

**REPORT**

**On**

**How To create a web server in Raspberry pi & Host A Website on It.**

**Course Title:*****Computer Networks***

**Course Code: *CSE 317***

**Submitted To:**

**MD. ATAULLAH BHUIYAN***Lecturer Department of CSE  
City University*

**Submitted By:**

***Md Pias Hossain - 163432563***

***Md Rasel Hossain- 163432521***

***Mehedi Hasan - 163432564***

***Alamgir Hossain - 163432584***

***Submission Date: 15-7-2019***

**Abstract**

The above mentioned features encourage the organizations and individual users to shift their applications and services to the cloud. Even the critical infrastructure, for example, [power generation and distribution](https://www.sciencedirect.com/topics/engineering/power-distribution) plants are being migrated to the cloud computing paradigm. However, the services provided by third-party [cloud service providers](https://www.sciencedirect.com/topics/computer-science/cloud-service-provider) entail additional [security](https://www.sciencedirect.com/topics/engineering/security) threats.

The migration of user’s assets (data, applications, etc.) outside the administrative control in a [shared environment](https://www.sciencedirect.com/topics/computer-science/shared-environment) where numerous users are collocated escalates the security concerns. This survey details the security issues that arise due to the very nature of cloud computing. Moreover, the survey presents the recent solutions presented in the literature to counter the security issues. Furthermore, a brief view of security vulnerabilities in the [mobile cloud computing](https://www.sciencedirect.com/topics/computer-science/mobile-cloud-computing) are also highlighted. In the end, the discussion on the open issues and future [research directions](https://www.sciencedirect.com/topics/computer-science/research-direction) is also presented.

Mainly, the practice of ‘cloud computing’ is clearly defined and explained, together with a variety of seroptions. One particular option is looked at in-depth: establishing a private cloud using a Raspberry Pi. The reader is guided with a step-by-step tutorial for setting up the necessary configurations. This setup is then compared with other popular commercial cloud services.

# **Table of content**

1. [INTRODUCTION 1](#_TOC_250025)
   1. [CLOUD COMPUTING DEFINED 1](#_TOC_250024)
   2. [INDUSTRIAL CONTEXT 2](#_TOC_250023)

1.2.1 CURRENT ISSUES FOR CORPORATE IT --------------------------------------------------------------------------------3

* 1. [SECURITY CONCERNS 3](#_TOC_250022)

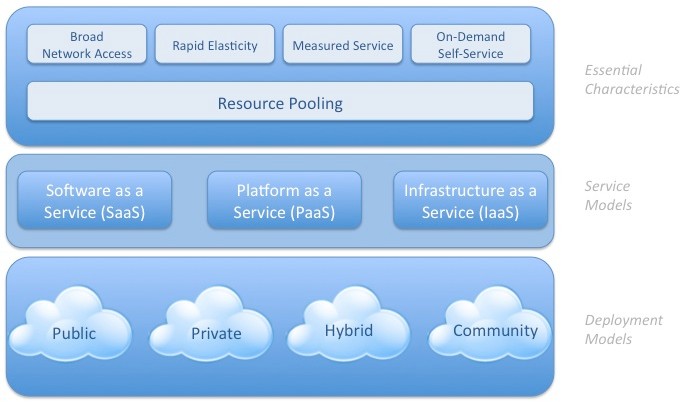
1. [PROJECT HARDWARE AND SOFTWARE 3](#_TOC_250020)
   1. [RASPBERRY PI 3](#_TOC_250019)
   2. [OPERATING SYSTEM: RASPBIAN 4](#_TOC_250018)
   3. [FILE HOSTING SERVICE: OWNCLOUD 5](#_TOC_250017)
   4. [WEB SERVER: NGINX 5](#_TOC_250016)
   5. [SCRIPTING LANGUAGE: PHP5 5](#_TOC_250015)
   6. [DYNAMIC DNS SERVICE: NO-IP 6](#_TOC_250014)
   7. [FILE SHARING PROTOCOL: AFP (NETATALK) 6](#_TOC_250013)
   8. [DATA REDUNDANCY: RSYNC](#_TOC_250012) 6
2. [METHODOLOGY 7](#_TOC_250011)
   1. [HARDWARE USED](#_TOC_250010) 7
   2. [INSTALLING OWNCLOUD 7](#_TOC_250009)
      1. INSTALLING RASPBIAN (OS) ON THE RASPBERRY PI 7
      2. FIND IP OF RASPBERRY PI AND LOGIN OVER SSH---------------------------------------------------7
      3. SETTING UP THE RASPBERRY PI SERVER AND PREPARING FOR OWNCLOUD 8
3. [RESULTS AND CONCLUSION 1](#_TOC_250006)0
   1. [RASPBERRY PI SETUP VS TO COMMERCIAL CLOUD 1](#_TOC_250005)0
   2. [COMPARISON TO NIST DEFINITION FOR CLOUD COMPUTING 1](#_TOC_250004)0
4. [LIMITATIONS 1](#_TOC_250003)0
   1. [AUTOMATIC SYNCHRONISATION WITH LOCAL FOLDER 1](#_TOC_250002)0
   2. [PROTOCOLS 1](#_TOC_250001)0
5. [CONCLUSION 1](#_TOC_250000)1
6. REFERENCES 11

# **Introduction**

## Cloud Computing defined

A cloud is massive collection of storage of various amounts of data gathered together at some servers, which helps client to connect or access the data from the server. As the data keeps on increasing there is a massive need of storage server which can hold tracks of client, their profiles, their data such as photos, document file, videos, etc. Moreover, there is risk, if our server crashes there might be loss of personal data of someone or even if the data gets theft. Sometimes the user forced to see the advertisements of products and deals on the screen while browsing which user have search for on third party applications or sites, this means somewhere in system the user data is been shared and level of privacy is not maintained. Since this shows that, there is need of storage server of own as well as privacy maintenance and security.

Our project own Cloud using Raspberry Pi provides solution for both the above given lacking qualities of cloud. Firstly, your data will be available to you all the time i.e. your personal hard drive will act as your own cloud server and secondly your data will get maximum about of security so that it won’t get theft or crashing of server won’t be a problem anymore. Our project will also allow you to share your server with others with proper authentication giving to them.



*Figure 1 : Visual Model of NIST Definition of Cloud Computing*

Another distinction must be made regarding the deployment models between public, private or hybrid clouds. In public clouds, the providers offer their physical resources to multiple

organizations and/or consumers simultaneously. It can host both a collection of services and individual services. In private clouds, these resources are not shared. In this case, the service providers dedicate specific computing power, storage and other resources to a single client. We also talk about private clouds when a company/consumer has created its own cloud by investing in resources such as a datacenter. It can then be referred to as an on-premises cloud. Public clouds can be offered more cheaply since the providers can make use of economies of scale. However, compared to private clouds, the clients have less control over things such as resource allocation and other configurations. Solutions exist that combine the cost- effectiveness of public clouds with the security and configurability of private clouds. These are called hybrid clouds.

Characteristics that apply to all cloud service models and deployment models are on-demand self-service, resource pooling, broad network access rapid elasticity and measured service. According to the NIST definition of cloud computing, a service must have these essential characteristics to be labeled as a cloud solution. ‘On-demand self-service’ is the ability to hide and adapt the limitations of the system so that it appears to be infinite to the users. ‘Resource pooling’ means that, for multiple users, the resources are combined and assigned dynamically to improve efficiency. With ‘broad network access’, the resources are available over the Internet through multiple platforms. ‘Rapid elasticity’ is achieved when the resources can be scaled easily to cope with changing demand. Next to that, the characteristic of ‘measured service’ is one that can be seen on any cloud service. The user receives certain limited resources based on the subscription model. The usage of these resources is shown to the user.

## Industrial Context

In the current fast-paced markets, it is of the essence that companies are flexible to adapt to new opportunities and threats. Therefore, focusing on the core business is increasingly important. Services that merely exist to support the delivery of the core products and core services become subject to outsourcing. In many cases a specialized company can provide the same service more cost-effectively, qualitatively and efficiently. When this is the case, outsourcing is done when it doesn’t pose a threat to the company’s intellectual property. For almost all companies, IT infrastructure makes up the backbone of their entire operation. Sometimes it includes an in-house built cloud. Factories and other facilities across borders rely on secure connections for day-to-day communication and operation. Establishing and maintaining such infrastructure requires significant investments and in-house knowledge. Although important, in most cases it is not the core business of a company. Cloud vendors today are able to offer the same, and better, infrastructure in a much more cost-effective way.

* + 1. Current issues for corporate IT

Data Centers require large investments and are thus fairly fixed assets. With a fixed capacity, it cannot cope with the unbalanced workload that arises from the complexity of current global markets. Also, for startups and SME’s, it is difficult, if not impossible, to finance such an investment. As business applications are costly to create/buy and integrate within the company, many of them are outdated and lack the ability to communicate with other application platforms outside the company. This, in turn, can put a strain on overall business performance.

## Security concerns

Many of the basic features that make cloud services so appealing, also make up the foundation of some major security concerns). Only some concerns are described here. Cloud solutions offer scalability by pooling resources. An intruder might compromise the whole service from this weak link. These shared resources together with scalability inevitably lead to one client using resources

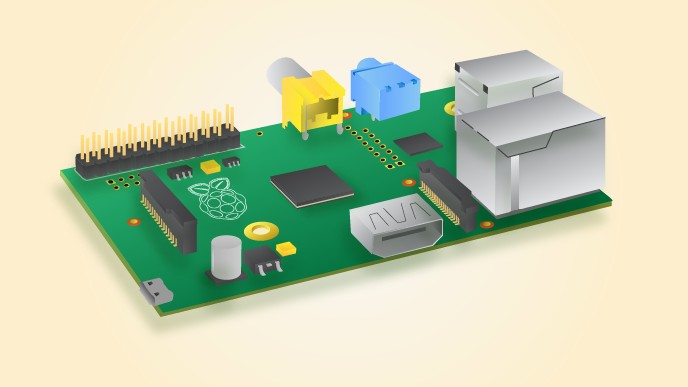
Next to that, the different cloud service models are interdependent. IaaS can be seen as the basis upon which the Paas is built. Similarly, the applications of the Saas are developed using the PaaS tools. Intruders can take advantage of these dependencies by attacking one model and compromising the other service models that are left exposed. A more obvious risk source are the service providers themselves. Employees with bad intentions might be granted sufficient access to pose a potential threat.

# **Project Hardware and Software**

The following section explains the components needed for a private cloud. Next to that, it specifies the options and the choices that were made. For this project, a Raspberry Pi 1 Model B is used, running Raspbian as operating system. There is an NGINX server running on it, as well as the file-hosting service OwnCloud and the AFP file-sharing service Netatalk.

## Raspberry Pi

The Raspberry Pi is a low cost, low energy mini-computer that is aimed at teachers and students. With it, they are introduced to programming and computers in general. It can be used as any other computer to browse the web, read e-mails etc., but in most cases it will be used for specific projects and for prototyping. When correctly set-up, it can be used as a NAS box, very much like the of-the-shelve-boxes.



*Figure 2 : Raspberry Pi 1 model B*

Chip: Broadcom BCM2835 Processor: ARMv6 single core Processor speed: 700 MHz

Power consumption: 600 mA @ 5V Memory: 512 MB SDRAM @ 400 MHz

2 USB 2.0 ports 1 SD card slot 26 GPIO pins

Ethernet: 10/100mb

GPU: Dual Core VideoCore IV Multimedia Co-Processor

## 

## Operating System: Raspbian

There are operating systems out there that are specifically tailored for building home storage devices. They are slimmed-down versions of general-purpose operating systems, focusing on file sharing protocols and features (Whitson, 2014).

FreeNAS is a sophisticated home server OS, based on FreeBSD. It supports RAID and many other things but requires at least 1GB RAM per TB of storage that you install, which the Raspberry Pi 1 does not have. An alternative would be NAS4Free, which is an old, less sophisticated version of FreeNAS. It is more suited for low-powered devices.

## File hosting service: ownCloud

ownCloud is a free and open-source suite of file hosting services with which your data is synchronised among all your connected device. It offers the same features as Dropbox and many more such as contact synchronisation. Since it is open-source, applications can easily be made to expand the platform. Examples include in-browser video streaming and a photo gallery. Alternatives for ownCloud are Pydio, Sparkleshare and Seafile. Pydio has a nicer GUI than ownCloud, better performance for large data but fewer documentation at the moment. Sparkleshare is more focused towards documents, as it offers build in version control as well as a revision history. Seafile is still very much under development and has less support for file sharing protocols.

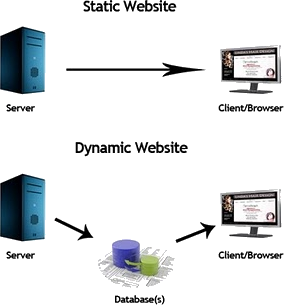
ownCloud is basically an application running on the operating system. To be available online, a web server must be installed as well.

## Web Server: NGINX

For accessing files that are stored on your Pi over the Internet via an Internet browser, we will need a web server. It stores, processes and delivers web pages to clients. Basically, a client, being a web browser, asks for a web page or a specific resource by sending an HTTP GET request to the server. There are two HTTP servers that are so popular that they make up half of the traffic on the Internet. These are Apache and NGINX (pronounced Engine X). Both are open source and can provide the same basic server solutions. However, they both have their unique features that make them better suited in specific situations .

Dynamic content delivery (such as PHP requests), however, is a task that is passed on to other modules. This means that NGINX needs to communicate with the processor via some protocol such as memcache. The two HTTP servers also differ on many other aspects such as their module system, their mapping of requests to resources and their configuration systems.

## Scripting language: PHP5

The documents can be delivered statically or dynamically. With static web pages, the user simply views a page but cannot alter the information on that page. There is no interaction. The web page is an HTML document that exists at the server-side and which is transferred to the browser when a request comes in. However, Dynamic web pages provide a solution for this. They create a live user experience. They can be provided in two ways, either by client-side scripting or server-side scripting.

## Dynamic DNS service: no-ip

Every Internet-connected household receives an IP address from the Internet service provider (ISP). These IP addresses are mostly dynamic, meaning that the service provider redistributes the IP addresses after a period of time, using the Dynamic Host Configuration Protocol (DHCP). However, this poses a problem when trying to access your private cloud from outside the local network. All Internet domain names are linked to an IP address, using a table called the Domain Name System (DNS). However, when the IP address of the server, running that domain name changes, the table is not adjusted automatically. After such a change has occurred, you won’t be able to connect anymore.

## File sharing protocol: AFP (Netatalk)

SMB (“Server Message Block”), has been the main Windows File Sharing protocol since the 90s. The protocol, although developed by IBM, is proprietary to Microsoft. However, it was revers engineered to create the Samba protocol. Samba allows unix-based systems to share files in a Windows environment and was implemented by Apple in OS X 10.2. Apple itself however, already used another protocol for file-sharing among macs (unix), called AFP (“Apple[Talk] Filing Protocol”). AFP outperforms SMB in numerous ways including read/write performance and sleep suspend support (JPY, 2015).Other file-sharing options include the older FTP, NFS and WebDAV. FTP “File Transfer Protocol

## Data redundancy: Rsync

Rsync (Remote sync) is used to copy and/or synchronise directories and files, both locally and remotely (Shrivastava, 2013). It minimizes the required effort by transferring only the differences between two sets of files. It finds where the differences are located within a file and transfers only those data blocks. The blocks that need to be transferred, are first compressed to consume less bandwidth. By default, rsync compares files and directories on both ends based on the ‘last modified date’ and the size of the file. This will miss differences that don’t appear in these two attributes. The probability of these type of differences occurring is very slow in a normal user environment. In environments where this occurs often, or where no mistakes are tolerated, the option --checksum offers a thorough checksum comparison. This requires more time to process.

# **Methodology**

This section is written as a tutorial in order to provide clarity in case this project needs to be reproduced. It is based on multiple tutorials with similar goals in mind. First, ownCloud is installed (Gus, 2015a; Koff1979, 2012; Mayank, 2016; Young, 2015) and made available over the Internet (Gus, 2015b). Next, direct file sharing is enabled (Damontimm, 2010; Thijxx, 2012).

## Hardware used

* Raspberry Pi 1 Model B
* SD card (4 GB Minimum)
* Ethernet cable
* 2 USB thumb drives
* Mac to SSH onto the Raspberry Pi

## Installing ownCloud

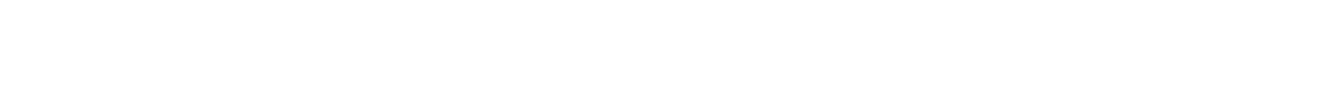
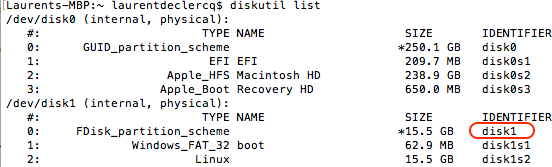
Installing Raspbian (OS) on the Raspberry Pi

After downloading the Raspbian zip file from https://[www.raspberrypi.org/downloads/,](http://www.raspberrypi.org/downloads/) it should be decompressed to a .img file. After that, it can be written to the SD card. For this, a script called Pi Filler can be used or the process below can be followed. The steps below are a slight adaptation of a tutorial on the official Raspberry Pi website.

Open the terminal.

Run the following command to identify the disk number of your SD card:

diskutil list

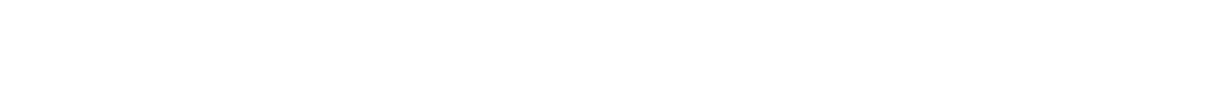
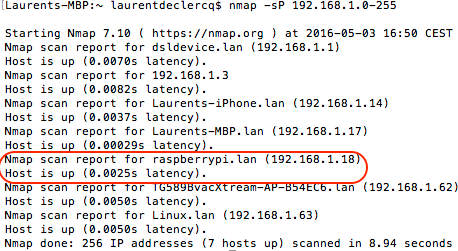


*Figure 4 : diskutil list command*

Find IP of Raspberry PI and Login over SSH

SSH uses the IP address of the device you want to connect to. it is important to know the IP address of the Raspberry Pi since this will be needed in future steps. If you don’t know the IP address of your router by heart, it will also be listed. This will be needed later on as well. Normally it should have 192.168.1.1 or 192.168.0.1 as the default IP address.

nmap -sP 192.168.1.0-255



*Figure 5 : nMap command*

Connect to it with SSH, default username is *pi* and the default password is *raspberry.* In the line below, use the IP address of your raspberry:

or simply

ssh pi@raspberrypi

ssh [pi@192.168.1.18](mailto:pi@192.168.1.18)

Setting up the Raspberry Pi Server and preparing for ownCloud

Copy and paste the following and replace the IP on the line of server\_name, with the IP of your Pi. In this case, the Raspberry Pi is located at 192.168.1.18. After that, save and exit.

*upstream php-handler { server 127.0.0.1:9000;*

*#server unix:/var/run/php5-fpm.sock;*

*}*

*server { listen 80;*

***server\_name 192.168.1.18;***

*return 301 https://$server\_name$request\_uri; # enforce https*

*}*

*server {*

*listen 443 ssl;*

***server\_name 192.168.1.18;*** *ssl\_certificate /etc/nginx/cert.pem; ssl\_certificate\_key /etc/nginx/cert.key; # Path to the root of your installation root /var/www/owncloud;*

*client\_max\_body\_size 1000M; # set max upload size fastcgi\_buffers 64 4K;*

*rewrite ^/caldav(.\*)$ /remote.php/caldav$1 redirect; rewrite ^/carddav(.\*)$ /remote.php/carddav$1 redirect; rewrite ^/webdav(.\*)$ /remote.php/webdav$1 redirect; index index.php;*

*error\_page 403 /core/templates/403.php;*

*error\_page 404 /core/templates/404.php; location = /robots.txt {*

*allow all; log\_not\_found off; access\_log off;*

*}*

*location ~ ^/(?:\.htaccess|data|config|db\_structure\.xml|README) { deny all;*

*}*

*location / {*

*# The following 2 rules are only needed with webfinger*

*rewrite ^/.well-known/host-meta /public.php?service=host-meta last;*

*rewrite ^/.well-known/host-meta.json /public.php?service=host-meta-json last; rewrite ^/.well-known/carddav /remote.php/carddav/ redirect;*

*rewrite ^/.well-known/caldav /remote.php/caldav/ redirect; rewrite ^(/core/doc/[^\/]+/)$ $1/index.html;*

*try\_files $uri $uri/ index.php;*

*}*

*location ~ \.php(?:$|/) { fastcgi\_split\_path\_info ^(.+\.php)(/.+)$; include fastcgi\_params;*

*fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name; fastcgi\_param PATH\_INFO $fastcgi\_path\_info;*

*fastcgi\_param HTTPS on; fastcgi\_pass php-handler;*

*}*

*# Optional: set long EXPIRES header on static assets location ~\* \.(?:jpg|jpeg|gif|bmp|ico|png|css|js|swf)$ {*

*expires 30d;*

*# Optional: Don't log access to assets access\_log off;*

*}*

*}*

Next, the PHP configuration file needs to be modified so that it will allow reasonable file sizes.

# **Results and Conclusion**

## Raspberry Pi setup vs to commercial cloud

Having followed all the steps provided above, has led to a product that can serve as a true alternative to commercial cloud services.the administrator has a wide set of tools to create user groups and appropriate permissions, link different calendars and contacts, offer live-collaboration, include chat, and many more. This way, ownCloud outperforms competitors such as Google Drive.As explained, the disadvantage of having the files reside on your network, is that the speed at which you can access them externally is limited to your upload speed. Externally, commercial cloud services clearly outperform the Raspberry Pi.

## Comparison to NIST definition for cloud computing

In order to be identified as a cloud solution, the Rasperry Pi configuration should answer certain criteria as stated in section 1.1. The comply with the NIST definition, the following characteristics need to be fulfilled: on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service. The Raspberry fulfils limited on-demand self- service as it requires only adaptations from the administrator to add extra storage capacity when the current configuration seems to become insufficient. This means that, this

# Limitations

## Automatic synchronisation with local folder

Here, we assume that the user wants to use a separate folder for files that need to be made available online. The user modifies the files and folders directly on the ownClouddrive itself. It would however, be equally possible to synchronise any local folder with the ownClouddrive using rsync in a similar fashion as how it is used to provide data redundancy. This way, the user keeps a local copy of the file, which is synchronized with the server in a background process. This has the advantage of independence of a reliable Internet connection. On the other hand, this requires local storage capacity.

## Protocols

Here, AFP has been chosen because the Raspberry Pi acts in a mac-only environment. However, when multiple users want to interact with the system, it is more than likely that windows devices will be involved. In that case, SMB should be used for a reliable service.

1. **CONCLUSION**

Having followed all the steps provided above, has led to a product that can serve as a true alternative to commercial cloud services. The Raspberry offers a convenient way to access and share your files at home and on the go. With the own Cloud platform, the administrator has a wide set of tools to create user groups and appropriate permissions, link different calendars and contacts, offer live-collaboration, include chat, and many more. This way ownCloud outperforms competitors such as Google Drive, that don’t provide these administrator tools to adapt the service to your needs. As explained, the disadvantage of having the files reside on your network, is that the speed at which you can access them externally is limited to your upload speed. Externally, commercial cloud services clearly outperform the Raspberry Pi.

With the AFP file-sharing protocol, the Raspberry Pi offers the possibility to access your files anywhere right within your file explorer as if they were located on your hard drive. At home, this means that you can work at LAN speeds. These are, under normal circumstances, higher than download or upload speeds. Also, unlike Dropbox, the files are not stored on your local drive, leaving storage capacity for other data. Although less safe as Dropbox, the daily backup provides a simple way to avoid losing data in case one of the drives crashes. Added to this is the security argument of controlling your data in a private environment. No privacy issues can come up and as owner of your data, you know exactly where your data is and who might have access to it.

The use of the low-cost, low-powered Raspberry Pi means that interaction with the system can take quite some time. The latest version of the Raspberry Pi, the Pi 3, provides a much faster 1.2GHz 64-bit quad-core ARMv8 CPU, doubles the RAM memory to 1GB, adds connectivity features such as BLE and doubles the USB ports (4). This model is much more suited for the computationally expensive cloud tasks such as video streaming and photo editing.

Another problem with AFP is that it works slowly when connecting over a slow or unreliable connection. Here again, it would be wise to use another protocol such as WebDAV.

# **References**

## Abstract <http://doi.org/10.1016/j.ins.2015.01.025>

Cloud Computing– <http://doi.org/10.2139/ssrn.2212051>

Raspberry Pi info [http://www.howtogeek.com/139433/how-to-turn-a- raspberry-pi-into-a-low-power-network-storage-device/](http://www.howtogeek.com/139433/how-to-turn-a-%20raspberry-pi-into-a-low-power-network-storage-device/)

Raspberry Pi <http://www.makershed.com/pages/raspberry-pi-comparison-chart>

Raspberry Pi-powered cloud service [http://www.techradar.com/how-to/computing/how-to-set-up-a-raspberry- pi-powered-cloud-service-1316017](http://www.techradar.com/how-to/computing/how-to-set-up-a-raspberry-%20pi-powered-cloud-service-1316017)