

1. Introduction

Open source software development area have tremendously grown up due to greater transparency, facility of modification and customization in product features. This is the reason, why many software organizations and developers leading towards open source technology which is the core component for today's businesses and researches. Supporting technologies such as cloud, which provide broad platform to open source developers, plays vital role to build backbone of open source products. Due to effect of this, giant companies like Microsoft, Redhat, IBM, VMware, Amazon, etc. demands for research in the area of open source. By considering the importance of this evolution we would like to contribute and introduce ourselves in this area of open source software development.

1.1 Problem Statement

Design and develop Kickstart server configuration tool using Glade UI.

1.2 Objectives

1. To study system working and understand the different system administration strategies.
2. To learn open source software development strategies.
3. To implement a software which automate system installation.

2. Literature Survey

Gartner is American advisory firm known for representing the maturity, adoption and social application of specific technologies. Accounting 2007's open source hype cycle produced by Gartner, Open source development tool could capture broad market applicability.

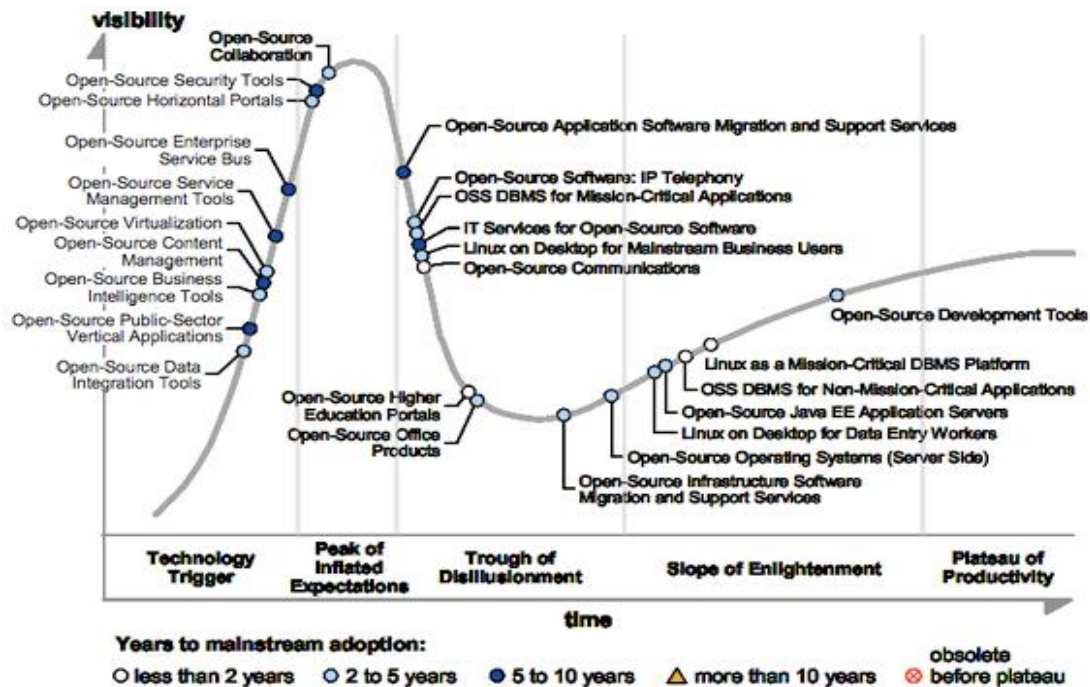


Fig 3.1 Gartner Hype Cycle for Open Source Software.

After 10 years from then, open source development tool fulfilled its all expectation and currently way beyond plateau of productivity.

International status of this project reflected from following existing software's:

- **Cobbler:** Cobbler is a Linux provisioning server that facilitates and automates the network-based system installation of multiple computer operating systems from a central point using services such as DHCP, TFTP, and DNS. It is made by Redhat Inc. and used in Fedora and Redhat Enterprise Linux.
- **Cobbler with JOOMLA:** It is open source tool made by developers generally used as cobbler having extra facility of platform independency. It can be run by browsers and supported to every linux platform.
- **PyKickstart:** PyKickstart is a python library that is used for reading and writing kickstart files. It provides a common data representation, a parser, and a writer. PyKickstart is designed to be easy to use for developers. It is easy to extend and embed in your own programs. It is developed under Fedora project for creating synchronize kickstart support between the two main programs - Anaconda and system-config-kickstart.

Design and Implementation of Kickstart Server Configuration Tool Using Python Flask

This survey proves all importance aspects of this project and provide guidelines for implementation.

3. Software and Hardware requirements

3.1 Software Requirements :

Software	Usage
Microsoft Office	Documentation and Presentation.
LaTeX	Documentation and Presentation.
Kickstart Server	To analyze installation processes.
Glade UI	GUI Framework for Designing.
PyCharm	IDE for Scripting.
IPython	External Command Shell for command execution.
VirtualBox	Testing server configuration for different VM.

3.2 Hardware Requirements :

Development system having configuration

- Least 4GB RAM
- x64 Processor CPU
- Least 20 GB Hard disk
- Networked LAN

4. Software Requirement Analysis

4.1 Product Perspective

This project software product is an implementation of new software for Kickstart server configuration. This product will have adaption of some features from existing product like Kickstart configurator and Cobbler with addition of some useful and interesting and new features. Kickstart Configurator is a simple GUI based Kickstart server configuration tool from Redhat. Cobbler is a Linux provisioning server that facilitates and automates the network-based system installation of multiple computer operating systems. From a central point using services such as DHCP, TFTP, and DNS. It can be configured for PXE, reinstallations, and virtualized guests.

The following diagram describe how product interact with the environment.

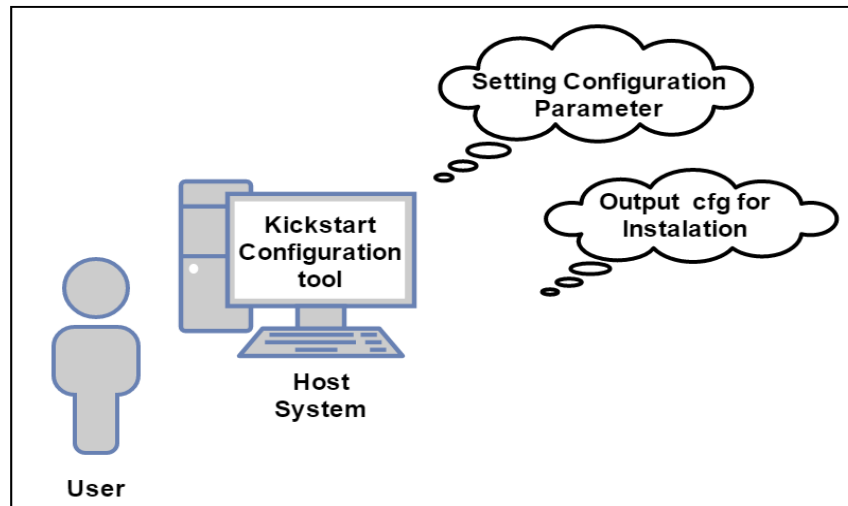


Fig. 4.1.1: Working of Product (Product User, Host System on which product is running).

4.2 Product Functionality

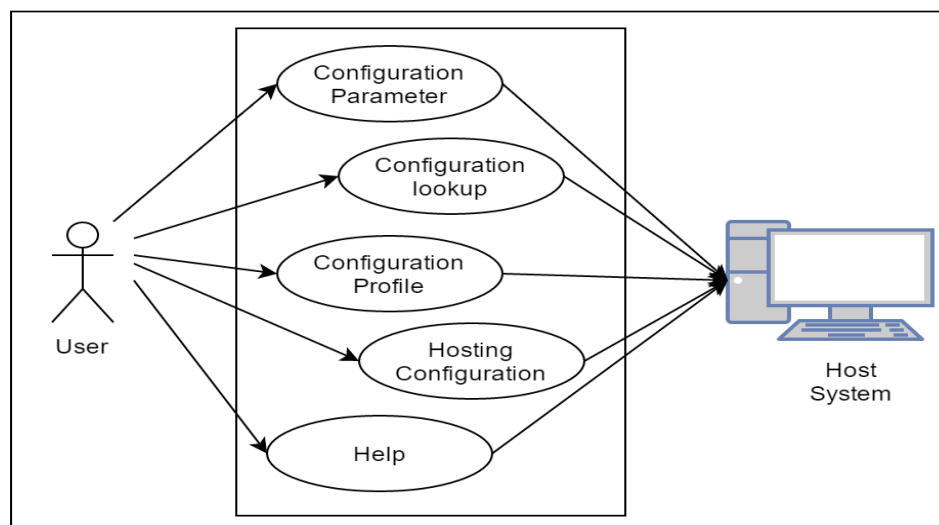


Fig 4.2.1: Use Case Showing Major functionality of the Product.

Design and Implementation of Kickstart Server Configuration Tool Using Python Flask

This product main functionalities are:

Setting Configuration parameter.

Configuration lookup.

Profile Management.

Host System Configuration.

Help

4.3 Users and Characteristics

The implementing software product don't have any user variations. That's why there is no any privilege pattern for users. The common user for this software is only one user who can do the configuration for his unattended installation. All the functionality of the software will be accessible by host system user without any authentication as a software like file editor Gedit, document writer Libre Office, Apache Open Office and Media Player etc.

4.4 Operating Environment

The minimum system architecture requirement are:

1 GB RAM

32 bit Processor

This project contain implementation of GUI using GLADE to run the project the user require the following decencies

Python 2.5 or later version

GTK+ 3 Libraries

The application and library interface is as follows

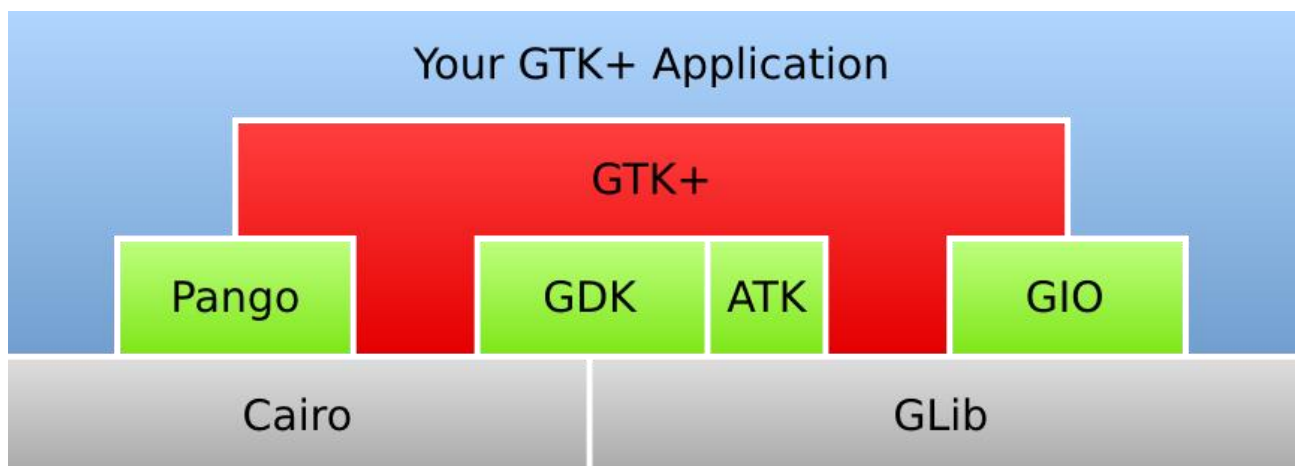


Fig. 4.4.1: GTK + and its Libraries

Design and Implementation of Kickstart Server Configuration Tool Using Python Flask

4.5 Design and Implementation Constraints

This project implementation is based on the python interpreter (python) and GTK+ libraries and Linux packages and repositories. Therefore the design and implementation constraints is as follows:

- Python Crash: Some libraries in python getting trouble during execution or libraries are not found, not responding etc.
- GTK+ Libraries Problem: Like python crash the GTK libraries have the same problem.
- Shell Command Variations: Since the product software of this project produce a configuration file to Kickstart. The file contain command if due to any processing mistake the command get modified by any empty or special character in it.
- Repository Variation: If the repository is changed by externally then configuration file will not work.
- Performance Analysis module: Since the performance analysis of a system is depend on the complexity and resource utilisation of the software. If a software is less complex and having optimum resource utilisation in its updated version which is not specified in this project. Then the performance analysis may differ.

4.6 User Documentation

For this project the user documentation is

- Standalone support or local support.
This is the documentation contain information of each module and it's working in software. This is attach with software while development.
- Online Support.
This is the documentation contain standalone documentation as well as the new issues related to software with the help of FAQ (frequently asked question) and other mode.

4.7 Assumptions and Dependencies

The assumption and dependencies for this project is as follows:

- Network Configuration:
In this project the important assumption is the host computer having Pre-boot Execution Environment (PXE) configure on it and the targeted system on which the operation system is going to install is in the same network. The host system is accessible from target system.
- Target system architecture:
The target system have the configuration specified in the configuration file.
- Network speed:
The configured network have the minimum speed of 100 mbps.
- Repo configuration: The required packages are store in the repository and configured in the network may be on host system or other system for fast access.

4.8 External Interface Requirements

4.8.1 User Interfaces

GUI Standards

This project is based on CRAP designing principal. CRAP principal aims in increase usability with four following techniques:

- **Contrast** - Make things that are different look different
- **Repetition** - Repeat the same style in a screen and for other screens
- **Alignment** - Line screen elements up! Yes, that includes text, images, controls and labels.
- **Proximity** - Group related elements together. A set of input fields to enter an address should be grouped together and be distinct from the group of input fields to enter credit card info.

This project also following Web Content Accessibility Guidelines 1.0 by the World Wide Web Consortium which focus on adoption of content, fonts and typefaces in user applications.

UI Style Guide

Each panel of user interface contain objects taken from product's style guide. This style guide made up from objects of GTK toolkit because this project follow CRAP designing techniques which contains buttons, textboxes, browse menu standards.

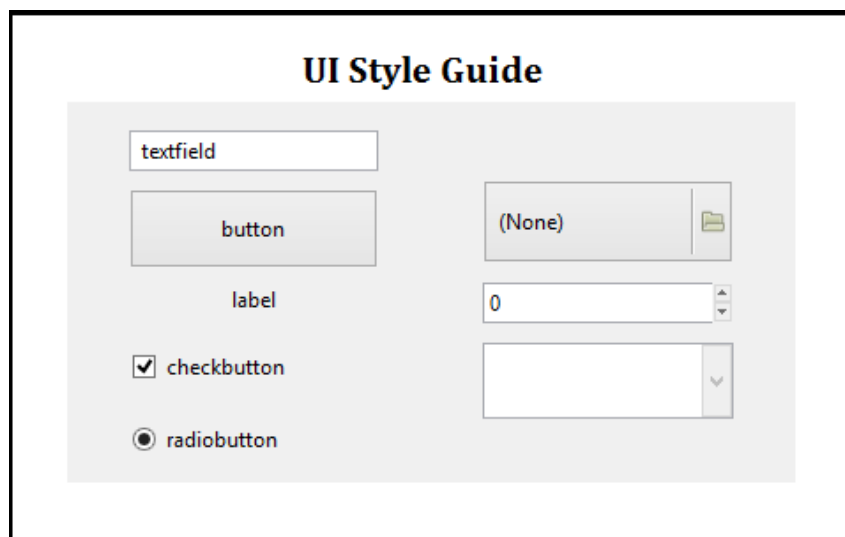


Fig. 4.8.1.1: Style Guide Toolkit of User Interface.

Screen Layout Constraints

User interface of product follows fixed layout constraints. This constraints are build and designed with consideration of research based project and may change upon requirements

Design and Implementation of Kickstart Server Configuration Tool Using Python Flask

Current layout constraints look like this

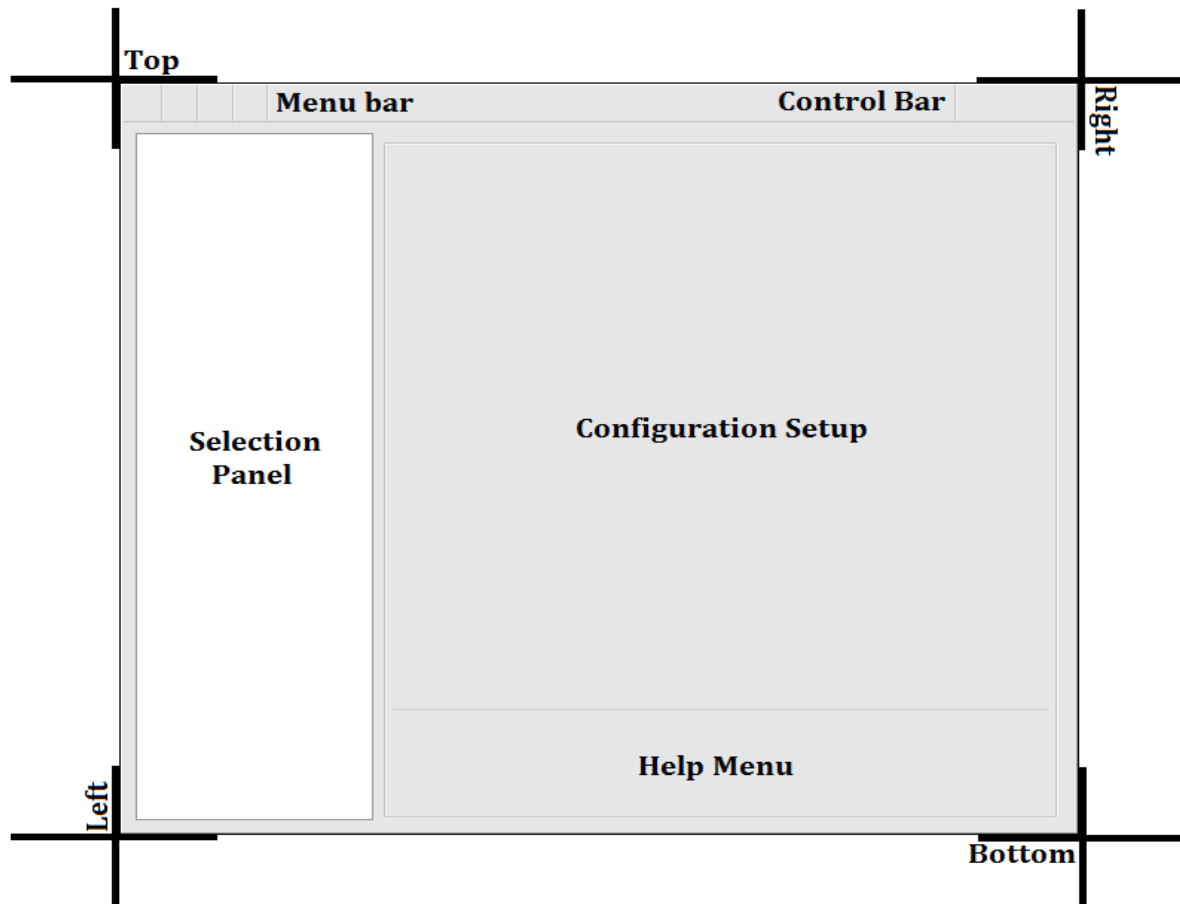


Fig. 4.8.1.2: Assumed Layout Constraints.

Assumed Prototype

Assumed product have following objects:

- Buttons
- Textboxes
- Scrollbars
- Labels
- Checkboxes
- Panels
- Menu
- Browse Functionality
- Spin button

File

Basic Configuration (required)

Default Language: English (USA)

Keyboard:

Mouse:

☐ Buttons

Time Zone:

☒ Encrypt password

Root Password:

Confirm Password:

☒ Reboot system after installation

Language Support:

Target Architecture:

Help

Perform installation in text mode (graphical is default)

Fig. 4.8.1.3: Assumed Prototype.

Hardware Interfaces

Nature of Data

The procedure of unattended installation is straightforward and sophisticatedly easy. In first step, user required to create .cfg file and provide it to pre-boot-execution environment (PXE). PXE read instruction of response_interrupt and perform customized installation on targeted system. This tool have aim to create a file builder having UI to reduce work of remembering countless commands.

Thus nature of a data is in UTF-8 text format and can be stored with .cfg extension. .cfg is a configuration file format used for storing settings. CFG files are created by many programs to store information and settings that differ from the factory defaults. CFG files usually appear as text documents, and can be opened by word processors though it is not recommended.

Supported Target Device Types

Design and Implementation of Kickstart Server Configuration Tool Using Python Flask

After precise and error free configuration of output .cfg file, operating system can be installed with following architectures:

- x86 Architecture
- AMD64
- Intel EM46T

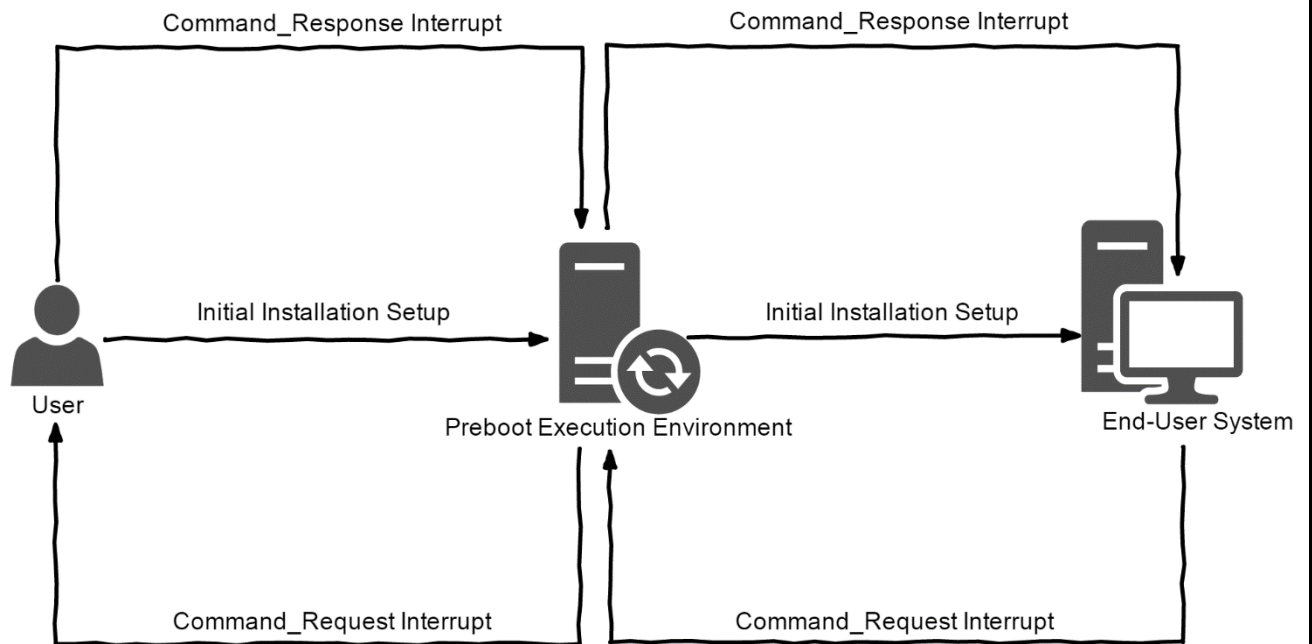


Fig. 4.8.1.4 Earlier attended installation process.

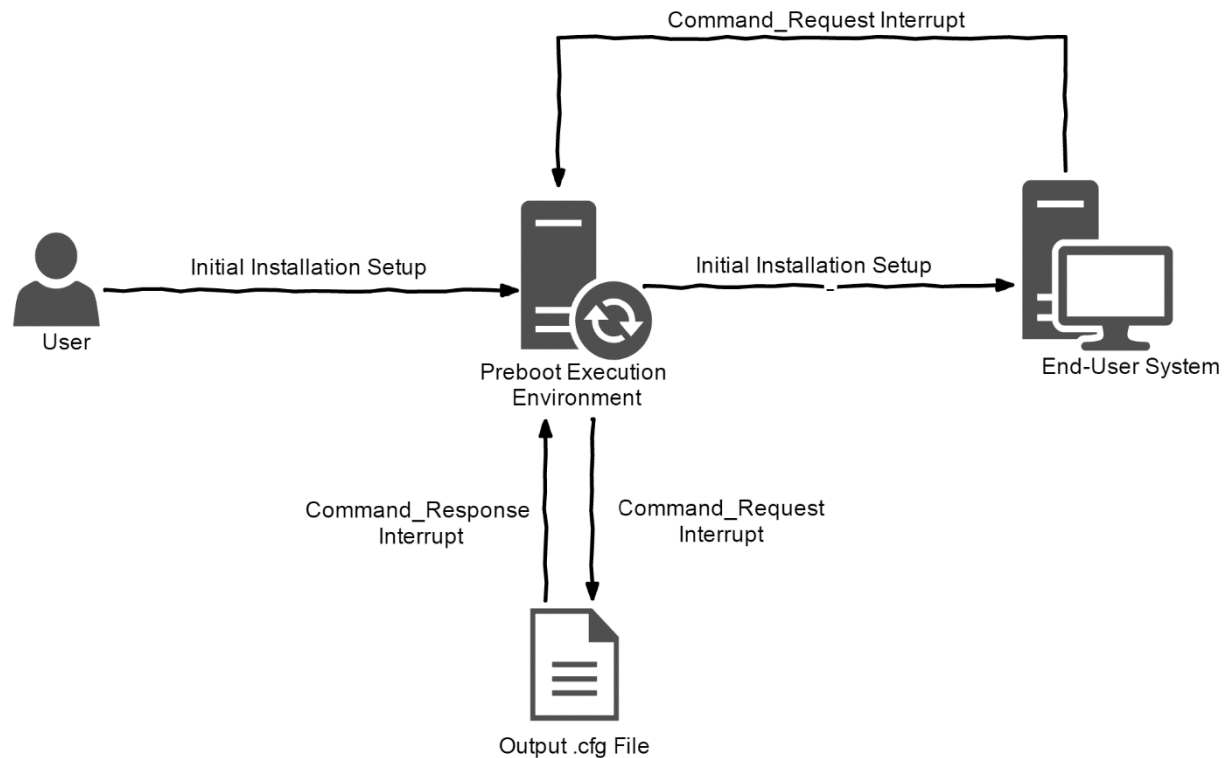


Fig. 4.8.1.5: Proposed unattended installation process.

Software Interfaces

User Interface plays key role to transform user interaction into corresponding system function. Earlier in 90's, before GTK came a long, user interaction was ridiculously difficult and done with command line interface or faulty, buggy UI frameworks.

In 1998, The GNOME project published new framework library called GIMP (GNU Image Manipulation Program) Tool Kit (GTK) and announced supportability with common Linux distributions. Due to sophisticated behaviour and attractive UI objects, GTK grows tremendously faster than any other open source UI framework.

This kickstart tool interact with GTK+ library version 3.13 which contains 203 active and 37 deprecated widgets. Glade placed in a top of the library and use to assemble GTK+'s objects with XML. Later python 3 use to manipulate tool with interactive action listeners.

Design and Implementation of Kickstart Server Configuration Tool Using Python Flask

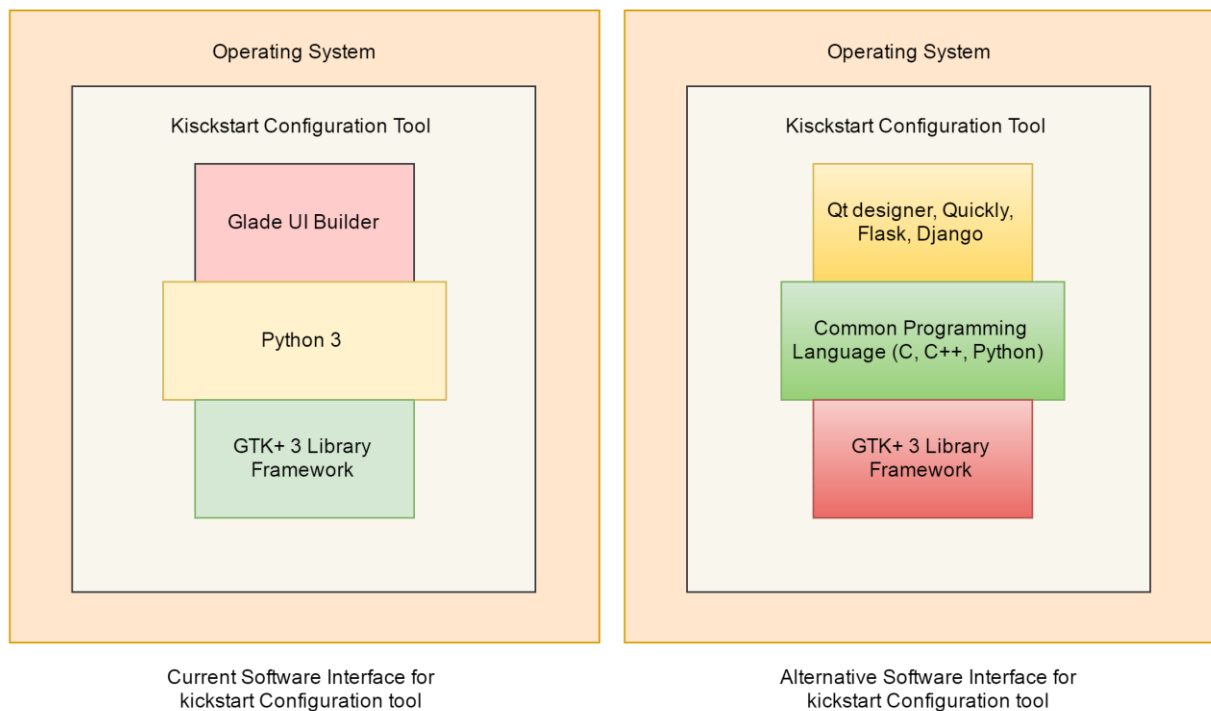


Fig. 4.8.1.6: Software Interface.

Communication Interfaces

As earlier described, Kickstart configuration tool depend on GTK+ framework were user interface built in XML. This product is heavier to work and chances of unresponsiveness with less RAM so alternative solution to this is to make product with browser supportability as cobbler with JOOMLA does.

Communication standard used

To interact with Pre-boot-Execution Environment (PXE) this tool need to follow client-server architecture. The connection which join all other client systems is need to configure with Dynamic Host Configuration Protocol (DHCP). Hence, whenever client need connect to server, it need to put request on server system, and upon request server assign dynamic IP address to corresponding client. Later, for pre and post installation of packages in targeted system is done by uploading packages to local or global repository. This repository need to make up with File Transfer Protocol (FTP) or Hyper Text Transmission Protocol (HTTP) so that user can provide link which installation in Reboot-Execution Environment (PXE).

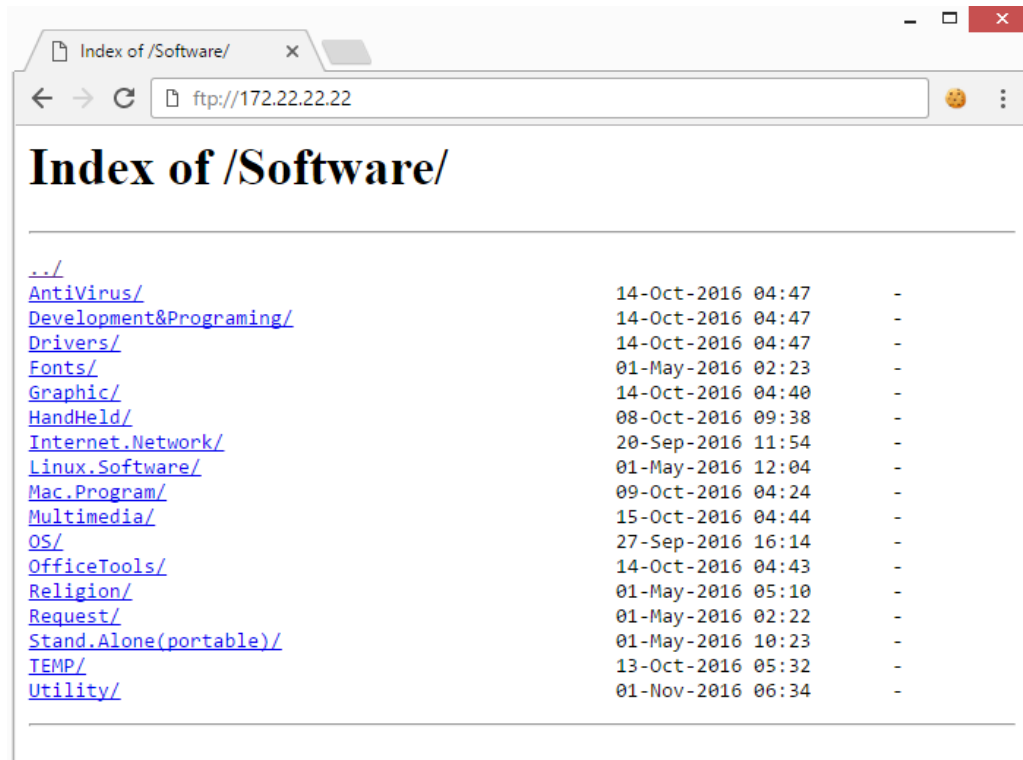


Fig. 4.8.1.7: Software Repository Uploaded via HTTP interface.

4.9 Functional Requirements

4.9.1 Basic Configuration

It include Choose of language to use during the installation and as the default language to be used after installation from the Default Language menu. Enter the desired root password for the system in the Root Password field. If the encryption option is selected, when the file is saved, the plain text password that you typed is encrypted and written to the Kickstart file.

Choosing Target Architecture specifies which specific hardware architecture distribution is used during installation. Kickstart installations are performed in graphical mode by default. To override this default and use text mode instead, select the perform installation in text mode. You can perform a Kickstart installation in interactive mode. This means that the installation program uses all the options pre-configured in the Kickstart file, but it allows you to preview the options in each screen before continuing to the next screen.

4.9.2 Installation Method

The Installation Method screen allows user to choose whether to perform a new installation or an upgrade.

If user choose upgrade, the Partition Information and Package Selection options are disabled and allows to configure- Basic configuration, Boot Loader Options, Network Configuration, Pre and Post installation script. New installation allow all configuration which disabled in upgrade.

Both new installation and upgrade allows **installation source** to choose type of installation and upgrade from following options:

- **DVD** - Choose this option to install or upgrade from the Red Hat Enterprise Linux DVD.
- **NFS** - Choose this option to install or upgrade from an NFS shared directory. It has two field NFS server and NFS directory, NFS server contain a fully-qualified domain name or IP address. NFS directory contain the name of the NFS directory that contains the variant directory of the installation tree.

For example,

If the NFS server contains the directory `/mirrors/redhat/i386/Server/`, enter `/mirrors/redhat/i386/` for the NFS directory.

- **FTP** - Choose this option to install or upgrade from an FTP server. The FTP server field contain fully-qualified domain name or IP address. For the FTP directory, enter the name of the FTP directory that contains the variant directory.

For example,

If the FTP server contains the directory `/mirrors/redhat/i386/Server/`, enter `/mirrors/redhat/i386/Server/` for the FTP directory. If the FTP server requires a username and password, specify them as well.

- **HTTP** - Choose this option to install or upgrade from an HTTP server. HTTP server field contain the fully-qualified domain name or IP address. HTTP directory contain the name of the HTTP directory that contains the variant directory.

For example,

If the HTTP server contains the directory `/mirrors/redhat/i386/Server/`, enter `/mirrors/redhat/i386/Server/` for the HTTP directory.

Design and Implementation of Kickstart Server Configuration Tool Using Python Flask

- **Hard Drive** - Choose this option to install or upgrade from a hard drive. Hard drive installations require the use of ISO images. Enter the hard drive partition that contains the ISO images (for example, /dev/hda1) in the Hard Drive Partition text box. Enter the directory that contains the ISO images in the Hard Drive Directory text box.

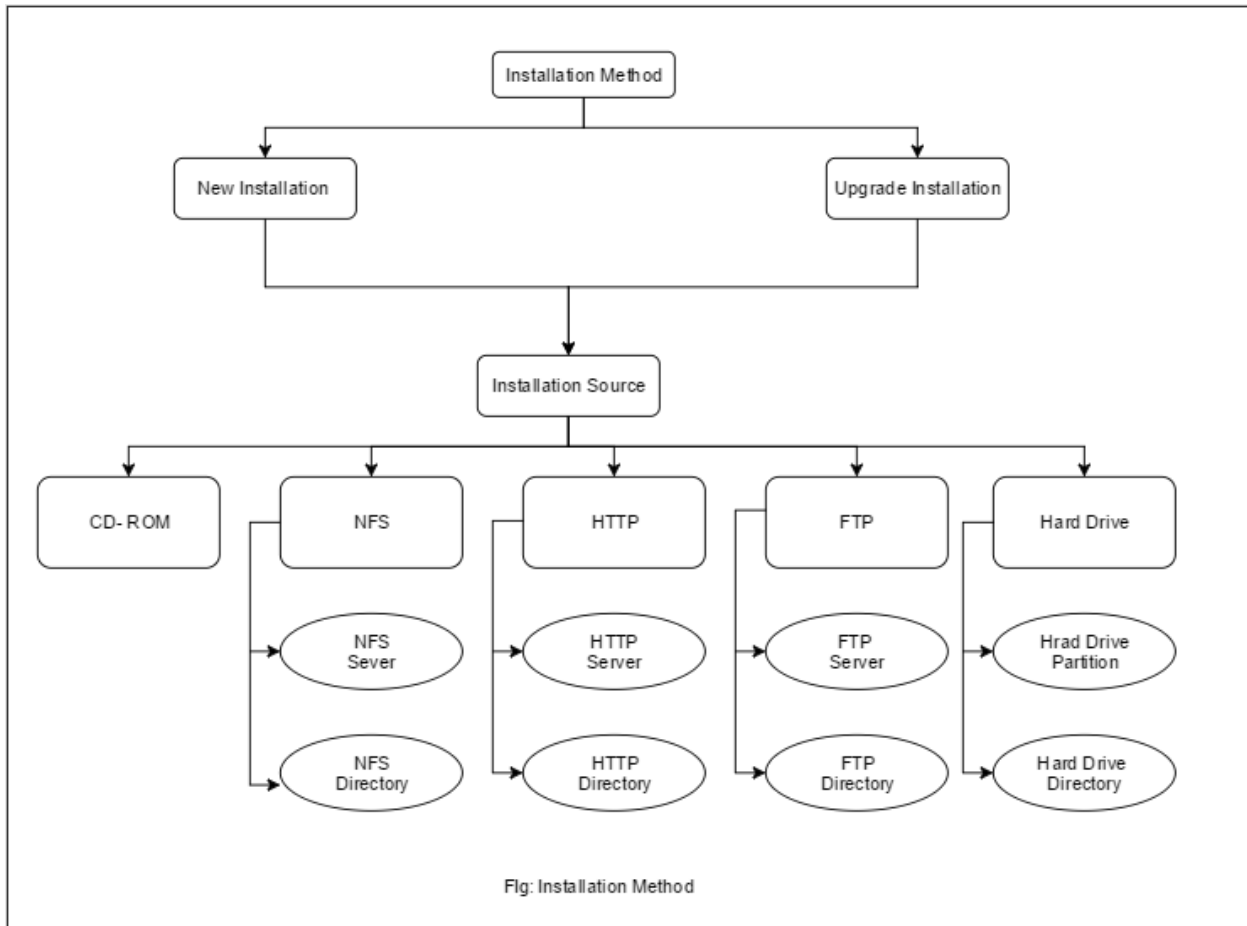


Fig. 4.9.2.1 Installation Method.

4.9.3 Boot Loader Options

- These Boot Loader options provide three feature:
 - I. Do not install boot loader.
 - II. Install new boot loader.
 - III. Upgrade existing boot loader.
- A program that loads an operating system when a computer is turned on. These Configuration available for target architecture x86 / x86_64 and disabled for Intel and IBM target architecture.

Design and Implementation of Kickstart Server Configuration Tool Using Python Flask

- GRUB is the default boot loader for Red Hat Enterprise Linux on x86 / x86_64 architectures. If user do not want to install a boot loader select **do not install a boot loader**. Which indicate that user have another way to boot their system, such as a third-party boot loader.
- If user choose to **install new boot loader** then user can either install boot loader on Master Boot Record (MBR) or on the first sector of the /boot partition.
- To pass any special parameters to the kernel to be used when the system boots, enter them in the Kernel parameters text field. For example, if user have an IDE CD-ROM Writer, user can tell the kernel to use the SCSI emulation driver that must be loaded before using cdrecord by configuring hdd=ide-scsi as a kernel parameter (where hdd is the CD-ROM device).
- During installing boot loader to increases security user can configure GRUB password. These also allow to store password in encrypted format. If the encryption option is selected, when the file is saved, the plain text password that user typed is encrypted and written to the kickstart file. If the password user typed was already encrypted, uncheck the encryption option.
- The upgrade existing boot loader is used for upgrading an existing installation in installation method to upgrade the existing boot loader configuration, while preserving the old entries.

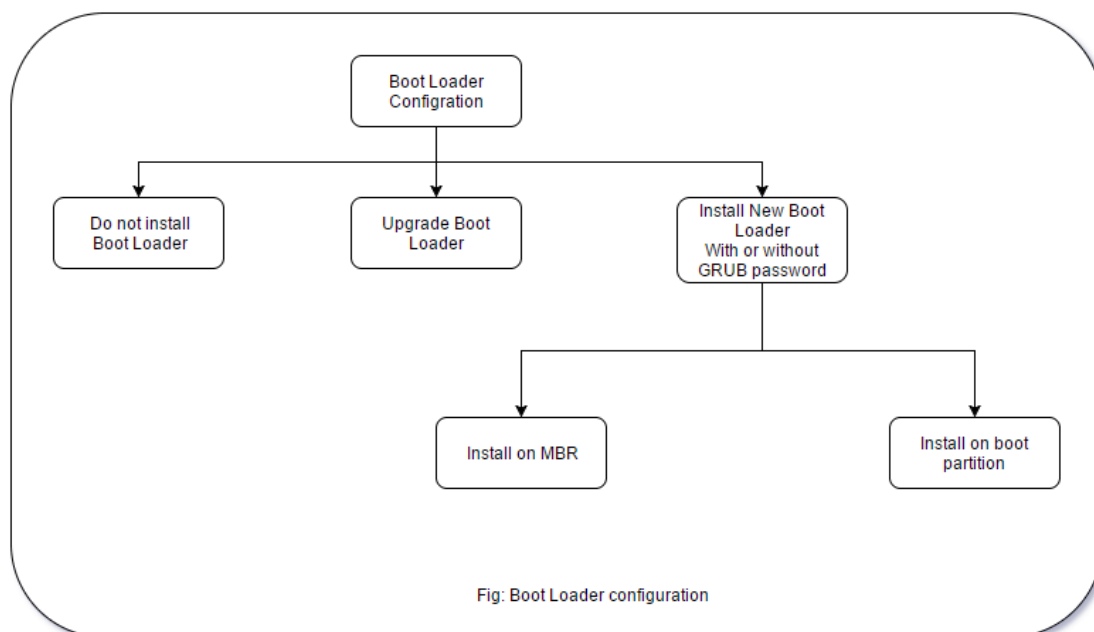


Fig. 4.9.3.1: Boot Loader Configuration.

4.9.4 Network Configuration

- If the system to be installed via kickstart does not have an Ethernet card, do not configure one on the Network Configuration page.
- Networking is only required if user choose a networking-based installation method (NFS, FTP, or HTTP). Networking can always be configured after installation with the Network Administration Tool (system-config-network).
- For each Ethernet card on the system, user can add Network Device and configure the network device and network type for the device. Select eth0 to configure the first Ethernet card, eth1 for the second Ethernet card, and so on.

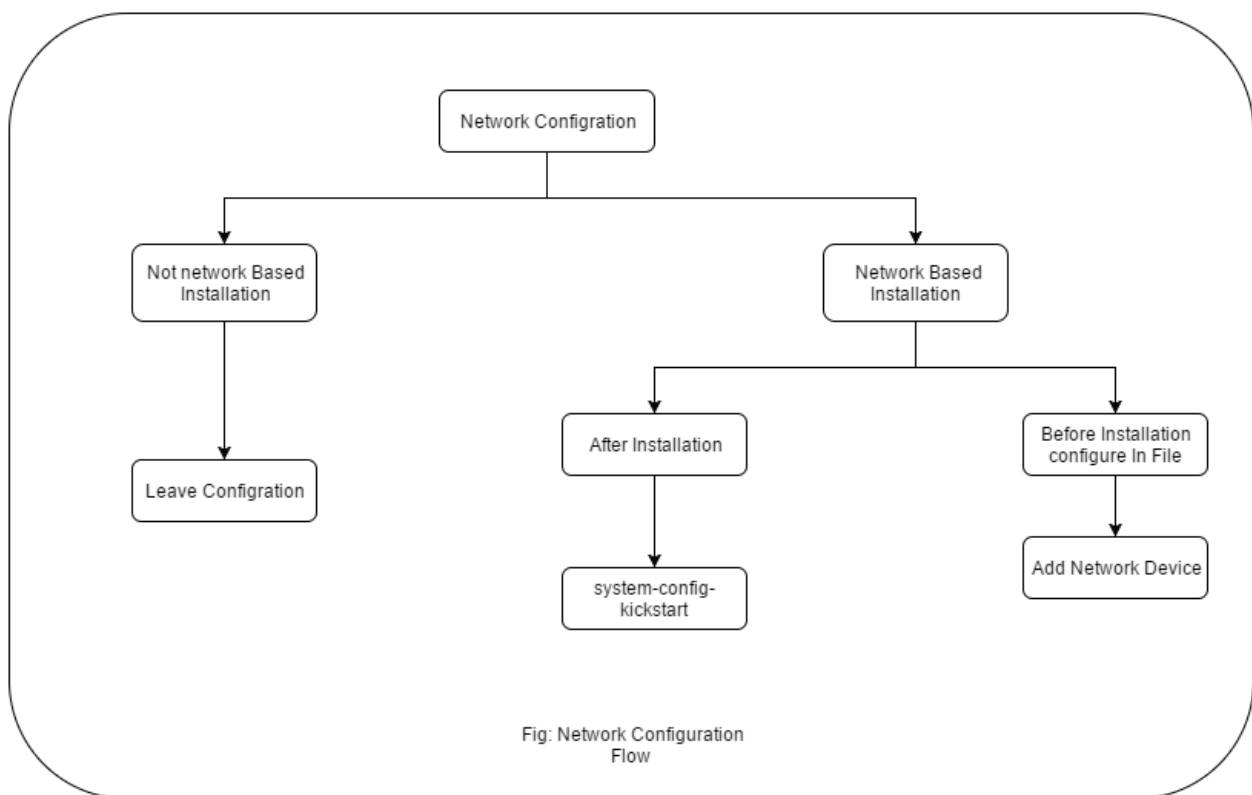


Fig. 4.9.4.1: Network Configuration Flow.

4.9.5 Partition Information

It contain 4 fields which are:

- i. Master Boot Records.

It allow user to configure clear or not master boot records.

- ii. Partition.

It allow user to remove all existing partitions, remove all existing Linux partitions, or preserve existing partitions. Preserve existing partition hide label feature.

iii. Label.

To initialize the disk label to the default for the architecture of the system (for example, msdos for x86), select Initialize the disk label if user are installing on a brand new hard drive.

iv. Layout.

It allow user to create new partition and create software RAID partition.

a) Create partition:

It allow user to create new partition by selecting mount point like /, /boot, /home, /var, /temp, /usr and filesystem of type ext2, ext3, ext4, swap, vfat, xfs. User can configure to make partition on specific drive or existing partition. User configure size in MB or fill all existing space on disk.

b) Software RAID:

It allow user to combine several disk into larger RAID device. The RAID device can be configured to provide additional speed and reliability compared to using an individual drive. Create more than one RAID partition then create RAID device by configuring mount point, filesystem type, RAID device name and RAID member.

4.9.6 Firewall Configuration

- The Firewall Configuration allows user to configure firewall settings for the installed system.
- If firewall is disabled then system allows complete access to any active services and ports. No connections to the system are refused or denied.
- If firewall configuration is enabled then system reject incoming connection are not in response to outbound request. User also allow only specific services through firewall.
- User allow trusted services such as HTTP, FTP, SSH, Telnet, SMTP. If a service is selected in the trusted services list, connections for the service are accepted and processed by the system.
- These configuration also allow additional ports pass through firewall by writing ports directly in text field. Use following format in text field as “**port:protocol**”. For example, IMAP allow through firewall by specifying **imap: protocol**.
- These allow user to set selinux policy as enforcing, permissive or disabled mode. Enforcing check context then allow service it if possible. Permissive only check context and allow service even if not possible. Disabled not check context and allow service.

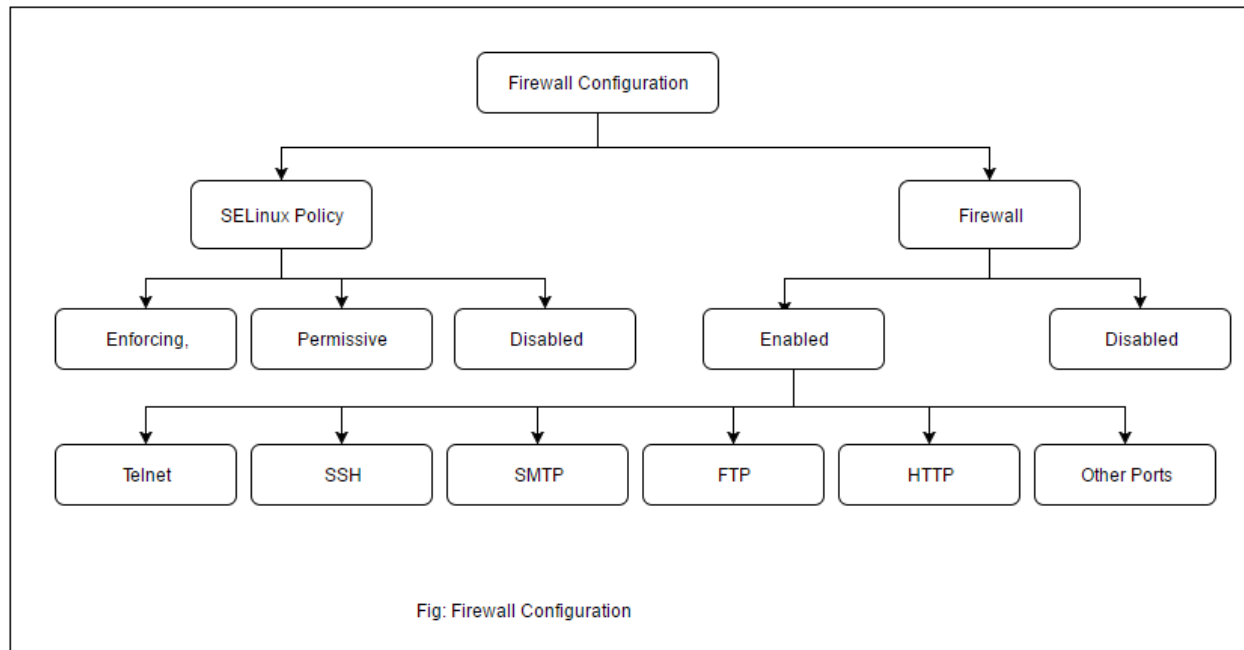


Fig. 4.9.6.1: Firewall Configuration.

4.9.7 Pre-Script Configuration

User can add commands to run on the system immediately after the Kickstart file has been parsed and before the installation begins. If user have configured the network in the Kickstart file, the network is enabled before this section is processed. To include a pre installation script pass script to this field.

4.9.8 Post-Installation Script

You can also add commands to execute on the system after the installation is completed. If the network is properly configured in the Kickstart file, the network is enabled, and the script can include commands to access resources on the network.

4.10 Behavior Requirements:

Use Case view:

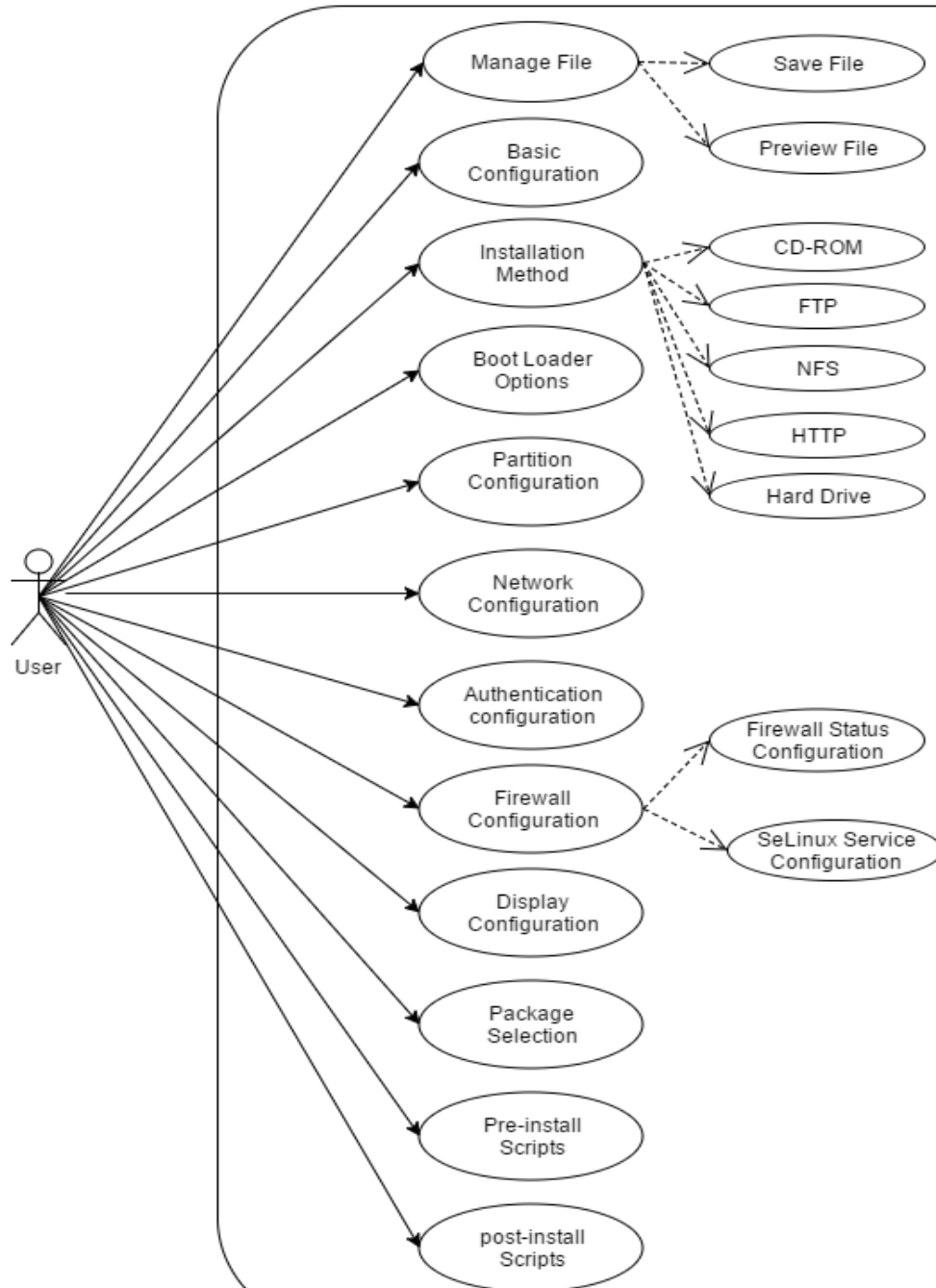


Fig: Use Case Diagram

Fig. 4.10.1: Use Case Diagram (Configuration Parameter).

4.11 Performance Requirements

The performance requirement for this software product is as follows:

- **Fast Start-up:** The software must be loaded in a main memory at most within 20-25 seconds.
- **Software Responses:** The software is running and user setting configuration parameter. In this case the parameter must be high responsive within 2-3 seconds.
- **Functionality Responses:** When user switch from one functionality to another functionality of a software. Then it must response within 5-6 second.
- **Performance Analysis Functionality:** In this functionality the result must be available to user within 30-35 second.
- **Output Configuration file:** The output configuration file must be available to user within 10-12 second when user request it.

4.12 Safety and Security Requirements

4.12.1 Encryption Standards

Some part of configuration have sensitive user data such as system passwords, LDAP passwords, etc. Password saved in plain text are too much risky and quickly recovered with middle man attack scenario, so kickstart server configuration tool required to follow some encryption mechanism to maintain security and privacy of user data.

This tool is using MD5 encryption standards, so whenever user saves sensitive data, it is encrypted with proper MD5 hashes-salt mechanism and maintain proper security leads to reduce effect of middle man attack.

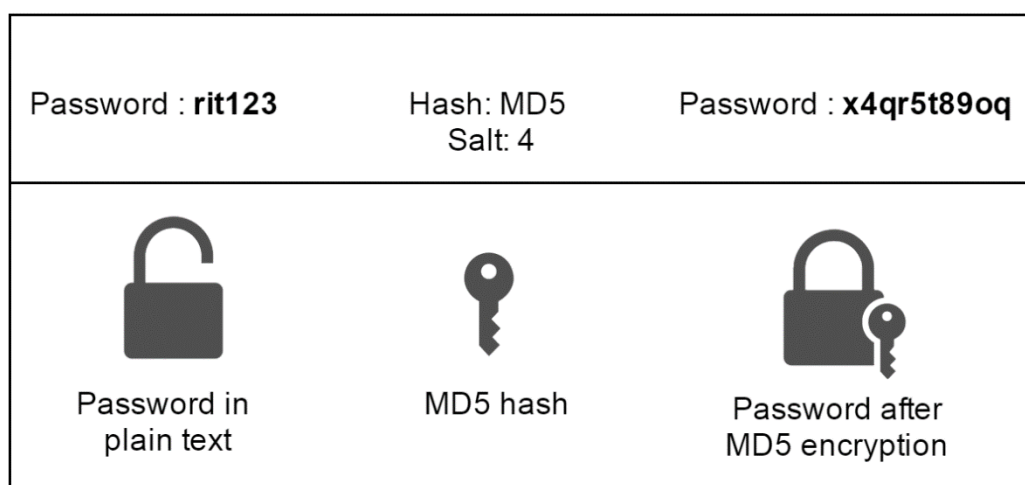


Fig. 4.12.1.1: Encryption Standards.

4.13 Software Quality Attributes

The quality attribute for this software product is as follows:

- **Availability**

The software will be available to end user when it is set up on the user system. It will launch quickly when it is run by user. The software set up will be available on internet with its modified versions. The online help documentation will be avail to user 24x7.

- **Correctness**

The software main function is to produce configuration file for unattended installation. The produced configuration file will be 100 % correct and will be checked by software product for its correctness.

- **Flexibility**

The functionality possess by software product will be flexible. There is easy switch between functionality. User will able use any one functionality at a time.

- **Interoperability**

The software will be highly interoperable it will use the system call for browser to give online help and to save an output file in the disk space.

- **Reliability**

The software will be reliable. For same configuration it will produce the same content fil again and again.

5. Coding /Code Templates

The Linux Automation Scripts:

Configure SSH:

```
#!/usr/share/python
import sys
import os
import subprocess

HOST='root@server0.example.com'
COMMAND='scp root@server0.example.com:/root/kiran2 kiosk@foundation1.ilt.example.com'

ssh = subprocess.Popen(['ssh','%s' %HOST , COMMAND], shell=False, stdout=subprocess.PIPE,
stderr=subprocess.PIPE)
result = ssh.stdout.readlines()
if result == []:
    error = ssh.stderr.readlines()
    print >> sys.stderr,"ERROR: %s" %error
else:
    print result
```

Shutdown:

```
import os
sys_no=1
while sys_no <= 20:
    d = "ssh root@172.25." + str(sys_no) + ".11" + " 'init 0'"
    #print d
    os.system(d)
    sys_no += 1
```

Sambaserver:

```
#!/usr/bin/python
import os
import sys
import crypt
a = raw_input("Enter No. Of user :")
```

Design and Implementation of Kickstart Server Configuration Tool Using Python Flask

```
def createuser (useradd1,passwd1):
    epass = crypt.crypt(passwd1,"22")
    return os.system(" useradd -p"+epass+ "-s" +"/bin/bash"+"-d"+" /home/" +useradd1+"-m"+"-c
\\\" +useradd1)
def sambaserver(user1):
    f = raw_input ("Enter Directory name:")
    net = raw_input("Enter Networ :")
    sb = raw_input("Enter you samba share name :")
    os.system('yum install samba samba-client cifs-utils -y')
    os.system("mkdir %s"%f )
    os.system("chcon -t samba_share_t %s "%f)
    os.system('echo [%s] >> /etc/samba/smb.conf"%sb)
    os.system('echo comment = public >> /etc/samba/smb.conf')
    os.system('echo path = %s >> /etc/samba/smb.conf"%f)
    os.system('echo public = yes >> /etc/samba/smb.conf')
    os.system('echo browsable = yes >> /etc/samba/smb.conf')
    os.system("echo valid users = %s >> /etc/samba/smb.conf"%user1)
    os.system("echo host allow = %s >> /etc/samba/smb.conf"%net)
    os.system('firewall-cmd --permanent --add-service=samba')
    os.system('firewall-cmd --reload')
    os.system("smbpasswd -a %s"%user1)
    os.system('systemctl restart smb nmb')
    os.system('systemctl enable smb nmb')
for x in range(a)
    useradd = raw_input("Enter User name :")
    passwd = raw_input("Enter password :")
    createuser(useradd,passwd)
    sambaserver(useradd)
```


6. Applications

Education- This Product Software will serve an interesting part of study for student. Since its implementation contain scripts which can trigger from internet. This software will be used by many lab assistance for setting their lab for particular exam or course.

Industries- The software industries require to update their system according to their project requirement. With the help of this software and Kickstart server the number of system can be set up for new project work in minimum time.

Research- This project introduce researcher a new topic that is automate system task.

7. Conclusion

Open source software development is an interesting task due to huge support of programming language like python. The implementation require the detail knowledge of Software development strategies and programming language python.

8. Future Work

This is the area of automation. Automation is the need of every one in today's life. We can extend this software at the stage where system administrator some task can be automated. Since the non-technical person can easily able to do kickstart server configuration with this project. It will reduce configuration complexity which earlier was the main reason to avoid kickstart.

This project provide users a simple and friendly UI to do this configuration, this will increase the use of kickstart server in many different areas.

9. Bibliography / References

- 1.https://www.centos.org/docs/5/html/Installation_Guide-en-US/ch-kickstart2.html
- 2.https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Installation_Guide/ch-redhat-config-kickstart.html
- 3.<https://www.gartner.com/doc/3371817/hype-cycle-opensource-software>
4. <http://cobbler.github.io/manuals/quickstart/>
5. <https://fedoraproject.org/wiki/Pykickstart>

10. Appendix A

Glossary [Define terms, acronyms, and abbreviations used]

- Boot loader - A boot loader, also called a boot manager, is a small program that places the operating system (OS) of a computer into memory.
- Software Requirements Specification for <Kickstart Server Configuration Tool> Page 2
- PXE - The Pre-boot Execution Environment is an industry standard client/server interface that allows networked computers that are not yet loaded with an operating system to be configured and booted remotely by an administrator.
- FTP - The File Transfer Protocol (FTP) is a standard network protocol used to transfer computer files between a client and server on a computer network. FTP is built on a client-server model architecture and uses separate control and data connections between the client and the server.
- HTTP - The Hypertext Transfer Protocol (HTTP) is an application protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web. Hypertext is structured text that uses hyperlinks between nodes containing text.
- DHCP - Dynamic Host Configuration Protocol (DHCP) is a network protocol that enables a server to automatically assign an IP address to a computer from a defined range of numbers (i.e., a scope) configured for a given network.
- GRUB - GNU GRand Unified Bootloader or just GRUB is a boot loader package that supports multiple operating systems on a computer. During boot-up, the user can select the operating system to run.
- MBR - A master boot record (MBR) is a special type of boot sector at the very beginning of partitioned computer mass storage devices like fixed disks or removable drives intended for use with compatible systems and beyond.
- LDAP - Lightweight Directory Access Protocol, a set of protocols for accessing information directory. LDAP enables anyone to locate organizations, individuals, and other resources such as files and devices in a network, whether on the public Internet or on a corporate intranet.

Design and Implementation of Kickstart Server Configuration Tool Using Python Flask

- Kerberos - Kerberos is a computer network authentication protocol that works on the basis of 'tickets' to allow nodes communicating over a non-secure network to prove their identity to one another in a secure manner.
- Firewall - A firewall is a network security system designed to prevent unauthorized access to or from a private network. Firewalls can be implemented in both hardware and software, or a combination of both.
- Pre and Post Installation Scripts - Redhat Package Manager Supports a script run prior to installation is known as Pre-Installation script and script run after installation known as Post-Installation script.
- GTK+ - GIMP Toolkit is a cross-platform widget toolkit for creating graphical user interfaces. It is one of the most popular toolkits for the windowing systems.
- GLADE - GLADE is graphical user interface builder for GTK+, with additional components for GNOME.
- Rapid application development - Rapid application development is a software development methodology that uses minimal planning in favor of rapid prototyping. A prototype is a working model that is functionally equivalent to a component of the product.