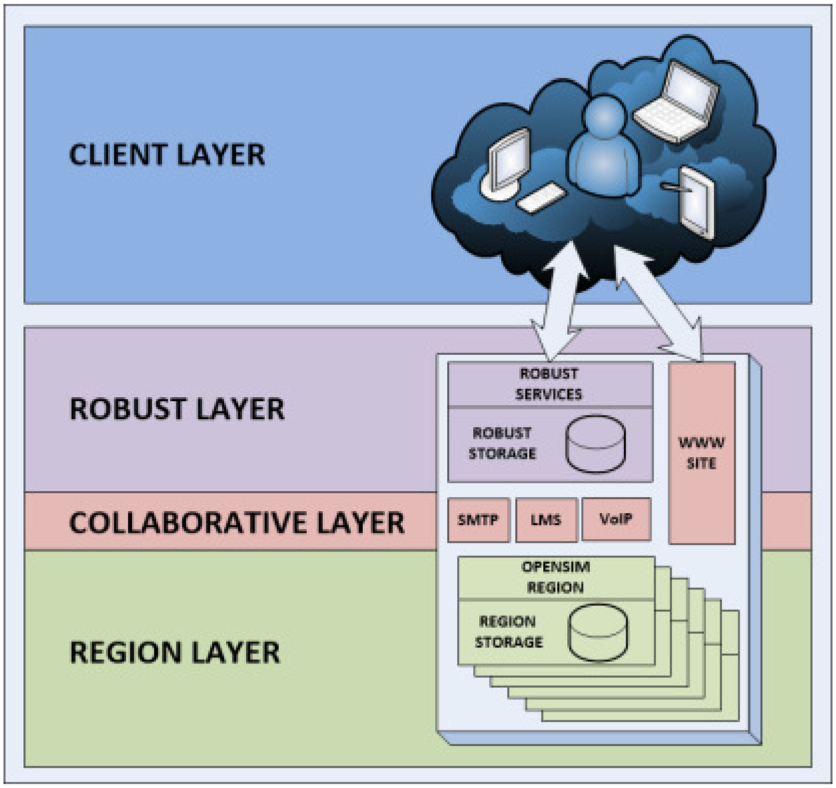

Figure 1: Solution Architectural Layers

Client Layer: The client layer encompasses the end-users virtual world experience; this consists primarily of the client viewer that is loaded on the end-users computer. The client viewer is the application that provides the end-user with a window that allows them to experience the virtual environment; essentially, it frames the end-users experience and defines their interactions. At present, the OpenSimulator application distribution does not come with its own client viewer (OpenSimulator.org - Connecting, 2012); however, several client viewers have been developed to date that support viewing OpenSimulator virtual world spaces (OpenSimulator.org - Compatible Viewers, 2012).

While evaluating the solution options, several client viewers were identified and tested for functionality with the Imprudence viewer (Kokua, 2011) retaining the functionalities that meet established requirements outlined in Table 1. Other than their ability to view OpenSimulator virtual world content, each client viewer was evaluated for their ability to:

1. Access key collaborative functions including voice chat and media viewing in-world;
2. Present detailed virtual content including mesh support;
3. Import, export and manipulate virtual content while interacting within the virtual world (i.e. in-world).

It should be noted here that two browser-based experimental clients (Xenki and 3Di viewer) were evaluated but failed to meet either stability or minimal functionality requirements (Capodieci, Martella, & Paiano, 2011); (Berntsson, Lin, & Dezso, 2009); (OpenSimulator.org - Connecting, 2012).

Robust Layer**:** Currently, an OpenSimulator grid deployment provides for a ROBUST (Redesigned OpenSimulator Basic Universal Server Technology) service module that manages fourteen (14) frontend, client-facing services. For scalability, ROBUST is capable of running as a single application managing all its related services or can be separated into several ROBUST server instances, each running one or more services. In this case, all ROBUST services were initially maintained on a single VM instance utilizing a separate robust storage VM for data management (see Figure 2); however, initial penetration testing highlighted the need to move all but the login service behind the firewall within the solution's own private IP domain, thus splitting the ROBUST services into two separate VM's. Future consideration was made to further separate out the grid and asset services into separate VM's once the grid began to build-out content and began to realize an increased concurrent user level.

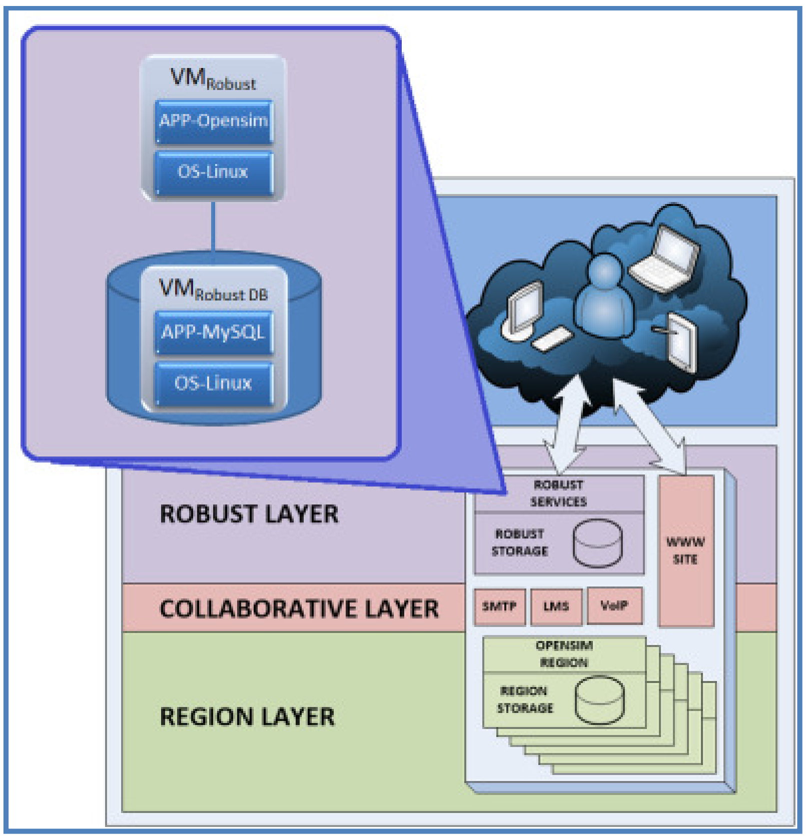

Figure 2: ROBUST Layer

The bulk of the solution security concerns are managed at the ROBUST layer. Identity management, access rights management and content ownership management are all initialized at the ROBUST layer. Identity management within a given avatar session is based on the capability concept where each entity (avatar, content object, texture, etc.) receives a unique URL for each given session of avatars' interaction; these capabilities are dynamically generated UUID based URLs that act as session tokens for a given period of time, based on the activity. Access rights management or Role Based Access Control (RBAC) is controlled by the use of various privilege user categories (i.e. 'God' (highest role), 'Region Owner', 'Parcel Owner', etc.). Finally, content ownership management is based on the establishment and use of two distinct in-world roles: the Creator and the Owner (Perera, Allison, & Miller, 2010).

***Collaborative Layer*:** There are several collaborative service capabilities that, by default, are not included with the OpenSimulator distribution (OpenSimulator.org - Services, 2012). Based on the functional requirements identified in Table 1, there are four collaborative services deemed essential to the case virtual world solution: email (SMTP); learning management solution (LMS); voice chat (VoIP); and website (WWW) services. For the case solution, each of the four collaborative services receives their own VM. The following open-source solutions were selected to meet these collaborative requirements (see Figure 3):

- Website Service (WWW): Aside from the need to provide a basic frontend to present grid information to the public (i.e. grid location and address, grid activity, end-user setup, grid support, etc.) the website provides the grid with a platform for handling various end-user functions that are not included with the OpenSimulator distribution including: account creation and updates; password recovery; end-user inventory management; welcome page for logging in; and online administrator site management tools.

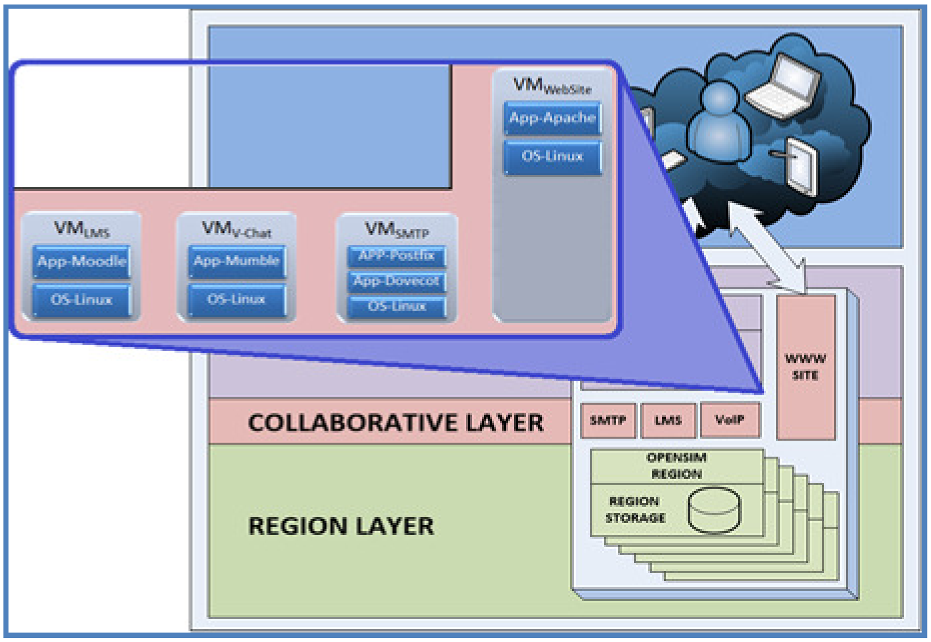

Figure 3: Collaborative Layer

Since the operating environment selected for this case is Linux-based, a default apache web service was installed and WIFI was selected to provide a basic web account management front-end. WIFI is an add-on module that provides an embedded Web application for handling user registrations. It supports web-based functionalities to include: Account creation; configurable default avatars for new accounts; account updates by both users and administrator; account deletion; password recovery via email; and basic management of user inventory (OpenSimulator.org - Wifi, 2012).

- Learning Management Service (LMS): The LMS functionality and integration with OpenSimulator are supported by Moodle and SLOODLE respectively. Moodle is an e-learning management solution designed to assist academics in creating online course offerings with a focus on interaction and collaborative construction of content (Moodle, 2012); (Gongalez-Crespo, Aguilar, Ferro-Escobar, & Torres, 2012). The SLOODLE (Simulation Linked Object Oriented Dynamic Learning Environment) project provides the integration of OpenSimulator with the Moodle learning-management solution (Bloomfield, 2012); (Konstantinidis, Tsiatsos, Demetriadis, & Pomportsis, 2010); (Gongalez-Crespo, Aguilar, Ferro-Escobar, & Torres, 2012).

- Email Service (SMTP): Simple Mail Transport Protocol Services (SMTP) is required for several of the grid service needs including password recovery, in-world email and SLOODLE collaborations. Several viable Linux-based Mail Transfer Agent (MTA) solutions are available, however, for this case Postfix was selected (Ubuntu.com, 2012).
- Voice-over-IP Service (VoIP) or Voice Chat: The open-source application Whisper that is based on Mumble was selected as the voice-over-IP (VoIP) application to provide voice chat functionality to the virtual world solution. The voice chat solution consists of an OpenSimulator region module and Murmur which is the Mumble voice server. The region module, referred to as MurmurVoiceModule, communicates with Murmur in order to open channels and to register users (Gaessler, 2010). Although the Mumble solution is currently limited to a select number of client viewers it was selected over other open-source options for its avatar lip-sync support and speaker indication functions.

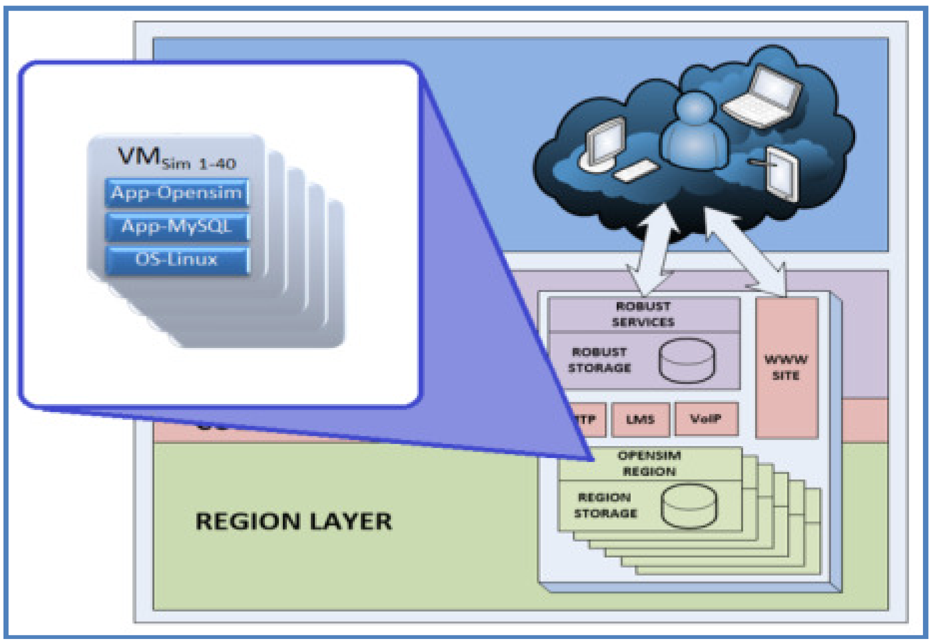

Figure 4: Region Layer

***Region Layer*:** As noted earlier, OpenSimulator supports one or more regions. A region in OpenSimulator is the virtual space that is seen when the end-user logs in; it is the environment where all virtual activity occurs and may contain a flat piece of land, an island, mountains, a plain, buildings, a combination of all of these, or simply be a vast ocean space. For this case solution, each region and its data store were encapsulated within their own VM (see Figure 4).

By default, each region has a virtual land surface of 256 meters squared; however, OpenSimulator has the ability to merge multiple regions into a single region that is referred to as a mega-region. Larger scale virtual workspaces consisting of multiple OpenSimulator regions are referred to as grids. The content within each region can contain upwards of 15,000 primitive (referred to as prims) objects (Farooq & Glauert, 2011).

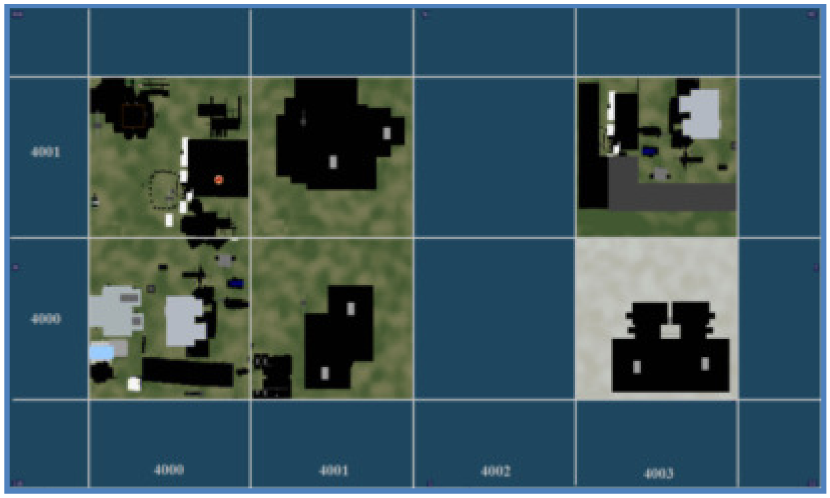

Figure 5: Sample grid layout with (6) regions

Regions are organized by way of a two-dimensional coordinate system that allows for the precise locating of each region. Figure 5 presents a sample grid layout showing six regions where in this example the region in the lower left corner would be at coordinates (4000,4000). When placing regions within a grid, they can be bordered by up to eight other regions (one for each point of the compass), or regions can be completely separate from other regions as indicated by the two regions set off to the right in the example in Figure 5. When two regions border each other, an end-user has the ability to cross the border between them having their avatar, and its related inventory transferred from the region they were in to the new region they moved into. Grid coordinates have a range of 0-65,535 allowing for a vast range of regional layout possibilities.

For this case, an area that represented the size of the universities main campus was utilized to demonstrate the scalability of the solution. This amount of virtual space required a grid with 63 regions as outlined in Figure 6. All regions received their own VM in an effort to better evaluate the system's resource demands realizing that varying levels of avatar activity and content would be maintained within each region. The value of virtualization and it scalability as well as its cloning capabilities allow for the rapid addition of regions in this case.

Figure 6: Sample campus grid layout with (6) regions

Scalability of the solution is further demonstrated by the use of two of OpenSimulator's built-in archive functions. At the region level, OpenSimulator supports the archiving of specific content/inventory for the entire region. The OpenSimulator Archive (OAR) function saves an entire region's content to a single file so that it can be later reloaded to any select region. The Inventory Archive (IAR) function saves selected content from an avatars inventory to a single file so that it can be later reloaded either to the same avatars account or loaded to another avatar's inventory account (Allison C. , Miller, Sturgeon, Perera, & McCaffrey, 2011); (OpenSimulator.org - Archives, 2012).

5. Considerations Moving Forward

In 1982, Rey Oldenburg and Dennis Brissett identified what they collectively referred to as "third places" or "great good places," as the public places on neutral ground where people are able to gather and interact (Oldenburg & Brissett, 1982). They noted that in contrast to first places (home) and second places (work), third places allowed people to set aside their concerns and enjoy the company and dialog of those around them. Third places were locations that hosted "the regular, voluntary, informal, and happily anticipated gatherings of individuals beyond the realms of home and work." (Oldenburg, 2000) Now fast forward three decades and through the advent of current virtual world technologies coupled with a new digital media-focused learner who has unprecedented global reach to information resources and instantaneous, or near-instantaneous, communication with others, enabling geographic independence; we now have the beginning of a new online virtual third place (Soukup, 2006) or "Virtual Third Space".

Now with the advent scalable virtual third space learning configurations by way of virtual world technologies, academics can now begin to realize new social consequences of online and mobile learning. The new ecology of virtual third space learning has the promise to support synthetic reason, augmented knowledge and culture, as well as immersive virtual reality and blended or hybrid virtual delivery modalities. However, to achieve the required functionality for architecting viable, scalable, academic virtual world grids as outlined earlier in this presentation, it is crucial that we create useable solutions that minimize the technical challenges on both academics and their institutions.

The results of the case are highlighted in Table 2 outlining the proposed solutions to each functional requirement and indicating the architectural model layer that each was realized at. Based on these criteria, solutions were found to meet the functional need and an architectural model was developed and presented to support this case solution.

Table 2: Functional Requirements by category, layer and solution

	FUNCTIONAL REQUIREMENT	CATEGORIES	LAYER	SOLUTION
1	Integrate with a learning management solution.	Accessibility	Collaborative	Integration with Moodle, Sloodle
2	Access various online social networking toolsets external to the virtual world solution.	Accessibility	Collaborative	Integration with Mumble, Apache, Postfix
3	Support multiple avatar's synchronously within a single grid.	Capability	ROBUST	Built-in Opensimulator functionality and scalability of ROBUST services.
4	Provide online collaborative capability to include synchronous text and voice chat.	Capability	Collaborative	Built-in Opensimulator text-chat functionality and establishment of Murmur VoIP services.
5	Enable end-users to create, manipulate, interact with 3D virtual objects and places, including mesh objects.	Capability	Region	Built-in Opensimulator functionality and region configuration.
6	Provide scalability of virtual grid space.	Scalablity	ROBUST & Region	Virtualization allows for rapid addition of regions as needed. ROBUST layer also scalable.
7	Provide reusability and cloning capability of virtual creations and grid spaces.	Scalablity	Region	Use of .oar and .iar files and VM cloning.
8	Maintain a secure, persistent virtual environment rendered in 3D.	Security	ROBUST, Collaborative & Region	Network configuration and penetration testing.
9	Control access into and out of virtual environment via secure authentication and controlled user account assignment	Security	ROBUST & Region	ROBUST layer configuration manages Grid access and each region has ability to limit accessibility further.
10	Group avatar accounts and permissions.	Security	ROBUST & Region	Configuration of group module at ROBUST and Region levels

Moving forward, an area for near term study is the concept of the hypergrid. A hypergrid is an extension to OpenSimulator that allows end-users to link OpenSimulator grids to similar virtual workspaces or grids on the Web. Although the concept of hypergrid has been worked on for the past several years it is still early in its development and testing (Scacchi, Brown, & Nies, 2012). Although not established as a core functional requirement for this case, the ability to support multiple grids or connects between other grids can add significantly to the scalability of the solution but adds to the complexity of the solution as well. With the OpenSimulator application, movement from one grid to another can be accomplished by creating virtual world hyperlinks (similar to hypertext links on the Web) between the two grids. The premise behind the hypergrid is that each virtual workspace creates hyperlinks on their world map to other OpenSimulator compatible grids. Once those hyperlinks are established, users interact with those regions in exactly the same way as they interact within their own local grid by teleporting from one grid to another. This capability is currently managed through OpenSimulator's grid service (OpenSimulator.org - HyperGrid, 2012); (Lopes, 2011).

The layered approach presented here provides the reader with one plausible approach to architecting an OpenSim virtual world solution. It is clear that each set of institutional needs is unique and may require other approaches. As the OpenSim solution matures further, this approach will require reevaluation and updating where appropriate. The challenges are there; this approach hopes to at least provide a conduit for follow-on efforts and dialogue towards the addition of virtual world technologies onto our list of collaborative academic media toolsets.

References

Allison, C., & Miller, A. (2012). *Open virtual worlds for open learning.* St. Andrews, UK: Higher Education Academy.

Allison, C., Miller, A., Sturgeon, T., Perera, I., & McCaffrey, J. (2011). The third dimension in open learning. *Frontiers in Education Conference* (pp. T2E-1). Rapid City, SD: IEEE.

Ausweb. (2012, December 12). *What is a virtual machine?* Retrieved December 12, 2012, from ausweb.com: http://www.ausweb.com.au/web-hosting/information/what-is-a-virtual-machine.html

Baldi, P., & Lopes, C. (2012, April 12). The Universal Campus: An open virtual 3-D world infrastructure for research and education. *eLearn Magazine: Education and Technology in Perspective*.

Bartle, R. (2004). *Designing Virtual Worlds.* Indianapolis, IN: New Riders Publicating.

Bell, M. (2008). Toward a Definition of “Virtual Worlds”. *Journal of Virtual Worlds Research*, 1-5.

Berntsson, J., Lin, N., & Dezso, Z. (2009). ExtSim: A Flexible Data Mapping and Synchronization Middleware for Scientific Visualization in Virtual Worlds. *Journal of Virtual Worlds Research, Volume 2, Issue 5*, 1-13.

Blair, J. (2011). An Approach for Integrating 3D Virtual Worlds with Multiagent Systems . International Conference on Advanced Information Networking and Applications (WAINA), 2011 IEEE Workshops of International Conference (pp. 580-585). Baniff, AB, Canada: IEEE Xplore.

Bloomfield, P. R. (2012). *SLOODLE Blog >> About.* Retrieved January 5, 2013, from Sloodle.org: http://www.sloodle.org/blog/?page_id=2

Capalini, Z. (2009, January 5). *Hypergrid, or how to teleport between WORLDS using OpenSim*. Retrieved May 31, 2012, from zonjacapalini.wordpress.com: http://zonjacapalini.wordpress.com/2009/01/05/hypergrid-or-how-to-teleport-between-worlds-using-opensim/

Capodieci, A., Martella, A., & Paiano, R. (2011). Web 2.0 services interoperability for E-learning. *The 9th International Conference on Education and Information Systems, Technologies and Applications: EISTA 2011* (pp. 1-8). Madrid, Spain: EISTA.

Dalgarno, B., Lee, M., Carlson, L., Gregory, S., & Tynan, B. (2011). An Australian and New Zealand scoping study on the use of 3D immersive virtual worlds in higher education. *Australasian Journal of Educational Technology, Volume 27, Issue 1*, 1-15.

Dawson, M. E., & Saeed, I. A. (2012). Use of Open Source Software and Virtualization in Academia to Enhance Higher Education Everywhere. In C. Wankel, & P. Blessinger, *Increasing Student Engagement and Retention Using Immersive Interfaces: Virtual Worlds, Gaming, and Simulation (Cutting-edge Technologies in Higher Education, Volume 6)* (pp. 283-313). Bingley, United Kingdom: Emerald Group Publishing Limited.

Farooq, U., & Glauert, J. (2011). Scalable and consistent virtual worlds: An extension to the architecture of OpenSimulator. *Computer Networks and Information Technology (ICCNIT), 2011 International Conference* (pp. 29-34). Abbottabad, Pakistan: IEEE.

Felix, U. (2005). E-learning pedagogy in the third millennium: the need for combining social and. *ReCALL, volume 17, Number 1*, 85-100.

Fishwick, P. A. (2009). An Introduction to OpenSimulator and Virtual Environment Agent-Based M&S Applications. *Proceedings of the 2009 Winter Simulation Conference*, pp. 177-183.

Gaessler, V. (2010, December 27). *Whisper Source Code is Available*. Retrieved January 6, 2013, from VCOMM Solutions: http://whisper.vcomm.ch/forum/viewtopic.php?f=2&t=62

Gongalez-Crespo, R., Aguilar, S. R., Ferro-Escobar, R., & Torres, N. (2012). Dynamic, ecological, accessible and 3D Virtual Worlds-based Libraries using OpenSim and Sloodle along with mobile location and NFC for checking in. *International Journal of Artificial Intelligence and Interactive Multimedia, Volume 1, Number 7*, 62-69.

Heudin, J.-C. (2000). *Virtual Worlds.* Berlin, Germany: Springer Publishing.

Hogan, R. (2010). Societal Issues, Legal Standards, & International Realities Universities Face in the Distance-Learning Market. In H. Song, *Distance Learning Technology, Current Instruction, and the Future of Education: Applications of Today, Practices of Tomorrow* (pp. 284-301). Hershey, PA: Information Science Reference.

Jensen, S. S., Phillips, L., & Strand, D. L. (2011). Virtual worlds as sites for social cultural innovation. *COnvergence: the International Journal of Research into New Media Technologies, Volume 18, Issue 1*, 3-10.

Kelton, A. (2008). Virtual worlds? “Outlook good”. *EDUCAUSE Review, Volume 43, Issue 5*, 15-22.

Kokua. (2011, January 23). *Imprudence:About - Kokua Wiki*. Retrieved January 9, 2013, from kokauviewer.org: http://wiki.kokuaviewer.org/wiki/Imprudence:About

Konstantinidis, A., Tsiatsos, T., Demetriadis, S., & Pomportsis, A. (2010). Collaborative Learning in OpenSim by Utilizing SLoodle. *Sixth Advanced International Conference on Telecommunications (AICT)* (pp. 90-95). Bercelona, Spain: IEEE Xplore.

Li, J., D'Souza, D., & Du, Y. (2011). Exploring the Contribution of Virtual Worlds to Learning in Organizations. *Human Resource Development Review, Volume 10, Issue 3*, 264-285 .

Lopes, C. (2011). Hypergrid: Architecture and Protocol for Virtual World Interoperability. *IEEE Internet Computing, Volume 15, Issue 5*, 22-29.

Malaby, T. M. (2009). *Making Virtual Worlds: Linden Lab and Second Life.* Ithaca, NY: Cornell University Press.

Margaryan, A., Littlejohn, A., & Vojt, G. (2011). Are digital natives a myth or reality? University students’ use of digital technologies. *Computers & Education, Volume 56, Issue 2*, 429-440.

Martin, S., Diaz, G., Sancristobal, E., Gil, R., Castro, M., & Peire, J. (2011). New technology trends in education: Seven years of forecasts and convergence. *Computers & Education, Volume 57, Issue 3*, 1893-1906.

Moodle. (2012). *Moodle is an Open Source Course Management System (CMS).* Retrieved January 5, 2013, from moodle.org: https://moodle.org/about/

Narten, T., Sridharan, M., Dutt, D., Black, D., & Kreeger, L. (2012). *Overlays for Network Virtualization.* Fremont, CA: Internet Engineering Task Force.

Noor, A. K., & Lobeck, W. E. (2009). Disruption from the Virtual World. *Mechanical Engineering, Volume 131, Issue 11*, 22-28.

Oldenburg, R. (2000). Celebrating the Third Place: Inspiring Stories about the "Great Good Places" at the Heart of Our Communities. New York: Marlowe & Company.

Oldenburg, R., & Brissett, D. (1982). The Third Space. *Qualitative Sociology, Volume 5, Issue 4*, 265-284.

OpenSimulator.org - Archives. (2012, September 23). *OpenSim Archives - OpenSim*. Retrieved January 7, 2013, from OpenSimulator.org: http://opensimulator.org/wiki/OpenSim_Archives

OpenSimulator.org - Compatible Viewers. (2012, December 14). *Compatible Viewers - Opensim.* Retrieved January 4, 2013, from OpenSimulator.org: http://opensimulator.org/wiki/Compatible_Viewers

OpenSimulator.org - Connecting. (2012, December 12). *Connecting - Opensim*. Retrieved January 4, 2013, from OpenSimulator.org: http://opensimulator.org/wiki/Connecting

OpenSimulator.org - HyperGrid. (2012, March 4). *Hypergrid.* Retrieved August 6, 2012, from OpenSimulator.org: http://opensimulator.org/wiki/Hypergrid

OpenSimulator.org - Services. (2012, December 31). *Services - Opensim.* Retrieved January 6, 2013, from OpenSimulator.org: http://opensimulator.org/wiki/Services

OpenSimulator.org - Wifi. (2012, December 17). *Wifi - OpenSim*. Retrieved January 8, 2013, from OpenSimulator.org: http://opensimulator.org/wiki/Wifi

Perera, I., Allison, C., & Miller, A. (2010). Secure Learning in 3 Dimensional Multi User Virtual Environments – Challenges to Overcome. *Proceedings of the 11th PGNet symposium* (pp. 1-6). Liverpool, UK: PGNet.

Rapanotti, L., & Hall, J. G. (2011). Design concerns in the engineering of virtual worlds for learning. *Behaviour & Information Technology , Volume 30, Issue 1*, 27-37.

Rymaszewski, M., Au, W. J., Wallace, M., Winters, C., Ondrejka, C., & Batstone-Cummingham, B. (2007). *Second Life: The official guide.* Hoboken, New Jersey: Wiley Publishing.

Scacchi, W., Brown, C., & Nies, K. (2012). Exploring the potential of virtual worlds for decentralized command and control. *17th Int. Command and Control Research and Technology Symposium (ICCRTS)* (pp. 33-47). Monterey, CA: Naval Post Graduate School.

Shi, X., Jin, H., Pan, X., Huang, D., Yu, B., & Jiang, H. (2012). Toward scalableWeb systems on multicore clusters: making use of virtual machines. *J Supercomput, Volume 61*, 27-45.

Soukup, C. (2006). Computer-mediated communication as a virtual third place: building Oldenburg's great good places on the world wide web. *New Media & Society, Volume 8, Issue: 3*, 421-440.

Thompson, C. (2011). Next-Generation Virtual Worlds: Architecture, Status, and Directions . *IEEE Internet Computing, Volume 15, Issue 1*, 60-65.

Trescak, T., Esteva, M., & Rodriguez, I. (2011). VIXEE an Innovative Communication Infrastructure for Virtual Institutions. *Tenth International Conference on Autonomous Agents and Multiagent Systems* (pp. 1131-1132). Taipei, Taiwan: IFAAMAS.

Ubuntu.com. (2012, May 10). *Postfix - Community Ubuntu.* Retrieved January 4, 2013, from help.ubuntu.com: https://help.ubuntu.com/community/Postfix

Warburton, S. (2009). Second Life in higher education: Assessing the potential for and the barriers to deploying virtual worlds in learning and teaching. *British Journal of Educational Technology, Volume 40, Issue 3*, 414-426.

Wasko, M., Teigland, R., Leidner, D., & Jarvenpaa, S. (2011). Stepping into the Internet: New Ventures in Virtual Wolrds. *MIS Quarterly, Volume 35, Issue 3*, 645-652.

Winkler, S. E. (2013). Opening the Content Pipeline for OpenSim-Based Virtual Worlds. In IGI Global, *Digital Rights Management: Concepts, Methodologies, Tools, and Applications* (pp. 1030-1042). Hershey, PA: Information Science Reference.