ClearOS Marketplace – Developer White Paper

March 2013

Version 1.1

Table of Contents

Introduction To ClearOS

ClearOS is an operating system based on Linux. The distribution follows closely with upstream source code released by RedHat for its RedHat Enterprise Linux family of products. The benefits of this model cannot be understated – RedHat is by far the most significant (by sales) Linux solutions vendor in the world - achieving this statue by consistently providing a secure, stable platform to its customers and own community (Fedora). By 'piggy-backing' atop RHEL source code, ClearOS inherits the security, stability and ecosystem of compiled packages (RPM-based) - both Open Source and from proprietary ISVs.

The ClearOS architecture is designed first around providing an open platform. This goal is achieved by providing compiled packages (and the source code) to the ClearFoundation so that images can be created to target specific target markets.

In its most generic form, the ClearFoundation produces and maintains ClearOS Community Edition. The Community Edition is non-specific in terms of its target market appeal. Hobbyists, home users and even businesses use the Community Edition to help them achieve common goals around the Network, Gateway and Server layers through the deployment of ClearOS Community Edition.

As of August, 2012, one additional edition has been created from the core packages provided to all via the ClearFoundation – ClearOS Professional Edition. ClearOS Professional Edition is maintained by ClearCenter and is a distribution targeted towards business (small/medium), distributed enterprise, government sector, educational sector and not-for-profit organizations.

So, just what are ClearFoundation and ClearCenter and how do they co-exist and interrelate? The next couple of sections will help to clarify this question and how it relates to app development.

ClearFoundation

The ClearFoundation is a for-profit organization registered in New Zealand. Its goal is to provide non-partisan oversight to the source code that comprises the core software. The Foundation also provides an ecosystem for users and developers to correspond, share and contribute to the good of the entire community through documentation, forums, bug/feature tracking, software version control etc.

Today, ClearFoundation is reliant to a large extent upon the sponsorship and resources provided to it by ClearCenter. Indeed, the majority of the developers contributing code to ClearFoundation are employed directly via ClearCenter. However, it is the hope and desire that this reliance and the developer pool will diversify over the next few years – indeed, it is already happening.

The ClearFoundation website can be found at http://www.clearfoundation.com.

ClearCenter

ClearCenter is a registered, US-based company, with team members residing in various countries around the world. ClearCenter created, markets and maintains ClearOS Professional Edition, a network/gateway/server distribution that is designed to provide businesses and organizations hybrid technology and choice. Hybrid technology allows an organization to decide whether to run services in the cloud or locally, on premise – or a combination, commonly termed *private cloud*.

The ClearCenter website can be found at http://www.clearcenter.com.

What Constitutes an App?

In order for an app to qualify to be listed in the ClearOS Marketplace, it must meet minimum requirements in each of the following criteria listed below:

1. **Function**
2. **Packaging**
3. **Maintenance**
4. **Support**

Function

By function, we mean the app must *do something* to enhance the functionality of the core platform. Do something? That's a pretty broad stroke...perhaps the following example will help with this definition.

If your company provides security services for Linux gateways, including penetrations tests and consulting services to ensure a level of security, you may be very interested in marketing your services the ClearOS install base. One way of creating an app that would be *declined from entry* to the Marketplace would be to create a simple app that provides your company's logo, contact info and website link with a description of the services you provide. Sorry – wrong approach.

Consider, however, if your company took the time to create an app that, either free of charge or for a fee, performed an automated pen-test and provided a report back to the user on a daily or weekly basis. This app would certainly meet the minimum 'function' element by providing the user with something of value. As a by-product, your app would help drive awareness of your organization's expertise and services and market to the ClearOS install base.

Packaging

Being derived from RHEL, ClearOS obviously uses the RPM package manager. A ClearOS app consists of at least two packages:

1. app-<basename>
2. app-<basename>-core

Where <basename> depends on the naming convention of the app (eg. app-mail-smtp would be expected to provide the Marketplace MVC components to manage an SMTP server – in this case, Postfix).

You don't have to become an RPM packaging expert to become a ClearOS developer...the heavy lifting for the creation of the spec file is done by automation scripts available as developer tools on ClearOS.

Depending on any additional software that may be dependent on your app, additional packages may be required. In many cases, these are available through one or more ClearOS repositories that are maintained by the ClearFoundation and fed by upstream sources.

If you do require code that has not been packaged in RPM file format, you will need to do so or engage ClearCenter's consulting services to take on this task for you. We do not accept apps that cannot do version control via RPM – scripts using wget to pull down tarballs from 3rd party websites...we're looking at you.

Maintenance

ClearCenter strives to provide a high-quality, secure and stable Marketplace. Developers and ISV's creating apps that are introduced to the Marketplace need to maintain a base minimum to ensure software best-practices are met. Apps (both free or paid) need to meet the following minimum requirements or they will be pulled:

1. If a flaw is reported that compromises the security of the system, this flaw will be fixed and back-ported.
2. If a bug is reported that affects the core function of the app, best efforts will be made to resolve the issue in a timely manner(within 30 days).
3. If a sole software dependency exists to another library or project, that project will be monitored by the app developer for updates fixing security exploits or significant bugs. The app developer will be responsible for providing the upgrade to the ClearOS community.

Support

The support requirement varies depending on whether your app falls in the category of free, paid or subscription.

Free – There is no onus or responsibility for a developer providing a free app into the Marketplace to provide support to anyone who considers installing the app. Good karma if you share/dedicated some time to the user forums if questions arise.

Paid – If an app enters the Marketplace as a paid submission, the developer or organization accepts *a degree* of responsibility to provide some assistance on open user forms (via ClearFoundation.com or elsewhere). As a paid app, the developer is essentially agreeing to become 'champion' of a specific feature which the app provides – sharing their knowledge and helping users who experience difficulties.

There is no Service Level Agreement or guarantee entitling users of a paid app to support and this will be clearly defined in the apps support policy.

Developers are free to create their own SLA (and charge more for the additional time/investment). For example, two apps sharing similar complexity might be priced rather differently...one at $10 and another at $150. The developer off the $150 app may have priced the app in such a way as to be able to include free technical support for 60 days, ensuring users of a good first impression and experience.

Subscription – Subscription apps with recurring fee structure are generally reserved for two scenarios:

1. A recurring service – for example, providing IDS signature updates on a weekly basis for as long as the subscription is renewed
2. Support – an app that continues with a continuous support policy to be able to seek commercial, professional support from the apps developer.
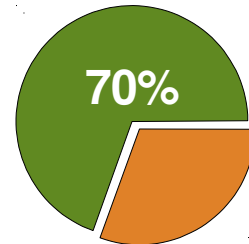
> Depending on the app complexity, ClearCenter reserves the right to decide whether an app in the Marketplace (at the time of submission) falls under the coverage of the commercial (and thus supported) ClearOS Professional subscriptions.

Marketplace Revenue Model

Revenue Share

ClearCenter is committed to creating an ecosystem where app developers can generate sustainable revenues from their submissions. To this end, the standard revenue share is split 70:30 – Developer to ClearCenter.

70%

There are cases where this share percentage is negotiated, such as:
- joint venture where shared development resources are used
- ISV's reselling software via the ClearOS Marketplace

Please contact ClearCenter *before starting development* to discuss the revenue share opportunity for an app you or your organization are thinking of creating for submission to the Marketplace.

Payment Schedule

Unless otherwise negotiated, revenues will be paid out by ClearCenter to the developer/organization on a quarterly basis if total revenues (of all apps in submission by developer) do not exceed $US 1,000, and on a monthly basis if revenue exceed $US 1,000.

Payment Transfers

Unless otherwise agreed upon, payment transfers from ClearCenter to a developer will be done via PayPal.

Currency

ClearCenter's Marketplace currently accepts payments in USD only. All submissions will be priced in US dollars.

Minimum Price

The minimum app price in the ClearOS Marketplace is US $10.

Product Bundles

ClearCenter may want to invoke 'product bundles' using an app created by you. In line with 'bundle theory', the cost of the bundle may be less than the sum of the individual apps.

Bundles can help with your apps adoption in creating higher value for customers and adding ease of use. Consider the mail stack on the ClearOS platform. There are over a dozen apps that would benefit from being bundled into 3 or 4 offerings. For example:

OSE Bundle – mail-smtp, mail-imap, mail-greylisting, mail-antimalware, mail-antispam
Home Bundle – mail-smtp, mail-greylisting, mail-antimalware, mail-antispam, zarafa-community
SMB Bundle - mail-smtp, mail-greylisting, mail-antimalware, mail-antispam, zarafa-small-business, kaspersky-mail

ClearCenter may request inclusion of your app into a bundle. If your app is paid, a negotiated revenue share less than the selling price of the individual app will need your agreement prior to inclusion.

Marketplace Exclusivity

A developer can choose whether the app is released to all ClearOS platforms or is exclusive to one or more editions (eg. ClearOS Professional Edition). Under certain circumstances, ClearCenter may require limiting the availability of an app to specific platforms.

Revenue and/or Usage Reports

A developer or organization can track installation/purchase history via the ClearCenter portal. Once your account is setup as a Developer, an additional heading ("Developer") will be visible. Select "Unit/Sales Tracking" from the sub-menu. Apps written by the developer can be selected along with a toggle to display by unit installs or revenues. Obviously, if an app is free, tracking of these apps can only be done by unit installs.

Use the sales tracking feature to invoice ClearCenter on either a monthly or quarterly basis.

Support Policy

Each app entering the Marketplace will be reviewed by ClearCenter and the app developer to determine the correct support policy. The support policy provides users of the app (and who have potentially paid for it) a clear and concise summary of what they can expect in terms of receiving support, should they require it.

To aid in the summary, ClearCenter has introduced a color-code system which quickly denotes to users what support options are available and which are not. This system is in place to set user expectations and ensure ClearCenter and its Marketplace developer and ISV community are meeting the requirements around support.

Public Forum Supported - Casual

Developer or organization may occasionally browse publicly available forums (possibly ClearFoundation's) to assist users having specific questions on the app. Developer is under no obligation to assist, however, it is a great way to earn good karma and new customers.

ClearFoundation Forum Supported - Dedicated

Developer will make best-effort to assist users by way of the ClearFoundation's community forums. There is no obligation to provide responses within a specified time.

Developer/ISV Supported

Users are provided with a link to a landing page where a support ticket can be submitted or instructions for initiating contact can be made. The landing page should also include the terms of service clearly defined. For example, users should be able to determine:
- if support is included in the apps price or available on a payment per incident basis
- what are the support hours of operation
- what kind of response time can be expected
- what costs are involved
- what is the scope of support
- etc.

ClearCenter Professionally Supported

At the discretion of ClearCenter *and* with approval from the ISV/developer, ClearCenter may offer support included under its commercially supported platform, ClearOS Professional Edition. If the ClearCenter technical team has adequate knowledge to support and app and the app is considered core to the ClearOS Professional platform, ClearCenter will provide support for an app within reasonable scope under the terms of the Professional subscription assigned to the system (eg. Professional LITE, Basic, Standard and Premium).

ClearCenter Consulting Services

At the discretion of ClearCenter *and* with approval from the ISV/developer, ClearCenter may offer support on a consulting basis (fee-based, hourly rate) for an app.

Documentation

Developers submitting apps to the ClearOS Marketplace are responsible for creating and maintaining up-to-date documentation for an app. Developers will receive access to the Wiki-based ClearCenter Documentation project. An example of a help page can be viewed here:

http://www.clearcenter.com/support/documentation/marketplace

In cases where documentation for the app resides elsewhere (eg. an ISV's website, down-loadable PDF, FOSS website etc.), providing ClearCenter with a link to the support landing page is sufficient.

Getting Started

If you've read this far, you're potentially interested in getting your software into the ClearOS Marketplace – be it an existing application (ISV), open-source project integration or an app built from the ground up using existing tools and libraries available for ClearOS.

The three most common methods (in no order of preference) of deploying an app into the Marketplace are:
1. In-house software development
2. ClearCenter consulting services
3. 3rd party outsourcing

The following sections will briefly summarize the development models listed above, including skill-set requirements, budget processes and advantages/disadvantages.

In-House Software Development

Overview
Obviously, if you are a single person creating apps for ClearOS Marketplace (either free or paid) you are your "own house". If your organization has software development team, this option may also be the preferred method to create apps or integrate your organization's application into ClearOS.

Skill-Set Requirements
Having prior knowledge of the following developer tools is essential:
- PHP (CodeIgniter MVC framework is a huge advantage)
- Programming RESTful API
- AJAX, JQuery
- SVN

Advantages
- Lower cost to entry
- Increased control over app development
- Access to ClearOS Marketplace "Mentor"

Disadvantages
- Applying resources to 'non-core' development
- Requires specific skill-set

ClearCenter Consulting Services

Overview
ClearCenter offers Consulting Services for those ISV's wanting to deploy their application onto ClearOS. Our development team has over a century of combined experience in developing software and over half of that working with the ClearOS API specifically.

By using Clearcenter's Consulting Services, you will be assigned a project manager to work with one or more of your organization's project leads to define, scope, develop, test and deploy your application on ClearOS via the Marketplace. Depending on the scope of your integration, we will assign 1 or more developers to work on a prearranged time-schedule. For more information on consulting services, please contact ClearCenter using the contact information provided at the end of this white paper.

Skill-Set Requirements

There are no skill-set requirements.

Advantages
- Remain focused on core competency
- Faster development cycle
- Developer knowledge of platform, tools and best-practices

Disadvantages
- Initial cost outlay

3rd Party Outsourcing

Please contact us to learn more about 3rd party outsourcing.

Developer Tools and Documentation

Documentation

Documentation changes rapidly and is therefor maintained on-line in Wiki format. A good place for developers new to app development for ClearOS is the Developer Overview page located on the ClearFoundation Website:

http://www.clearfoundation.com/docs/developer/start

Here, you will find everything you need to get started, including creating a developer account, creating a developer environment, sample code etc.

Developer Tools

Please refer to the ClearFoundation website for information on developer tools that are available:

http://www.clearfoundation.com/docs/developer/development_environment

API

Documentation for the core framework and app library's will be available onlne in Q4, 2012.

Resources

IRC

Developer IRC – available at http://freenode.net/ on channel #clearfoundation, developers convene regularly on Friday's from 12pm – 3pm EST to discuss technical matters related to ClearOS app development.
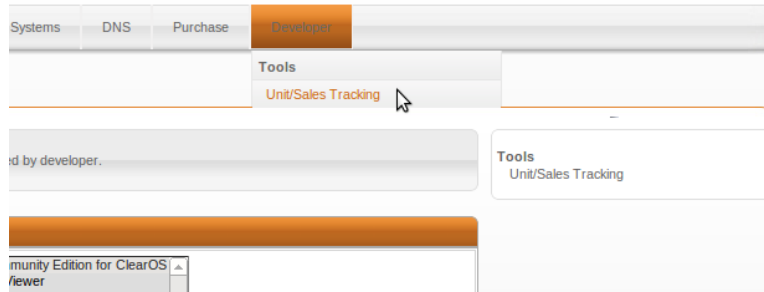
Mailing List

To get on the developer mailing list, please contact developer@clearcenter.com.

Sales Revenue & Installed Unit Tracking

Your ClearCenter account is used to track sales and/or units installed. Once configured, an additional menu will be available from the ClearCenter portal, named "Developer".

Use this tool to track revenues generated from your apps. This tracker is also used to submit revenue-share invoices to ClearCenter for payment.

App Version Control

Version control is the sole responsibility of the developer. Maintaining software plays a key role in maintaining compliance in the ClearOS Marketplace and ensuring your app is available.

If your app is built around an Open Source project and relies on additional packages or source code , you may be responsible for ensuring any upstream updates are monitored and made available in the event of a newly discovered security or severe bug fix.

> If software is external to mainstream repositories supported by either ClearCenter or the ClearFoundation, you are required to monitor and provide updates to any required (dependent) packgages in addition to your own.

Automated Updates

Automated updates are possible if you are using the ClearFoundation SVN revision control system in addition to the plague build system to natively build packages for ClearOS. Using these tools, new packages will enter ClearFoundation's public repositories or ClearCenter's private repositories, depending on whether your app is free or paid.

If you are using these systems to create your app, new versions will automatically be scanned and added to the Marketplace back-end database to notify users of pending updates. If a user's system is set to automatically update, depending on what repositories are enabled, your updates will be automatically downloaded and applied to registered systems.

You will be added to a Marketplace software version email report similar to the one shown to the right. This report will contain a list of all apps, current versions and will highlight any changes or missing information deemed important.

Marketplace Synchronization Report: 2012-11-02 17:33:53-0700

ClearOS Community, version 6.3.0

clearos

account_import	1.1.0-1	Up to date
accounts	1.2.4-1	Up to date
antiphishing	1.1.0-1	Up to date
antivirus	1.1.0-1	Up to date
bandwidth	1.0.10-1	Up to date
base	1.2.5-1	Up to date
certificate_manager	1.2.2-1	Up to date
clearcenter	1.2.2-1	Marketplace exemption
configuration_backup	1.1.0-1	Up to date
content_filter	1.1.2-1	Up to date
dashboard	1.1.0-1	Up to date
date	1.1.0-1	Up to date
dhcp	1.1.0-1	Up to date
disk_usage	1.1.0-1	Up to date
dmz	1.1.0-1	Up to date
dns	1.1.0-1	Up to date
edition	1.2.4-1	Marketplace exemption
egress_firewall	1.1.1-1	Up to date
firewall_custom	1.1.0-1	Up to date
flexshare	1.1.2-1	Up to date
ftp	1.1.1-1	Up to date
graphical_console	1.1.0-1	Marketplace exemption
greylisting	1.2.2-1	Up to date
groups	1.2.2-1	Up to date
incoming_firewall	1.1.0-1	Up to date
intrusion_detection	1.1.0-1	Up to date
intrusion_prevention	1.1.0-1	Up to date
language	1.2.2-1	Up to date

Manual Updates

Developer Access and API Keys

Keys are used to restrict access to the developer web service. Access control can be secured further to restrict by source IP, however, currently, any IP is allowed if you have a valid API key. Each developer will be issued their own key. For example:

techapps-LKJF34JPO093GDFC9712

The name prefixed in front is for logging purposes...all changes will be logged, with the first part of the key used to identify the developer.

The webservices reside at:

https://secure.clearcenter.com/ws/1.1/developer

You can either access the webservice directly (useful for integrating updates directly into your own development tools and methodologies) or use a script available in the developer tools that makes life easier (more for 'human interaction with web services).

The script can be installed by installing the developer tools:

yum install app-devel

The script name is `marketplace_version_ctl.sh` and resides in /usr/sbin/ path. Below, we'll go over the usage of the script with a couple of examples.

Using marketplace_version_ctl.sh

All web service calls using marketplace_version_ctl.sh require the following common options:

Parameter	Example	Notes
--apikey	MYNAME-12345678910	Required
--action	create_template	Required

The following web services are defined by the use of the --action parameter (eg. --action=get_versions).
get_sales
1. create_template
2. add_version
3. get_versions
4. update
5. details

Each action will be examined in more detail in the sections below.

get_sales
Gets sales numbers for a particular app.

Parameter	Example	Notes
--basename	zarafa_community	Required

Returns :
List of user purchases.

create_template
Copies a sample template to your ClearOS's temp directory. This template file can then be modified and used to create a new version.

Parameter	Example	Notes
--basename	content_filter	Required

Creates a template file in the */var/clearos/framework/tmp/* with filename:
<basename>-<userid>.template

For example:
content_filter-johndevel.template

add_version
Submit your new version to the SDN.

Parameter	Example	Notes
--basename	content_filter	Required
--template	/tmp/bob.template	Optional. If not specified, it will use the filename that gets created using the create_template method.

get_versions
Lists all versions of an app.

Parameter	Example	Notes
--basename	content_filter	Required

update
Updates an existing version.

Parameter	Example	Notes
--basename	content_filter	Required
--id	56	The app ID. The id can be retrieved from either the get_versions() or details() web service calls.
--param	state=1000&released=2012-03-01	Key/value pairs, identified with the equals sign and separated by the ampersand. Any number of fields can be updated at once. On the

		CLI, this options must be inside double quotes -ie. --param="state=1000&released=2012-03-01"

The following table defines which fields can be updated:

• version • release • released • repo_name • md5	• copyright • arch • size • packager • type	• state • code • change_log • notes

Note on the 'state' field. Three values are valid in the state field.
1000   Active
2000   Pending
3000   Disabled/Deleted

The app version will not be visible in Marketplace *unless* the state is set to 1000, ACTIVE.

details
Lists the details of a selected app version. This is useful for checking that your addition or update went according to plan.

Parameter	Example	Notes
--basename	content_filter	Required
--id	App version ID reference	Optional - if not privided, the most recent version will be displayed.

Example
A working example is probably the easiest way to see how to use the script. Let's take the 'date' app.

Looking at the current FTP site, version 1.0.4 is currently available:

app-date-1.0.4-1.v6.noarch.rpm	17-Feb-2012 17:39	9.1K
app-date-core-1.0.4-1.v6.noarch.rpm	17-Feb-2012 17:39	30K
app-dhcp-1.0.4-1.v6.noarch.rpm	17-Feb-2012 17:39	13K
app-dhcp-core-1.0.4-1.v6.noarch.rpm	17-Feb-2012 17:39	165K

Using the 'get_versions' web service call lists the current version.

```
[benjamin@devel packaging]$ ./marketplace_version_ctl.sh  --apikey=BENJAMIN-AAIUOWEM742DGHAD0F --action=get_versions  --basename=date

Version history - date
======================

ID     Version Release State    Released        Repo
3      1.0.5   1       1000     2012-03-15      clearos

[benjamin@devel packaging]$
```

> The API key used in the sample code above is for example only…This key will not work against production or test web services operated by ClearCenter.

As you can see, version 1.0.5-1 is listed as the only app in Marketplace, and is in the 'clearos' repository…a repo that all users have enabled by default. We'll need to use the *update* call, setting the version to the correct 1.0.4.

The options we'll provide are:

--apikey=BENJAMIN-AAIUOWEM742DGHAD0F
--action=update
--basename=date
--id=3
--param="version=1.0.4"

We can now check to see if our update 'took' by listing the versions:

```
[benjamin@devel packaging]$ ./marketplace_version_ctl.sh  --apikey=BENJAMIN-AAIUOWEM742DGHAD0F --action=get_versions  --basename=date

Version history - date
======================

ID      Version Release State   Released      Repo
3       1.0.4   1       1000    2012-03-15    clearos
```

But what if we release an update…in another repository? To continue our example, we find a date app which is version 1.0.5 in the update-testing repository:

Index of /clearos/enterprise/testing/6.2beta3.1/updates-testing /x86_64/RPMS

Name	Last modified	Size	Description
Parent Directory		-	
app-base-1.0.6-1.v6.noarch.rpm	09-Mar-2012 23:22	45K	
app-base-core-1.0.6-1.v6.noarch.rpm	09-Mar-2012 23:22	52K	
app-content-filter-1.0.6-1.v6.noarch.rpm	09-Mar-2012 23:22	21K	
app-content-filter-core-1.0.6-1.v6.noarch.rpm	09-Mar-2012 23:22	21K	
app-date-1.0.5-1.v6.noarch.rpm	08-Mar-2012 05:15	9.1K	
app-date-core-1.0.5-1.v6.noarch.rpm	08-Mar-2012 05:15	30K	

So, knowing that we have a data app, version 1.0.5, we need to create this instance. First, we'll download a template:

After running:

marketplace_version_ctl.sh --apikey=BENJAMIN-AAIUOWEM742DGHAD0F --action=create_template --basename=date

I should have a file here:

/var/clearos/framework/tmp/date-benjamin.template

Editing this file, I will set the version to 1.0.5 and the repo name to 'updates-testing'.

All I have changed is those two parameters, plus add a mandatory comment in the change_log field. Ideally, it should be a little more descriptive, and in time, when things become automated, this can be the SVN comment.

Now that we have our new template, time to upload/save it.

marketplace_version_ctl.sh --apikey=BENJAMIN-AAIUOWEM742DGHAD0F --action=add_version --basename=date

Should do the trick, and we can confirm that with the get_versions call.

```
created = 2012-03-15
packager = ClearFoundation
type = 0
release = 1
notes =
app_id = 3
arch =
md5 =
version = 1.0.5
modified = 2012-03-15
code = 0
license = GPL
change_log = Fixed bug
license_library = LGPL
source_rpm =
app_eula_id = 0
repo_name = updates-testing
released = 2012-03-15
state = 1000
size = 0
copyright = ClearFoundation
```

```
[benjamin@devel packaging]$ ./marketplace_version_ctl.sh  --apikey=BENJAMIN-AAIUOWEM742DGHAD0F --action=get_versions  --basename=date

Version history - date
======================

ID      Version Release State   Released      Repo
63      1.0.5   1       1000    2012-03-15    updates-testing
3       1.0.4   1       1000    2012-03-15    clearos
```

This tool and webservice only allows versioning control...a new app that has never been in the Marketplace still must go through me to initially get entered.

> This tool and webservice only allows versioning control. A new app that has never been in the Marketplace must go through the Marketplace submission review. Contact bchambers@clearcenter.com.

Summary

We hope this introduction to some of the fundamentals around developing Network, Gateway and Server apps and services for ClearOS provides clarity on many of the questions that arise when first considering developing for a new platform.

ClearCenter has a clear vision for ClearOS - to become the premier hybrid technology platform for the Network, Gateway and Server. By integrating the best of Open-Source and proprietary software, ClearOS is about offering choice – what services to run, where to deploy those services (on-premise, in the cloud or private cloud) and with whom.

Additional Reading

- • ClearFoundation Website: http://www.clearfoundation.com
- • ClearCenter Website: http://www.clearcenter.com
- • ClearOS User Forums: http://forums.clearfoundation.com
- • ClearOS Developer Resources: http://www.clearfoundation.com/docs/developer/start
- • ClearOS Live Demo: http://www.clearcenter.com/Software/clearos-live-demo.html

- ClearOS Bug/Feature Tracker: http://tracker.clearfoundation.com
- ClearOS Marketplace: http://www.clearcenter.com/marketplace/
- ClearOS Documentation: http://www.clearcenter.com/support/documentation/start

Contact Information

If you would like more information on developing your app for ClearOS or would like to get started and speak with a ClearCenter team representative, please contact Benjamin Chambers at +1.801.851.5562 or bchambers@clearcenter.com.