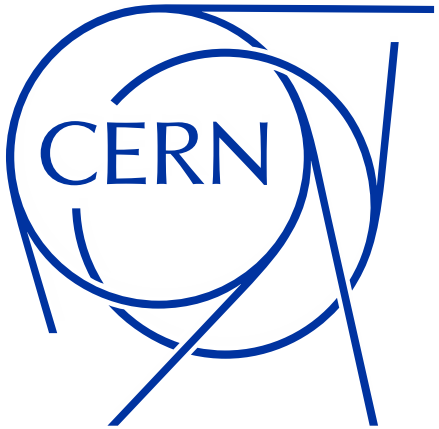
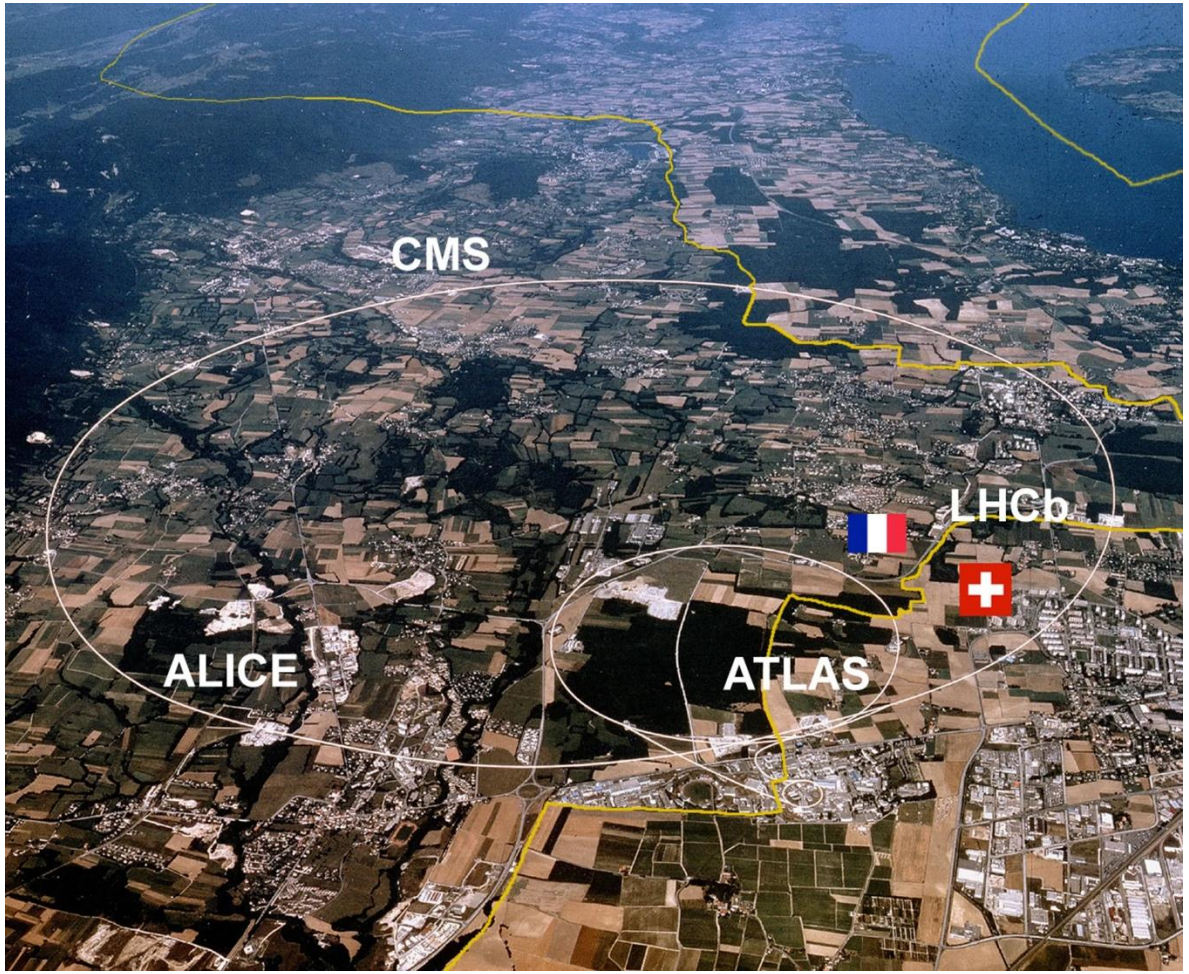


Keeping the LHC colliding: Providing Extended Lifecycle support for EL7

Ben Morrice – on behalf of the CERN IT Linux team

03.04.2025

What is the LHC?



“The largest machine in the world...”

“Comprised of 26 659 meters of underground tunnel, straddling Geneva and bordering France”

“Accelerating particles around this circumference 11245 times per second at 99.9999991% the speed of light”

Sounds… complicated.

How do you control (bend) a beam of particles traveling at near the speed of light?

What is the LHC? (continued)

Answer (extremely simplified): **magnets** (around 9000 of them)

Superconducting magnets steer particles in the LHC operate at **-271.3°C**, colder than outer space.

A significant cooling system uses over **10,000** tons of liquid nitrogen before switching to liquid helium for the final cooling stages.

The distribution system for the liquid helium stretches over **40,000** meters throughout the LHC.

This level of complexity requires thousands of embedded systems that control physical hardware which directly interacts with the operation of the beam:

Introducing the **FEC (Front End Computer)**

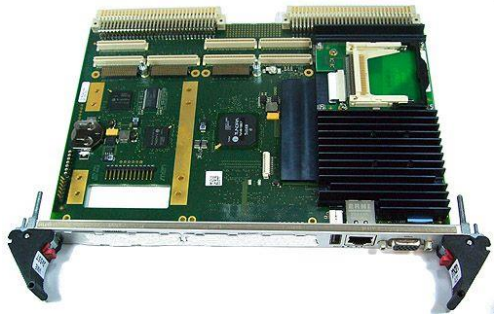
What is a Front End Computer (FEC)?

- ~2100 industrial grade computer gateways, scattered around the LHC
- Many are underground and in difficult to access places
- Configured with network boot to facilitate operations
- Large array of diverse hardware types are interfaced through a FEC
- Some hardware is off the shelf, other silicon is built at CERN
- Mostly PCI interface, some PCIe, VMEbus, USB, etc

FEC hardware

Several manufacturers providing industrial grade systems

- MEN A20
- K762
- IPC647E



One small problem:

For their operating system, they are currently utilizing **CERN CentOS 7** (Enterprise Linux 7) which went into end-of-life **30.06.2024** …

FEC OS upgrade

So … Easy – Just upgrade the OS to CERN IT's recommendation?

AlmaLinux 9 or **Red Hat Enterprise Linux 9** ?

Not so fast …

FEC breakdown versus x86_64 microcode support

2155 FECs in total

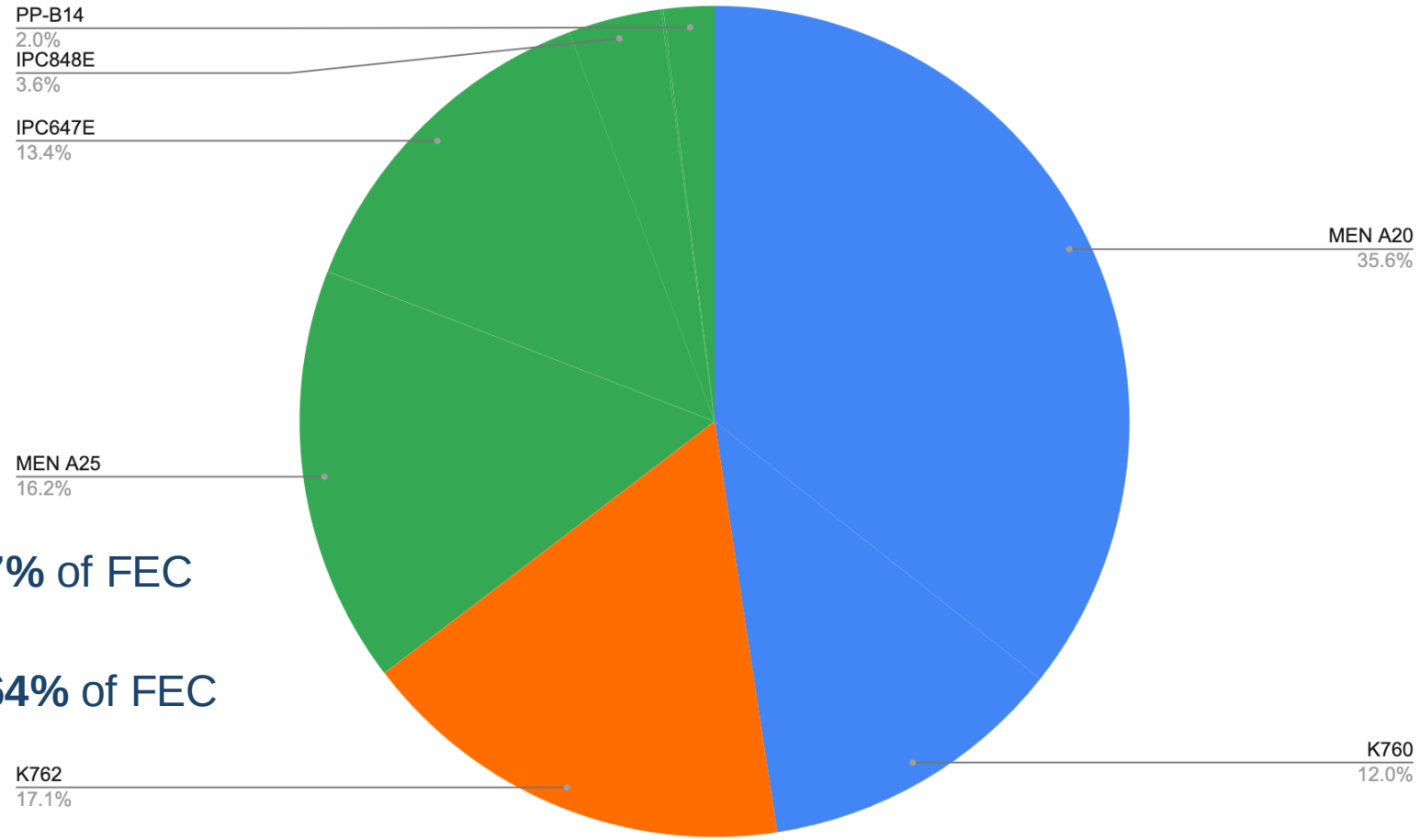
CPU maximum support x86_64-v1

CPU maximum support x86_64-v2

CPU maximum support x86_64-v3

RHEL9 requires x86_64-v2, meaning **47%** of FEC systems cannot run it

RHEL10 requires x86_64-v3, meaning **64%** of FEC systems cannot run it



OS replacement, or is there another way…?

- Cost prohibitive to replace FEC hardware to x86_64-v3 (6M CHF)

But … wait!

There is a Red Hat product that provides security updates for Red Hat Enterprise Linux 7:

Red Hat "Extended Lifecycle Support" 7

Extended Lifecycle Support (ELS)

- Addon for Red Hat Enterprise Linux (7) licensed systems
  - Red Hat support is provided for these systems
- Only security updates are released that are deemed ‘important’ by Red Hat
- Fixed duration of four years (ending support in **30.06.2028**)

Maybe this will work …

Providing ELS7 at CERN

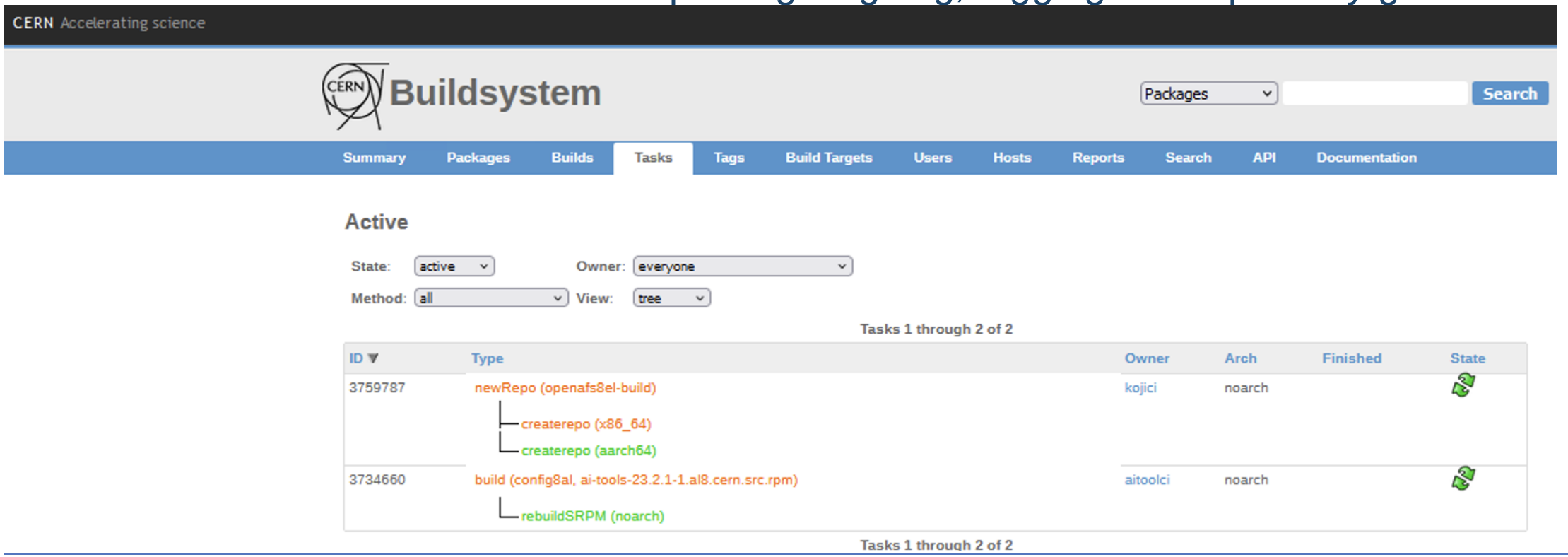
- CERN has the resources and infrastructure to build RPM packages
- CERN has staffing to support FEC systems internally

ATS (Accelerators and Technology Sector) / IT agreement EDMS

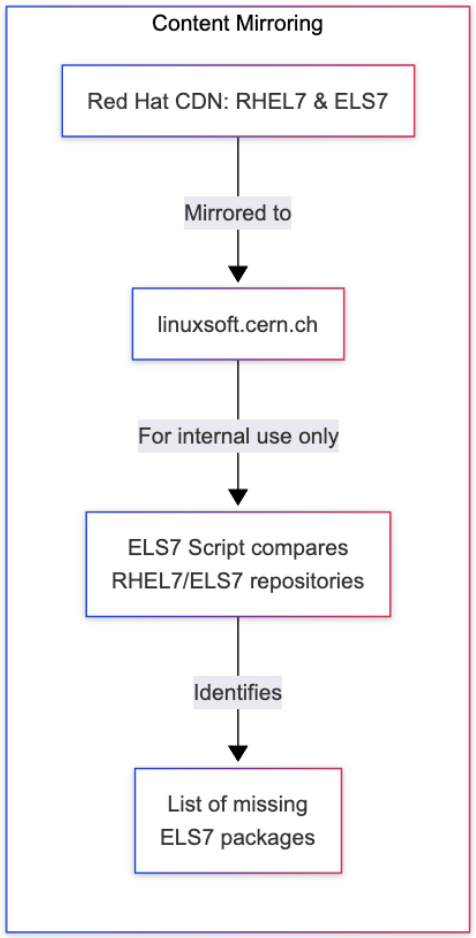
- It was agreed that CERN IT will rebuild the ELS7 product from Red Hat source RPMs and provide the binary RPMs internally for FEC systems to consume
  - This agreement will be time limited, *“during RUN3”*
  - Only for **FEC** systems, all other systems and software is out of scope

How RPMs are built at CERN

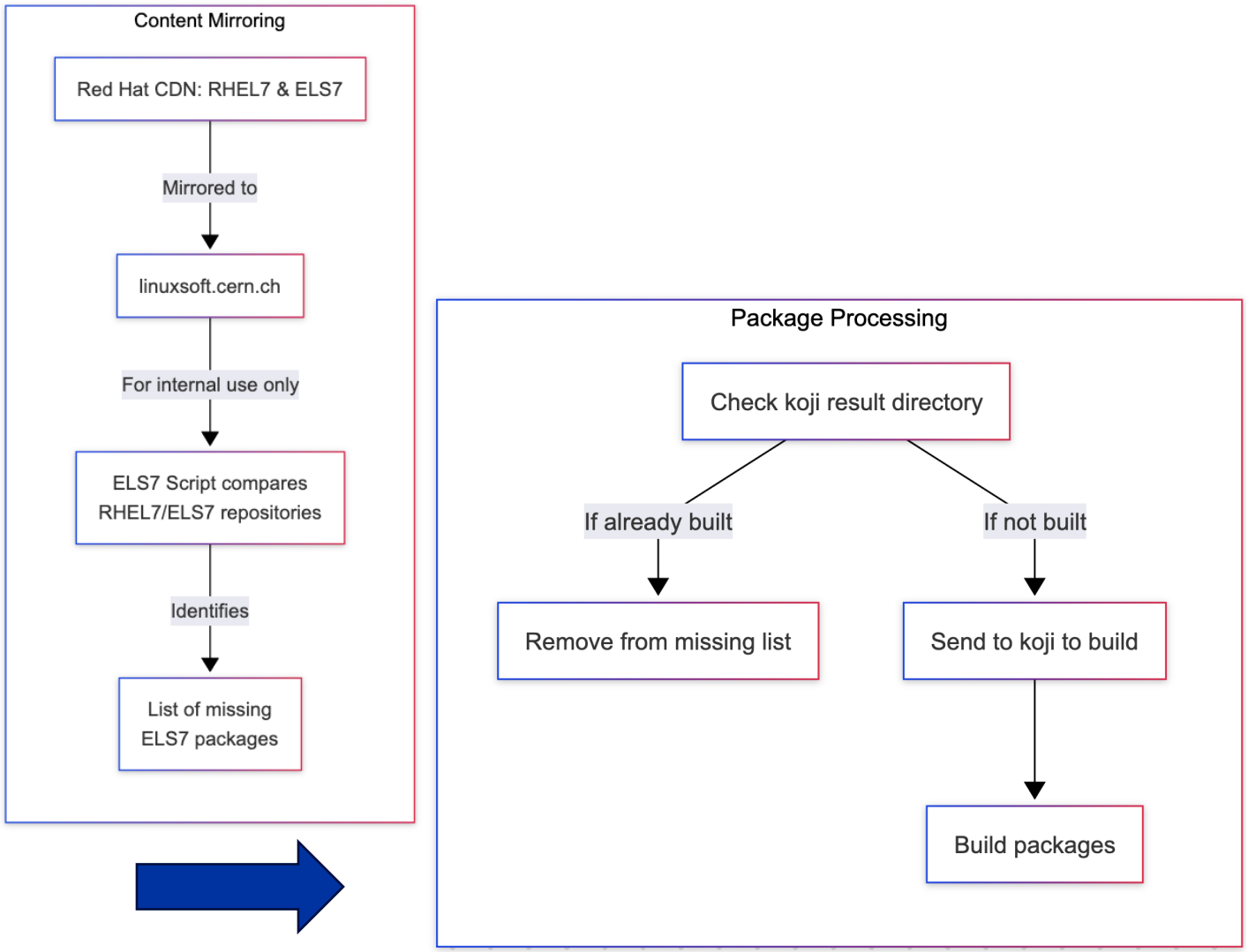
- CERN IT manages a KOJI instance
- A user can upload a source RPM (src.rpm) and KOJI will build from this a binary RPM
  - Our KOJI instance also handles package signing, tagging and repository generation



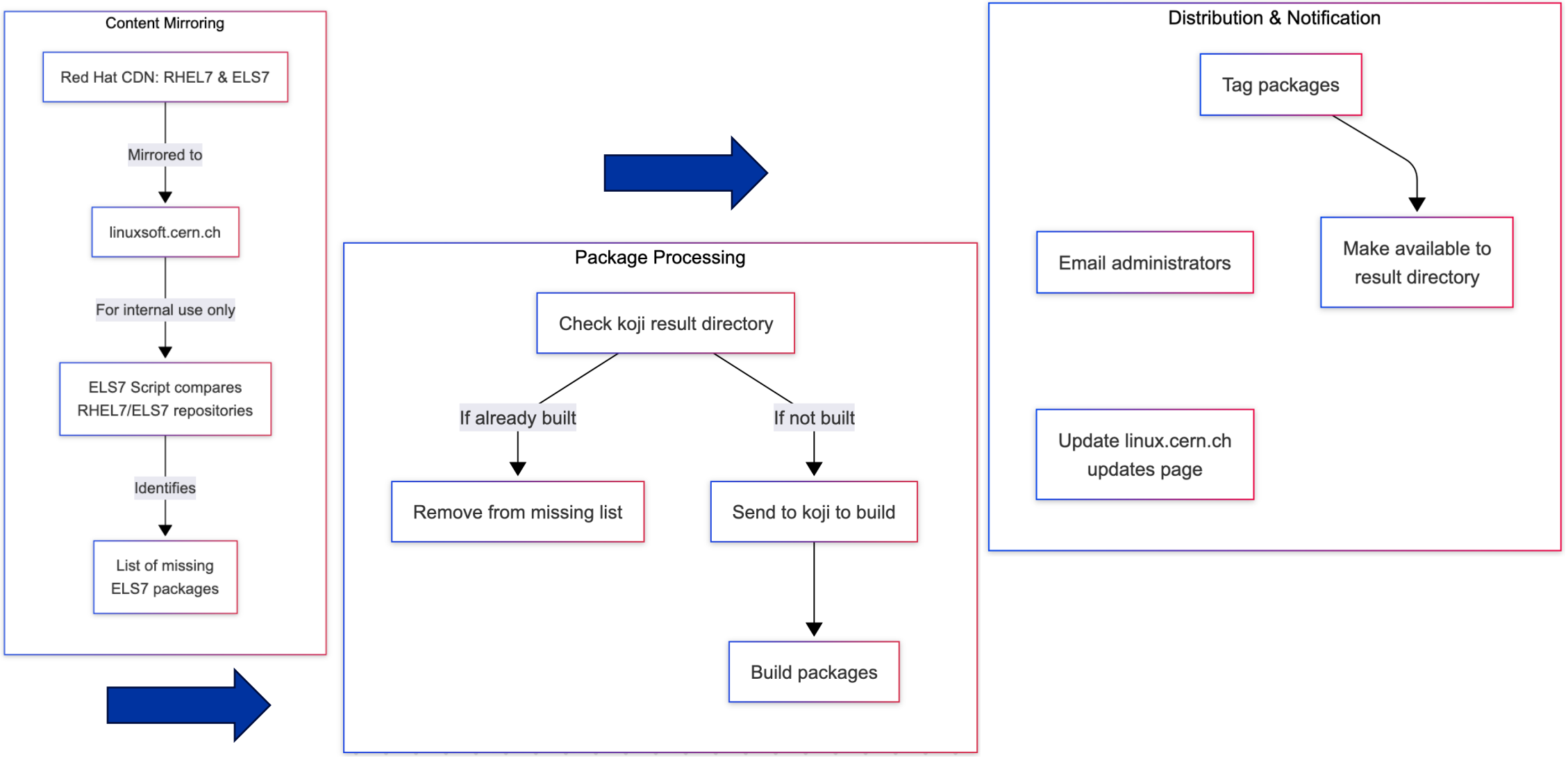
How is ELS7 built at CERN



How is ELS7 built at CERN



How is ELS7 built at CERN



How is ELS7 built at CERN

From: Linux Support <Linux.Support@cern.ch>
Sent: Tuesday, April 1, 2025 5:44 AM
To: lxsoft-admins (Mailing list for the administrators of Linux.Support inst...) <lxsoft-admins@cern.ch>
Subject: ELS7: New packages

Dear admins,

Today there were 3 new packages, of which 0 failed to build. The full list is as follows:

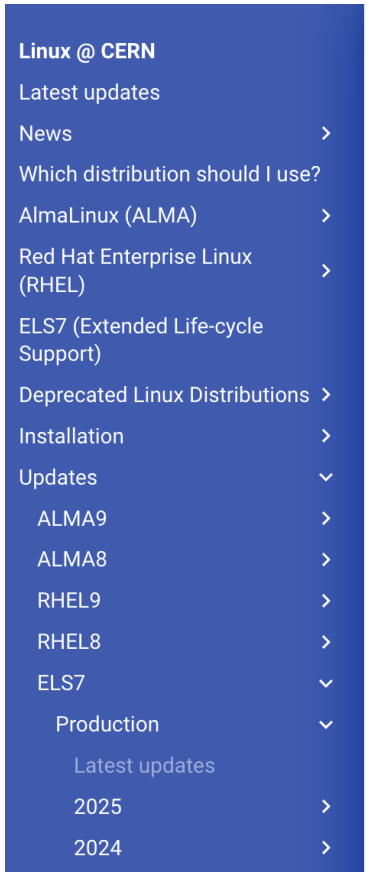
Built:
freetype-2.8-15.el7_9.1.src.rpm https://koji.cern.ch/taskinfo?taskID=3778966
tzdata-2025b-1.el7.src.rpm https://koji.cern.ch/taskinfo?taskID=3778969
grub2-2.02-0.87.el7_9.15.src.rpm https://koji.cern.ch/taskinfo?taskID=3778968

Failed:

As usual, all ELS7 packages can be found at https://linuxsoft.cern.ch/repos/els7-stable

Have a nice day.

Best regards,
CERN Linux Droid
(on behalf of the friendly humans of Linux Support)



Latest updates

2025-04-01

Package	Version	Advisory	Notes
freetype	2.8	RHSA-2025:3395	Security Advisory (CVE-2025-27363)
freetype-demos	2.8		
freetype-devel	2.8	RHSA-2025:3395	Security Advisory (CVE-2025-27363)
grub2	2.02	RHSA-2025:3396	Security Advisory (CVE-2025-0624)
grub2-common	2.02	RHSA-2025:3396	Security Advisory (CVE-2025-0624)
grub2-efi-ia32	2.02	RHSA-2025:3396	Security Advisory (CVE-2025-0624)

This magic script runs daily and does all the work for us….
Until it doesn't ☺

ELS7 build problems – garbage in, garbage out

Problem:

Koji refused to build:

`tigervnc-1.8.0-33.el7_9`

- We implemented a `$SKIP_LIST`, allowing us to omit this package from being included as a build candidate
- We informed Red Hat that their source RPM was not possible to be built

Solution:

- Red Hat eventually replied and released `tigervnc-1.8.0-3`**`4`**`.el7_9` that could be built

```
patching file Xi/xiproperty.c
Reversed (or previously applied) patch detected!
Assume -R? [n] n
Apply anyway? [n] n
Skipping patch.
2 out of 2 hunks ignored -- saving rejects to
file Xi/xiproperty.c.rej
patching file randr/rrproperty.c
Reversed (or previously applied) patch detected!
Assume -R? [n] n
Apply anyway? [n] n
Skipping patch.
2 out of 2 hunks ignored -- saving rejects to
file randr/rrproperty.c.rej
error: Bad exit status from /var/tmp/rpm-
tmp.1inYp8 (%prep)
```

ELS7 build problems – building cutting edge code on 14 year old distros

Problems:

- `firefox` is an application that is built with `rust` > 1.73
- The `rust` package is not even provided by Red Hat (EL7)
- EPEL7 does provide `rust:1.72`
  - Note: EPEL7 is archived and removed from active mirror network
- `firefox` also requires recent(ish) versions of `clang, llvm, llvm-toolset`

ELS7 build problems – building cutting edge code on 14 year old distros

Solution:

- Build a recent version of `rust` for EL7
- Utilise the `llvm-toolset-14.0` **software collection** to get recent versions of `clang, llvm, llvm-toolset`
- Provide special treatment to `firefox`

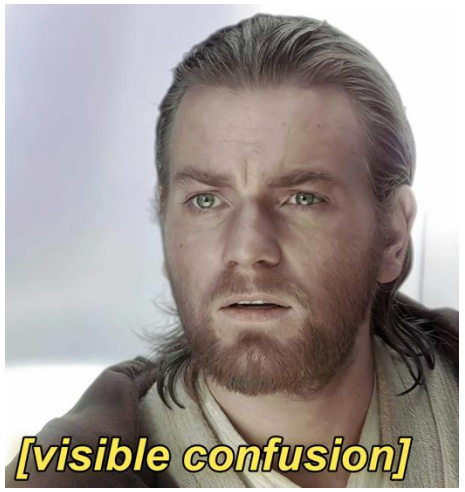
```
+    # firefox gets special treatment as the RPM needs llvm-toolkit-14 to be configured
+    # firefox also requires recent versions of rust/cargo - these are satisfied via https://
  gitlab.cern.ch/linuxsupport/rpms/rust
+    if [ "$PACKAGE" == "firefox" ]; then
+      rpm -ivh $FULL_ELS_PACKAGE_PATH
+      sed -i '1i %global ___build_pre %{___build_pre}; source scl_source enable llvm-toolset-14.0
  || :' ~/rpmbuild/SPECS/firefox.spec
+      sed -i 's/BuildRequires:  clang clang-libs llvm llvm-devel/BuildRequires:  llvm-toolset-14.0-
  clang-libs llvm-toolset-14.0-clang llvm-toolset-14.0-llvm-devel llvm-toolset-14.0-llvm/' ~/rpmbuild/
  SPECS/firefox.spec
+      rpmbuild -bs --define "dist .el7_9" ~/rpmbuild/SPECS/firefox.spec
+      FULL_ELS_PACKAGE_PATH=~/rpmbuild/SRPMS/${NVR}.src.rpm
+    fi
```

ELS7 build problems – finding bugs in our koji

Problem:

Our users reported that the `freeradius` package was not installable:

```
$ rpm -qp freeradius-3.0.20-1.el7_9.1.x86_64.rpm --requires |egrep
'package|is|not|installed'
installed
is
not
openssl >= package
```



The `freeradius` package is giving some really weird requirements. There are no such packages: “`is`”, “`not`”, “`installed`” and “package” is not an appropriate version for `openssl`

ELS7 build problems – finding bugs in our koji

Digging deeper. The error "***package openssl is not installed***" is the result of this line in the `freeradius` spec file:

```
Requires: openssl >= %(rpm -q --queryformat '%%{EPOCH}:%%{VERSION}' openssl)
```

However, there is also a `BuildRequires: openssl` line - so `openssl` should exist in the rpm build root. What gives?

More debugging pointed to an internal koji misconfiguration

ELS7 build problems – finding bugs in our koji

- We had for a long time (4 years) configuration to disable the bootstrap feature of `mock`
  - This configuration was originally deployed during CS8 as it was the recommendation at the time
  - With bootstrap disabled, `mock` uses the koji builder's `dnf` binary (AlmaLinux 9) to install packages in the `mock` chroot
  - In this ELS7 example, the `rpm` EL7 binary could not read the RPM database that had been created with the `rpm` binary from EL9
    - In EL9 (from `rpm` version 4.16), the RPM database changed from **Berkeley DB** to **SQLite** 

ELS7 build problems – finding bugs in our koji

Solution:

- reconfigure EL7 builds to use the bootstrap feature (default)

Final result:

```
$ rpm -qp freeradius-3.0.20-1.el7_9.1.1.x86_64.rpm --requires
|egrep 'package|is|not|installed|openssl’

openssl >= 1:1.0.2k
```

ELS7 build problems – providing multilib support
Problem statement

- Another user enquired why `krb5-libs.i686` was missing

ELS7 build problems – providing multilib support
Problem statement

- Another user enquired why `krb5-libs.i686` was missing

Wait – what is a `i686` package?

ELS7 build problems – providing multilib support
What is multilib?

Multilib in RHEL 7 refers to the ability run both **32-bit** (i686) and 64-bit (x86_64) applications on a 64-bit system. Multilib is important for environments that require backward compatibility with older applications or for running specific commercial software that's only available in 32-bit versions.

Multilib in RHEL 7:

- Allows 64-bit systems to run **32-bit** applications by providing the necessary **32-bit** libraries alongside the 64-bit versions
- **32-bit** packages are typically named with the "`.i686`" suffix, while 64-bit packages use "`.x86_64`"
- The offering of **32-bit** packages is minimal – only key development and libraries are provided
- Both architectures are contained within in the **same repository**

ELS7 build problems – providing multilib support (Back to the problem statement)

- Another user enquired why `krb5-libs.i686` was missing
  - Why? Because we conveniently forgot to build for multilib packages, but no problems, let's just add i686 to the list of architectures to build for?
    - `koji edit-tag --arches "x86_64 i686" els7-build`

Why this doesn't work:

- The repositories we use for koji ELS7 build dependencies don't actually have 32-bit (`i686`) content
- A source RPM from Red Hat typically builds several "sub packages", so providing you fix the above point – you can build a source RPM for `i686`, however Red Hat doesn't release ALL sub packages for `i686`
  - The source package for `krb5` builds for `x86_64`: `krb5-devel, krb5-libs, krb5-pkinit, krb5-server, krb5-server-ldap, krb5-workstation, libkadm5`
  - The `i686` packages that exist for `krb5` are only: `krb5-devel.i686, krb5-libs.i686, libkadm5.i686`

ELS7 build problems – providing multilib support - Solving the initial problems

- Permit building against two separate koji targets, one for `x86_64` and one for `i686`
  - The `i686` target has "*CentOS 7 altarch `i686`*" (32-bit) repos defined
- Use `rpmspec` from the spec file of a source RPM and loop on the sub packages querying the **existing** RHEL7 repository, checking to see if the a `i686` package **existed** in `i686` for RHEL7 
  - If so – send the build also to the `i686` build target
- If an `i686` build was sent, download the binary `i686` packages from koji
- Profit

ELS7 build problems – providing multilib support - Solving the initial problems

Luckily, koji allows us to import a binary RPM (`koji import`) into the final “stable” tag of the build in question

This works, and seems like a nice solution as it allows a `x86_64` repository to house `i686` packages (multilib)

Or at least it **should** ….

ELS7 build problems – providing multilib support - Solving … more problems

- We use `koji dist-repo` to generate end-user repositories, but since now we had `i686` packages tagged along side `x86_64` – we needed to enable **multilib** during the dist-repo process
  - Sadly dist-repo **multilib** support is actually broken, and has been for many years
  - I raised a pull request upstream to "*fix*" it
    - We utilise this patch in our deployment
    - Spurred on some discussion
    - It's unclear if this will be fixed for good upstream

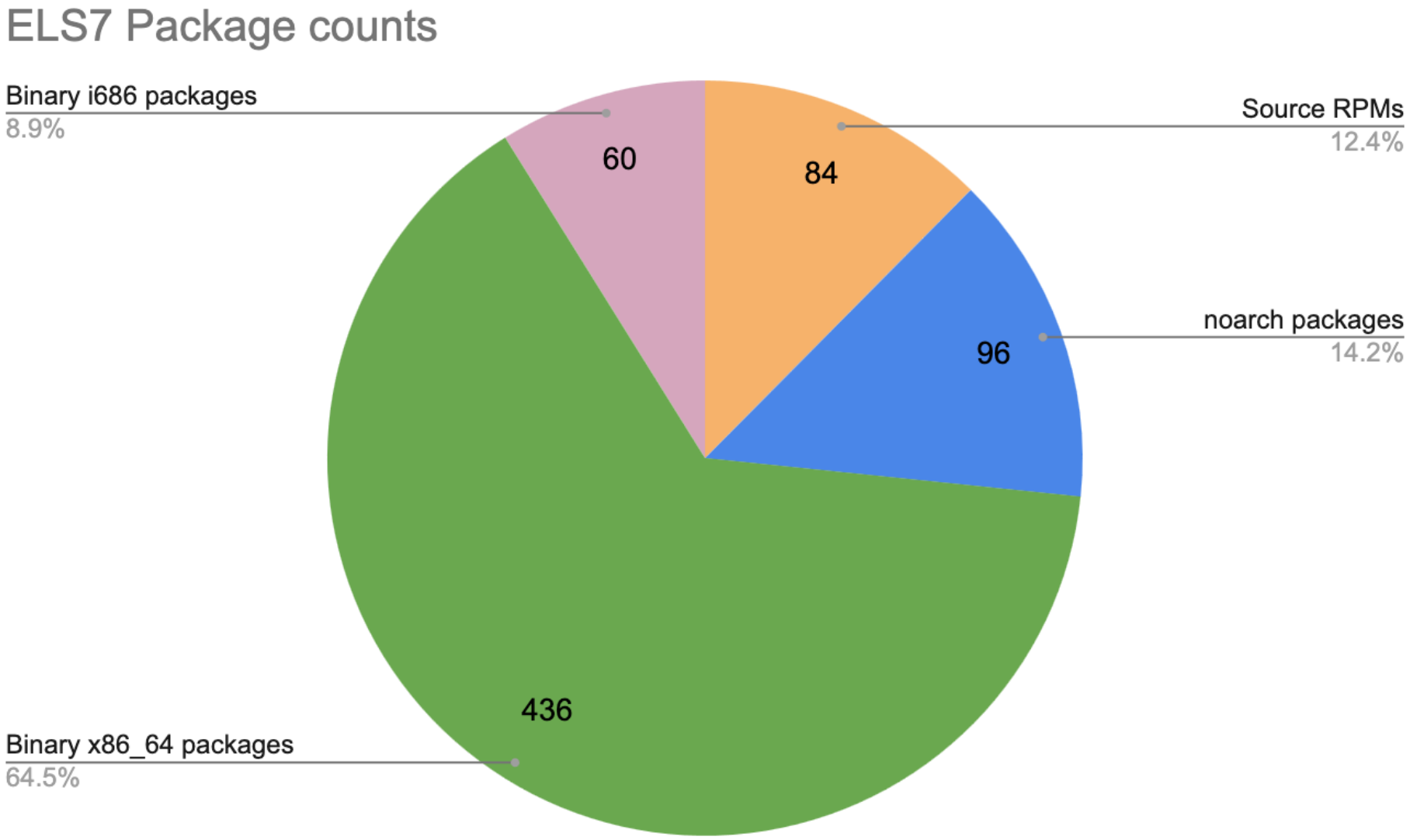
ELS7 lessons learned

- Just because the src.rpm comes from Red Hat, don't expect that it can be built
  - .feat `tigervnc`, `firefox`
- Just because we can build the RPM, don't expect that it can be installed
  - .feat `freeradius`
- *"Just rebuild the packages, it's easy"*
  - It's not. Be careful what you commit to your users, and for how long!
- Don't assume that nobody needs `i686` anymore
  - However … the RHEL9 release notes show:
    - *32-bit packages are **deprecated***
    - *Linking against 32-bit multilib packages is **deprecated**. The *.i686 packages will remain supported for the life cycle of Red Hat Enterprise Linux 9, but will be **removed** in the **next major** version of RHEL*
    - Just saying :grin:

ELS7 next steps

- Implement additional testing to the workflow, before releasing new updates into production
  - Avoiding such issues as seen with `freeradius`
  - Even though ELS7 is a temporary offering (scheduled end-of-life at CERN **30.06.2026**) validation of built packages is currently missing
  - Who knows what the future brings? We may one day need ELS8, ELS9 or ELS10
    - If so, much of the work has already been done
      - Note - CERN was already rebuilding ELS6 packages (from 01.07.2020 to 01.07.2025) via a similar method in a proof-of-concept capacity

ELS7 statistics



51 unique packages (some having successive version releases)

FEC OS future

- ELS7 at CERN ensures continuity of the LHC through **Run3 - June 2026**
- ELS7 as the upstream product ceases in **June 2028**
  - ELS at CERN ceases in **June 2026**
- Run4 starts in **June 2030**
- A next generation FEC OS project has been worked on for several years already
  - Allowing CERN to keep using the existing **x86_64-v1** hardware
  - Changing the OS to one that doesn’t restrict the microcode level

FEC OS future

- Currently 14 FECs are running **Debian** 12 (*“bookworm”*) in production and a large-scale migration of remaining FECs is foreseen in LS3
- **Debian** 13 (“*trixie*”) is slated for release sometime in 2025
- Freexian’s Extended LTS ensures 10 years of support (EOL 2035 taking us into LS4)
- Work is also ongoing to enable upgrades between major releases (eg: Debian **13** -> **14**) during operation

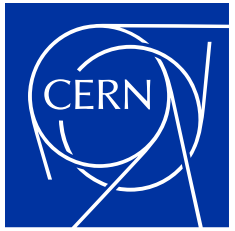
debian

ELS7 potentially useful(?) links

Or, how can I do this at my site?

- ELS7 build script: https://gitlab.cern.ch/linuxsupport/cronjobs/els7_release/
- Koji ELS7 tag configuration: https://gitlab.cern.ch/linuxsupport/lxdist-build/-/blob/master/bin/tags/els7.sh?ref_type=heads
- CERN package signing plugin: https://gitlab.cern.ch/linuxsupport/rpms/koji-hub-plugins-cern
- cern-els7-release RPM: https://gitlab.cern.ch/linuxsupport/rpms/releases/cern-els7-release
- Repository mirroring from Red Hat's CDN: https://gitlab.cern.ch/linuxsupport/cronjobs/reposync/-/blob/master/reposync/runreposync.sh?ref_type=heads
- Red Hat candlepin certificate synchronisation: https://gitlab.cern.ch/linuxsupport/cronjobs/reposync/-/blob/master/reposync/rhncheck.py?ref_type=heads
- Advisory database synchronisation: https://gitlab.cern.ch/linuxsupport/cronjobs/advisories

Thank-you for your attention!



home.cern