

Relax and Recover

Linux Disaster Recovery as a Service (with rear)

Gratien D'haese
IT3 Consultants
Belgium



Goals of today

- Learn the importance of having a Linux Disaster Recovery Procedure
- Learn about the steps required for a bare metal restore
- Learn about automating disaster recovery with relax and recover (rear)
- Learn about Linux disaster recovery as a service with rear-server

Disasters do these really happen?







- Fire
- Flood
- Earthquake

Seriously now?







Some Basics

- What is Disaster Recovery?

The process by which a business function is restored to the normal, steady state after a disaster

- What is Business Continuity?

The way that a business function will operate after a disaster, until such time as the normal, steady state is restored



Linux Disaster Recovery

Like any other UNIX Operating System, Linux is vulnerable for disaster to strike

The question really is “What shall I do if a disaster strikes?”

Dependent on:

- Hardware failure (e.g. boot disk lost)
- Lost everything (fire, water, earthquake, theft)
- The answer: “Act immediately (with a disaster recovery plan)”



Why are backups not enough?

- Backups of data are necessary!
- Are not enough in case of losing the complete Operating System (OS)!
- Reinstalling the OS from scratch takes hours
- Restoring the backups a few more hours
- Fine-tuning of configurations takes days
- Even months later issues pop up!
- It is absolute necessary to foresee an inventory of hard- and software



Disaster Recovery Plan (DRP)

- DRP addresses need to recover from an emergency with minimum impact to the enterprise
- Protects enterprise from major services failure
- Minimizes risk to enterprise from delays in providing services
- Guarantees reliability of standby systems by testing and simulation
- Minimizes personnel decision-making required during disaster recovery



DRP: main steps

- Risk Analysis
- What is the budget?
- Develop the DRP according
  - Required time to normal operations
  - Establish priorities
  - Inventorying equipment and software
  - Make checklists and test procedures
- Test the DRP (at least on yearly basis)



KISS Principle

- The best way to prepare for a disaster is to avoid the disaster.
- Therefore, look for any potential problems you can find, and correct them.
  - Implement data mirrors or RAID systems
  - Take backups and test restores!
  - Use System Inventory software (e.g. cfg2html)
  - Select a Disaster Recovery Program which takes care of bare metal recovery



Commercial or Open Source?

- There is no standard solution delivered with Linux as such
- The choice is do we go for a commercial or Open Source solution?
- Try before you decide !!
  - It doesn’t always work as promised
  - Do we have test equipment? Don’t try it on production without a real DR test first!



Open Source Solutions

- DR optional with Open Source backup software
  - Completely dependent on backup solution
  - E.g. bacula
- Image makers (cloning)
  - Disk to image or partition to image
  - E.g. partimage or clonezilla



Open Source Solutions (cont.)

- True DR Open Source software
  - No focus on pure backups (incremental)
  - Main focus on fast DR
  - No fancy GUIs can be expected (nobody cares about a GUI if a disaster strikes)
  - To name a few of the most known:
    - mondorescue,
    - mkCDrec and
    - Relax and Recover (ReaR)



Disaster Recovery Media

- Extremely important to use **external storage**
- Boot-able media: CD/DVD, USB, LAN, tape ...
- Mix of media is allowed for boot and backup
  - Boot-able CD/DVD, USB disk for data
  - LAN boot (via PXE) with backup data via NFS, CIFS
  - Boot-able tapes (OBDR – CD boot emulation)
- Separate boot media and backup data
  - Boot from USB key, CD/DVD image or LAN
  - Recover system with backup software (tar, rsync, ...)



Disaster Recovery in Practice

- Gather system information
- Store the disk layout
  - Partitioning, LVM and RAID configuration
  - File systems, file system labels ...
  - Boot loader (GRUB, LILO, ELILO)
- Make a system backup (OS and user data)
- Create boot-able rescue media with system configuration (and optional with backup data)
- All steps are done “online”



Disaster Recovery: rescue media

- Create “rescue linux” from running system
- Optimally compatible “tool box”
- Clone the system environment
  - Linux kernel and modules
  - Device driver configuration
  - Network configuration
  - Basic system software and tools
- Operate entirely in RAM (initrd)



Recovery Process in detail

- Boot system from rescue media
- Restore disk layout
  - Create partitions, RAID configuration and LVM
  - Create file systems (mkfs, mkswap)
  - Configure file systems (labels, mount points)
- Restore the backup data
- Restore the boot loader
- Reboot



Relax and Recover (rear)

- Proven solution at large enterprise customers
- Rear established as standard solution for Linux disaster recovery in data centers
- Shipping with Fedora and openSUSE
- Integrates with many “commercial” backup software solutions, e.g. TSM, DP, NBU, …
- Integrates with OS backup software solutions as well, e.g. GNU tar, rsync, bacula, ...
- Scales well with large amounts of servers



A bit of history

- Rear is a spin-off of 2 existing projects:
  - OpenVPN Gateway Builder (OGB) of *Schlomo Schapiro (Germany)*
  - Make CD-ROM Recovery (mkCDrec) of *Gratien D'haese (Belgium)*
- First released in July 2006
  - Thanks to modular concept the first release was written in 3 weeks time



Rear Features

- Focus on disaster recovery and not backup
- Tight integration with common backup software
- Simple full backup integrated
- Complements backup software
  - Backup software: data storage and retrieval
  - Rear: recover the system layout and make it work
  - Rear: use the backup software to restore data
- Methodology: use the best tool for the job



Rear Backup Software

- Supported solutions today include:
  - CommVault Galaxy
  - IBM Tivoli Storage Manager (TSM)
  - Symantec NetBackup (NBU)
  - HP DataProtector (DP)
  - Rsync and other 'external' methods
  - GNU tar archives on NAS shares (CIFS, NFS)
- Very transparent integration
- Other backup solutions can be added (bacula is in test phase)



Rear Network Integration

- Disaster recovery as part of network infrastructure
  - Backup software: file level backup storage using LAN or SAN
  - Rear: takes care of the system environment
  - Boot rescue media via PXE or virtual CD image
    - No physical media required
  - Very scalable: automated installation of entire disaster recovery data center
    - Rear distribution via company branded RPM
    - Use scheduler to automate the creation of rescue media



Rear Development

- Truly Open Source model – SVN, sourceforge
- Development model based on
  - Sponsoring
  - Patches, modules from other OS developers
- Modular Concept
  - Framework with many small Bash scripts
  - Code re-usability with library functions
- Documentation on-line (web site) and in SVN



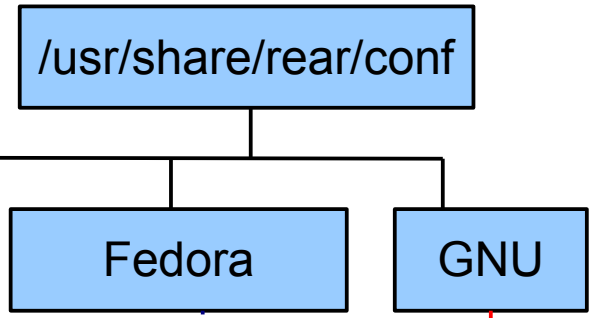
Architecture of rear

rear dump:

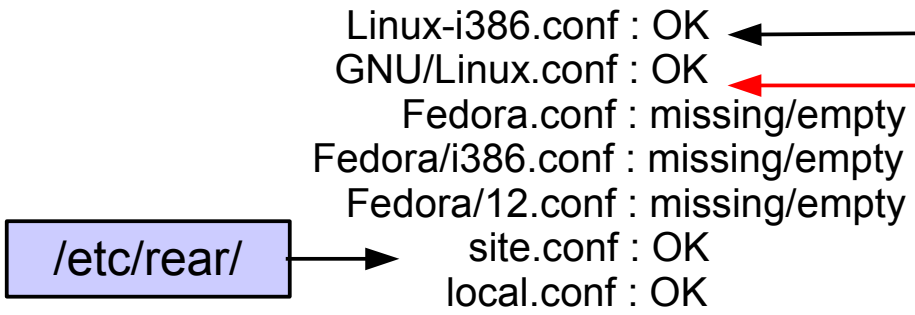
Dumping out configuration and system information

System definition:

ARCH = Linux-i386
OS = GNU/Linux
OS_VENDOR = Fedora
OS_VENDOR_ARCH = Fedora/i386
OS_VENDOR_VERSION = Fedora/12



Configuration tree:





Usage of rear

- Shell scripts are stored under /usr/share/rear
- Scripts are kept together according work-flows
  - mkrescue (only make rescue image)
  - mkbackup (including make rescue image)
  - mkbackuponly (excluding make rescue image)
  - recover (the actual recovery part)
- Easy to incorporate new scripts, e.g. for information gathering of Hard- and Software, or other goodies



Rear Configuration files

- SITE wide configuration file
  - /etc/rear/site.conf
  - Add here bash like settings which apply to all systems in your site (distribute from a local point)
- LOCAL (this system) configuration file
  - /etc/rear/local.conf
    OUTPUT=ISO
    BACKUP=NETFS
    #BACKUP=TSM|NBU|DP|GALAXY|EXTERNAL|...
    NETFS_URL=nfs://server.domain/backup-path
    MODULES_LOAD=(vmxnet)



Rear command options

```
rear [Options] <command> [command options ...]
Relax & Recover Version 1.7.24 / 2009-12-09
Build: e20b259e7bc9ce602b67cd5fe4397fc7
Copyright (C) 2006-2009
     Schlomo Schapiro
     Gratien D'haese, IT3 Consultants
Relax & Recover comes with ABSOLUTELY NO WARRANTY; for details
see the GNU General Public License at http://www.gnu.org/licenses/gpl.html

Available Options:
-V             version information
-d             debug mode
-D             debugscript mode
-S             Step-by-step mode
-s             Simulation mode (shows the scripts included)
-q             Quiet mode
-r a.b.c-xx-yy      kernel version to use (current: 2.6.31.12-174.2.22.fc12.i686.PAE)

List of commands:
dump                    Dump configuration and system information
help                    print out usage
mkbackup                Create rescue media and backup system.
mkbackuponly            Backup system without creating a (new) rescue media.
mkdeb                   Create DEB packages with this rear version
mkdist                  Create distribution tar archive with this rear version
mkrescue                Create rescue media only
mkrpm                   Create RPM packages with this rear version
mktar                   Create tar archive with this rear installation
mkvendorrpm             Create vendor RPM with this rear version
recover                 Recover the system
validate                Submit validation information
```



Work-flow backup (or rescue)

- mkbackup – mkrescue
  - Preparation (building the root file system layout)
  - Analyze (disaster recovery environment creation)
    - Creation of /var/lib/rear/recovery structure
  - Analyze (building the rescue system)
  - Build (copy all executables that are needed)
  - Pack (kernel and initial ram-disk)
  - Backup (optional)
  - Output (copy to destination, PXE, ISO,...)
  - Cleanup



rear -s mkbackup

```
Relax & Recover Version 1.7.24 / 2009-12-09
Simulation mode activated, ReaR base directory:
/usr/share/rear
Source conf/Linux-i386.conf
Source conf/GNU/Linux.conf
Source prep/default/01_progress_start.sh
Source prep/GNU/Linux/28_include_vmware_tools.sh
Source prep/ISO/default/30_check_iso_dir.sh
Source prep/ISO/default/32_check_cdrom_size.sh
Source prep/ISO/GNU/Linux/32_verify_mkisofs.sh
Source prep/ISO/Linux-i386/33_find_isolinux.sh
Source prep/default/99_progress_stop.sh
Source dr/default/01_mk_config_dir_recovery.sh
Source dr/default/09_only_include_vg.sh
Source dr/GNU/Linux/10_describe_physical_devices.sh
Source dr/GNU/Linux/11_describe_mountpoint_device.sh
Source dr/GNU/Linux/12_describe_filesystems.sh
Source dr/GNU/Linux/13_describe_swap.sh
Source dr/GNU/Linux/15_copy_proc_partitions.sh
Source dr/GNU/Linux/21_describe_md.sh
Source dr/GNU/Linux/23_describe_lvm2.sh
Source dr/GNU/Linux/29_find_required_devices.sh
Source dr/Linux-i386/31_describe_device_properties.sh
Source dr/GNU/Linux/80_copy_fstab_file.sh
Source dr/GNU/Linux/95_cfg2html.sh
Source dr/GNU/Linux/95_collect_hpacucli.sh
Source dr/GNU/Linux/96_collect_MC_serviceguard_infos.sh
Source rescue/default/00_remove_workflow_conf.sh
Source rescue/default/01_merge_skeletons.sh
Source rescue/default/10_hostname.sh
Source rescue/default/20_etc_issue.sh
Source rescue/GNU/Linux/30_dns.sh
Source rescue/GNU/Linux/31_network_devices.sh
Source rescue/GNU/Linux/35_routing.sh
Source rescue/GNU/Linux/39_check_usb_modules.sh
Source rescue/GNU/Linux/40_kernel_modules.sh
Source rescue/default/43_prepare_timesync.sh
Source rescue/GNU/Linux/50_clone_keyboard_mappings.sh
Source rescue/default/50_ssh.sh
Source rescue/default/90_clone_users_and_groups.sh
Source build/GNU/Linux/00_create_symlinks.sh
Source build/GNU/Linux/10_copy_as_is.sh
Source build/GNU/Linux/11_touch_empty_files.sh
Source build/GNU/Linux/13_create_dotfiles.sh
Source build/GNU/Linux/15_adjust_permissions.sh
Source build/GNU/Linux/39_copy_binaries_libraries.sh
Source build/GNU/Linux/40_copy_modules.sh
Source build/default/50_patch_sshd_config.sh
Source build/default/99_update_os_conf.sh
Source pack/Linux-i386/30_copy_kernel.sh
Source pack/GNU/Linux/90_create_initramfs.sh
Source output/ISO/Linux-i386/30_create_isolinux.sh
Source output/ISO/Linux-i386/80_create_isofs.sh
Source output/default/95_email_result_files.sh
Source cleanup/default/01_progress_start.sh
Source cleanup/default/99_progress_stop.sh
Finished in 4 seconds.
```



Work-flow recovery

- The same configuration files are used during the recovery work-flow
- Recovery Process:
  - Verify (integrity and sanity check)
  - Recreate (file system layout)
  - Restore (the backups including Operating System)
  - Finalize (install boot loader, dump recovery log into /tmp of the recovered system)



rear -s recover

```
Relax & Recover Version 1.7.24 / 2009-12-09
Simulation mode activated, ReaR base directory: /usr/share/rear
Source conf/Linux-i386.conf
Source conf/GNU/Linux.conf
Source setup/GNU/Linux/80_setup_hp_raid.sh
Source verify/GNU/Linux/05_sane_recovery_check.sh
Source verify/GNU/Linux/10_describe_physical_devices.sh
Source verify/Linux-i386/11_describe_device_properties.sh
Source verify/GNU/Linux/12_compare_physical_devices.sh
Source recreate/GNU/Linux/09_disable_lvm2_md.sh
Source recreate/Linux-i386/10_initialize_physical_devices.sh
Source recreate/GNU/Linux/21_create_md_devices.sh
Source recreate/GNU/Linux/22_create_lvm2_devices.sh
Source recreate/GNU/Linux/31_create_filesystems.sh
Source recreate/GNU/Linux/70_mount_filesystems.sh
Source recreate/default/98_show_disk_free.sh
Source restore/REQUESTRESTORE/default/20_prompt_user_to_start_restore.sh
Source restore/default/90_create_missing_directories.sh
Source finalize/default/01_prepare_checks.sh
Source finalize/default/10_populate_dev.sh
Source finalize/GNU/Linux/70_create_swapfiles.sh
Source finalize/default/88_check_for_mount_by_id.sh
Source finalize/default/89_finish_checks.sh
Source finalize/default/90_remount_sync.sh
Source finalize/default/98_good_bye.sh
Source finalize/default/99_copy_logfile.sh
Finished in 4 seconds.
```

Cfg2html: hard- and software details

- When **cfg2html** is installed and in local.conf "SKIP_CFG2HTML=" has been set

```
# rear mkrescue
Relax & Recover Version 1.7.24 / 2009-12-09
The preparation phase OK
Physical devices that will be recovered: /dev/sda
Collecting general system information
(cfg2html) OK
Creating root FS layout OK
Copy files and directories OK
Copy program files & libraries OK
Copy kernel modules OK
Create initramfs OK
Making ISO image OK
Wrote ISO Image /tmp/ReaR.iso (17M)
The cleanup phase OK
Finished in 488 seconds.

# ls /var/lib/rear/recovery/cfg2html/
localhost.localdomain.err
localhost.localdomain.partitions.save
localhost.localdomain.txt
localhost.localdomain.html
localhost.localdomain.tar
```





Log file /tmp/rear.log

```
2010-03-12 13:09:07 Using 'blkid' for vol_id
2010-03-12 13:09:07 Relax & Recover Version 1.7.24 / 2009-12-09
2010-03-12 13:09:07 Combining configuration files
2010-03-12 13:09:07 Skipping /etc/rear/os.conf (file not found or empty)
2010-03-12 13:09:07 Skipping /etc/rear/mkrescue.conf (file not found or empty)
2010-03-12 13:09:08 Including conf/Linux-i386.conf
2010-03-12 13:09:08 Including conf/GNU/Linux.conf
2010-03-12 13:09:08 Skipping /usr/share/rear/conf/Fedora.conf (file not found or empty)
2010-03-12 13:09:08 Skipping /usr/share/rear/conf/Fedora/i386.conf (file not found or empty)
2010-03-12 13:09:08 Skipping /usr/share/rear/conf/Fedora/12.conf (file not found or empty)
2010-03-12 13:09:08 Skipping /usr/share/rear/conf/Fedora/12/i386.conf (file not found or empty)
2010-03-12 13:09:08 Including /etc/rear/site.conf
2010-03-12 13:09:08 Including /etc/rear/local.conf
2010-03-12 13:09:08 Creating build area '/tmp/rear.10018'
2010-03-12 13:09:08 Running mkrescue workflow
2010-03-12 13:09:08 Running 'prep' stage
2010-03-12 13:09:08 Including prep/default/01_progress_start.sh
2010-03-12 13:09:08 Including prep/GNU/Linux/28_include_vmware_tools.sh
2010-03-12 13:09:08 Including prep/ISO/default/30_check_iso_dir.sh
2010-03-12 13:09:08 Including prep/ISO/default/32_check_cdrom_size.sh
2010-03-12 13:09:08 ISO Directory '/tmp' [/dev/mapper/VolGroup-lv_root] has 3087 MB free space
2010-03-12 13:09:08 Including prep/ISO/GNU/Linux/32_verify_mkisofs.sh
2010-03-12 13:09:08 Using '/usr/bin/mkisofs' to create ISO images
2010-03-12 13:09:08 Including prep/ISO/Linux-i386/33_find_isolinux.sh
2010-03-12 13:09:18 Including prep/default/99_progress_stop.sh
2010-03-12 13:09:18 Finished running 'prep' stage in 10 seconds
…
Done with: Ending Padblock                          Block(s)    150
Max brk space used 0
8427 extents written (16 MB)
2010-03-12 13:10:35 Including output/default/95_email_result_files.sh
2010-03-12 13:10:35 Finished running 'output' stage in 1 seconds
2010-03-12 13:10:35 Running 'cleanup' stage
2010-03-12 13:10:35 Including cleanup/default/01_progress_start.sh
2010-03-12 13:10:35 Including cleanup/default/99_progress_stop.sh
2010-03-12 13:10:35 Finished running 'cleanup' stage in 0 seconds
2010-03-12 13:10:35 Finished running mkrescue workflow
2010-03-12 13:10:35 Removing build area /tmp/rear.10018
2010-03-12 13:10:35 End of program reached
```



ReaR Status

- Stable software
  - i386 and x86_64 are well tested
  - ia64 and ppc less tested
- Released as RPM, TAR, DEB
- Rear ships with
  - SUSE Linux Enterprise HA extension 11 SP1
  - OpenSUSE 11.2 and Fedora 11, 12, 13 and 14
- Support available (community and/or commercial)
- Open for patch submissions by rear community



Current ReaR Development

- Go beyond just disaster recovery (v1.9.x)
  - System cloning
  - System migration (P2V, V2P, V2V, P2P)
  - Enhance the system toolkit for offline & rescue tasks



What is missing?

- Most customers miss a central component for ReaR that
  - Gathers information about rear
  - Stores rear boot images
  - Initiates Disaster Recovery
  - Makes rear information available for 3rd party

  - Linux Disaster Recovery as a Service
    - rear-server



rear-server version 1 functions

- Information gathering only
- Information pushed to server from rear as part of “rear mkrescue” boot image creation
- Optionally collect also boot images
- Display information about rear-protected systems
- Group information by hosts/domains and tags
- Scalability for thousands of systems
- Simple Web-GUI



implementation requirements

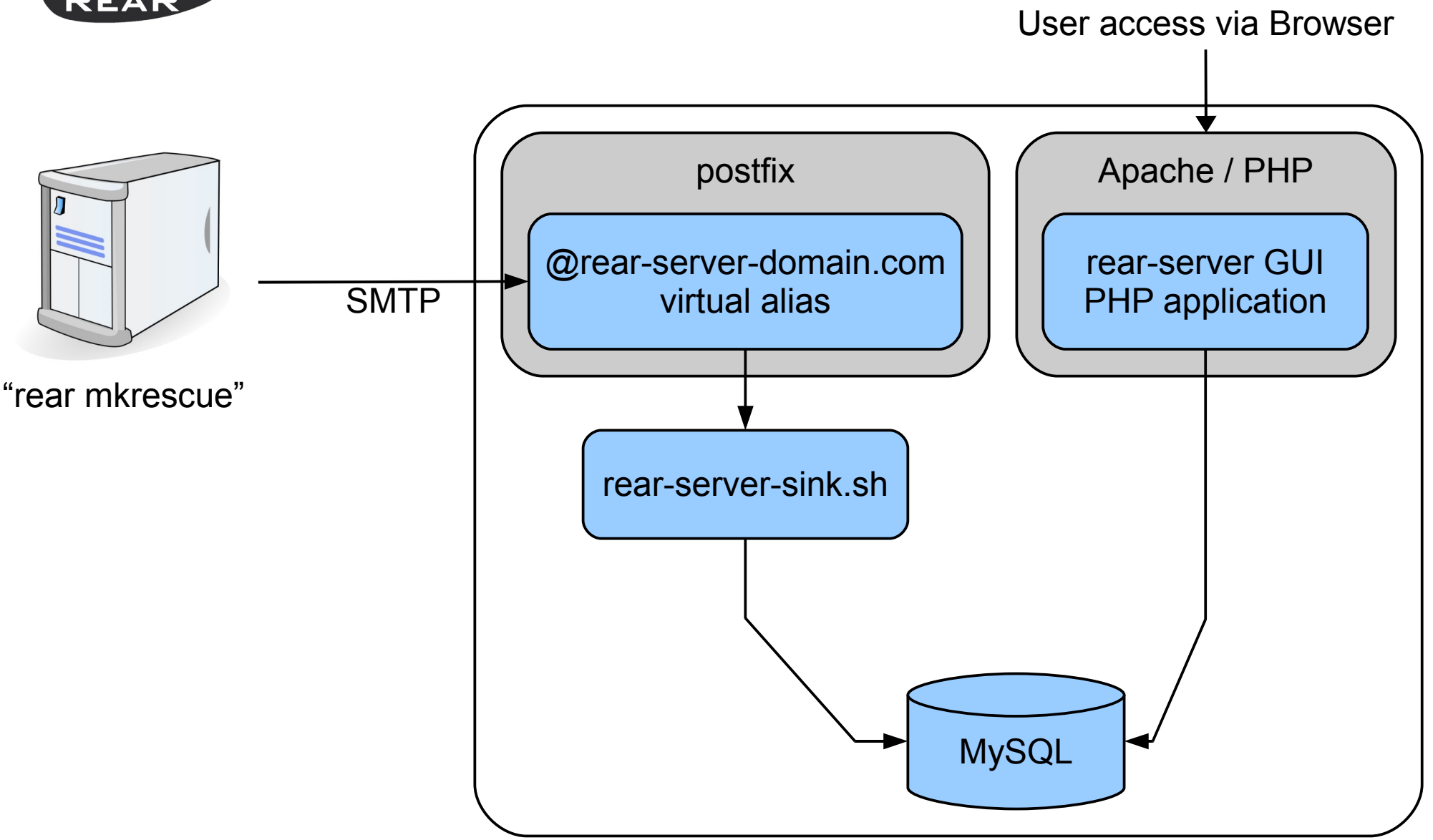
- Use existing network / routing / firewall infrastructure and standard protocols
- No direct connection between rear and rear-server required



rear changes required

- Additional configuration variables
  - REAR_SERVER
  - REAR_SERVER_SEND_RESULT
  - REAR_SERVER_TAGS
- Scriptlets to send the required information to the rear server (similar to existing email support)

rear-server Architecture





Design Considerations

- SMTP can be routed, is indirect
- Most corporate setups allow sending emails to internal system even for DMZ systems
- Based on standard components (postfix, Apache, PHP, MySQL)
- No “rear daemon” required, use existing daemons of Apache and postfix
- “Free” queuing for incoming requests by postfix
- SMTP can be easily secured



rear-server Web GUI

- Authentication handled by Apache
  - Support LDAP, AD, Kerberos etc.
  - For version 1 (information gathering) all access has the same security level (in any case only read-only)
- Multiple views:
  - Overview
  - Last 20 (with auto refresh like “tail -f”)
  - Grouped/Filtered by client domain
  - Grouped/Filtered by tags
  - Search



Delivery

- GPL Software
- Central configuration directory:
  - /etc/rear-server
- As RPM that installs rear-server and configures
  - postfix
  - LAMP
- Optionally vendor-based configuration tool
  - yast rear-server
  - system-config-rear



Roadmap rear-server

- 1.0
  - Basic functionality as described in this proposal
  - Depending on implementation issues without some advanced parts
- 1.2
  - Implement missing parts, optimize design
  - Implement features from customer installations
  - Optionally integrate with monitoring solutions (e.g. Nagios)
- 2.0
  - Trigger Disaster Recovery from Web GUI
  - Depends on integration with backup software
  - Integrate with system management environments



Contacts

Web-site: http://rear.sourceforge.net/

Gratien D'haese
ReaR Maintainer

gratien.dhaese@it3.be

Or

Schlomo Schapiro
ReaR Maintainer

schlomo@schapiro.org