@schlomo @schlomo
@schlomoschapiro @schlomo@floss.social

# Relax and Recover (ReaR):

1. Automated Linux Recovery
2. The Open Source Project

Schlomo Schapiro, Associate Partner / Principal Engineer, Tektit Consulting
18.06.2024, stackconf 2024, Berlin

## Problem To Solve

Fully & End-2-End Automated

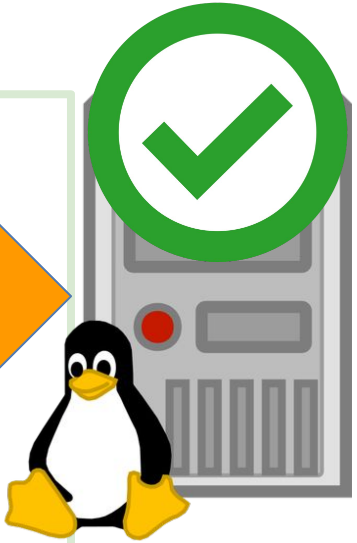
**Disaster Recovery / Bare Metal Restore**

Zero Knowledge, Hands Off,

Repeatable, Scaleable,

Universal, It-Just-Works,

Testable & Provable Quality

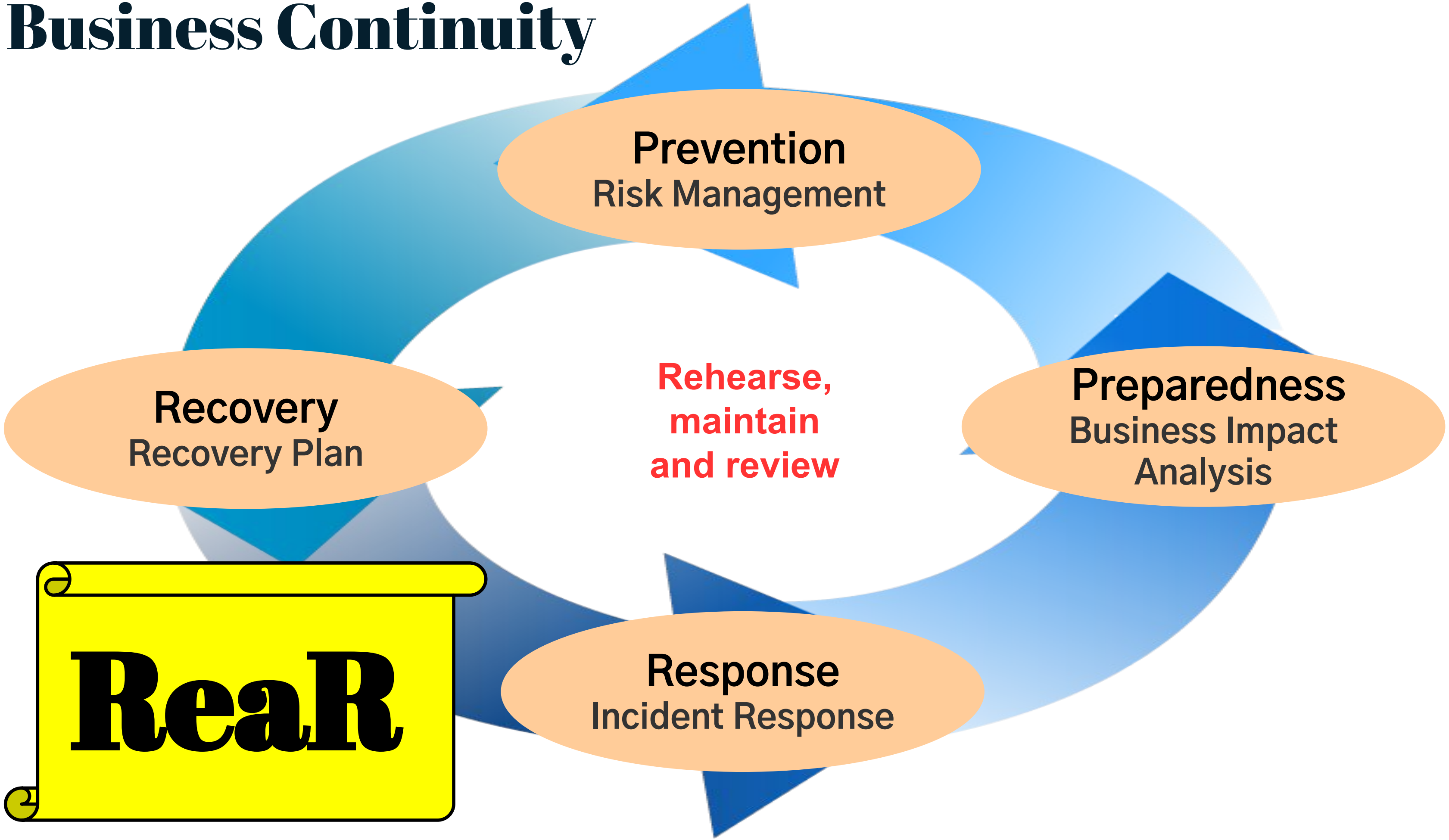


# Automated Linux Recovery

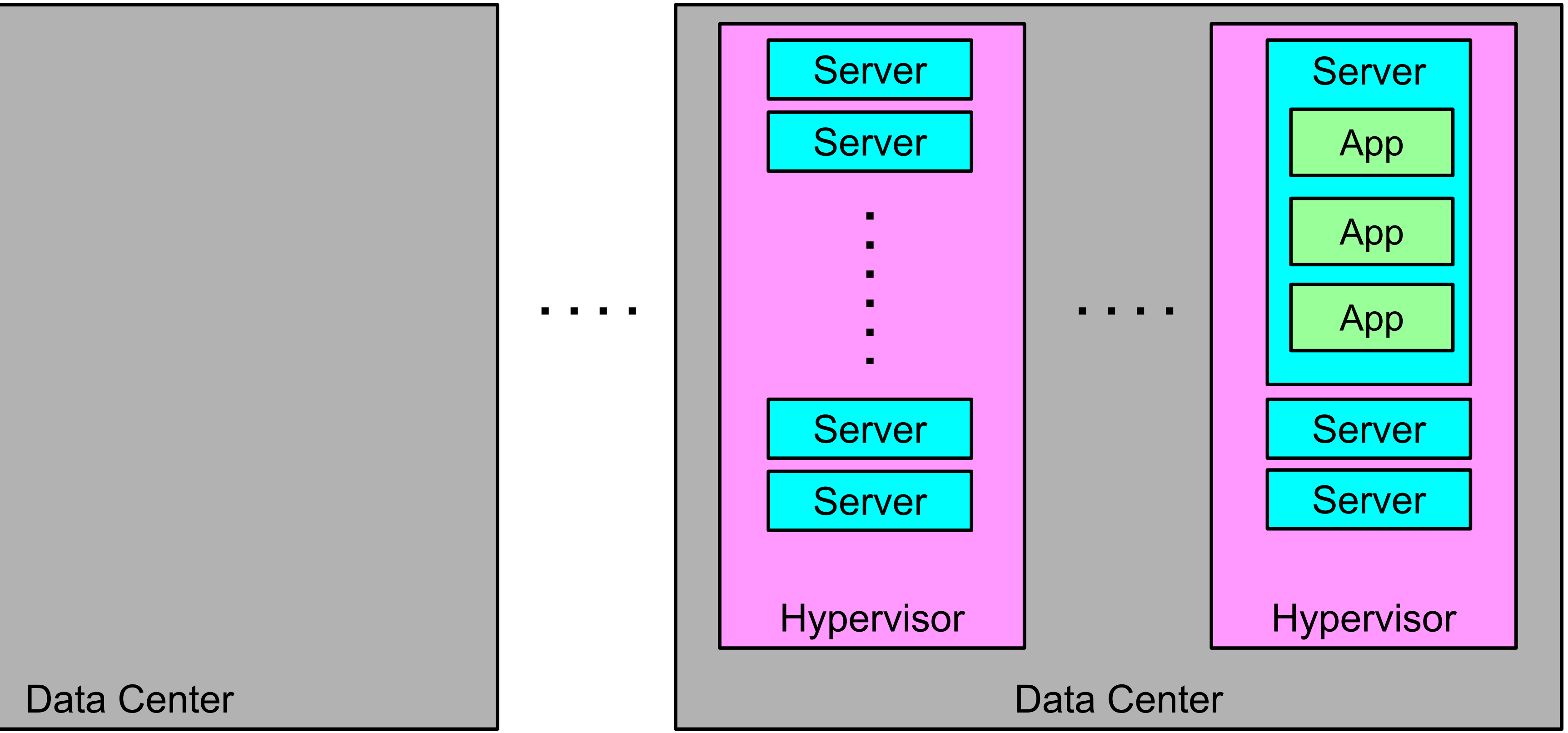
## Ask Yourself: Mean Time to Restore Service

- After deploying a bad software update or configuration?
  → On 1000 servers?
- After upgrading the Operating System to a faulty version?
  → On 50 servers? On 500 servers?
- After deleting the hard disk / SAN LUN of your main database?
  → After deleting 20 LUNs?
- After deleting the hard disk / SAN LUN of a hypervisor?
  → All the LUNs of a virtualization cluster?
- After flooding or burning the data center?
  → Getting a new data center?

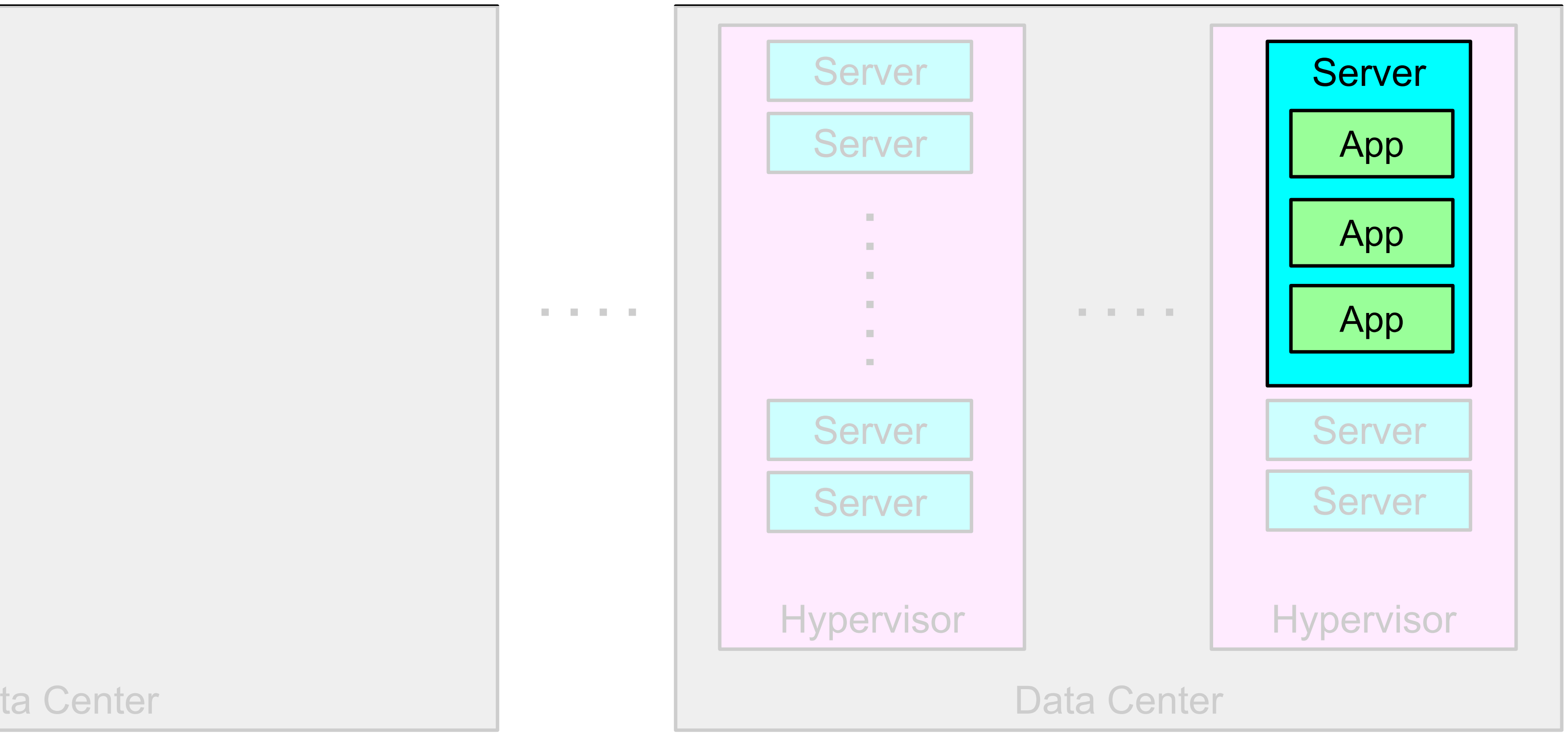
## Business Continuity



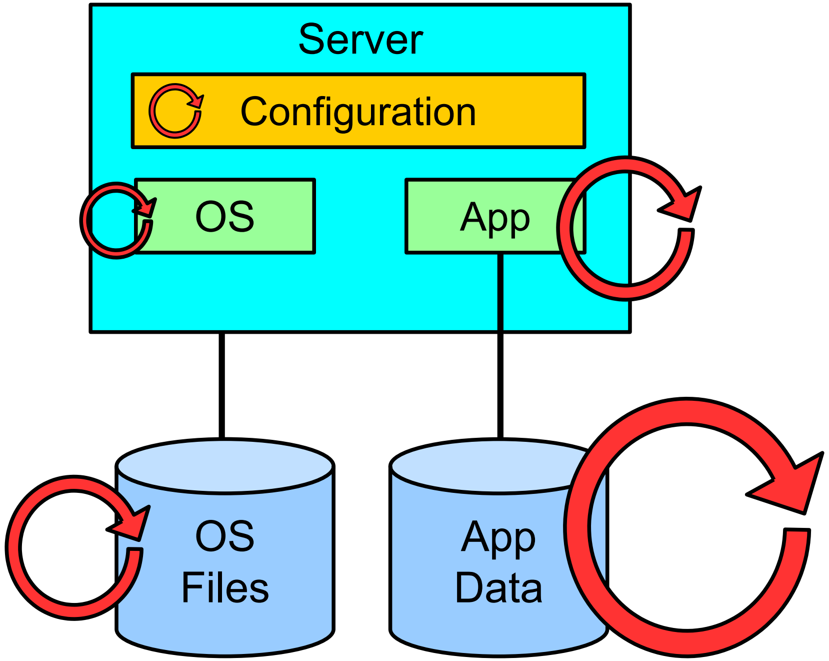
## Scope

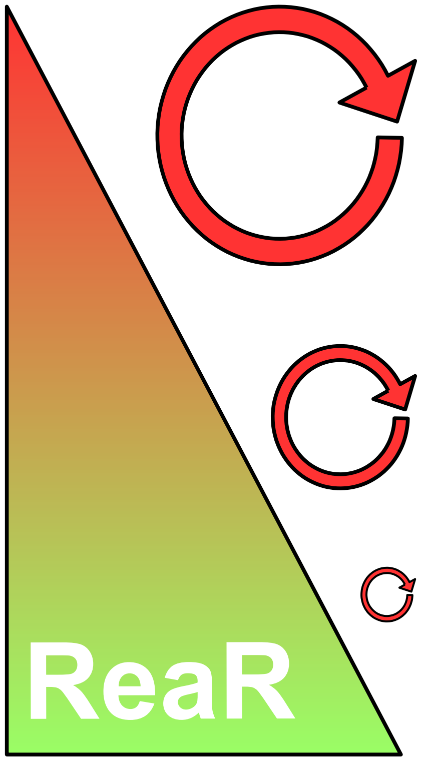

Server
Server
Server
Server
Server
App
App
App
Server
Server
Hypervisor
Hypervisor
Data Center
Data Center

# Scope of Relax and Recover



# Understanding Change Frequencies



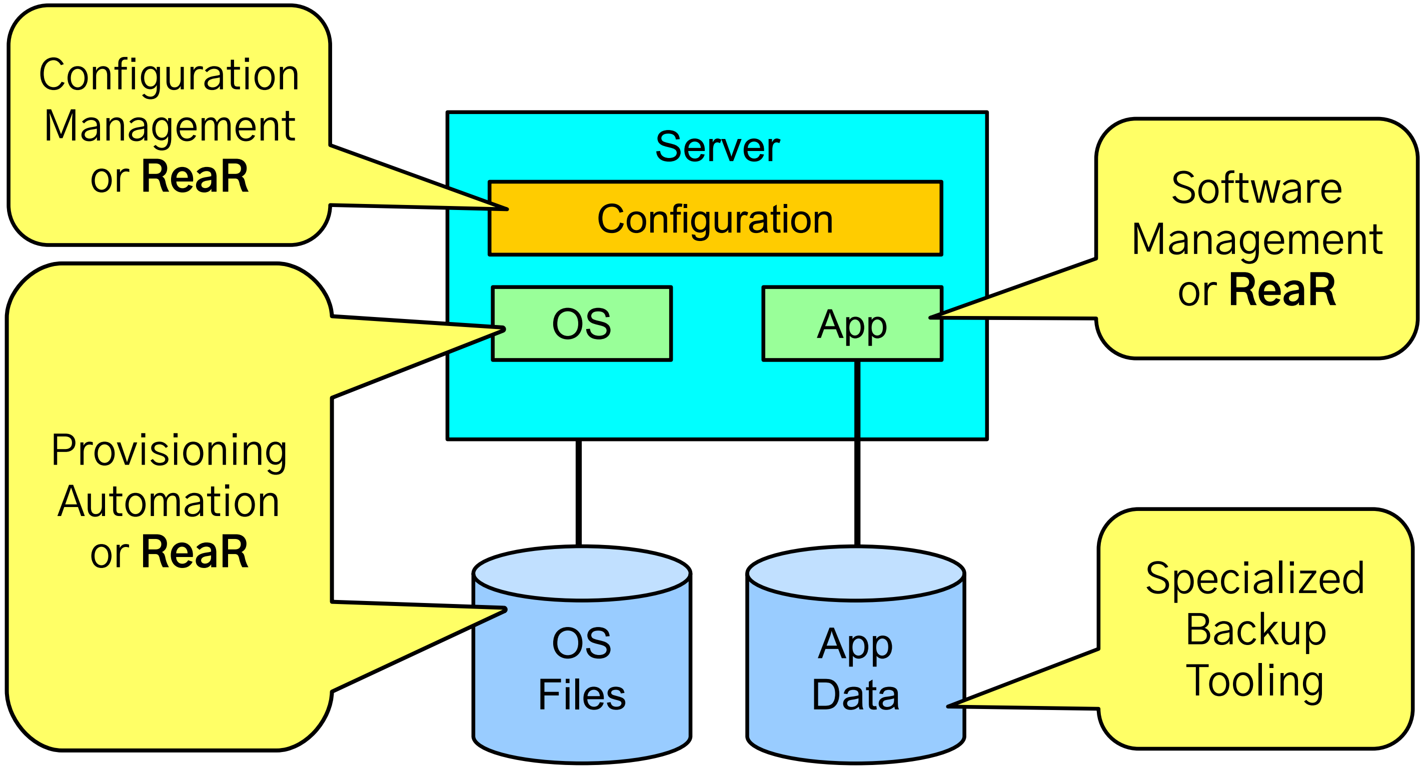


Constantly changing
Very high cost per time lost

Regularly changing

Seldomly changing

## Use the Best Tool for the Recovery / Restore Job


Configuration Management or **ReaR**
Provisioning Automation or **ReaR**
Server
Configuration
OS
App
OS Files
App Data
Software Management or **ReaR**
Specialized Backup Tooling

$$\underbrace{\text{Manual} \rightarrow \text{Automated} \qquad \text{Backup} \rightarrow \text{Restore} \rightarrow \text{Recovery}}_{\text{Relax and Recover}}$$

## Why are Backups not Enough?

- “We have a backup of everything” – really?
- Backups of data are necessary – but it is only a starting point!
- Are not enough in case of losing the complete Operating System (OS)!
- Reinstalling the OS from scratch takes hours
- Restoring the backups takes more hours or days
- Fine–tuning of configurations takes days
- Even months later issues pop up!

→ Complete system backups provide much better protection!

## 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 human decision–making required during disaster recovery

→ Automation creates a DRP that can scale out/up to cover everything!

## Relax and Recover as Linux DR solution

- Rear is a tool that implements a **DR workflow** for Linux
- Basically meaning:
  - Modular framework written in Bash
  - Easy to extend to own needs
  - Easy to deploy (set up and forget)
  - Integration for all common Linux technologies
  - Integration with most common back-up solutions
  - Attempts to make system recovery **as easy as possible**
- ReaR runs **"live"** during regular operation, no downtime required to create a DR rescue image

## Introducing Relax and Recover (ReaR)

- 100% Open Source
- Goal:

  *Fully automate everything related to Linux disaster recovery and bare metal restore*

## Basic ReaR Design Concepts

1. Every file on a Linux computer must be stored in a **Backup**
2. ReaR creates a bootable rescue media as **Output**
3. Rescue media boots on recovery computer
4. ReaR automates system recreation (disk layout, fileystems, networking …)
5. ReaR **facilitates** the **restore** of **all files from backup**
6. ReaR makes computer **bootable**
7. Reboot
8. Done ✅
9. 🎉 🥳 🍾 🎈 ❤️❤️❤️

## ReaR specializes on Linux System Recovery

### PREPARATION

**Backup Software**

Backup of all files
Incremental / differential backup
Quiescing and consistency of backups
Backup of databases
…

+

**ReaR**

Hardware configuration
Disk layout
Filesystems
Networking
Boot loader configuration
…

RESCUE

### RECOVERY

**ReaR**

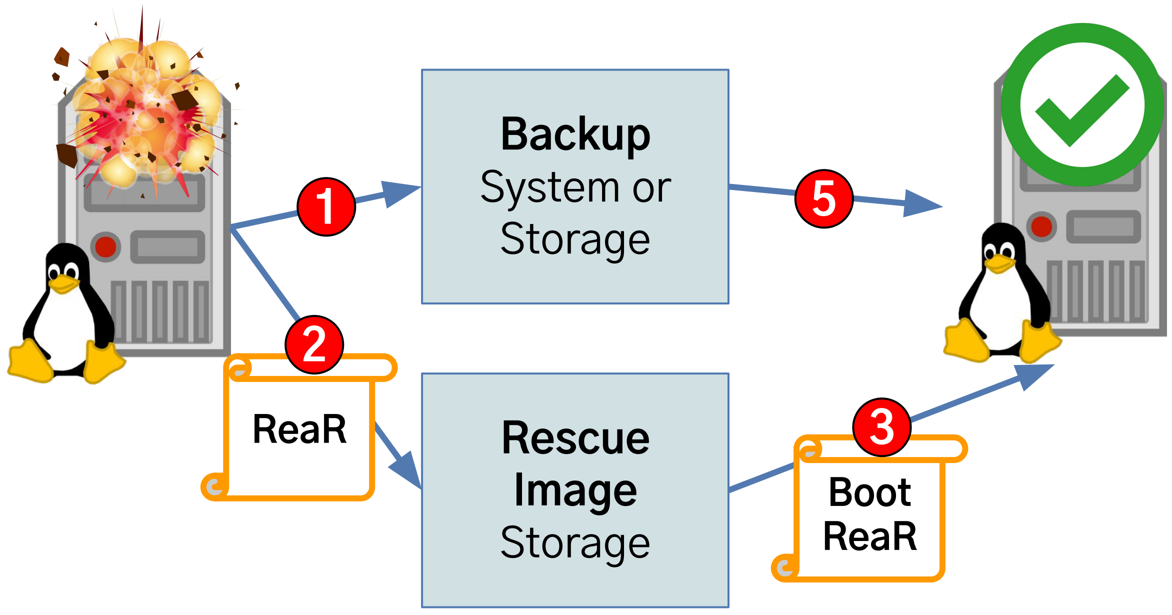
Boot rescue system
Configure Hardware
Recreate disk layout
Recreate filesystems
Configure networking
Configure drivers
Install boot loader
Adjust system for new hardware
…

+

**Backup Software**

Restore all files
Point-in-time restore of older backups

# Disaster Recovery with ReaR



4

**ReaR Rescue System:**

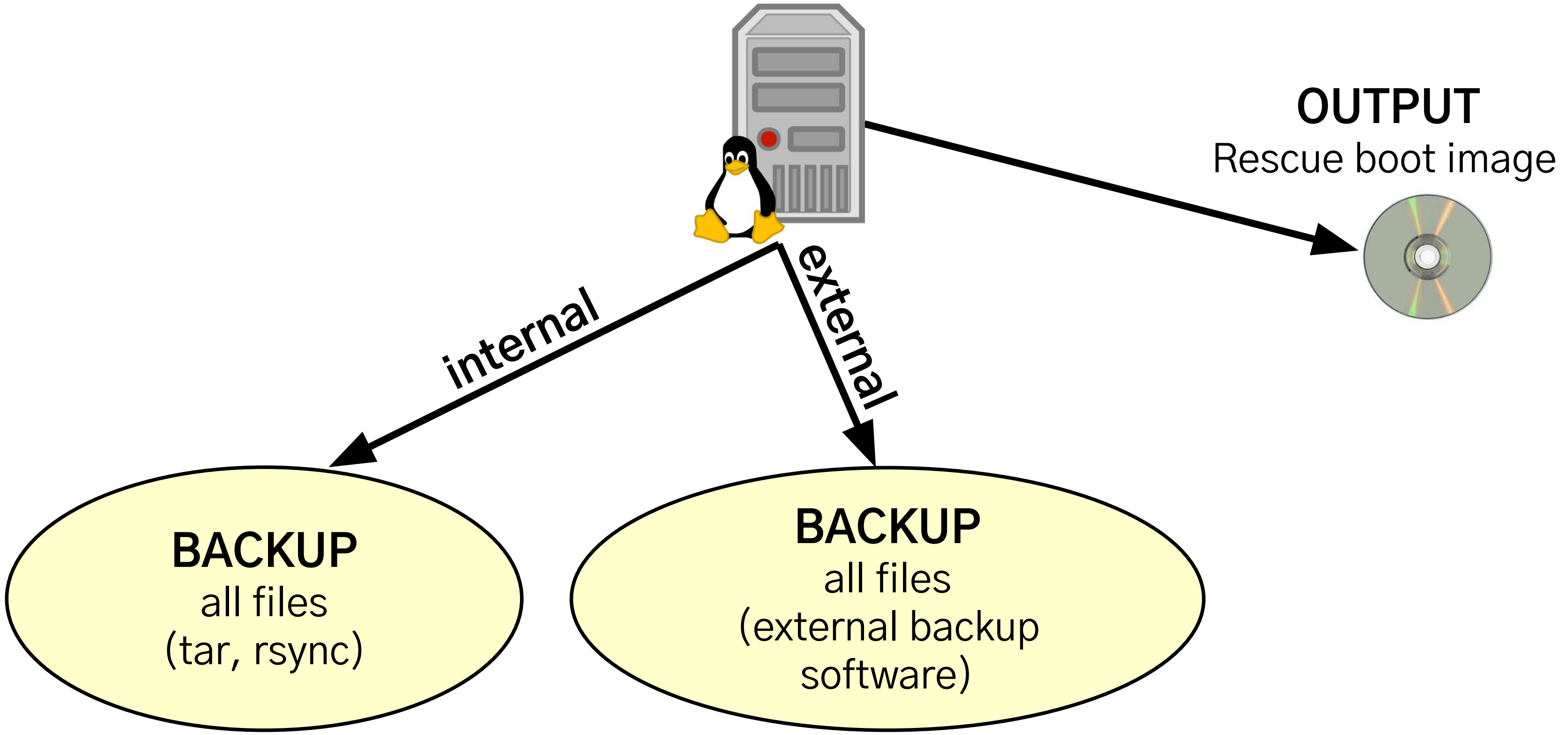
Same name/IP/login as original system 😁

1. Configure disks
2. Format filesystems
3. Mount filesystems
4. Restore files 5
5. Install boot loader
6. Adjust system configuration
7. Reboot
8. Done

6

TEKTIT CONSULTING

## DR Flow – BACKUP and OUTPUT



## Decide on DR strategy ⅓

- Which backup mechanism to use (BACKUP=)?
  - Internal backup: GNU tar, rsync
  - External backup: Bareos, Borg, a commercial backup solution

## Decide on DR strategy ⅔

- Which backup mechanism to use (BACKUP=)?
  - Internal backup:   GNU tar, rsync
  - External backup: Bareos, Borg, a commercial backup solution
- Where will the backups reside (BACKUP_URL=)?
  - NFS share, CIFS share, external USB disk, tape, local spare disk, cloud storage, DVD
  - Remote network and/or storage location

## Decide on DR strategy

- Which backup mechanism to use (BACKUP=)?
  - Internal backup: GNU tar, rsync
  - External backup: Bareos, Borg, a commercial backup solution
- Where will the backups reside (BACKUP_URL=)?
  - NFS share, CIFS share, external USB disk, tape, local spare disk, cloud storage, DVD
  - Remote network and/or storage location
- How shall we boot the rescue image (OUTPUT=)?
  - Via DVD (ISO image), tape (OBDR), network (PXE), USB disk
- How/where to collect the rescue images for all systems (OUTPUT_URL=)?
  - NFS or CIFS share, copy to remote location (FTP, SFTP, HTTPS ...)

## Disaster Recovery - Media

- Most important: External storage!
- Bootable media: CD/DVD/ISO image, USB storage, PXE, tape …
- Combined boot and backup media:
  - Bootable CD/DVD or USB storage with backup data included
  - LAN boot (PXE) with backup data via CIFS, NFS …
  - Bootable tapes – HP OBDR (CD emulation)
- Separation between boot media and backup data
  - Boot the system from a (small) USB storage, CD/DVD or LAN
  - Recover the system with backup software, tar, rsync …

# Usage of rear

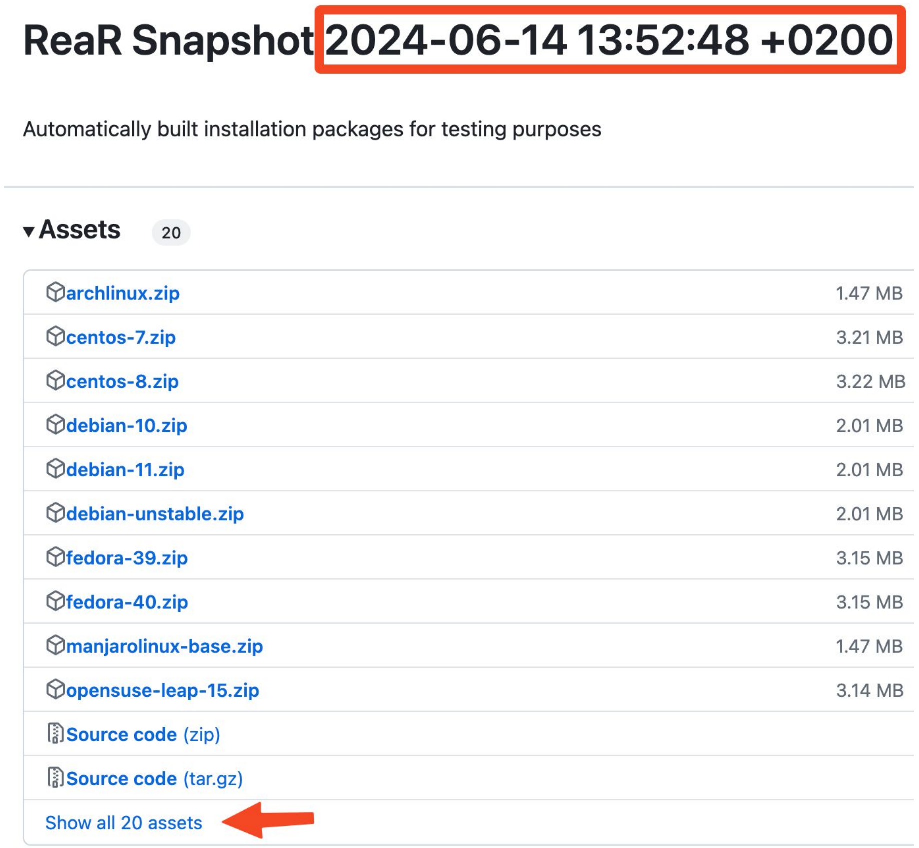
- Usage: `rear <Options> <Workflow>`
- Options:
  `-v, -d, -D --expose-secrets, --non-interactive ...`
- Workflows:
  - **`mkrescue`** (make rescue image)
  - **`mkbackup`** (create backup and rescue image)
  - **`recover`** (the actual recovery part, run from ReaR rescue system)
  - **`dump`** (show configuration)
  - more for special use cases:
    checklayout, udev, `format, mkopalpba, opaladmin, shell, validate` ...

## Installing ReaR

1. From your distro (old):
    a. `apt install rear`
    b. `dnf install rear`
    c. `zypper install rear`
    d. …

## Installing ReaR

1. From your distro (old):
   a. `apt install rear`
   b. `dnf install rear`
   c. `zypper install rear`
   d. …
2. From upstream (current development):
   a. [github.com/rear/rear/releases/](https://github.com/rear/rear/releases/)
   b. built automatically
   c. OS packages



**ReaR Snapshot 2024-06-14 13:52:48 +0200**

Automatically built installation packages for testing purposes

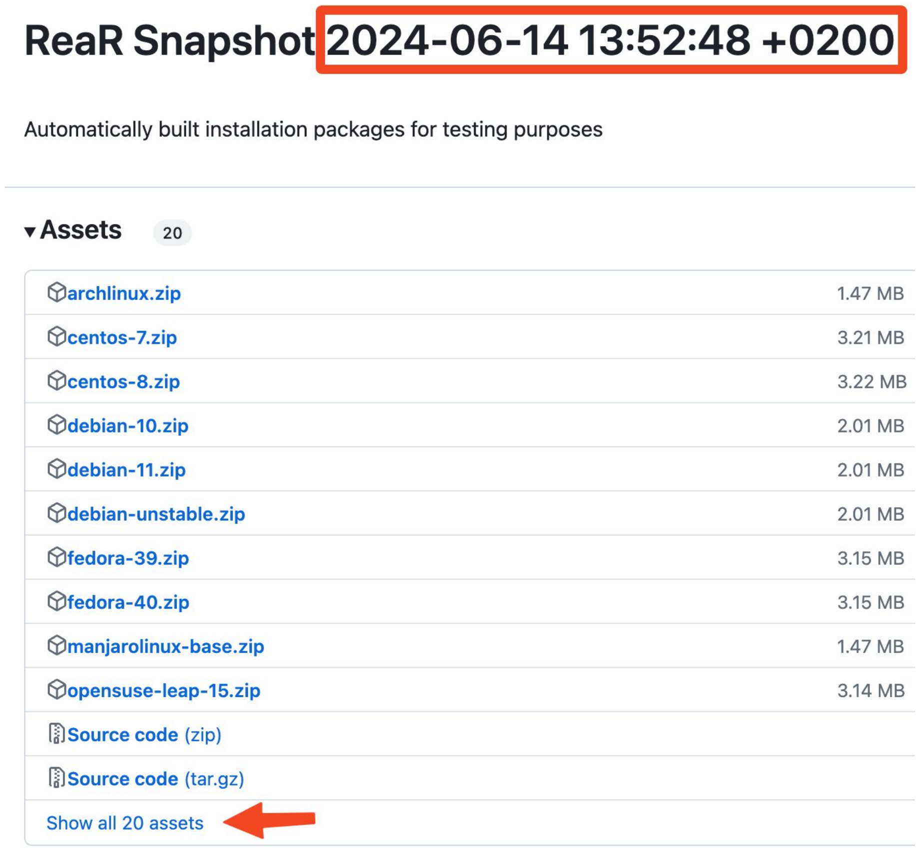
▾ **Assets** 20

| | |
|---|---|
| archlinux.zip | 1.47 MB |
| centos-7.zip | 3.21 MB |
| centos-8.zip | 3.22 MB |
| debian-10.zip | 2.01 MB |
| debian-11.zip | 2.01 MB |
| debian-unstable.zip | 2.01 MB |
| fedora-39.zip | 3.15 MB |
| fedora-40.zip | 3.15 MB |
| manjarolinux-base.zip | 1.47 MB |
| opensuse-leap-15.zip | 3.14 MB |
| Source code (zip) | |
| Source code (tar.gz) | |

Show all 20 assets

# Installing ReaR

1. From your distro (old):
   a. `apt install rear`
   b. `dnf install rear`
   c. `zypper install rear`
   d. …
2. From upstream (current development):
   a. [github.com/rear/rear/releases/](github.com/rear/rear/releases/)
   b. built automatically
   c. OS packages
3. From source:
   a. [github.com/rear/rear](github.com/rear/rear)
   b. `make install`
4. *Use from source (portable):* `./usr/src/rear`



ReaR Snapshot 2024-06-14 13:52:48 +0200

Automatically built installation packages for testing purposes

▾Assets 20

| | |
|---|---|
| archlinux.zip | 1.47 MB |
| centos-7.zip | 3.21 MB |
| centos-8.zip | 3.22 MB |
| debian-10.zip | 2.01 MB |
| debian-11.zip | 2.01 MB |
| debian-unstable.zip | 2.01 MB |
| fedora-39.zip | 3.15 MB |
| fedora-40.zip | 3.15 MB |
| manjarolinux-base.zip | 1.47 MB |
| opensuse-leap-15.zip | 3.14 MB |
| Source code (zip) | |
| Source code (tar.gz) | |
| Show all 20 assets | |

TEKTIT CONSULTING

## Configuring ReaR

- `/etc/rear/local.conf` → use for manual configuration
- `/etc/rear/site.conf` → use for configuration management
- `/usr/share/rear/conf/default.conf` → defaults & explanations
- We use Bash arrays, e.g.
  ```
  PROGS+=( mc )
  COPY_AS_IS+=( /{etc,usr/lib,usr/share}/mc )
  ```
- Configuration files are Bash scripts – use your fantasy
- Supply additional configuration files via `-C` option

## Four Core Configuration Variables

1. BACKUP variable defines the “backup” method:
   NETFS, RSYNC, DUPLICITY, PPDM, VEEAM, GALAXY11 ...
2. BACKUP_URL variable defines the location where to store the backup archive (for internal backup methods only)
3. OUTPUT variable defines the “output” method:
   ISO, PXE, OBDR, USB, RAWDISK
4. OUTPUT_URL variable defines the location where to store the rescue image (ISO image, pxe configuration, extlinux configuration)

Check `default.conf` for details of all other configuration variables

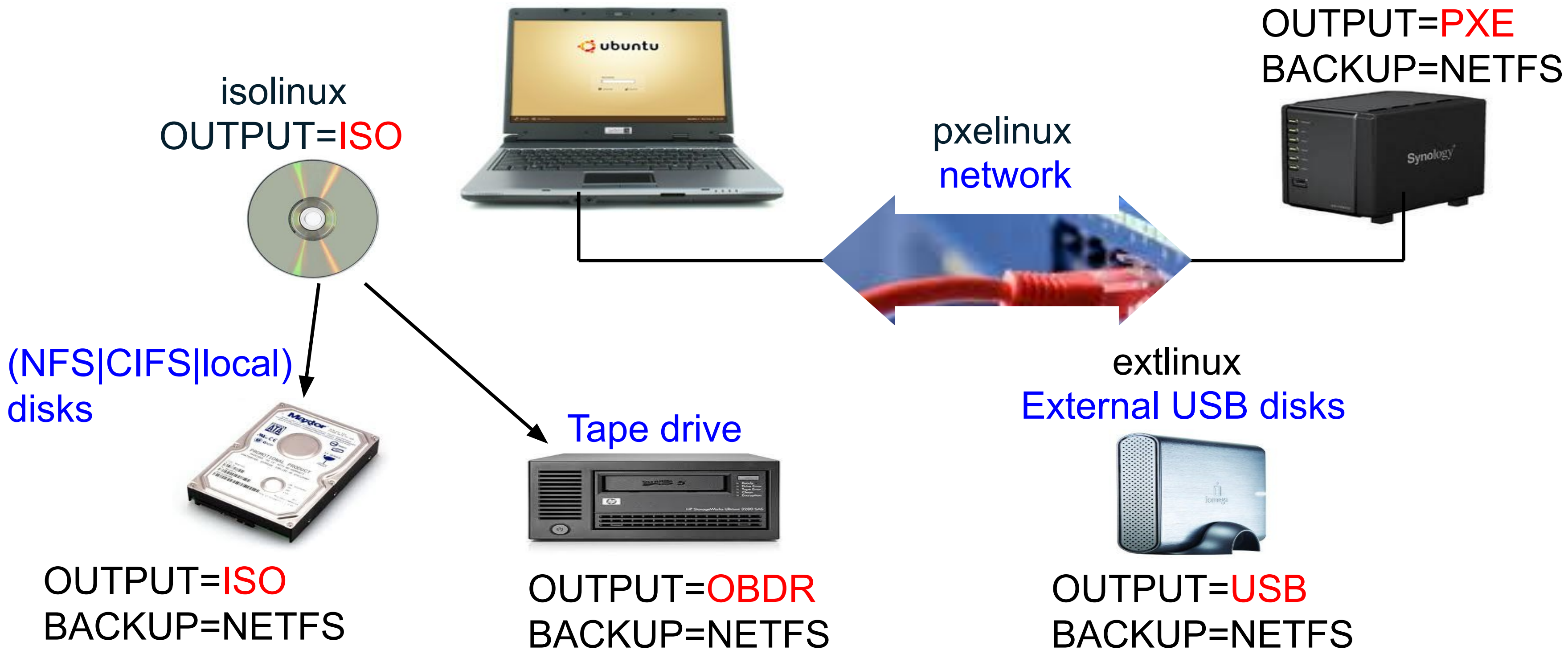
## ReaR Configuration Example - Production

```
OUTPUT_URL=ftp://server.domain/dr
BACKUP=TSM

SECURE_BOOT_BOOTLOADER=(
   /boot/efi/EFI/*/shimx64.efi
)
```

# ReaR Configuration Example - Development

```
OUTPUT_URL=nfs://server.domain/srv/scratch
OUTPUT=ISO

BACKUP=NETFS
BACKUP_URL=nfs://server.domain/srv/scratch

FIRMWARE_FILES=('no')
MODULES=('loaded_modules')
EXCLUDE_MD5SUM_VERIFICATION=all

AUTOSHRINK_DISK_SIZE_LIMIT_PERCENTAGE=80

SECURE_BOOT_BOOTLOADER=(/boot/efi/EFI/*/shimx64.efi)
```

## Internal Backup via BACKUP=NETFS



## BACKUP_URL sets backup locations

`BACKUP=NETFS`

- File type: `BACKUP_URL=file:///directory/`
- NFS type: `BACKUP_URL=nfs://nfs-server/directory/`
- CIFS type: `BACKUP_URL=cifs://samba/directory/`
- USB type: `BACKUP_URL=usb:///dev/disk/by-label/REAR-000`
- ISO type: `BACKUP_URL=iso://backup`
- Tape type: `BACKUP_URL=tape:///dev/nst0`

# Disaster Recovery for your Laptop (or Server)

- Configuration:

```
BACKUP=NETFS
BACKUP_URL=usb:///dev/disk/by-label/REAR-000
USB_DEVICE_FILESYSTEM_LABEL=REAR-000
OUTPUT=USB
UDEV_WORKFLOW=mkbackup
SECURE_BOOT_BOOTLOADER=(/boot/efi/EFI/*/shimx64.efi)
```

- Create bootable recovery media on USB storage:

```
rear format /dev/sdb
```

- Automatically trigger backup via udev rule:

```
ACTION=="add", SUBSYSTEM=="block", ENV{ID_FS_LABEL}=="REAR-000", RUN+="/usr/sbin/rear udev"
```

- Insert USB storage and wait till system beeps 📣 to signal “done”

## Advanced ReaR

- `COPY_AS_IS, PROGS`
  add custom files and programs
- `OUTPUT=PORTABLE`
  portable mode rescue image (~5 MB)
- `/etc/rear/mappings`
  pre-seed answers for changed hardware, e.g. network cards or host bus adapters, to facilitate an automated P2V or V2V mass migration
- `POST_RECOVERY_SCRIPT, PRE_RECOVERY_COMMANDS ...`
  add your own hooks to run custom code before or after recovery or backup
- `USE_DHCLIENT=yes`
  force DHCP on rescue system instead of static IP from original system
- `TIMESYNC=`
  activate time sync in rescue system (useful for hardware)
- `SSH_ROOT_PASSWORD=, TTY_ROOT_PASSWORD`
  activate password login for SSH or for local login in rescue system

## ReaR Internals

- **Bash** framework using Bash v4 features
- Main program: `/usr/sbin/rear`
- Configuration: `/usr/share/rear/conf/` and `/etc/rear/`
- Functions: `/usr/share/rear/lib/*-functions.sh`
- Workflows: `/usr/share/rear/lib/*-workflow.sh`
- Stages: `/usr/share/rear/*`
- Multi-dimensional script merging by `ARCH, OS, OS_VERSION, BACKUP, OUTPUT, BACKUP/OUTPUT` and more.
  Use `rear -s` (simulate) to see how it works.
- `tools/run-in-docker.sh` – run ReaR code in Docker on multiple distros

## Open Source Project

TEKTIT CONSULTING

## Open Source Project: Relax-and-Recover (ReaR)

### Linux Bare Metal Restore

### Linux Disaster Recovery

- 🌐 relax-and-recover.org
- 7 ReaR maintainers:
  Schlomo Schapiro, Gratien D’haese, Johannes Meixner, Vladimir Gozora, Sébastien Chabrolles, Renaud Métrich, Pavel Cahyna
- 193 other contributors

# ReaR Achievements Overview

**De–facto standard tool within Linux community**

- Easy to use and adjust for sysadmins (Bash)
- Shipping with all Linux distributions
- 100% Open Source code (GPL v3)
- Create bootable rescue media:
  ISO, USB, eSATA, PXE, OBDR
- Has plenty of backup software integrations:
  tar, rsync, Micro Focus Data Protector, FDR/Upstream, CommVault Galaxy, Symantec NetBackup, IBM Tivoli Storage Manager, EMC NetWorker (Legato), Dell PowerProtect Data Manager, Veeam, EMC Avamar, SEP Sesam, NovaStor DC, Rubrik Cloud Data Management, Rsync Backup Made Easy, Bareos, Bacula, Duplicity, BorgBackup

**Business Value for ReaR Users:**

- Pay 25 k€ **once** for ReaR **Open Source** integration instead of 100 k€ / **year** for **commercial** BMR tool
- Supports most commercial backup tools
- Complete end–2–end automation
- Easy configuration management

**Ecosystem Integration  based on ReaR:**

- System Recovery on Red Hat Enterprise Linux
- Disaster Recovery on SUSE LINUX Enterprise
- DRLM – Disaster Recovery Linux Manager
- SEP SESAM and Bareos use ReaR for BMR

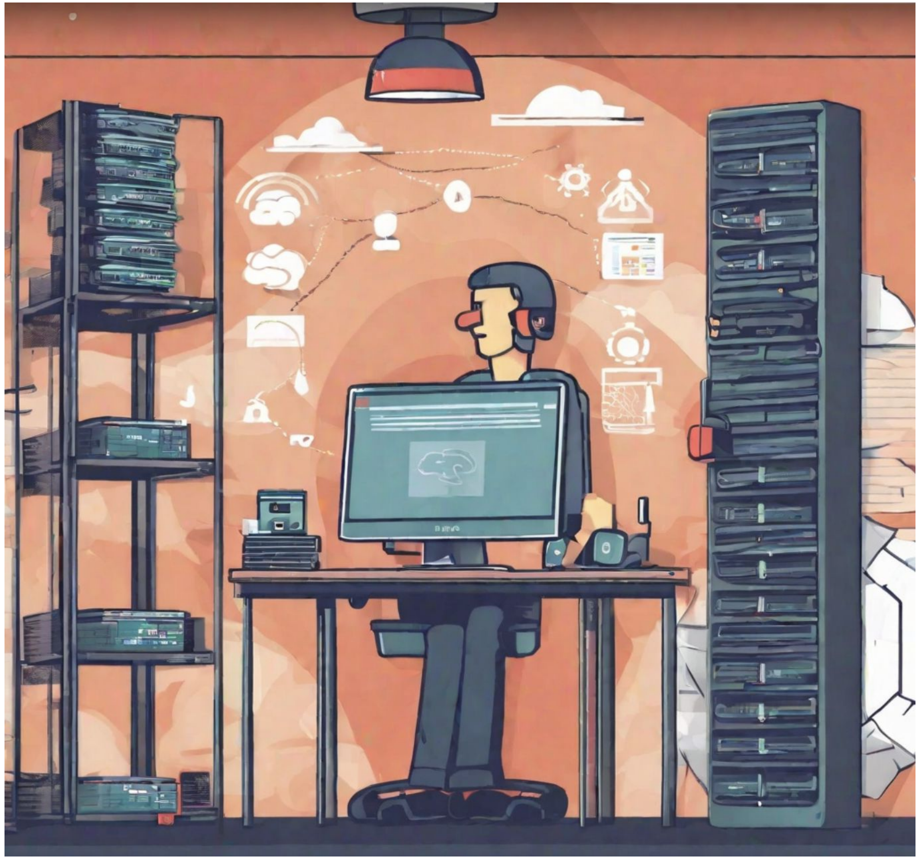
# ReaR History

## Make CD-ROM Recovery (mkcdrec)

- Started 2000 – ended 2012
- Predecessor of ReaR

## Relax-and-Recover (ReaR)

- Started 2006
- In 2010 Suse added it to their HA portfolio
- In 2015 RedHat added it in their main repository
- ... continuous development ...
- 2026 – 20 years ReaR 🎉🎉🎉🎉



# What goes actually well?

User Support (via GitHub issues)

End-users submitting issues **and smaller contributions** on GitHub

Interaction with major Linux vendors (SUSE, RedHat)

Code cleanup and bug fixes

Mature code base

Deprecating ancient legacy, e.g. Bash 3 for new releases

## What doesn't go well?

Documentation in general

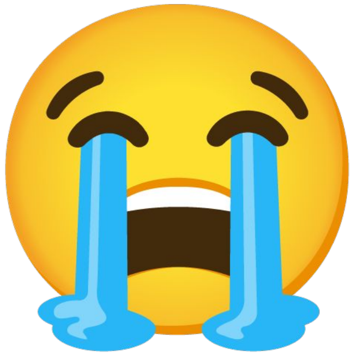
ReaR User Guide

Not enough capacity needed to add new features

Not enough capacity of developers to assist end-user support

Cross-architecture support & testing:
i386, x86_64, ia64, ppc64, s390, (arm)

Automated testing of new releases of ReaR
on different distros and platforms



## Relax-and-Recover (ReaR) Automated Testing

gdha.github.io/rear-automated-testing/ or
github.com/gdha/rear-automated-testing

- Sponsored project – currently *suspended* due lack of sponsoring

github.com/rear/rear-integration-tests

- RedHat (Pavel) tries to bring some live into
- Low activity (and limited to RedHat's testing platform)

- SUSE does manual testing

## Issues we Maintainers are Facing

Can’t test everything, especially backup software integration

Keeping contributors active and engaged

Lack of sponsoring to add new features

Lack of ReaR service contracts to pay for regular maintenance

**Release management**

## ReaR Release History

2024: ??? hopefully 3.0

2023: —

2022: 2.7

2021: —

2020: 2.6

2019: 2.5

2018: 2.4

2017: 2.3, 2.2, 2.1, 2.0

2016: 1.19, 1.18, 1.17.2

2015: 1.17.1

## What we are trying to do?

- Point to Snapshots as “rolling release”
- Getting the ”ReaR User Guide” on rails
- Find sponsoring
- Merge with similar projects?
    - Talks started with DRLM project
- More workshops?
- More commercial activities?

# Automated Linux Disaster Recovery with ReaR

Relax, and Recover!

**No Backup? No Mercy!**

🌐 relax-and-recover.org

# Q&A — How may I help you?

schlomo.schapiro.org

*We are not consultants. We are Partners, Coaches, Humans, Enablers, Catalysts, Sparring Partners, Experts … and sometimes a little annoying.*

I focus on IT strategy, IT governance, technology and architecture management, security and compliance automation, related organisational changes, business continuity, open source and cloud technologies – and I'm available as a Principal Engineer or Technical Product Owner for short-term / interim support.

Examples:

- **Business-IT alignment & leveraging**, developing required skills and abilities for 21st century IT, leverage AI
- **SaaS compliance & governance**, data possession vs. ownership, IAM, integrations, backup & DR, shadow IT
- **Compliance Automation**, finding the "golden path" to a "golden state"
- **Secrets Management** for Datacenter, Cloud Infrastructure, IaaS/PaaS/SaaS
- **Open Source**, from usage to contribution, writing policies, using SBOM, establishing Open Source Stewardship
- **Good Engineering Practices**, GitOps, test driven development, good architecture decisions, known tech strategy
- **Business Continuity and Disaster Recovery** for office, Cloud infrastructure, data center & SaaS, with quality assurance, emergency communication & collaboration, hot & cold standby, no-restore solution, ransomware protection, Linux Disaster Recovery / Bare Metal Restore with "Relax and Recover (rear)" Open Source tooling

schlomo.schapiro@tektitconsulting.com

TEKTIT CONSULTING