Abstract

Over the past decade, the technical marketing term “cloud computing” has led users and organizations to enter into contractual agreements that sometimes result in the inappropriate use of content without their knowledge and additional risk for breaches of privacy. These agreements also can end up “vendor locking” organizations into services that take on a subscription model as opposed to true data ownership from self-hosting. While this shift occurred, processors evolved at a pace that allowed the adoption of smartphones which now equal pocket computers. Many smartphones run on the ARM based processor platform, that facilitates powerful processing and relatively low power consumption.

With the arrival of the successful ARM processor based, educational microcomputer called the “Raspberry Pi”, a whole new market of credit card sized microcomputers has started to flourish. These microcomputers have the capability of running web server applications that can be used in libraries of all sizes. This potential “desktop datacenter” advancement is profound for small libraries like those found in churches and seminaries around the world. With low power consumption, these single board servers give the capability to service areas that have never had the opportunity to host application servers due to unstable electricity.

This paper illustrates the use of the single board computing platform named “BeagleBone Black” running Ubuntu Linux, to host the institutional repository application Omeka. This work guides a small library to build an IR site, supporting
Dublin Core metadata with Omeka, for under a $200 while assuring complete content ownership.
The Crossroads of Embracing Content Management

The Crossroads of Embracing Content Management:

Using the BeagleBone Black Microcomputer to Host Omeka.

Introduction

Technical innovation and library services have shared in an interesting evolution over the past 50 years. No matter the size or type of library, the last two decades have seen an accelerated pace of change in innovation relating to digital library services. In the 1960’s and 70’s, library non-profit organizations were highly innovative as they helped to usher in the “digital age”. However, by the late 1990’s, library organizations began to feel fiscal headwinds, and user expectation challenges. As Silicon Valley began to boom with a flood of venture capital, and web services such as Google started to take shape, the early signs of what we now call “the cloud” were forming. At the same time, libraries appeared to shift from being technically proactive, to becoming reactive when it came to hosting and owning their own digital platforms.

Today, the idea of libraries being digital service consumers has taken full hold as many libraries purchase “cloud” hosted, contracted services like integrated library platforms, content management systems, and institutional repositories. Examples are platforms such as OCLC’s WorldShare Management, ExLibris’ Alma, content management products like LibGuides, and hosted institutional repositories like bePress. In the past, though libraries did not directly program their ILS’s, or other software, they did at least take the responsibility to host and own the server platforms that ran the
service. With the advent of “the cloud”, this idea of owning and running in house systems has been replaced with a signature on a contract and an internet connection.

The purpose of this research was to offer small libraries an alternative to cloud hosted solutions. With the evolution of computer hardware and ARM based processor technology, it was decided that for this project, a digital repository could be created at a very low cost using the BeagleBone Black single board system. Using an open source operating system and application like Ubuntu Linux and Omeka, this work illustrates how a small library can insure ownership and secure institutional content locally. By following this model, libraries retain full ownership of digital material, and mitigate security risks that are not always apparent when entering a relationship with cloud a provider.

This paper does not seek to question the value of every cloud based service on the market today. However, this paper seeks to examine questions that surround the cloud, such as security, digital content ownership, and the fiscal advantages and disadvantages associated to such services. We believe these are very valid questions as another technical advancement runs parallel to the cloud. This advancement involves the basic premise that processor technology has also evolved at the same time as the cloud. This advancement in processor technology has not only resulted in the growth of “the cloud” and mobile technologies, but has also created small scale server platforms now referred to as “micro-servers”.

Many entry level micro-servers exist on the market today. The most popular of these is the Raspberry Pi, which was developed as an educational platform at the
The Crossroads of Embracing Content Management

University of Cambridge (Raspberry Pi, 2013). The BeagleBone Black is a close second in popularity which has been developed by Texas Instruments in collaboration with CircuitCo. The BeagleBone Black (BBB) is a more robust platform that is capable of running many server versions of the Linux operating system. For this proof of concept paper, we have selected the Ubuntu version of Linux to showcase this micro-server’s potential as a counter to a library cloud hosted service. We have successfully ported over the lightweight institutional repository platform Omeka, and propose its use in small theological library environments.

This proof of concept illustrates how libraries of all sizes can use the BeagleBone Black, at a cost of under $100, to host and own digital content for an institutional repository. In particular, for theological, church, and international mission libraries, this setup can be used to build a low cost platform that can host digital content, and also provide a means of preservation for the library and its customers.

Not only does this paper aim to give the reader a different idea about cloud based services and micro-server advancements, it will also illustrates our experiences building three working Omeka servers on the BeagleBone Black. A process of detailed documentation was undertaken through these server builds, to provide the reader with a framework to replicate this project. After one or two test builds, this process of building a working Omeka server can now be done by the authors in less than two hours. Most of that time is spent in downloading updates to the Ubuntu operating system. Therefore, conceptually after this research, a working institutional repository is more than possible
The Crossroads of Embracing Content Management

in less than two hours of work, with a cost that is a fraction of a typical organizational cloud contract.

Literature Review

Admittedly, there is little research available in library literature today regarding the use of single board micro-servers as a hosting platform for library services. There have been several articles written about using the Raspberry Pi as an educational platform in libraries, but finding academic scholarship around using micro-server devices to host content is limited. Instead, in this review, we will explore information at a broader level relating to institutional repository platforms and their use today. We will examine the history of the BeagleBone Black itself, the history of Omeka as a lightweight IR, and general technology advancements in the micro-server market. Application of these ideas in combination, will relate to the potential of this project for small libraries, such as those found in churches and theological institutions around the world.

Institutional Repositories Today and Potential for Theological Libraries.

Clifford Lynch (2003) defines institutional repositories as “a university-based institutional repository is a set of services that a university offers to the members of the community for the management and dissemination of digital materials created by the institution and its community members” (p.2). Ryam Crow (2002) notes the contribution of the institutional repository as “an immediate and valuable complement to the existing scholarly publishing model, while stimulating innovation in a new disaggregated
publishing structure that will evolve and improve over time” (p.4). Crow further explains the advantage of institutional repositories, “Moreover, they can be introduced by reallocating existing resources, usually without extensive technical development” (p.4). Andrew Keck (2008) notes some challenges for libraries in implementing digital repositories is to deal with the primary function in digital reposition, which is to discover the content, describe it thoroughly, and to seek new content (Keck, 2008, p.7).

There is a gap in the research on the use of institutional repositories in theological libraries. An implication for potential of research in this area exists. An institutional repository gives the theological library the potential to tell the story of the seminary through the years. The IR can be a platform to express the Christian faith throughout the world. It has the capability of allowing the student and faculty to get the Gospel message through their scholarly work outside the walls of the seminary. The opportunity for sermon manuscripts to be stored on the IR allows churches to download a student’s candidating works. Use of institutional repositories in theological libraries can impact services for both students and faculty. The institutional repository becomes the platform for the content of the seminary. Another appealing concept is that the possibilities are not limited to the United States, but can be viewed worldwide. A section for alumni would be beneficial so that alumni could use the seminary institutional repository for the work they do after graduation. The interesting feature of Omeka is that the seminary can have different item types such as Old and New Testament lessons and sermons. Through Omeka the theological library has the option to configure the collections in any way they prefer.
Andrew Keck (2008) in his work “Digital Repositories and Theological Libraries” illustrates advantages for theological libraries to setup an institutional repository with regard to benefiting the faculty and students' scholarship (Keck, 2008, p.6). An institutional repository at a theological library gives the faculty the opportunity to have a collection of their works that are published as well as unpublished in a digital format (Keck, 2008, p.6). Regarding student work such as theses and dissertations, an institutional repository provides an open access digital format for other students and researchers outside the institution (Keck, 2008, p.6). With regard to students, a possibility to consider would be when students apply for doctoral and graduate level program admissions. Representatives would access the institutional repository to view the student's work. It would be a central point for the student or alum to link prospective graduate studies programs or employers to past examples of their evaluated work.

History of the BeagleBone Black.

The current generation of single board ARM based processor technology includes the BeagleBone Black revision C among others. Logic Supply, a major supplier of the BeagleBone Black provides overview and purchasing information on the BeagleBone Black. The BeagleBone Black is made by CircuitCo and the BeagleBone is endorsed by Texas Instruments. The BeagleBone Black is open source and community focused (“Beagle Bone Black”, 2015). The elinux wiki website notes some history of the Beagle Bone Black. The amount of shipments for BeagleBone Black board was noted
The Crossroads of Embracing Content Management

being 227,612 through early 2015. The dates mentioned in total number of shipments ranged from October 2013 to February 2015.

The processor is made by Texas Instruments and is a 1GHz processor with 512MB of DDR3L RAM, an 800 MHZ 4GB eMMC onboard flash (“Beagleboard:BeagleBoneBlack”, 2015). The revision C, which is used for this project, has an upgraded onboard flash from the previous version which was 2GB and it is now 4GB. Revision C of BeagleBone Black, comes with the operating system Debian Linux (“Beagleboard: BeagleBoneBlack”, 2015). The BeagleBone Black Revision B received an upgrade from the prior model in the Texas Instruments processor. The first version of the BeagleBone Black Revision A4, showed promise, but the distribution was limited for this board (“Beagleboard:BeagleBoneBlack”, 2015). It is hopeful to see the progress that has been made with BeagleBone over the years. This progression of features on the board has advanced the technology to the next level, which now allows for application hosting for products like Omeka.

History of Omeka.

As the founder and director of the Center for History and New Media (CHNM) at George Mason University, Dr. Roy Rosenzweig was involved in many digital history projects relating to US and world history. After the September 11, 2001 attacks, he led the effort to digitally preserve the event in cooperation with the American Social History Project at the City University of New York, thanks to funding from the Alfred P. Sloan Foundation (Roy Rosenzweig Center for History and New Media and American Social
The Crossroads of Embracing Content Management

History Project/Center for Media and Learning, n.d.). This archive and many other
digital preservation projects are available through the Center for History and New Media
website today thanks to the Omeka software platform. Dr. Rosenzweig’s many
accomplishments in digital history were recognized in 2003 with the Richard W. Lyman
Award for outstanding achievement in the use of technology to advance scholarship in the
humanities (Roy Rosenzweig Center for History and New Media., 2007). Sadly, Dr.
Rosenzweig passed away in 2007 of lung cancer, therefore, not living long enough to see
Omeka’s initial release in 2008.

Aside from the Center for History and New Media at George Mason University
and his many digital history projects, Omeka is a large part of Dr. Rosenzweig’s legacy
that lives on (Bernstein, 2007). Since being released in 2008, Omeka is currently in
version 2.2.2, which was most recently updated in August 2014 (Roy Rosenzweig Center
for History and New Media George Mason University, “Omeka”, 2015). Though
typically considered a content management system, Omeka utilizes the Dublin Core
metadata standard to allow the cataloging and uploading of digital materials like many
mainstream institutional repositories. For use in small libraries, Omeka provides the
functionality required to serve as an institutional repository, without as many technical
hurdles to overcome (Wikipedia, "Omeka", 2014).

Though not the focus of our work, the Center for History and New Media at
George Mason University also offers a hosted cloud version of Omeka that maybe
functional for some. This service is available at several price points, but limits the user’s
ability to have access to the internal workings of the platform. With moderate coding
ability, a local install of Omeka allows the administrator to create an online repository without the need of heavy programming skills. The Center for History and New Media maintains a long listing of sites that use either the locally installed version of Omeka or the cloud hosted version using omeka.net (Roy Rosenzweig Center for History and New Media, George Mason University, "Sites Using Omeka", 2015).

Micro-server Development Today.

Running parallel to “the cloud” movement across industries today is the advancement of processing power and the decrease of server size. This scaling down of size and increase in power is somewhat a result of Moore’s Law, which predicts a doubling of processing power every two years (Computer History Museum, n.d.). The conceptual validity of Moore’s Law is not a topic of this work, and is only referenced here to illustrate that there is some level of theoretical momentum behind the fact that server technology is in fact decreasing in size at a greater rate over time. The technologist whether in a library or elsewhere, now has a fiscally viable choice between hosting services in “the cloud” with a service provider and their contractual assurances, or self-hosting with an extremely small scale server in house. These processor advancements are widespread across the IT industry, and will impact both self-hosting and cloud hosting no matter what.

In 2011, IBM’s Watson supercomputer competed and won on the gameshow Jeopardy. At that time, it was reported that the system was the size of a large bedroom. Three years later, in January 2014, IBM announced that they had scaled down the system
to the size of “three pizza boxes” (Love, 2014). The idea of having a supercomputer with the capability of Watson’s in such a small form factor, is mind blowing to anyone who has ever spent time in a datacenter or worked around large scale systems. Such computing power using much less electricity is an incredible feat with large scale implications. IBM has since announced that this cognitive computing platform is being used for medical purposes at the Memorial Sloan Kettering Hospital (Upbin, 2013).

In January 2014, Advanced Micro Devices (AMD) also announced the first true ARM based processor server for the IT market (Shimpi, 2014). This announcement confirmed that server platforms were now poised to become lower in cost, and have a much greener energy footprint than even the most efficient rack servers in use today. The AMD server does not have the same capabilities of Watson; however, it does scale server sizes down to approximately that of an Apple iPad Mini.

IBM and AMD are far from the only two technology developers looking to scale down systems to such a small level. In early 2015, Intel announced their new “Compute Stick” that provides a ready to go Windows or Ubuntu Linux install in a case that measures four inches in length. Thus, the system size is the equivalent to a USB flash drive. This concept allows the user to plug their computer into any display that has a USB port, providing a desktop capable PC that can be carried in a pocket ("Intel® Compute Stick", 2015).

In 2011, the Raspberry Pi Foundation based at the University of Cambridge released their first Raspberry Pi platform. Once released, the target audience for the platform quickly expanded beyond educators and students to a broader group of
hobbyists and open source technology professionals. Since 2011, many programmers have shared their work with porting over various Linux based operating systems, while also using the Raspberry Pi to create hardware oriented projects that were unimaginable when the units were release in 2011 (Raspberry Pi, 2013). With a cost of $35, the Raspberry Pi 2 was released in February 2015. Upon its release, this second model came with an upgraded processor that promised a six times great performance upgrade from the prior model (Raspberry Pi Foundation, 2015). The Raspberry Pi and the BeagleBone Black used in this project, can serve as an entry level to the single board server market and provide fiscal saving for libraries of all sizes.

Like the Raspberry Pi, the BeagleBone Black is a highly popular single board micro-server among open source enthusiasts. Even with the release of the Raspberry Pi 2, the BeagleBone Black still provides a superior platform for hosting and running web applications like Omeka. With a cost of under $100, both the BeagleBone Black and Raspberry Pi offer low funded libraries a great option for self-hosting institutional repositories. Remote libraries in developing countries having less than reliable electricity also can benefit from these low cost servers due to the fact that they require far less energy to operate.

Project Process and Outcomes

The goal of this project was to get the Omeka application installed on the BeagleBone Black. To accomplish this, we first needed to get Ubuntu installed on the device itself. Online evidence through various BeagleBoard support forums indicated
that this would be possible but questions did linger prior to making the purchase of our devices. Though a working install of Ubuntu would have been a nice accomplishment alone, ultimately, the purpose of this project was to get the Omeka application installed and running.

One point to emphasize about this project is, though you do not need to be extremely knowledgeable with Linux and networking to complete the work, you do need to have some basic understanding of these areas. Knowing Linux commands and networking basics, are a baseline requirement for completing this work. It is not within the scope of this paper to cover Linux basic operations or networking principles. If needed, one can consult many print and online sources for additional information in this area.

The following sections illustrate some of the top level issues we encountered through the install process of both Ubuntu and Omeka on the BeagleBone Black. Detailed information regarding the installation procedures are available online, but are not within the scope of this paper. The generalizations that follow are not meant to provide an exact blueprint of replication for this project. However, they are intended as commentary on the feasibility of the project and the working server that results.

Server Build 1: “OmekaBBB”.

The first BeagleBone Black device arrived in the fall of 2014, and was purchased through the vendor Logic Supply. This installation took place in the office Michael Wells, Systems Librarian at Northern Kentucky University and was solely intended to be
a proof of concept server to illustrate that Omeka could be installed and run on the BeagleBone Black. This server was not intended to be used by the University, and was for research purposes only.

Upon the arrival of the device, it was quickly discovered that without a very specific display, there was no realistic way to work directly on the BeagleBone Black by plugging in a keyboard and mouse. Instead, the device needed to be interfaced through a “headless” sign-on via an SSH connection from another computer using a software such as Terminal on the Mac OS, or Putty on Windows.

The BeagleBone Black has two means of communications when operating through a headless connection. One is the USB port (#1 - Figure 1) which connects directly to a PC or Mac, and the other is a CAT 5/6 network port (#2 - Figure 1). Initially, whether you are working with the device on a large or small network, it is advisable to connect the BeagleBone Black to both the USB and a network connection. The USB connection will automatically connect to your workstation with an IP address of 192.168.7.2, which will allow you to connect via SSH. Once you are connected via SSH,
you can then use the “ifconfig” command in the BeagleBone’s Linux environment to
determine the network IP address on the network interface port (#2 - Figure 2).

By default, the BeagleBone Black is preloaded with the Debian version of Linux.
Though there is some evidence of successful installs of Omeka on Debian when doing a
broad internet search, the documentation presented on the Omeka website details
recommended information for installing the application on Ubuntu Linux. Therefore, it
was determined not to deviate from this recommended version of Linux. When you
initially connect to the BeagleBone Black, you will be doing so with the default install of
Debian. The first objective in this setup is the install of Ubuntu that has been specifically
ported over to the BeagleBone Black.

Since the BeagleBone Black does not have a hard drive, it uses an eMMC chip for
onboard storage and running the operating system. Currently, the BeagleBone Black has
4GB’s of eMMC storage, which is more than enough for installing Ubuntu. To perform
the install of Ubuntu, you first need to download the OS image from an online source
("BeagleBoardUbuntu", n.d.). This image is specially designed for the BeagleBone
Black, and requires using another application for “flashing” the Ubuntu install file on a
MicroSD card ("Ubuntu Wiki", 2014). Once the Ubuntu OS is flashed to the MicroSD
card, you insert the card into the BeagleBone’s MicroSD drive (found on the opposite
side of the board from what is depicted in Figure 1) and then initiate a special reboot of
the device.
This special reboot requires holding down micro buttons on the BeagleBone Black itself to initiate the operating system load. The button primarily used in this process is located near the MicroSD card slot and is called the “boot select” button (see #4 - Figure 2). By holding this button down during the boot up process, it tells the BeagleBone Black to read the data on the MicroSD card. Pressing this button during boot up will initiate the script that flashes the OS on the BeagleBone Black’s eMMC storage. The boot select button (#4 - Figure 2) is held down while either pressing the “power” button (#2 - Figure 2) or by removing and replacing the power cable to the board (#3 in Figure 1). You only need to press the “boot select” button for five to ten seconds during this boot up process. If done correctly, the LED lights on the board (#1 - Figure 2) that indicate activity will cycle and then stay on solid once the Ubuntu image has been
flashed to the eMMC chip storage. After this procedure, when all the lights stay on after cycling, remove the MicroSD card from the card drive slot and reboot the device.

If everything has been done correctly, this reboot will result in a working install of Ubuntu. The website where this version of Ubuntu was downloaded contains information on the credentials required for initially signing on. After signing on for the first time, it is important to change the initial password and hostname of the device to be indicative of its purpose. For this first test, the device was named “OmekaBBB”.

After getting logged on to Ubuntu, it is also vital to install all updates and upgrades for the operating system. Continual updating and upgrading is a best practice for running any server, and on Linux this practice requires very little time if done daily or weekly. During this initial install, several highly publicized security vulnerabilities were disclosed, and patched on this server. This was a high priority because it was actively running on the Northern Kentucky University network.

Once Ubuntu was installed, credentials changed, and operating system secured, it was time to begin the install of Omeka itself. To begin, we created a new Ubuntu user named “omeka” this was the account that would be used for the remainder of the install. Over the course of the install with this user, several problems occurred relating to file rights. As a result of our testing, it was identified that the “omeka” user should be placed in the “sudoers” or super users group immediately after it is created. This will not fix every file rights issue that may occur, but it will help to mitigate some problems.

The next thing to do is download the most current version of Omeka to a desktop system. Unfortunately, for “OmekaBBB”, instead of installing the most current version
of Omeka (2.2.2) an older version was used (2.2). Version 2.2 had several published security vulnerabilities, which led to the decision to build a second version of the server. This second install, named “OmekaBBBNKU” is later documented in this paper.

Once the Omeka application file was downloaded, the .zip file that contained the software was placed on the server using another SSH application called Cyberduck. The .zip file needs to be placed in the /var/www/html directory on the Ubuntu server. This allows for an easier procedure of extracting the .zip and placing the files into their proper location. The one aspect that makes Omeka easy to install is that all the files are structured in a manner that requires little manipulation. As long as all the extracted files are placed in the “html” directory, everything should function properly.

Other steps in the install process include installing and setting up a MySQL database, setting up a basic firewall application, and setting Ubuntu to save documents properly to a device interfacing with the USB device port of the BeagleBone Black. This USB port is located opposite of the ports depicted in Figure 1. This is the device port that allows connection and expansion of the BeagleBone Black to other devices such as external hard drives.

The USB storage setup proved problematic for all installs. Without making sure that the external storage device is formatted with a Linux file system first, there was great difficulty in getting Ubuntu to read the device properly. For an application like Omeka, this is very important. Since Omeka takes in documents such as PDF’s and image files, when installed on the BeagleBone Black, it needs to save these files to an external device, so that the 4GB of eMMC chip storage will not be used for document storage.
Ultimately, once each external device had the proper file system formatting (Linux - ext4) saving to the device was not a problem. Figure 3 illustrates conceptually how Omeka saves files in this environment. Because Omeka uses a default location named “original” in the “html” directory for storage, a link needs to be created during the setup process. This Linux level file linking points the default location (/html/files/original) to the Ubuntu USB device directory (/media/usb) which ultimately saves to the external device.

Figure 3: Omeka file saving setup on the BeagleBone Black.

It should be repeated that this description is an extreme top level overview and does not illustrate the detailed steps for performing this install. Though there are many steps involved, once completed two or three times, the install of Omeka on the BeagleBone Black can be replicated very easily. The second server build that took place was named “OmekaBBBNKU” and took less than two hours to complete after becoming familiar with the install procedures used for OmekaBBB.
The Crossroads of Embracing Content Management

Server Build 2: “OmekaBBBNKU”.

The second install of Omeka was performed after about a month of testing the OmekaBBB initial install. This second server configuration was much easier to complete based on experience from the first test installation. With this second server there were improvements relating to file rights issues that occurred during the first install. These differences were positive, in that, the experience from the first install led to more efficient placement of the files and less troubleshooting overall.

Additionally, the USB device issues were better documented after going through the procedures a second time. By spending more time on the install documentation with this second build, steps were documented better around using the “fdisk” utility in Ubuntu to setup the storage partition on the USB device. In the initial install, the use of “fdisk” was very cryptically documented, and thus this second install allowed for more clarity around this very important process.

This server is still running and is being used for demonstration and testing purposes at NKU. For more than four months, system performance monitoring of this BeagleBone Black has not indicated any issues with this server or with the original BeagleBone Black used for OmekaBBB. OmekaBBB has since been redeployed as a working Drupal server in a testing environment at NKU.

Server Build 3: “BBBOmekaTX”.

One of the objectives for the third server build, “BBBOmekaTX” was to have a Theological Education Institutional Repository running on a BeagleBone Black. The
The Crossroads of Embracing Content Management

research question being asked in the “BBBOmekaTX” install is how can a BeagleBone Black running Omeka help a seminary tell its history and the Gospel of Christ? The BBBOmekaTX was purchased from LogicSupply in the fall of 2014. The purpose of this installation on the BeagleBone Black was to validate the documented procedure from the OmekaBBB install by Michael Wells at Northern Kentucky University.

The BBBOmekaTX was a challenging installation not only for the Ubuntu server, but also with the Omeka 2.2.2 application. It began with a bad image file that burned to the MicroSD card. After the user burns the image to the chip, the lights on the board are supposed to become solid, but after about five to ten minutes of waiting on the lights to turn solid, it seemed something was wrong. The flashing of the operating system to the board resulted in the four lights cycling and not becoming solid. Therefore, the decision was made to use a prior version of the operating system, which was the same one used for OmekaBBB. This produced a successful install of the image to the board.

The BeagleBone Black does not have a hard drive. Therefore, an external hard drive was purchased to connect to the USB connection on the BeagleBone Black. The procedure to mount the external storage was done on the server, but the storage device was not recognized. The BeagleBone Black USB port does not supply enough power to control an external USB hard drive. A hub with 4 USB ports that has a power switch for each port was purchased. After much troubleshooting of the 500 GB external drive, this installation had to resort to using a 16GB solid state flash drive. The 16GB flash drive had to be formatted to be a “Linux-swap” partition using “gparted” on an external Linux system. This is a limitation in the setup that was found for those who are not overly
technical. Further research needs to be done to identify if an external mechanical hard drive can be used on the BeagleBone Black.

Another difficulty experienced with this install was establishing a connection to the MySQL server during the Omeka setup. The other challenge with this install was the rights on the files and directories. The key with installing Omeka is to make sure the rights are correct at the file and folder level, plus making sure the owner is the Omeka user. The MySQL server needed to be uninstalled and re-installed on this Ubuntu server.

Again, the purpose of this Omeka install was intended for use with a research objective in theological education as an institutional repository. On this institutional repository it is setup to have Old and New Testament lessons within Omeka. Potential future use is to have a collection of sermons, especially to promote events such as a student preaching week.

Operating System Updates and Disaster Recovery Planning.

It is best practice to update your Ubuntu operating system at least weekly. Performing updates can be done manually from a command line, or scheduled through the cronjob capability in Ubuntu. During this project, several security vulnerabilities were disclosed with the operating system used. Updating the server was imperative to the success of each server installation. Keeping server installs current, along with a backup methodology is a responsibility taken on when self-hosting.

Devices fail and disasters happen therefore, it is always advisable to consider a plan of what to do if a BeagleBone fails while being used to host an application like
Omeka. Even though the device uses solid state storage and not a moving hard drive, it still could withstand an electric surge or catastrophic event. When considering a backup methodology for a production environment running Omeka, the setup used in this project, should be replicated with a second BeagleBone Black.

Having a backup copy of an Omeka server, installed on a second BeagleBone Black in the same way as done in this project serves as a failover server. By having a second unit setup the same way as what is done in this project, provides a “swappable” second server to put in place if your first unit were to fail. Using the BeagleBone Black in place of a traditional server creates an appliance-like device that can be quickly replaced in the event of a failure.

Content Ownership versus “The Cloud”.

This study is evaluating an alternative to the “cloud” allowing the organization to have ownership of content. It is implied the “cloud” maybe a better solution for the organization. However, there are many factors to take in consideration when making the decision to outsource an institutional repository or other services.

The National Information Standards Organization (NISO) in January 2015 released a grant proposal outlining a framework on how to protect patron data as it relates to a digital library. In the NISO document, it shows an extreme need for libraries to own content and be able to protect patron data (Carpenter, Lagace, Hinchliffe, Tijerina, & Zimmer, 2015 p.3). This NISO proposal identifies that libraries utilize services from outside providers such as publishers and vendor supplied systems (Carpenter, Lagace,
The Crossroads of Embracing Content Management

Hinchliffe, Tijerina, & Zimmer, 2015, p.3). This proposal further identifies that it is uncertain as to what occurs as primary and secondary vendors utilize patron data (Carpenter, Lagace, Hinchliffe, Tijerina, & Zimmer, 2015, p.4). The document discusses the fact that when libraries partner with third party vendors then there is loss of oversight with patron data. These are some valid points that libraries need to be aware of when contracting with vendor providers. The patron is entrusting the library with their data. Libraries need to take responsibility to make sure data is safe and this NISO proposal is a step in that direction.

Robin Hastings (2009) notes the benefits and problems when dealing with cloud computing (Hastings, Library Technology Reports, 2009, p.10-11). The benefits listed include, day to day technical operations are done by someone else, including backups of data (Hastings, Library Technology Reports, 2009, p.10). The problems that are associated with cloud computing include security and document versioning (Hastings, Library Technology Reports, 2009, p.11). There are advantages to cloud computing, but is it worth the security and privacy risk to store content elsewhere? It is always worthwhile to look at internal solutions that allow organizations to bear responsibility for data. With cloud computing, transmission of data is a high consideration as well. Organizations should consider how many hops it takes to get to data to its destination. This question of transmission introduces many players into the data model process, and further jeopardizes patron privacy.

Should the organization want someone else to be concerned about security? Vendors may assure the organization that they will provide security however this
responsibility may or may not be at the level it needs to be. Knowing that patron data
given to the library needs to be kept private, in the cloud, it may not be as private as we
think. There is a sense security with convenience, but is it worth it in the end?

Conclusion

When evaluating alternatives to hosting institutional repositories in the cloud, the
BeagleBone Black running Omeka should be considered for certain organizations. The
solution documented in this work provides an organization with a cost effective way to
host scholarly content. With this self-hosting responsibility comes data management and
security visibility that is lacking when reliant on a third party vendor. Omeka may not be
on the decision maker’s radar thanks to the marketing efforts of large hosting providers in
the current library technology environment. However, it is an open source institutional
repository platform that can be a viable solution for many institutions. Ultimately, an
organization needs to ask questions around owning and securing data and scholarly
content before seeking to replicate this work.

In researching the BeagleBone Black hosting Omeka, several technical challenges
occurred along the way. However, it became more evident through this work that this
solution could be an alternative for a theological library. This self-hosted solution is not
only advantageous on cost, but the BeagleBone Black is a powerful device that signals a
new era in computing. Coupling this extremely small micro-server with the Omeka
platform, provides a means of content management in house with the use of minimal
resources. Though there needs to be some technical skill in order to setup the BeagleBone
Black running Linux, it seems the advantages could outweigh the negatives as issues of security and privacy continue to grow.

If patron privacy and scholarly access are to remain important to the library profession, technical management evaluation of what it means to be committed to such ideals should occur at every level when transacting with data. In the cloud based environment, many institutions have already abandoned the concern of owning, managing, and controlling data in house for the sake of convenience. Without a commitment to in house data management, future librarian activities in the area of privacy become only rhetorical activism backed by contractual faith that cloud providers will insure safekeeping and transmission of data.

Contractual faith in the big data activities of the cloud, are currently not backed by legislative protection in the United States. Thus, many questions arise when contracting technology services to the cloud. For librarianship, one is left to wonder if the profession has simply evolved into reactive consumers and abandoned the role of technical innovator. This is a critical question to examine, especially as it relates to areas such as digital security and privacy. Privacy is a foundational ideal within librarianship and that does not change within a digital world.

We have long sought best practices for owning digital content, but often opted for the role of subscriber within the last decade. This has resulted in an abandonment of the concept of ownership for our organizations. In the age of the eBook, and other digital content, we are now subscribed consumers. Institutional Repositories have provided a counter to this movement by allowing libraries to curate and control content for their
institution. Moving that content to the cloud, even with a contractual assurance of portability and ownership, ultimately seems contrary to the purpose of a repository.
The Crossroads of Embracing Content Management

References


The Crossroads of Embracing Content Management


