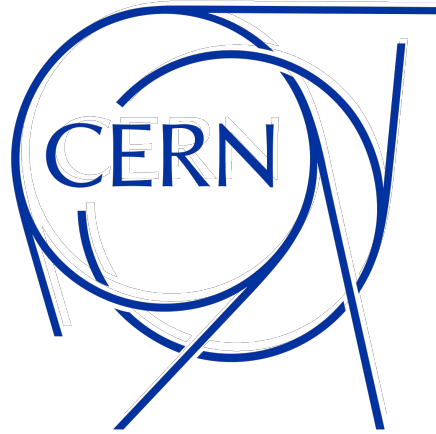




Controlling CERN's Accelerators with Debian

Debian's role in the world's largest particle accelerator

Federico Vaga & Nikos Tsipinakis

2026-08-30

About Us

Federico Vaga (federico.vaga at cern.ch)

- **At CERN since 2014, working in the control system infrastructure team**
- **Software engineer and project manager**
- **FOSS enthusiast - without free time**

Nikos Tsipinakis (IRC: nikos - email: nikos.tsipinakis at cern.ch)

- **Systems Engineer for the accelerator control system infrastructure**
- **Open Source maintainer**
- **Debian User for +10 years**
- **DC/DM for ~7 years**
- **DD - TBC :-)**

CERN In A Few Words

What is it, and what do we do?



WHERE THE WEB WAS BORN

In the offices of this corridor, all the fundamental technologies of the World Wide Web were developed.

Started in 1990 from a proposal made by Tim Berners-Lee in 1989, the effort was first divided between an office in building 31 of the Computing and Networking Division (CN) and one in building 2 of the Electronics and Computing for Physics Division (ECP).

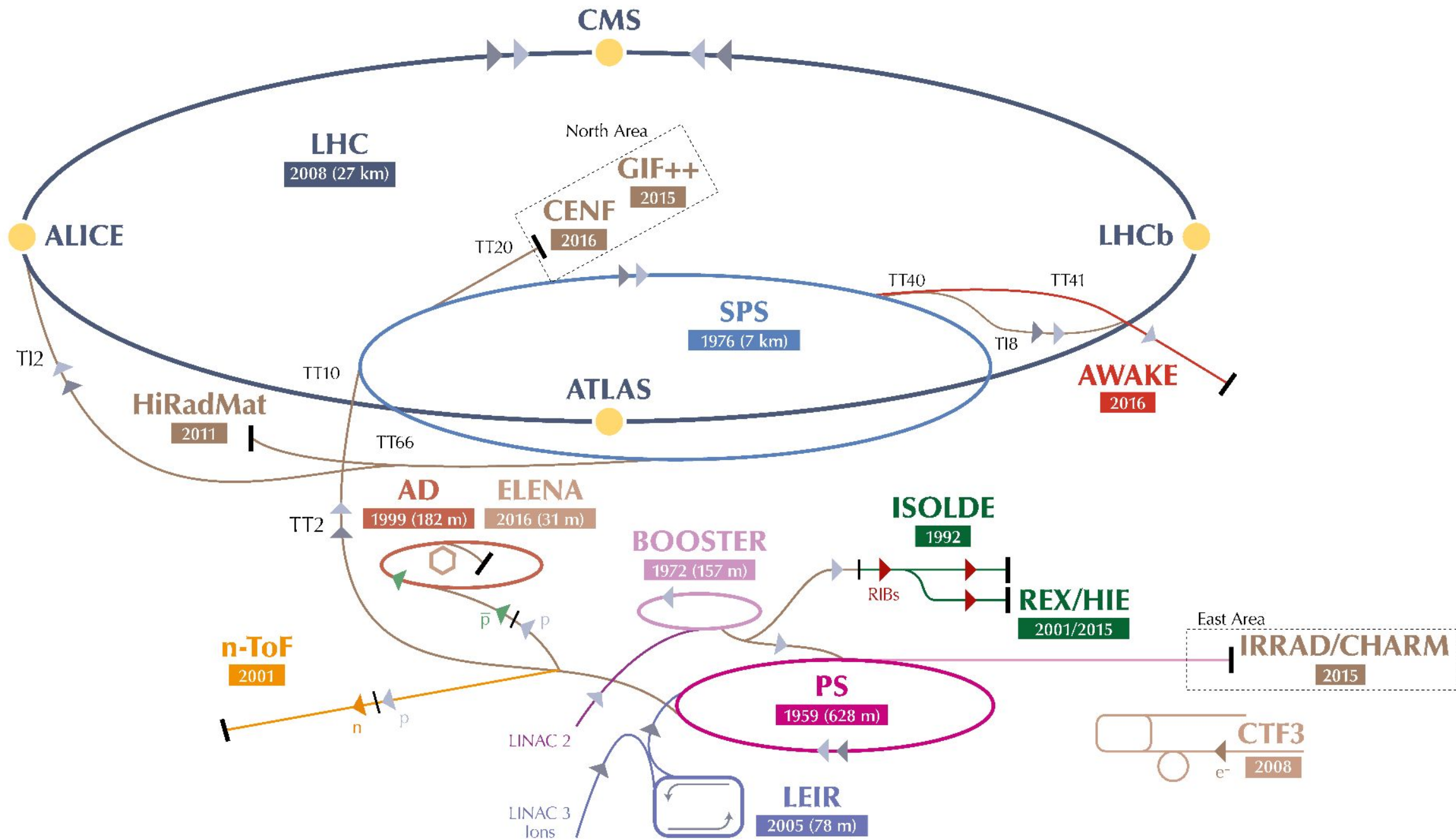
In 1991 the team came together in these offices, then belonging to ECP. It was composed of two CERN staff members, Tim Berners-Lee (GB) and Robert Cailliau (BE), aided by a number of Fellows, Technical Students, a Coopérant and Summer Students.

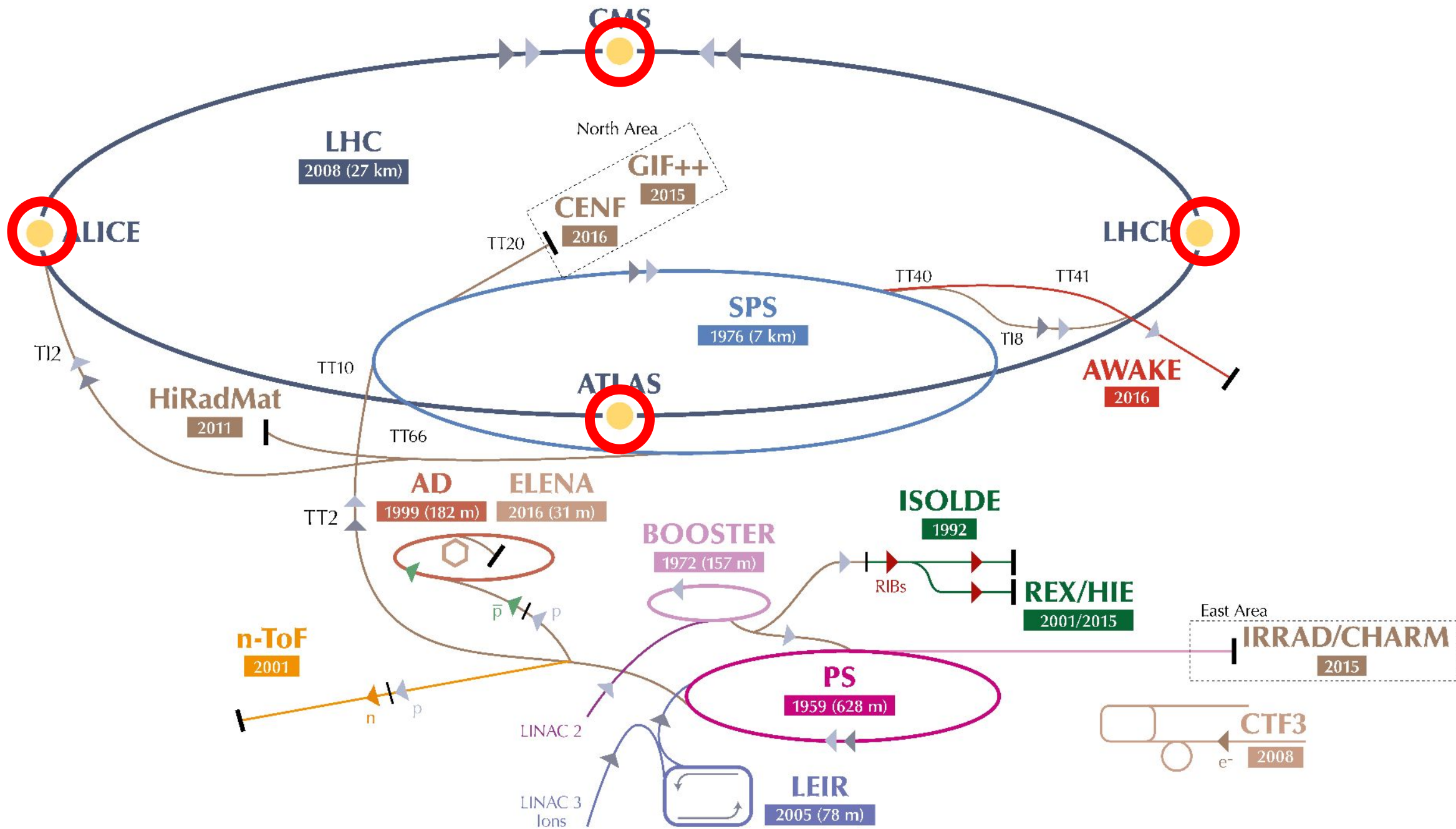
At the end of 1994 Tim Berners-Lee left CERN to direct the WWW Consortium (W3C), a world-wide organization devoted to leading the Web to its full potential. The W3C was founded with the help of CERN, the European Commission, the Massachusetts Institute of Technology (MIT), the Institut National pour la Recherche en Informatique et en Automatique (INRIA), and the Advanced Research Projects Agency (ARPA).

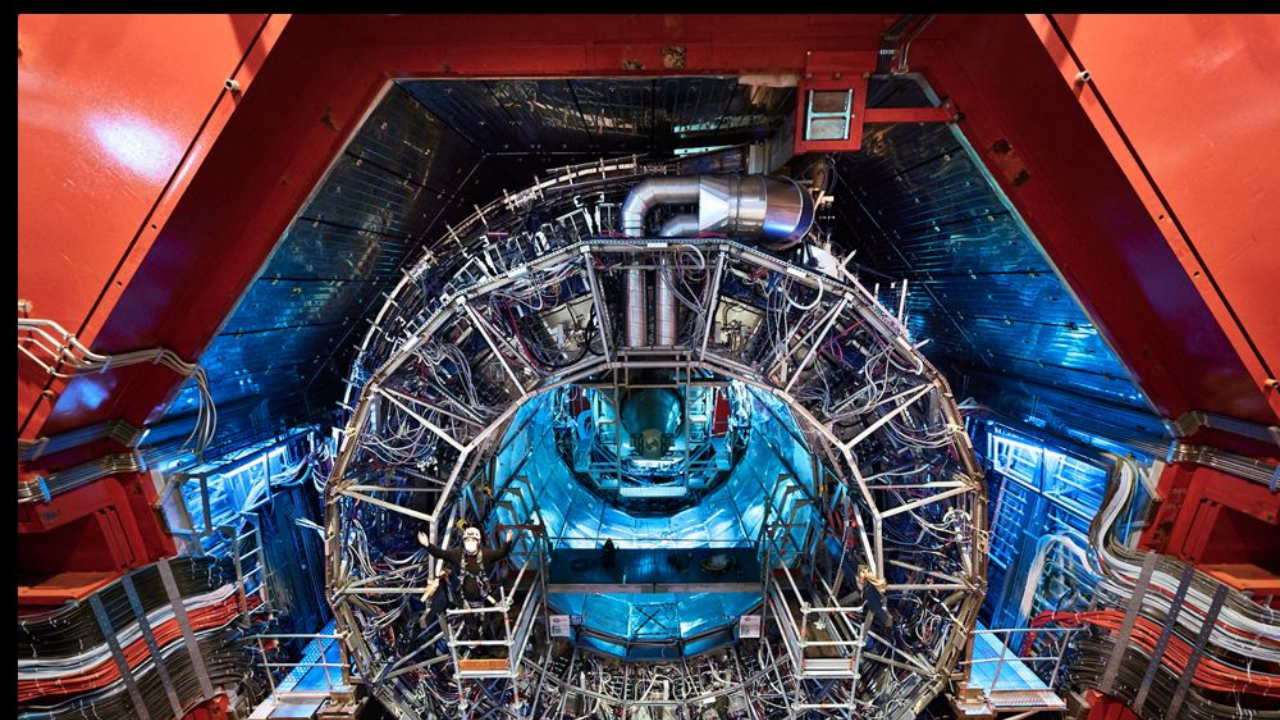
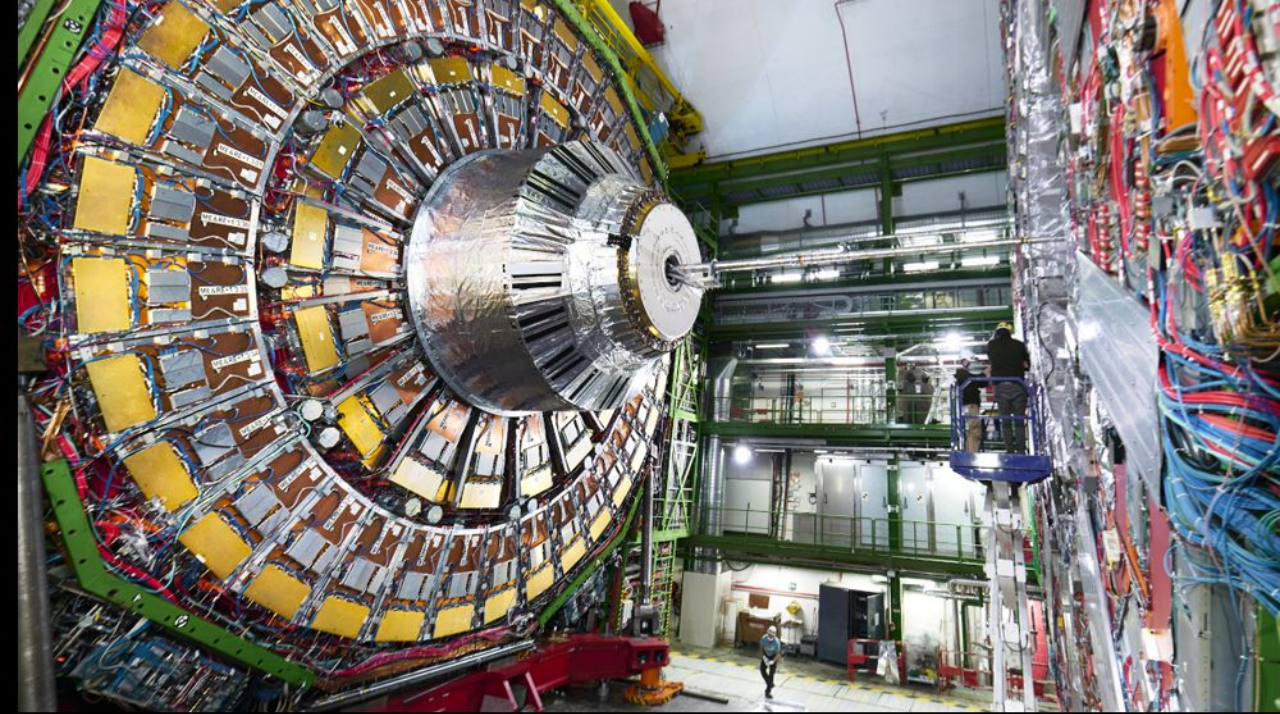
In 1995 Tim Berners-Lee and Robert Cailliau received the ACM Software System Award for the World Wide Web. In 2004, Tim Berners-Lee was awarded the first Millennium Technology Prize by the Finnish Technology Award Foundation.

The CERN Library
June 2004









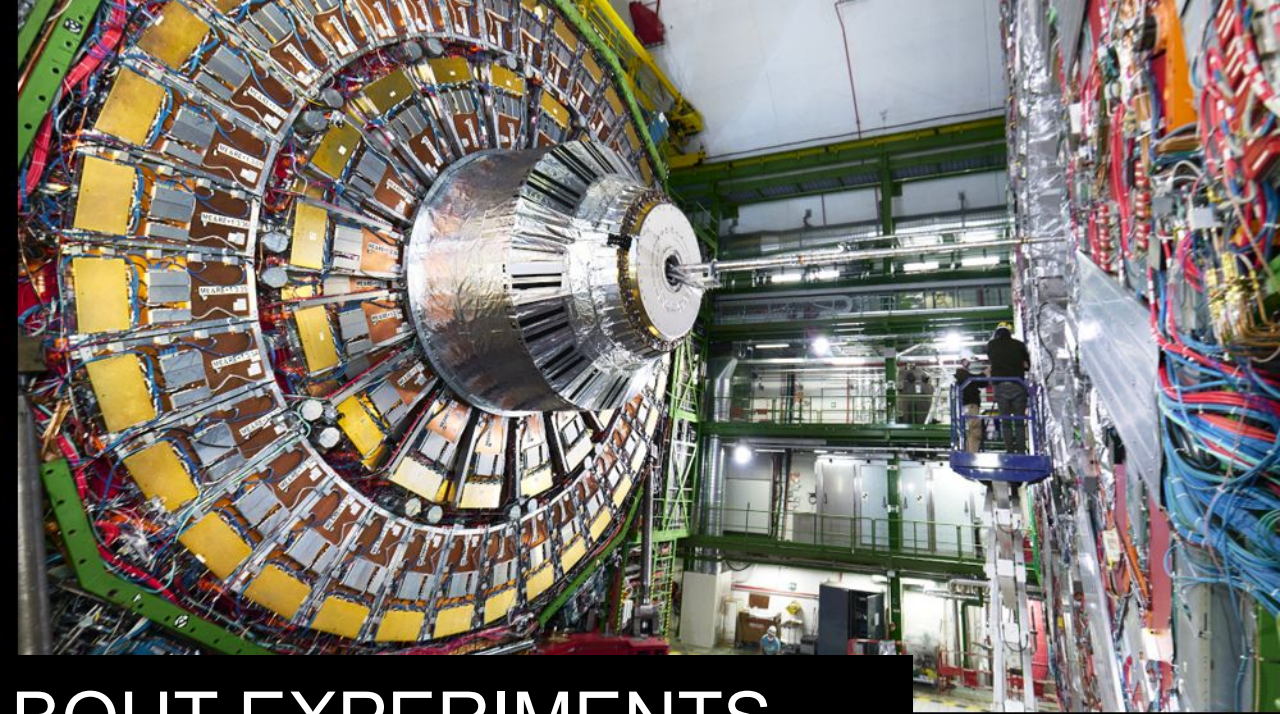


***“Without context, words and actions
have no meaning at all”***

~ Gregory Bateson



WE WILL NOT TALK ABOUT DATA CENTERS

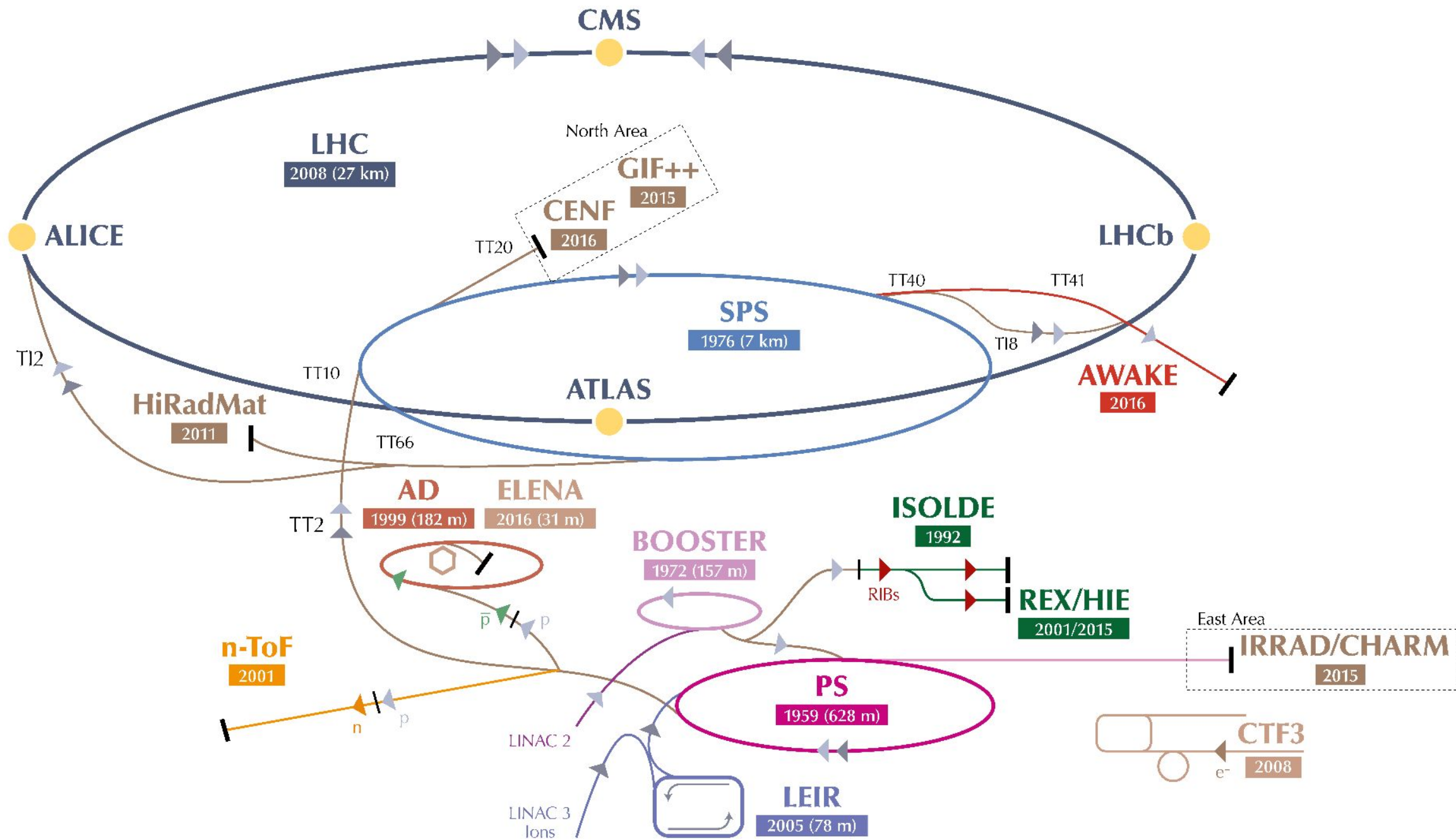


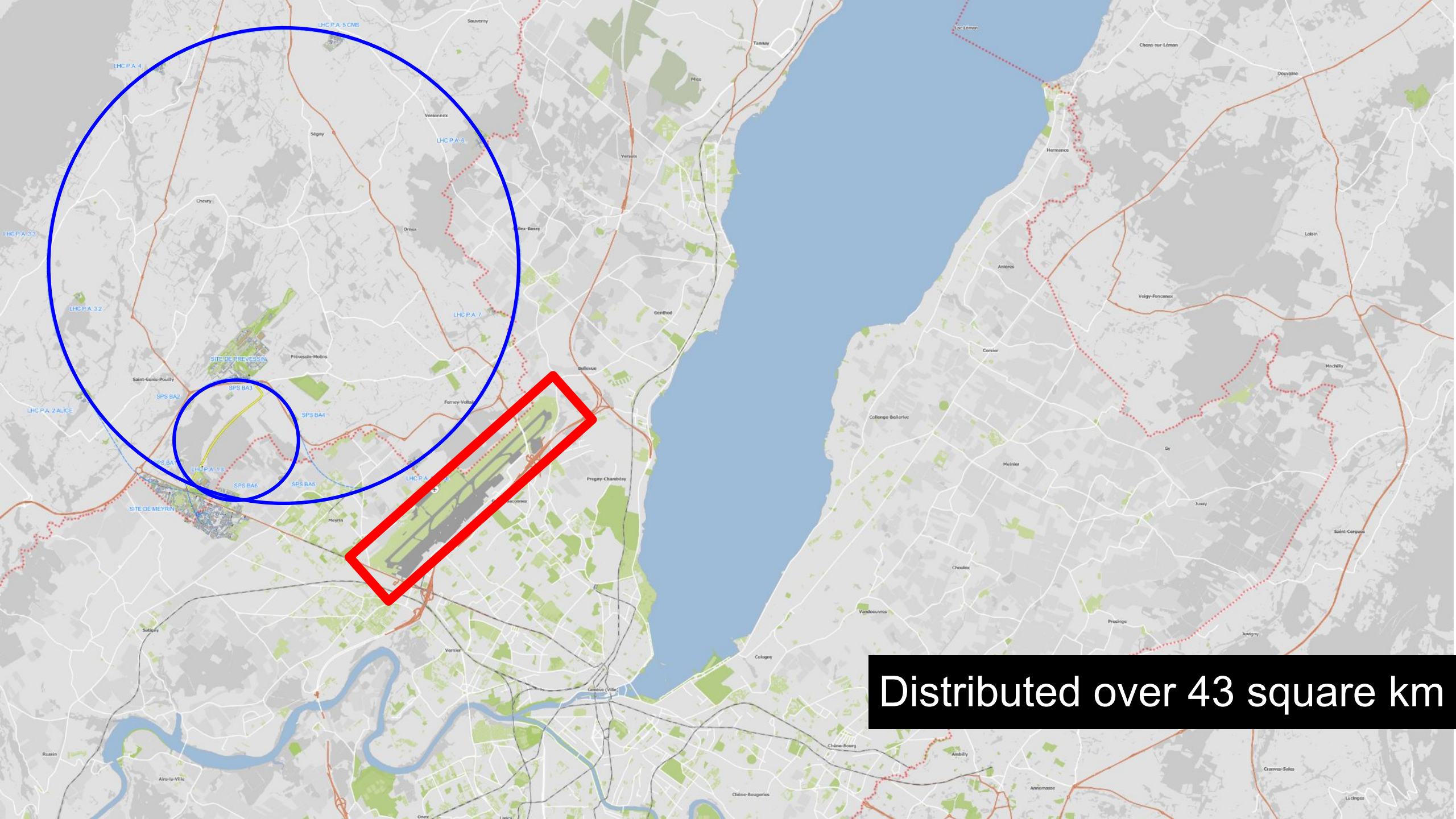
WE WILL NOT TALK ABOUT EXPERIMENTS



The Accelerator Context

A brief overview of what had an impact on our choice

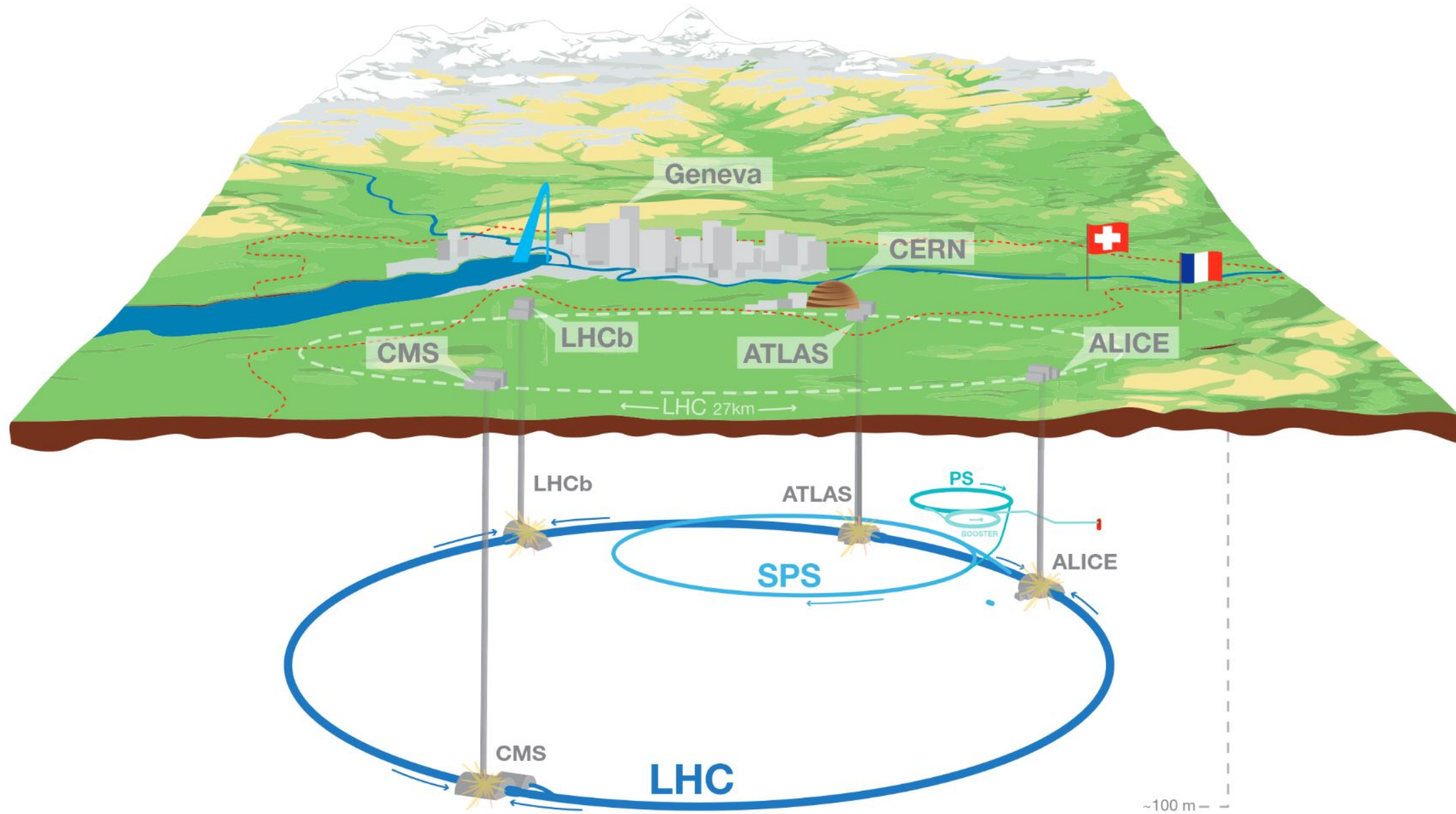


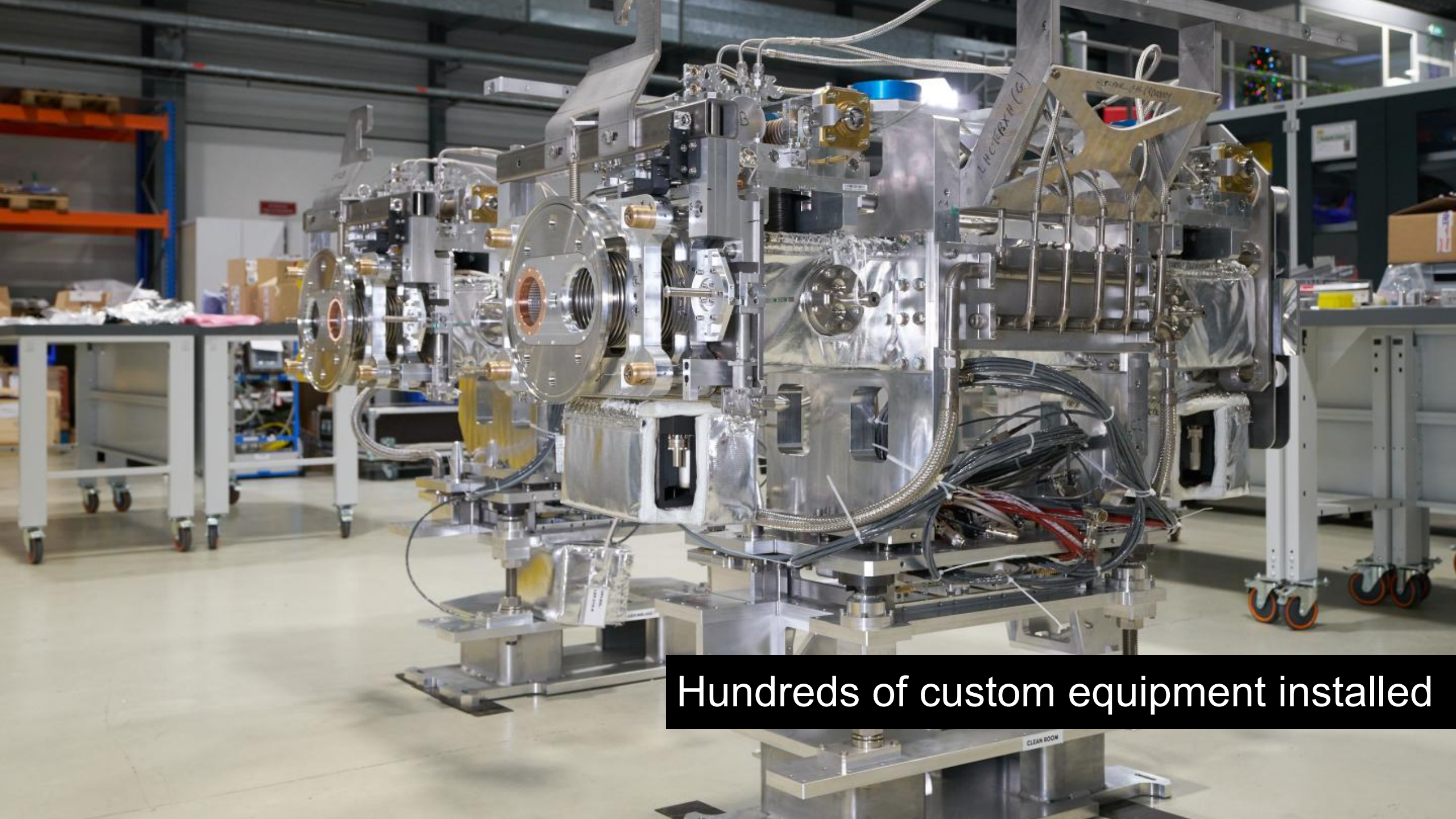


Distributed over 43 square km

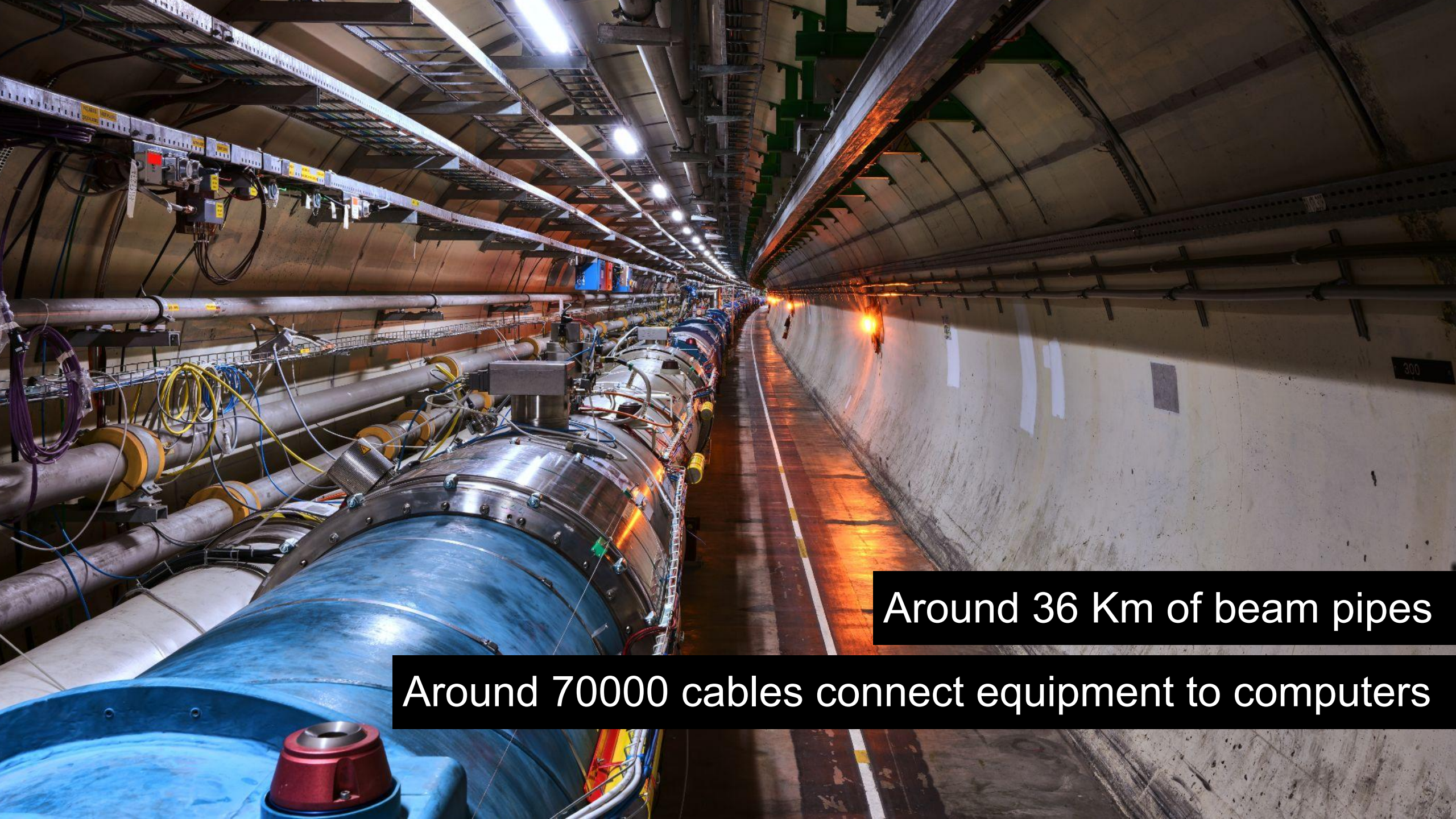


Most of it is underground



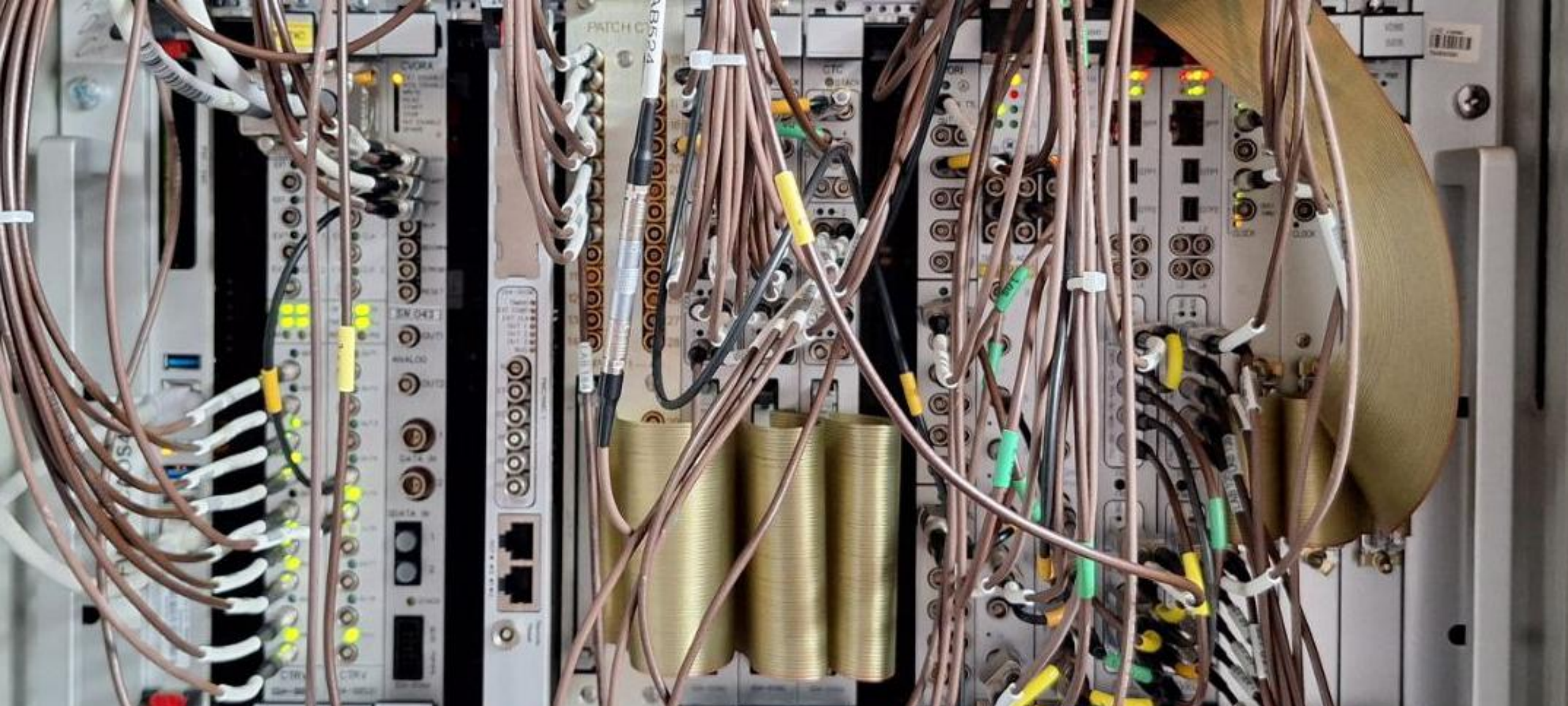


Hundreds of custom equipment installed



Around 36 Km of beam pipes

Around 70000 cables connect equipment to computers



Around 2200 computers, 17000 devices installed (and growing)







We need a stable environment

The Linux Context

Operating System history, Linux adoption, and usage

Real-Time Operating System History



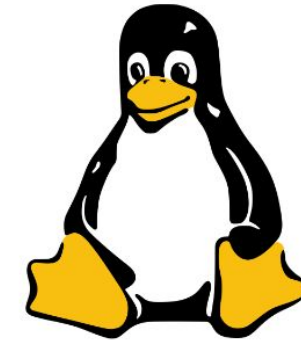
198x



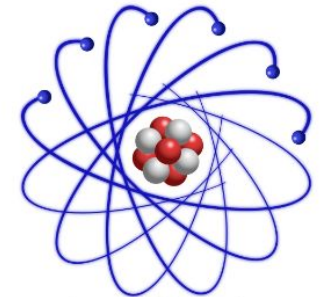
1992



Red Hat
Enterprise Linux



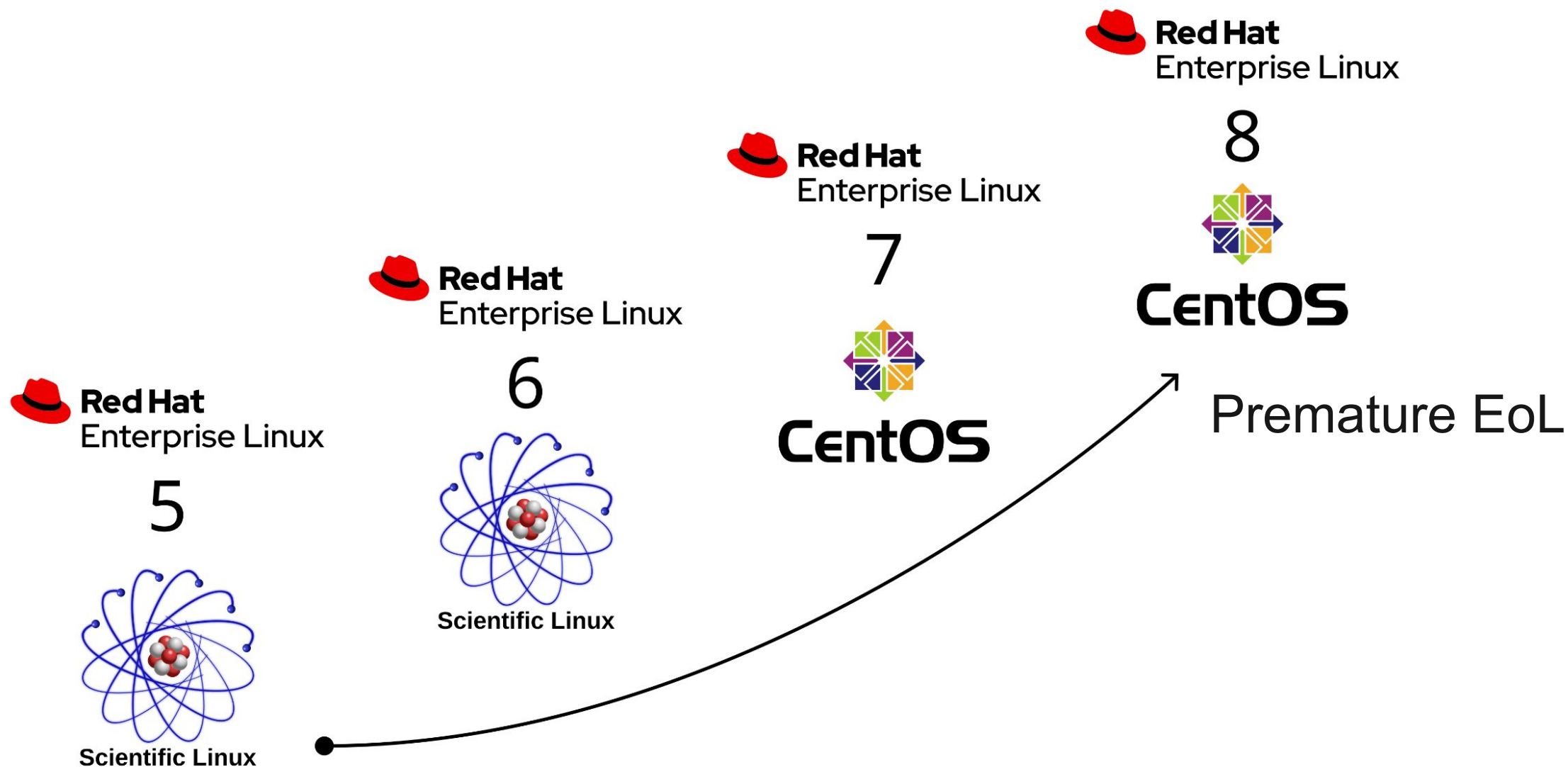
PREEMPT_RT



Scientific Linux

2008

Linux Updates On Autopilot

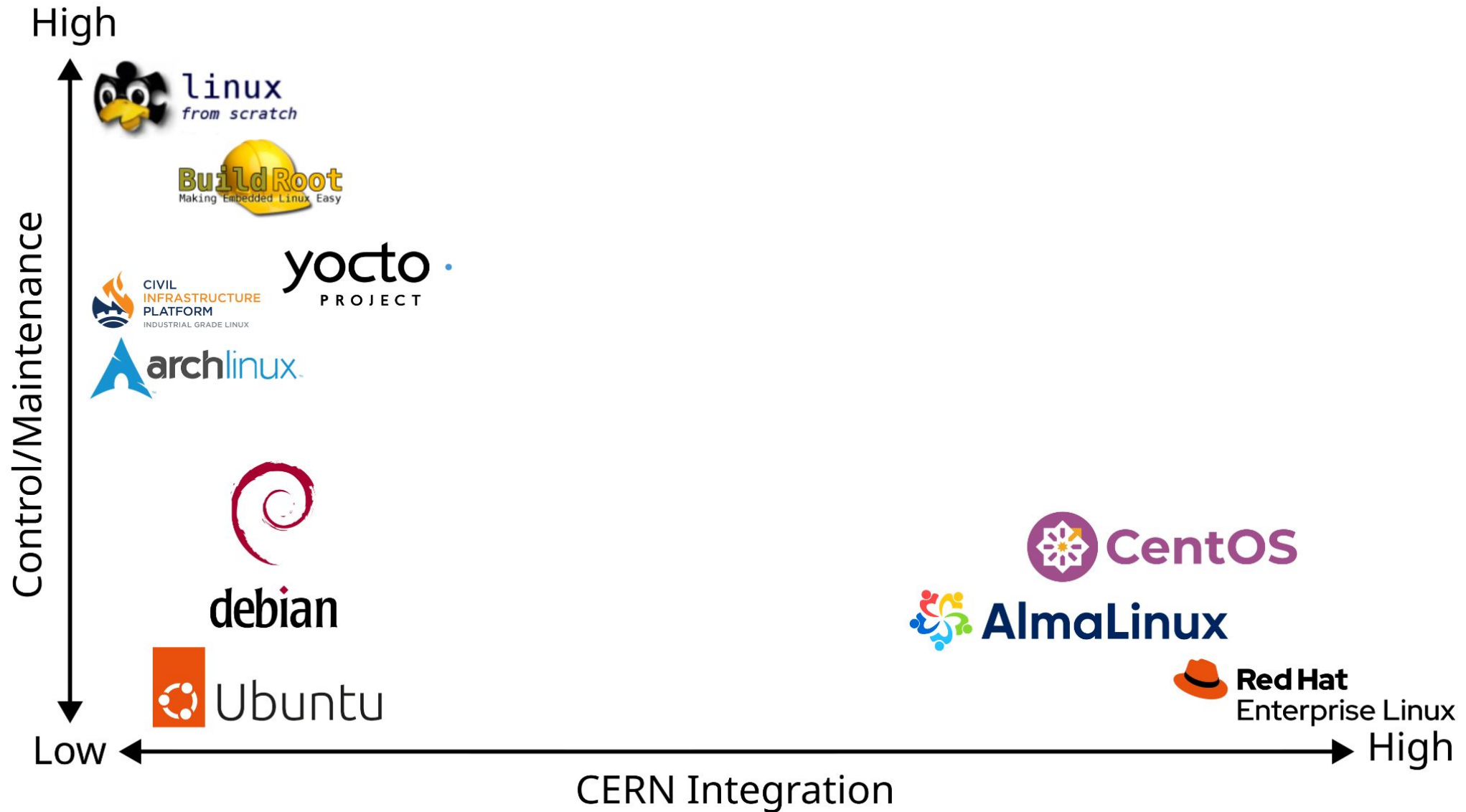


Any Would Do ...

... but not all integrates well in a given context



The Selection Process



[Try To] Stay In Red Hat Ecosystem

Use CentOS Stream

- **Avoid divergences within CERN**
- **Reuse as much as possible from the current infrastructure and services**
- **Start the validation on CentOS Stream 9**
- **Upgrade to CentOS Stream 10 as soon as available**
 - Likely upgrade to CentOS Stream 11 when available
 - To be used in production on the next accelerator restart until 2033

We can't afford new surprises

- **Make the CERN integration layer distribution agnostic**
- **Select and prepare a plan B Operating System: Debian**

The Straw That Broke The Camel's Back

What made us abandon Red Hat Enterprise Linux for the accelerator complex

Forcing Obsolescence With A Compiler Flag

```
gcc -Wall -Wextra -Werror -Wconversion -Wdouble-promotion \  
    -Wno-unused-parameter -Wno-sign-conversion \  
    -O3 -g3 -march=x86-64-v2 -std=c17 \  
    -o random-name.out random-name.c  
  
[ -march=x86-64-v3 ]
```

References:

<https://developers.redhat.com/blog/2021/01/05/building-red-hat-enterprise-linux-9-for-the-x86-64-v2-microarchitecture-level>

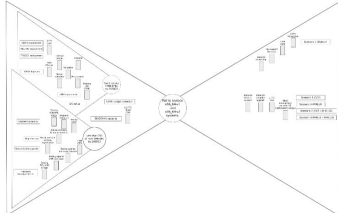
https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9/html/9.0_release_notes/architectures

<https://developers.redhat.com/blog/2021/01/05/building-red-hat-enterprise-linux-9-for-the-x86-64-v2-microarchitecture-level>

https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/10/html/10.0_release_notes/architectures

Estimating The Cost Of A Compiler Flag

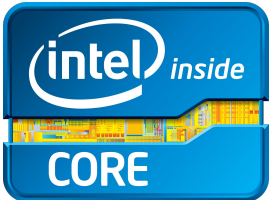
Risk Analysis (2023Q2): what if we had stayed in Red Hat ecosystem?



~47% x86-64

RHEL 9 -march=x86-64-v2

Ivy Bridge gen 3



~17% x86-64-v2

RHEL 10 -march=x86-64-v3

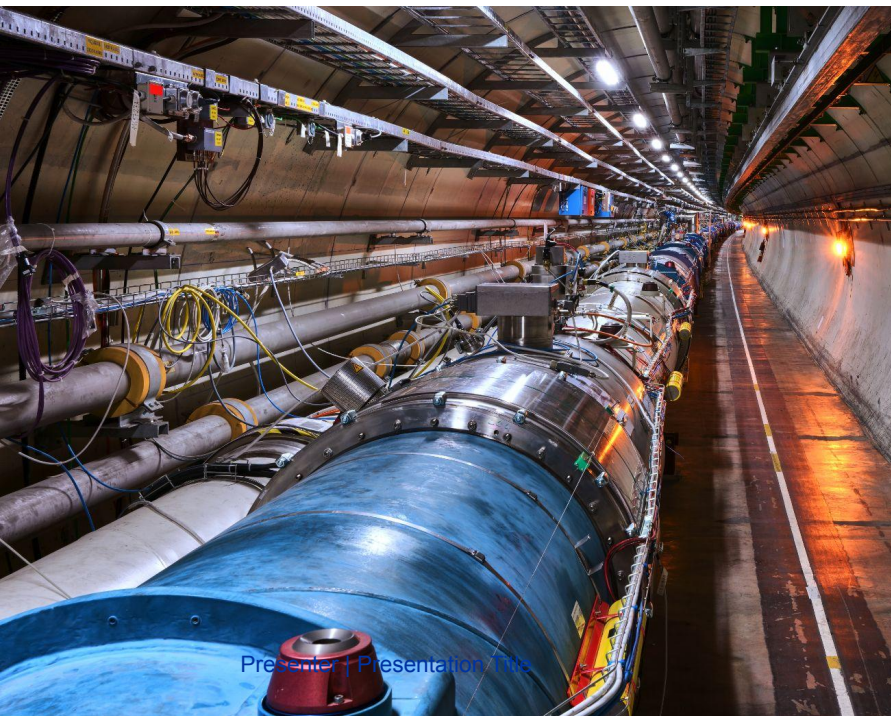
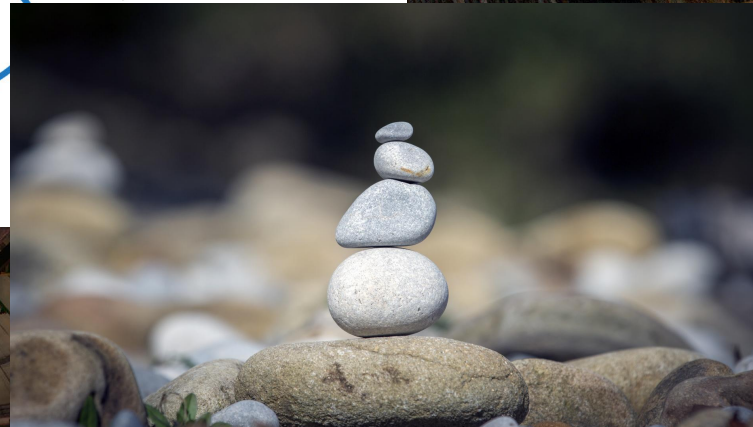
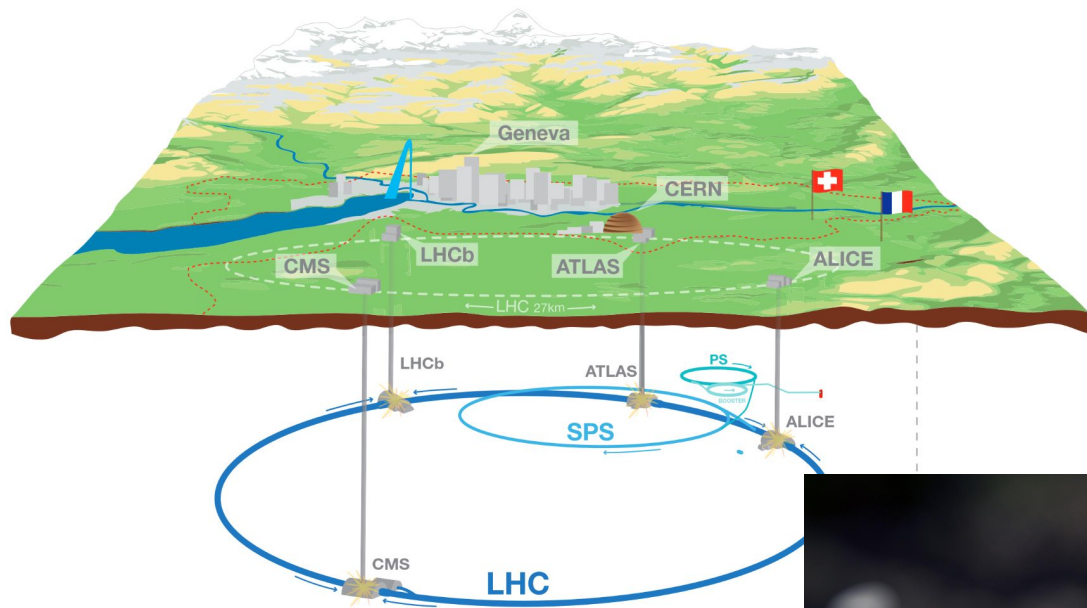


Single Board Computer



and more ...





Estimating The Cost Of A Compiler Flag

Risk Analysis (2023Q2): what if we had stayed in Red Hat ecosystem?

- **Hard deadline 2026Q4**
- **Estimated budget: 5.4 million CHF**
- **Hardware: around 11 boards to be redesigned**
- **Hiring: 2 electronic engineers, 2 software engineers, 2 technicians**
- **Installation: re-organize and re-cable racks**
- **Commissioning: most systems will be affected**
- **Estimated success rate: an optimistic 20% (assuming bug-free solutions)**

What did we do then?

- **Software solution for a software problem: no Red Hat Enterprise Linux**
- **Adopt Debian for the accelerator's computers**

The New Architecture

The Operating System and its surrounding infrastructure

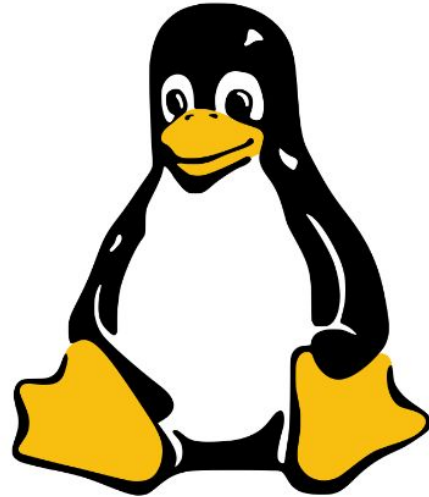
Distribution Decomposition

Decoupling the Linux kernel from its distributions

Bootloader



Kernel



With Real-Time patches

With all necessary drivers built-in

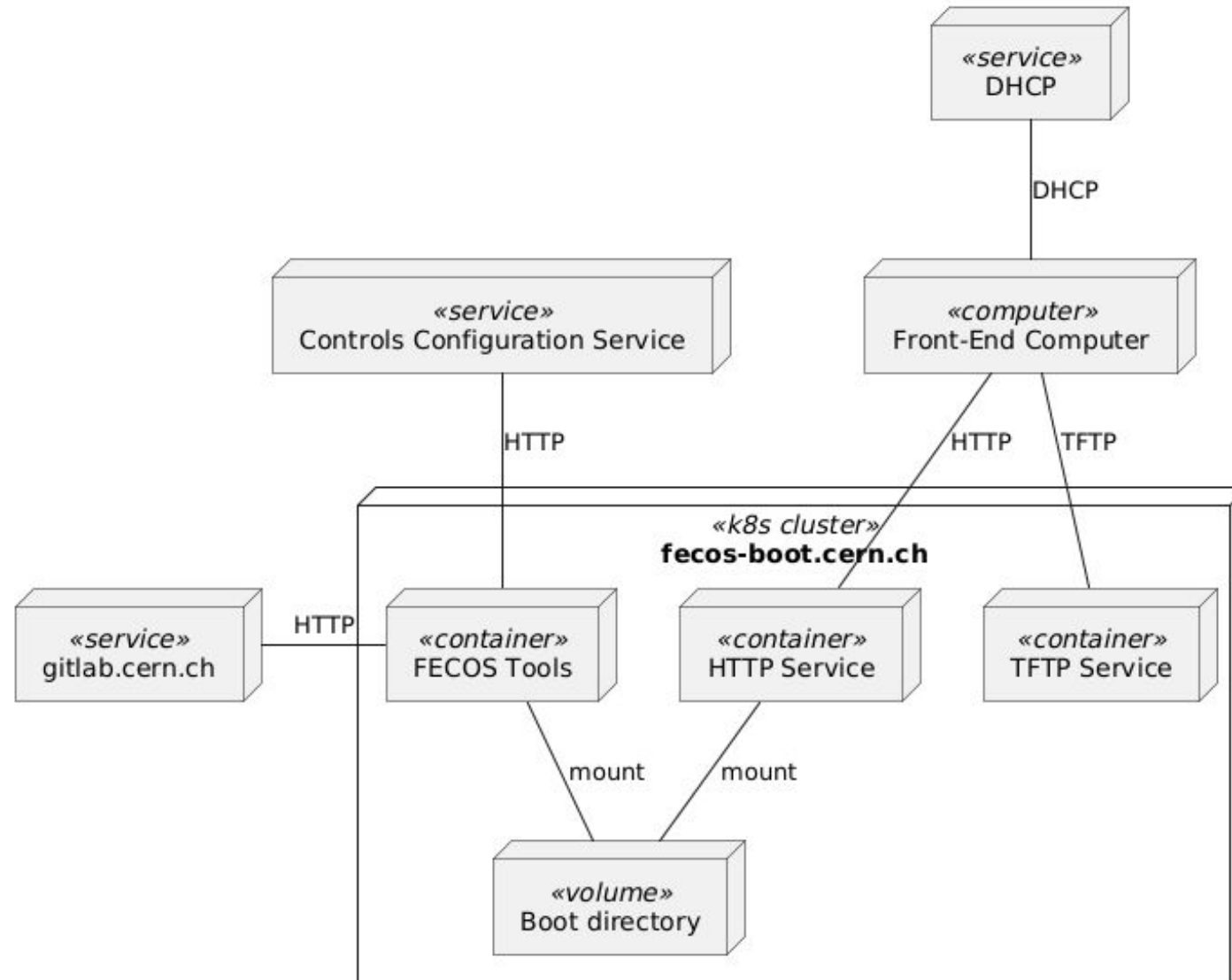
Init RAM Disk

DRACUT 
debian

Userspace



Distributing OS Components - Architecture



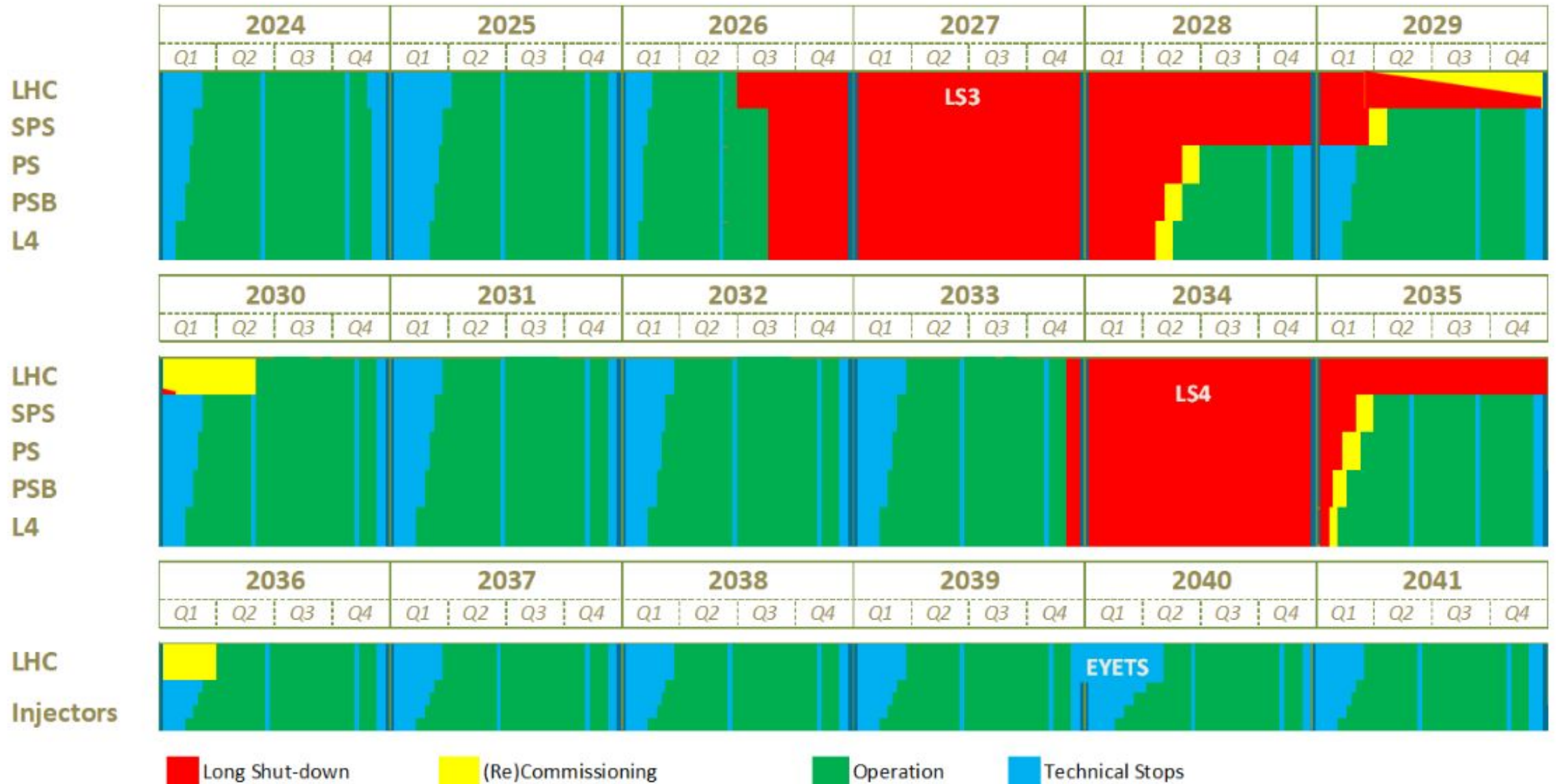
The Future of CERNs Control System ...is Debian



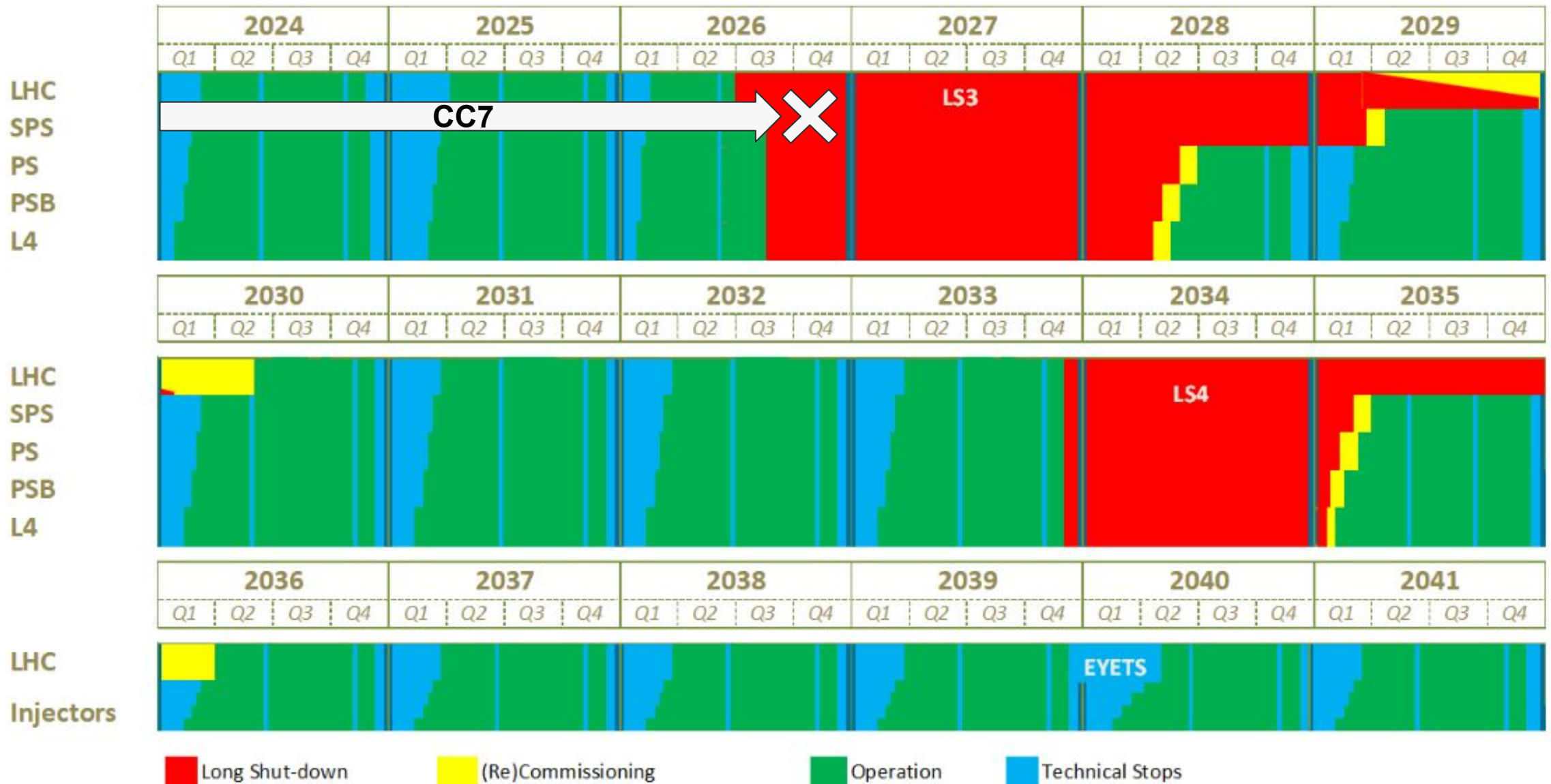
The Accelerator Schedule & Debian

Picking the right release so we don't have to hotpatch a live particle accelerator

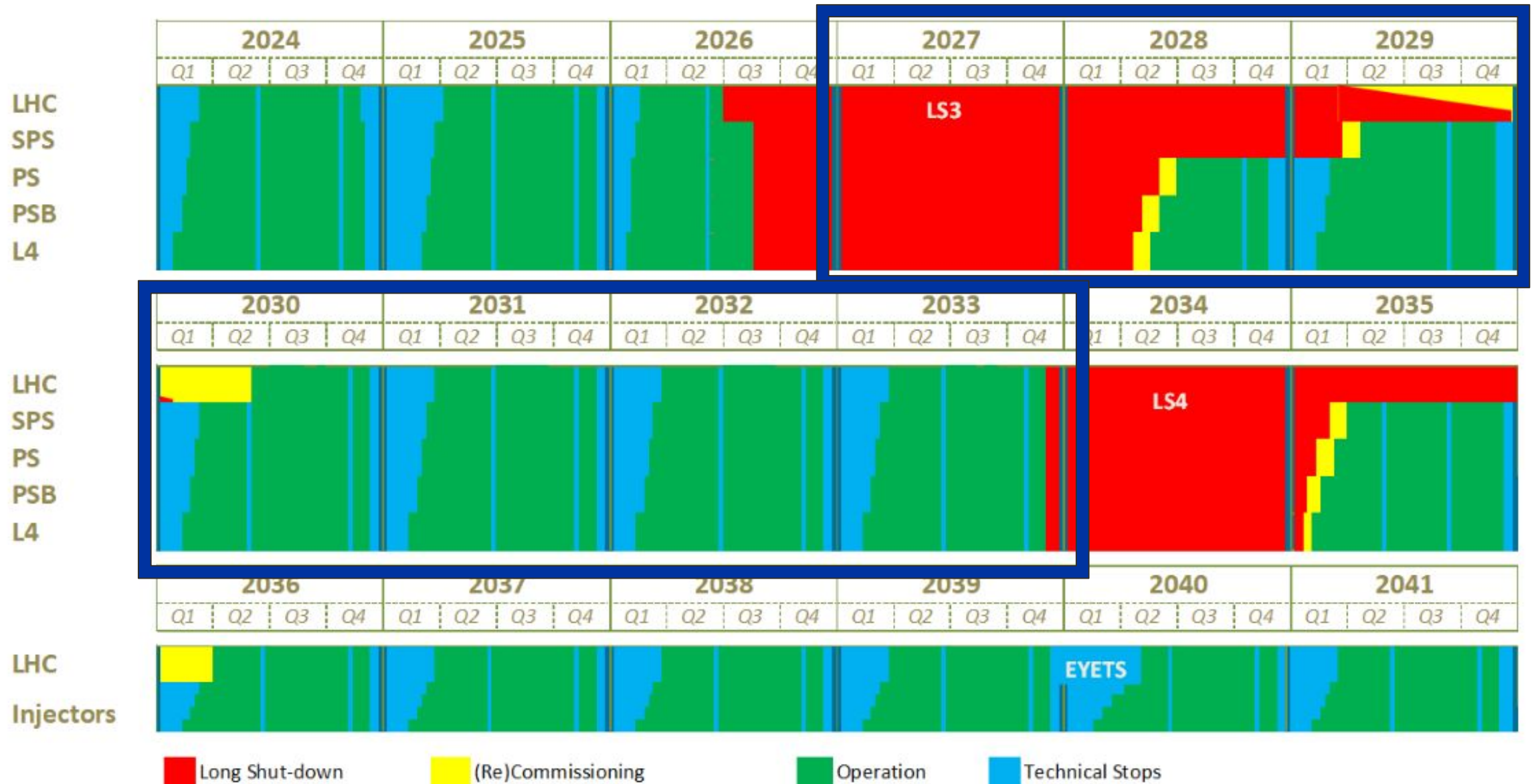
Long Term Schedule for CERN Accelerator complex



Long Term Schedule for CERN Accelerator complex

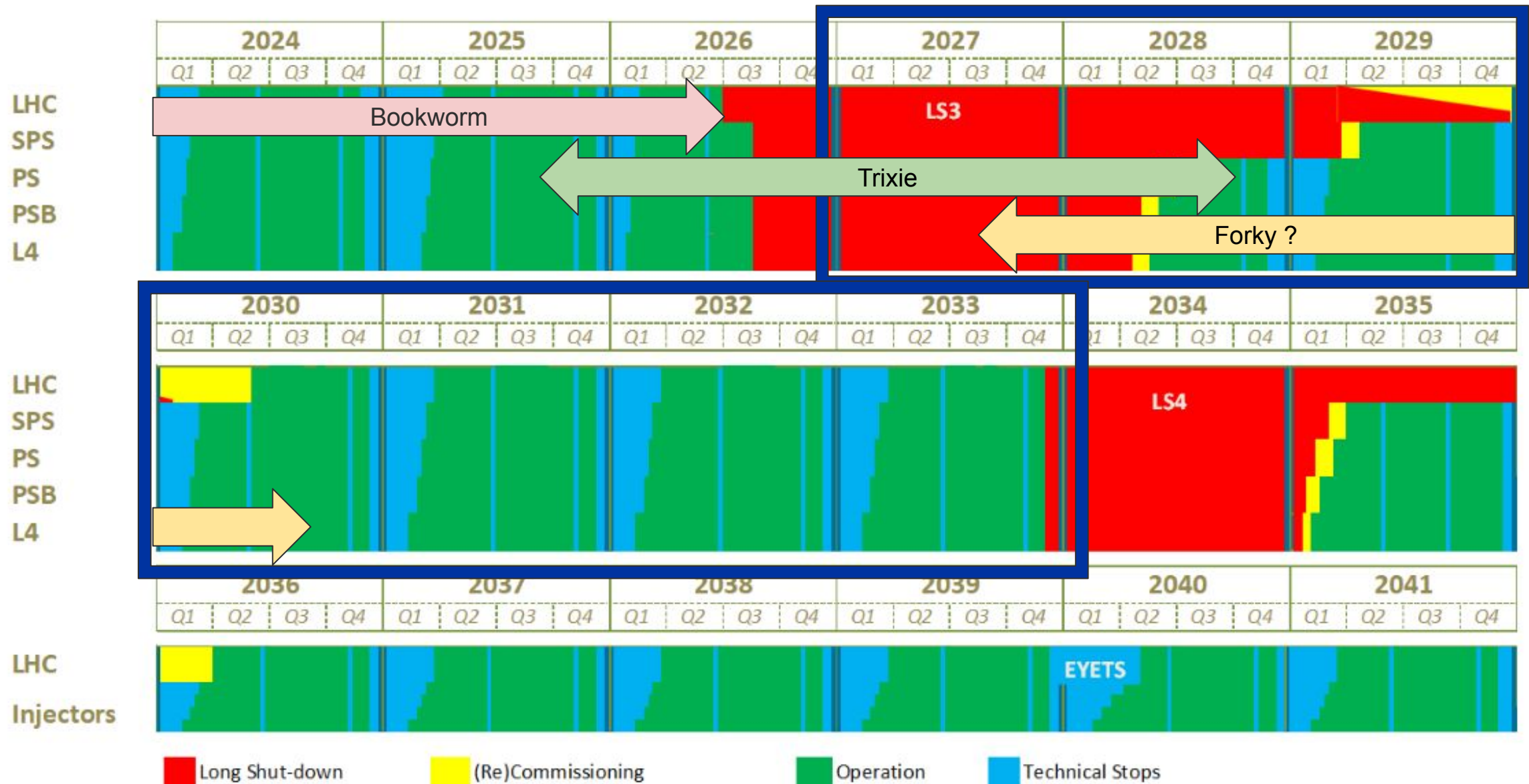


Long Term Schedule for CERN Accelerator complex



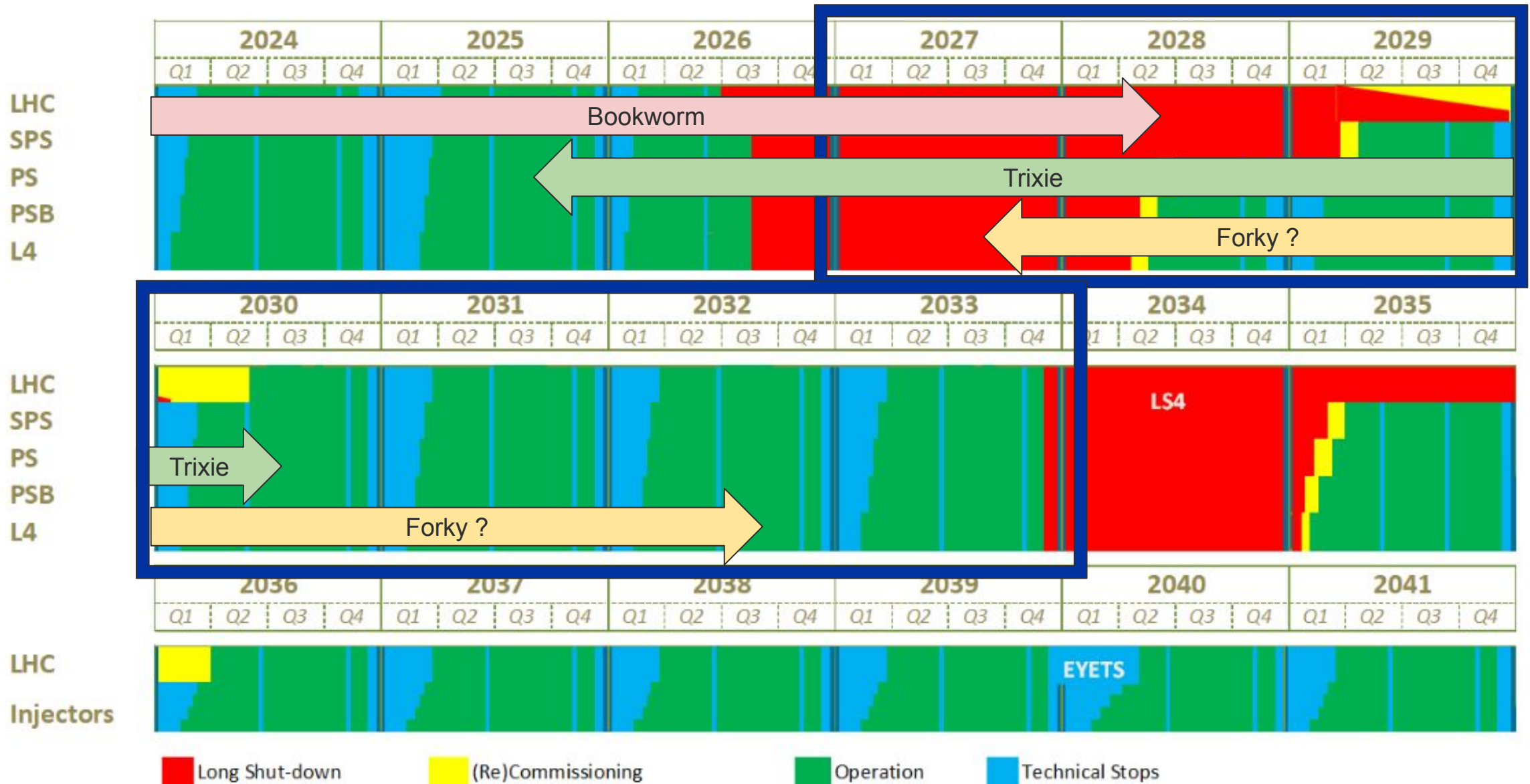
Stable Releases:

Long Term Schedule for CERN Accelerator complex



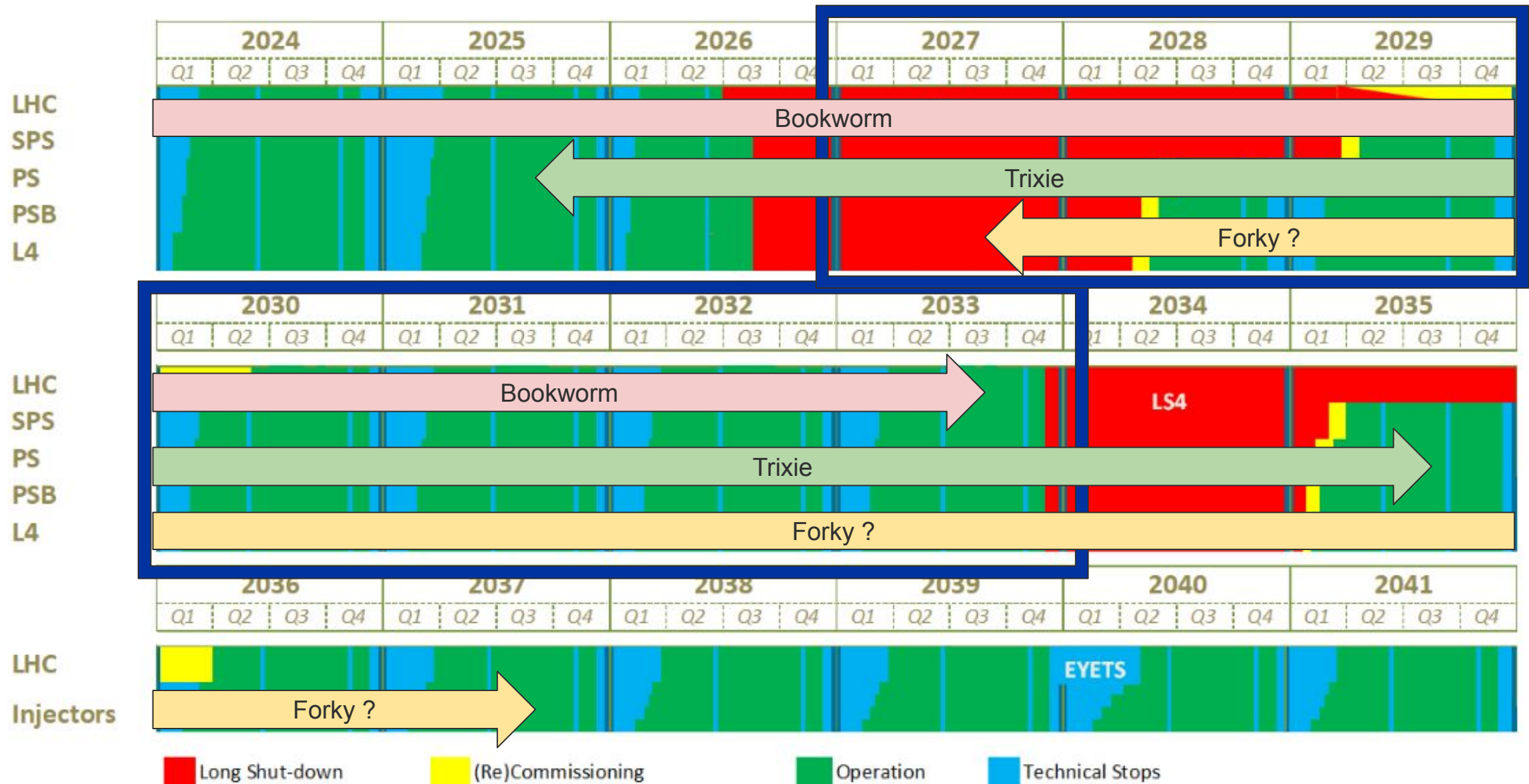
LTS Releases:

Long Term Schedule for CERN Accelerator complex



Freexian ELTS Releases:

Long Term Schedule for CERN Accelerator complex

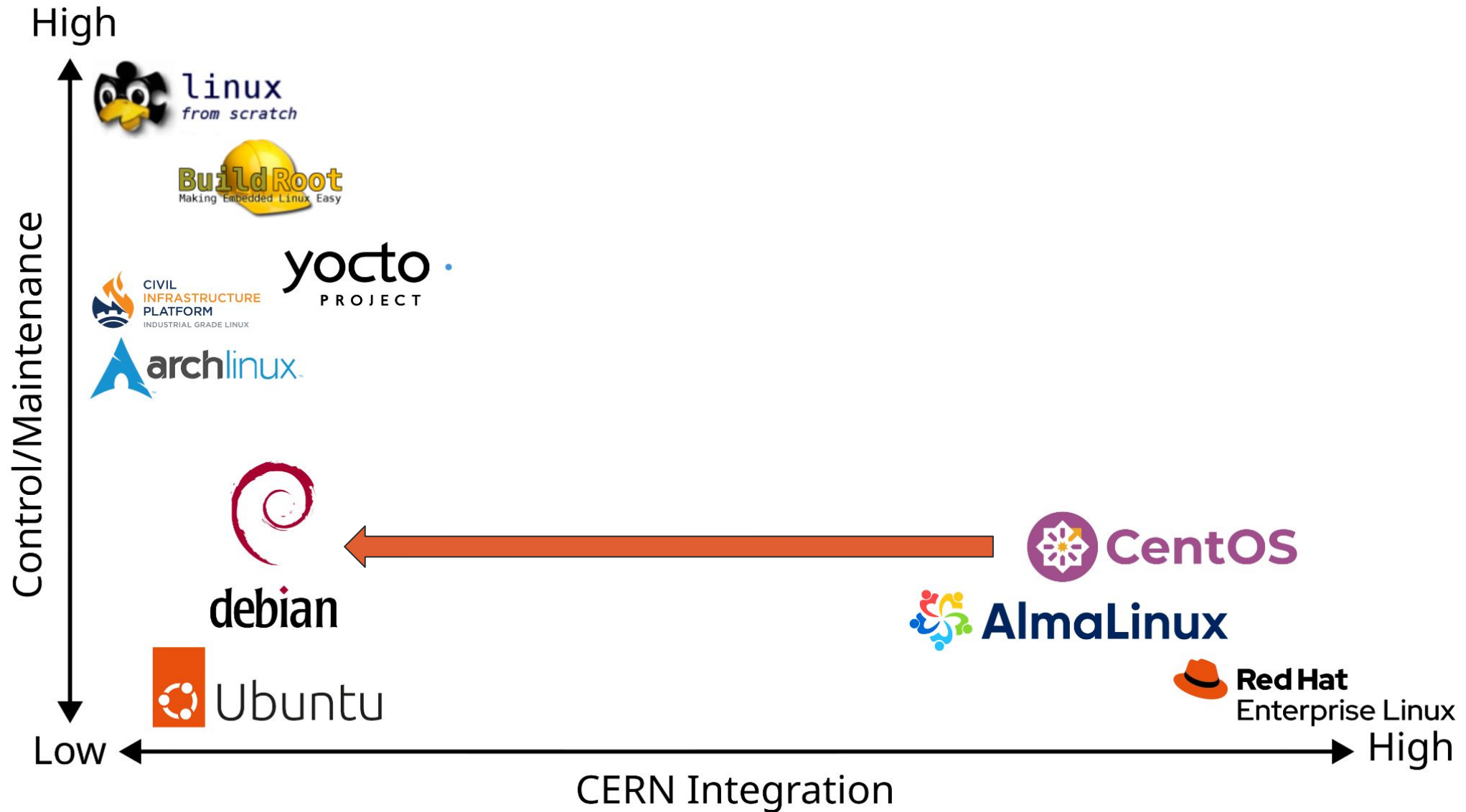


The Accelerator Schedule & Debian Stable

- **Plan A: Bookworm LTS -> Trixie LTS -> Debian 15**
 - Develop on Bookworm up to 2026 (LTS)
 - Deploy Trixie from 2026 up to 2030, (LTS)
 - Upgrade to Debian 15 in 2030
- **Plan B: Bookworm -> Trixie ELTS**
 - Develop on Bookworm
 - Upgrade & use Trixie in 2026 up to 2033, (ELTS)
- ~~**Plan C: Bookworm ELTS**~~
 - ~~Develop on & use Bookworm up to 2033, (ELTS)~~

Debian LTS (and maybe ELTS) are crucial for us. We have started sponsoring Freexian to strengthen the Debian community and ensure its longevity

The Selection Process



Anyone here knows about this Debian thing?

Anyone here knows about this Debian thing?



Sébastien Delafond

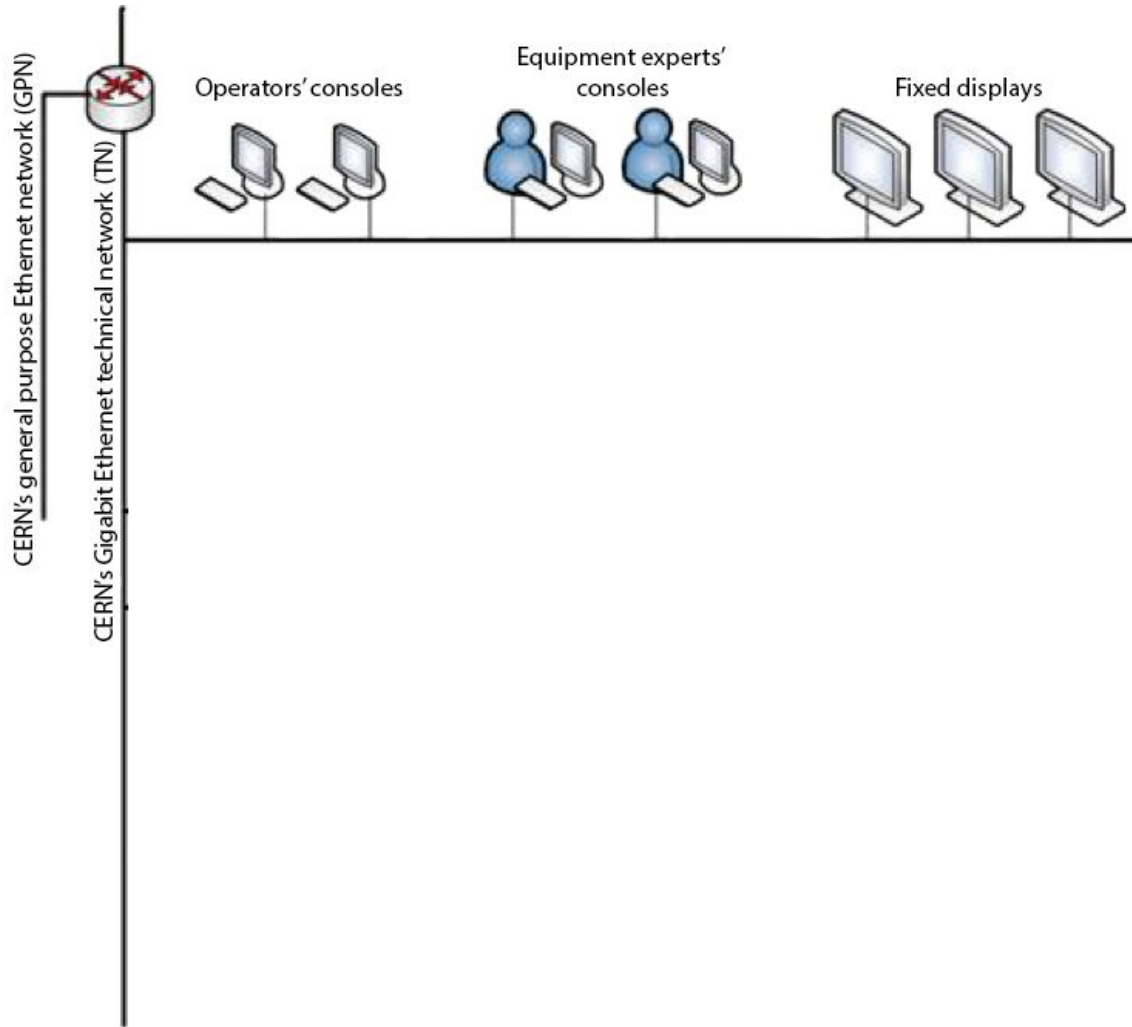
seb@debian.org & seb@freexian.com

CERN Debian training

Infrastructure overview

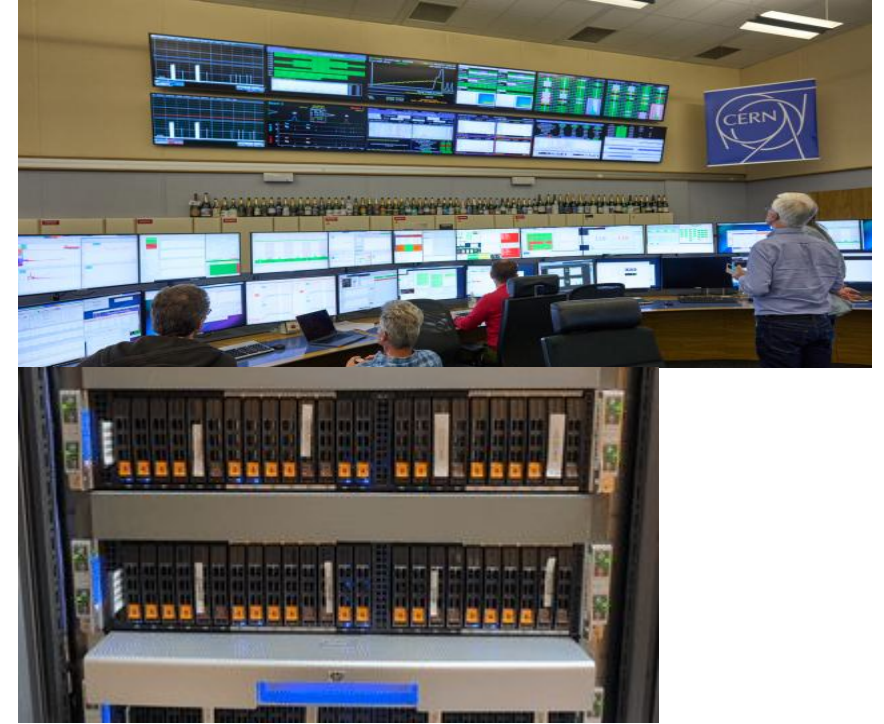
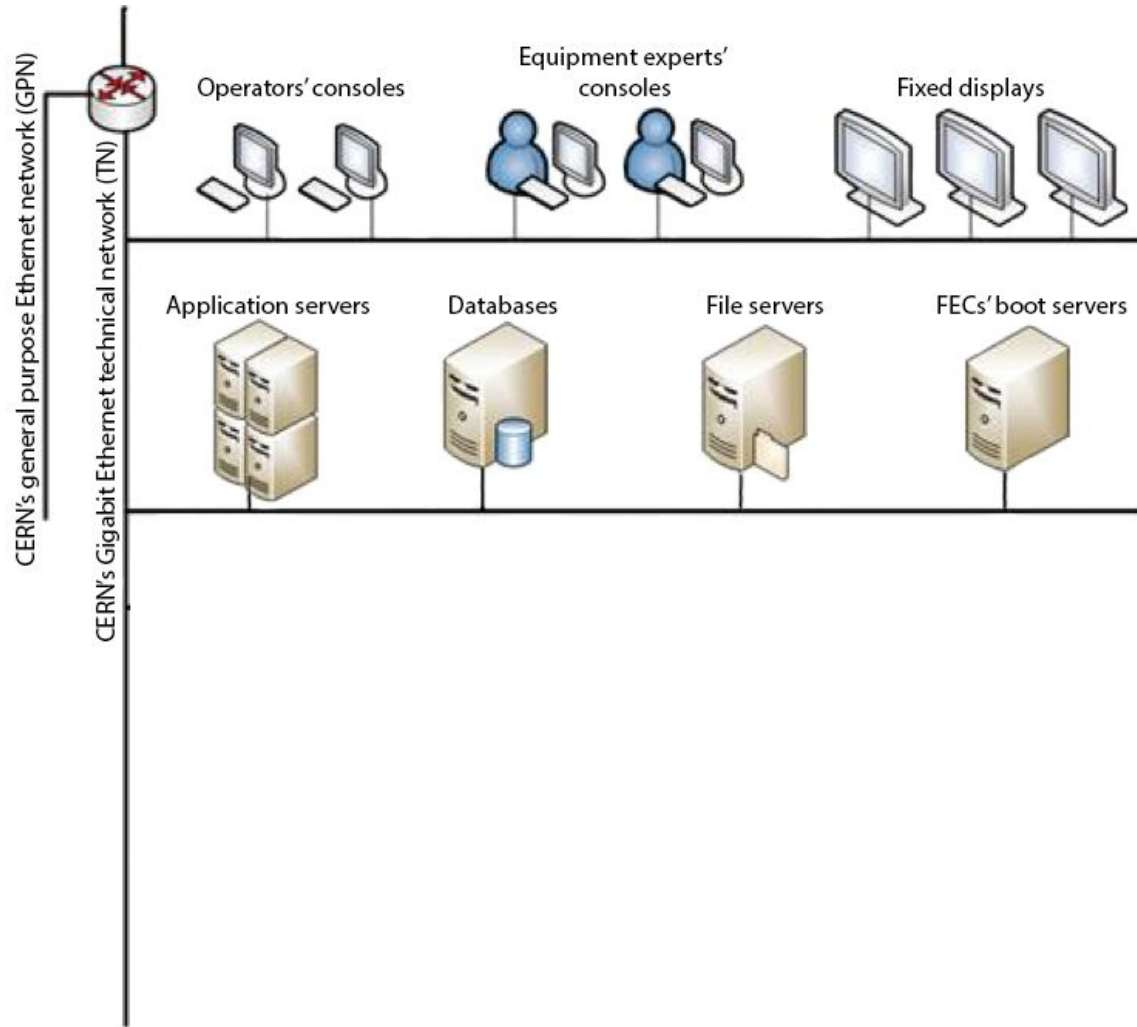
A Multi-tiered infrastructure

Now a multi-distribution infrastructure



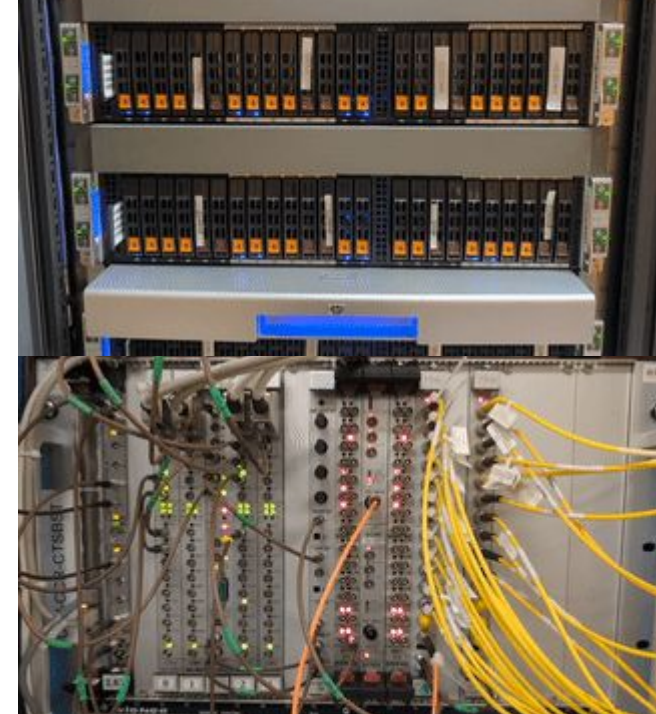
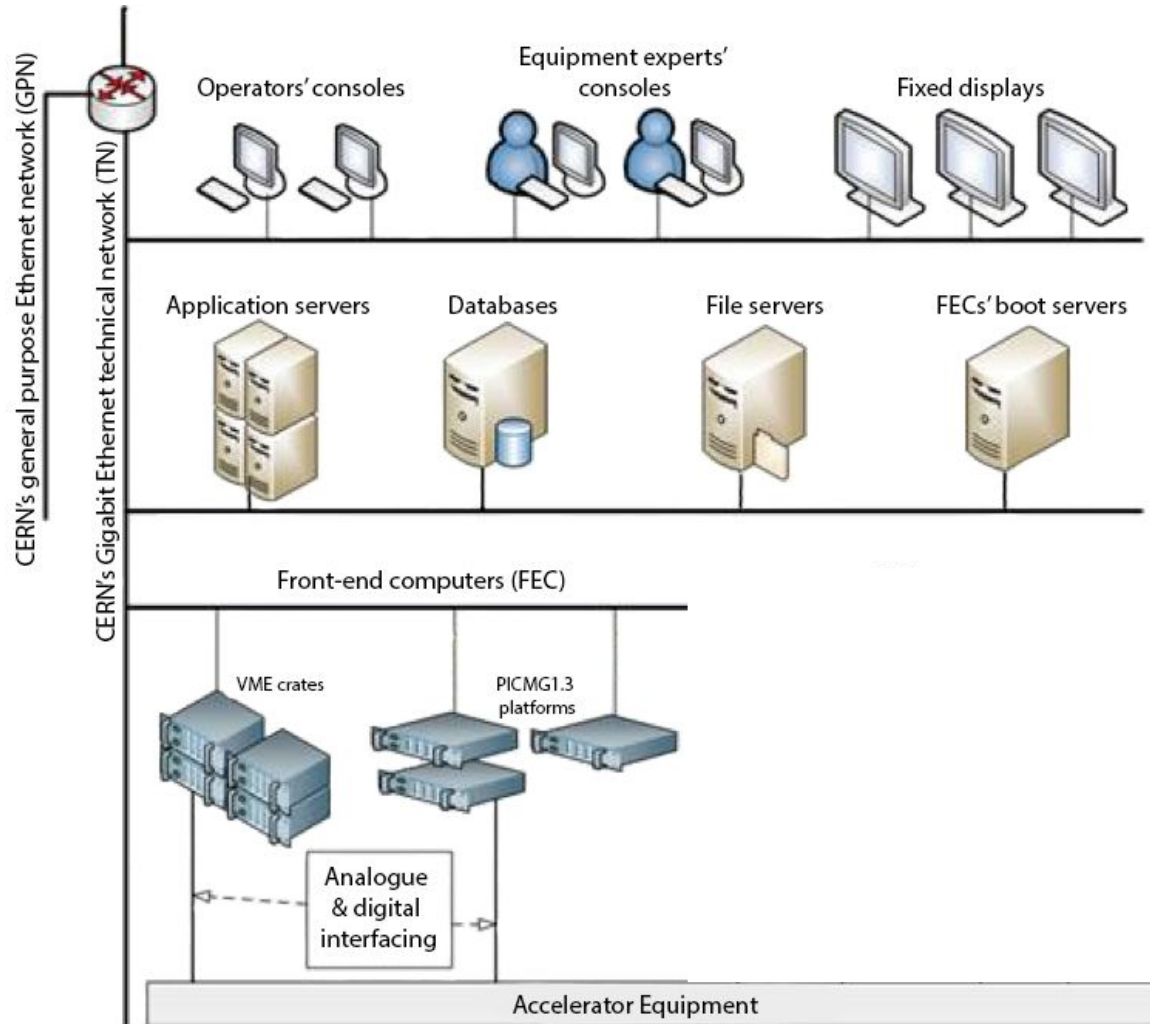
A Multi-tiered infrastructure

Now a multi-distribution infrastructure



A Multi-tiered infrastructure

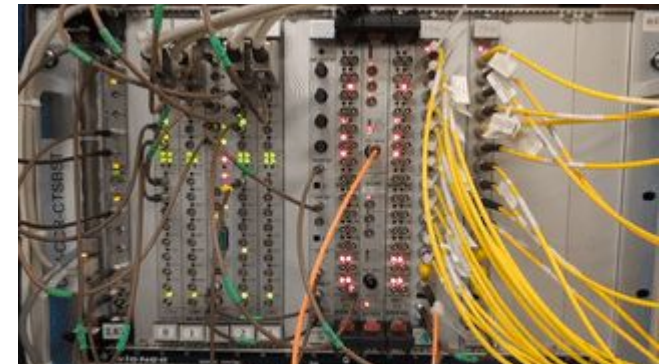
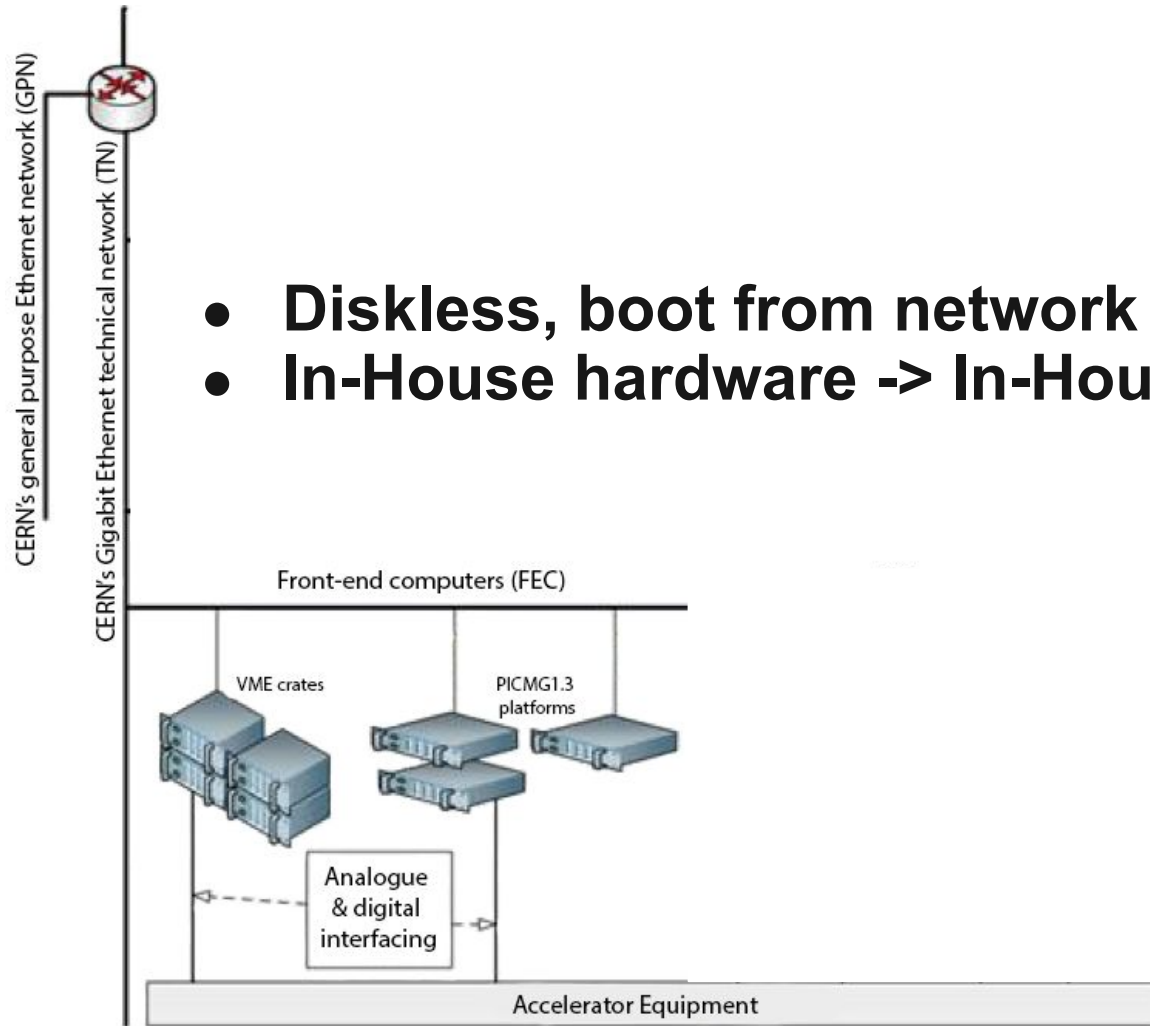
Now a multi-distribution infrastructure



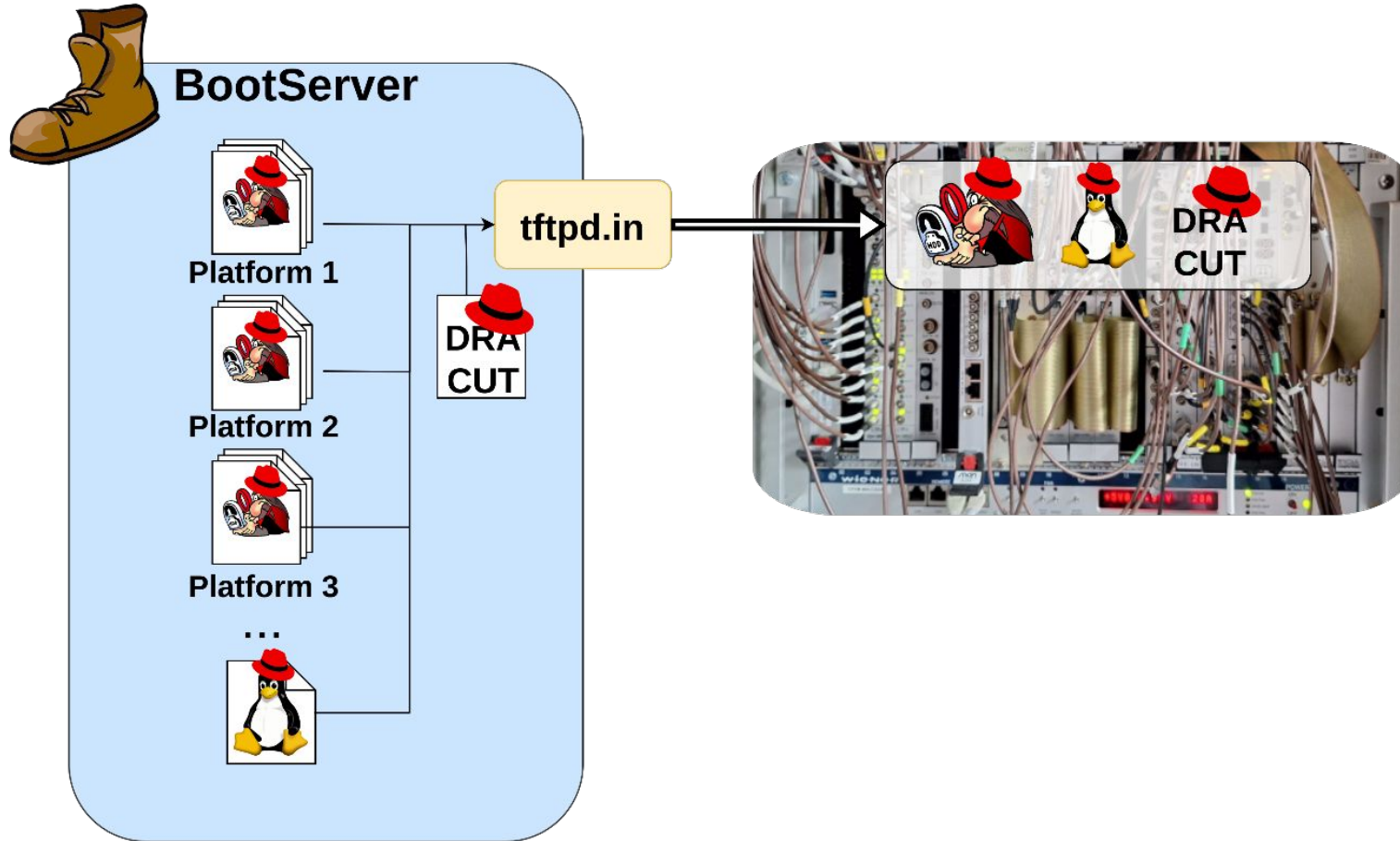
Peculiarities of FECs

The ones we cannot change

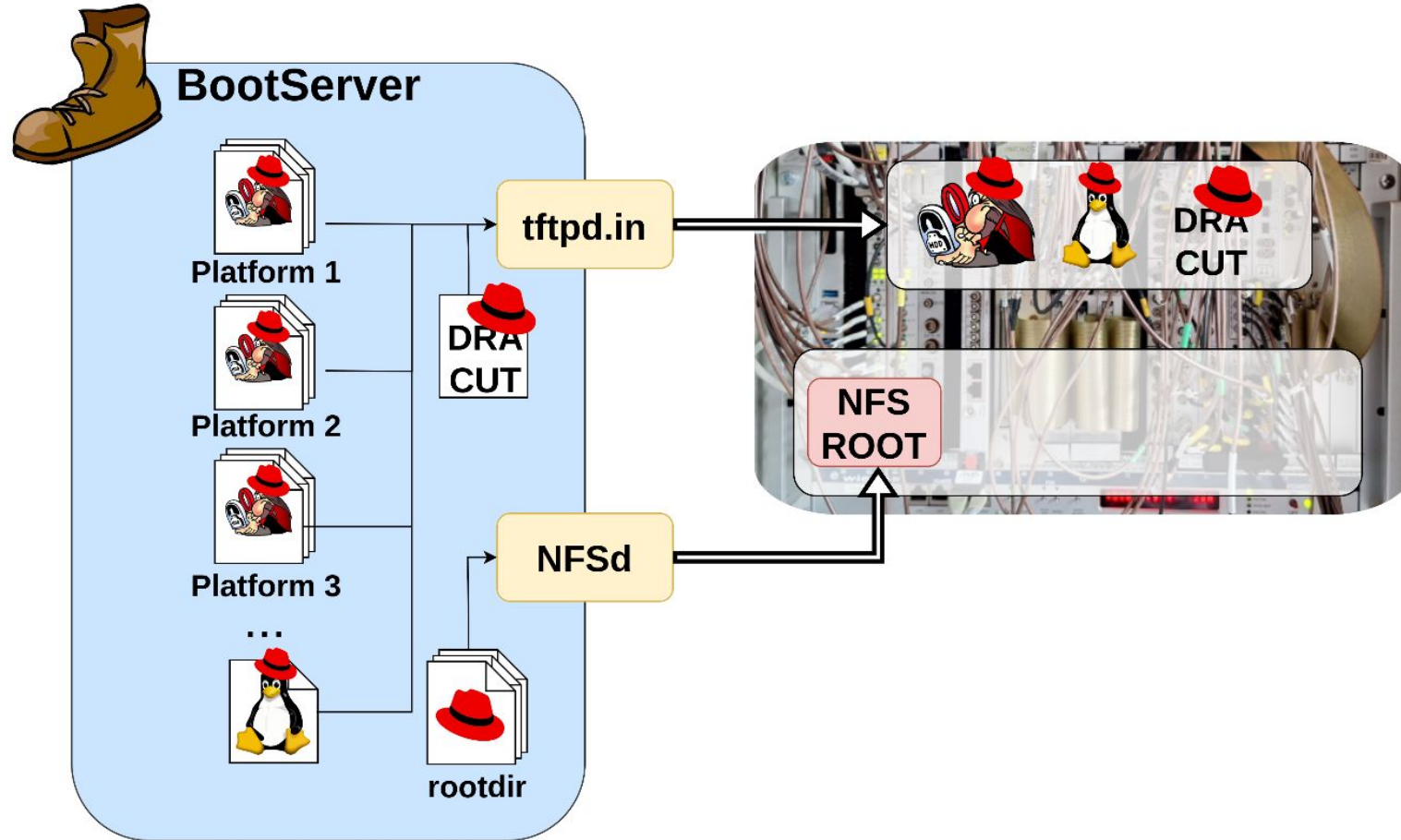
- Diskless, boot from network
- In-House hardware -> In-House kernel drivers



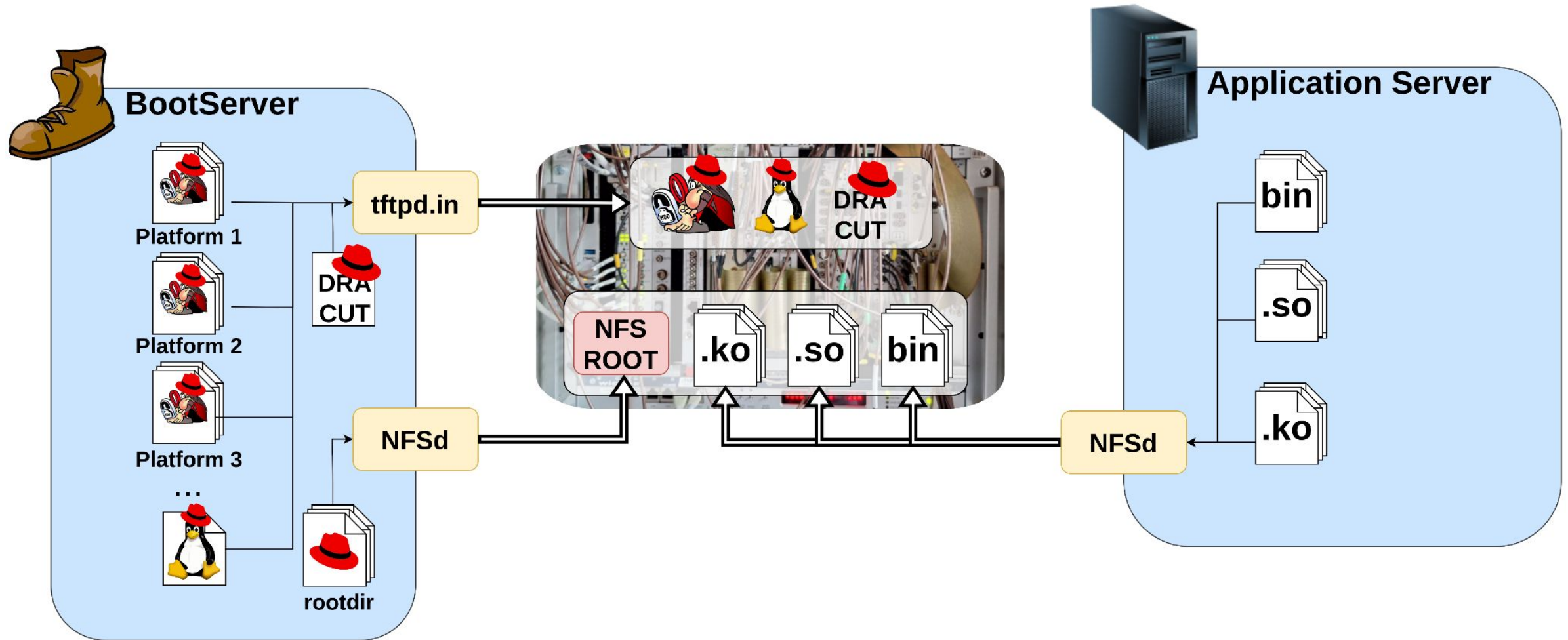
The Legacy Way: Bootserver



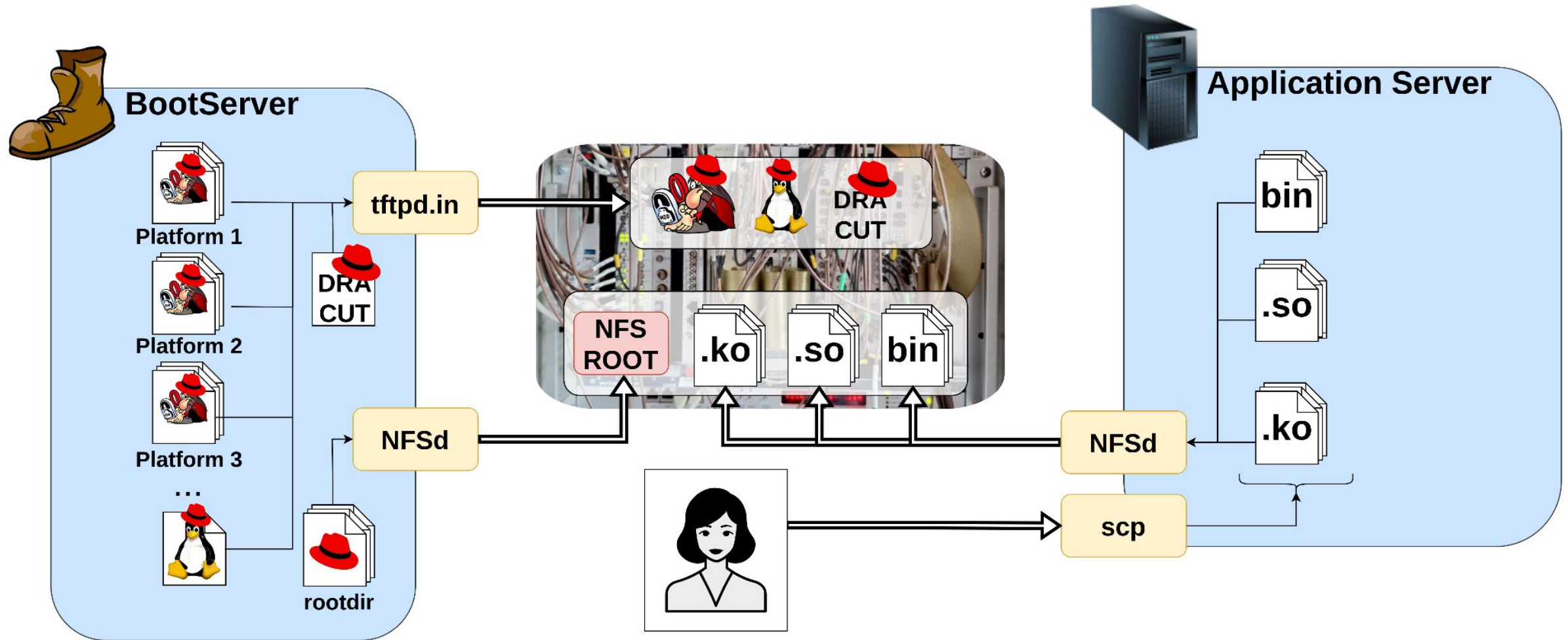
The Legacy Way: Bootserver



The Legacy Way: Application Delivery



The Legacy Way: Application Delivery



Architecture Redesign

Going back to the drawing board, skipping from 2005 to 2026 in one swoop

Building Debian



Table of Contents

[ELBE in a nutshell](#)
[ELBE technical Details](#)
[Downloads](#)
[Documentation](#)
[ELBE Support](#)
[Development](#)
[Imprint](#)
[News](#)

Embedded Linux Build Environment

Why ELBE was born

ELBE principles

Start Using ELBE

ELBEs usage of QEMU

Secure

ELBE reuses the security of Debian. An update-checker informs if security updates are available.

Reproducible Images

The Rootfilesystem is described in a single XML file.

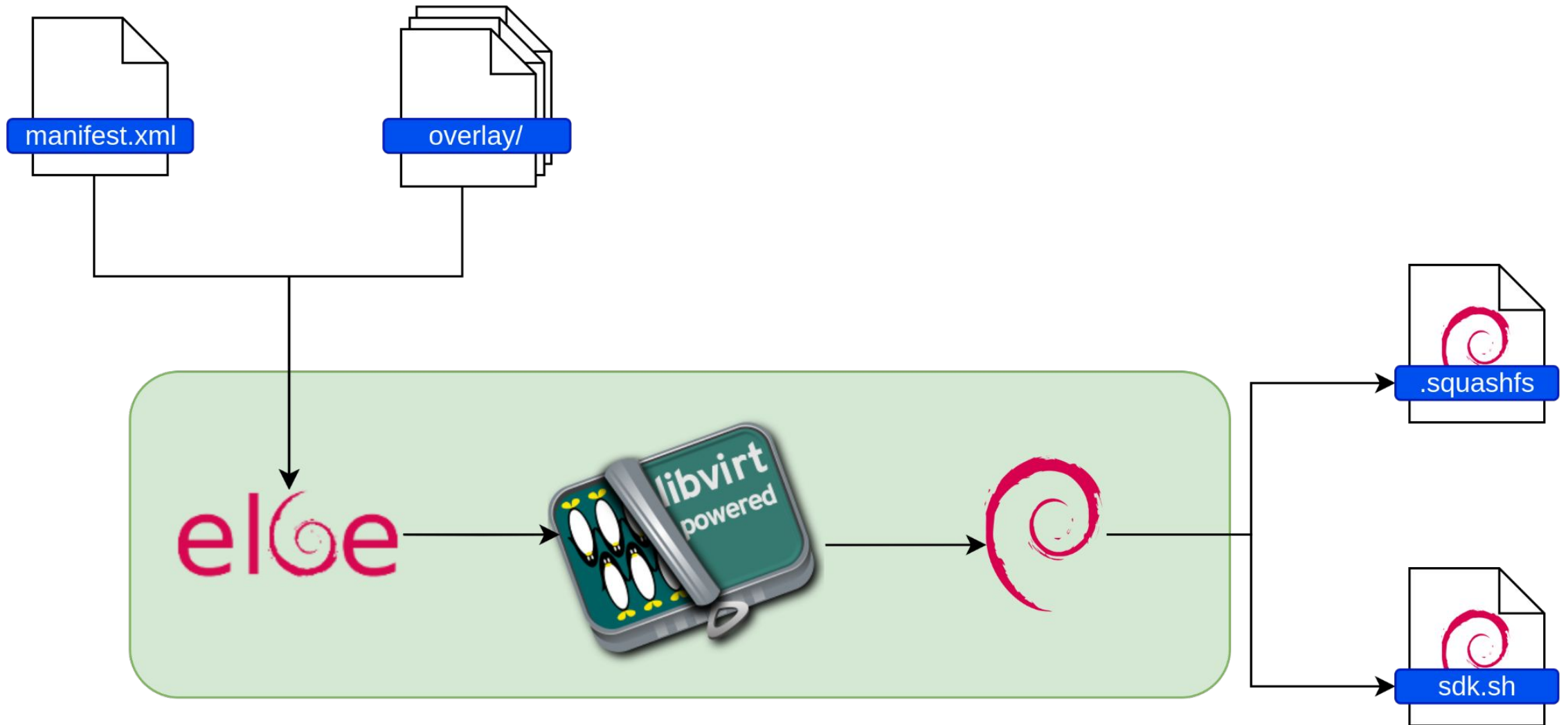
Based on Debian

No need to build all binaries. ELBE reuses Debian packages!

ELBE ADK

A native Application Development Kit is included

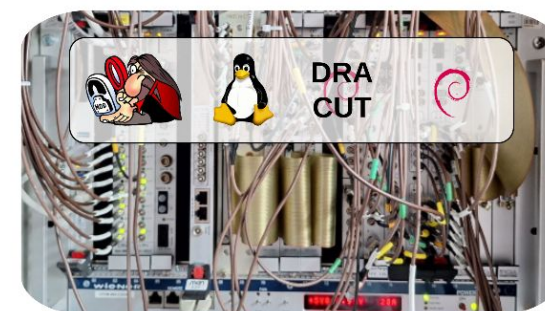
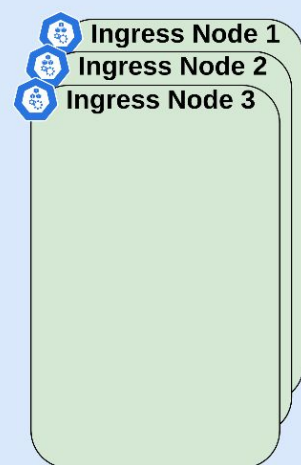
Building Debian

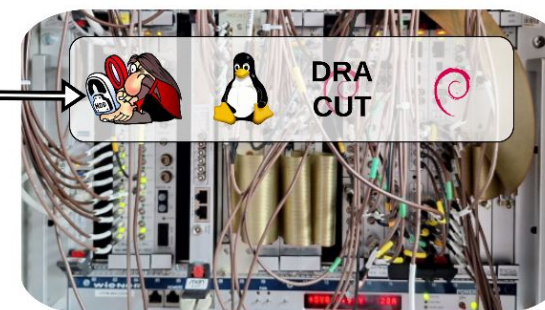
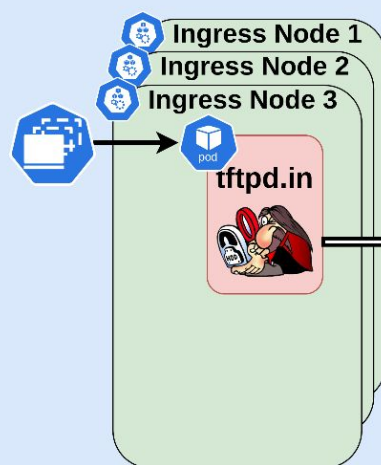


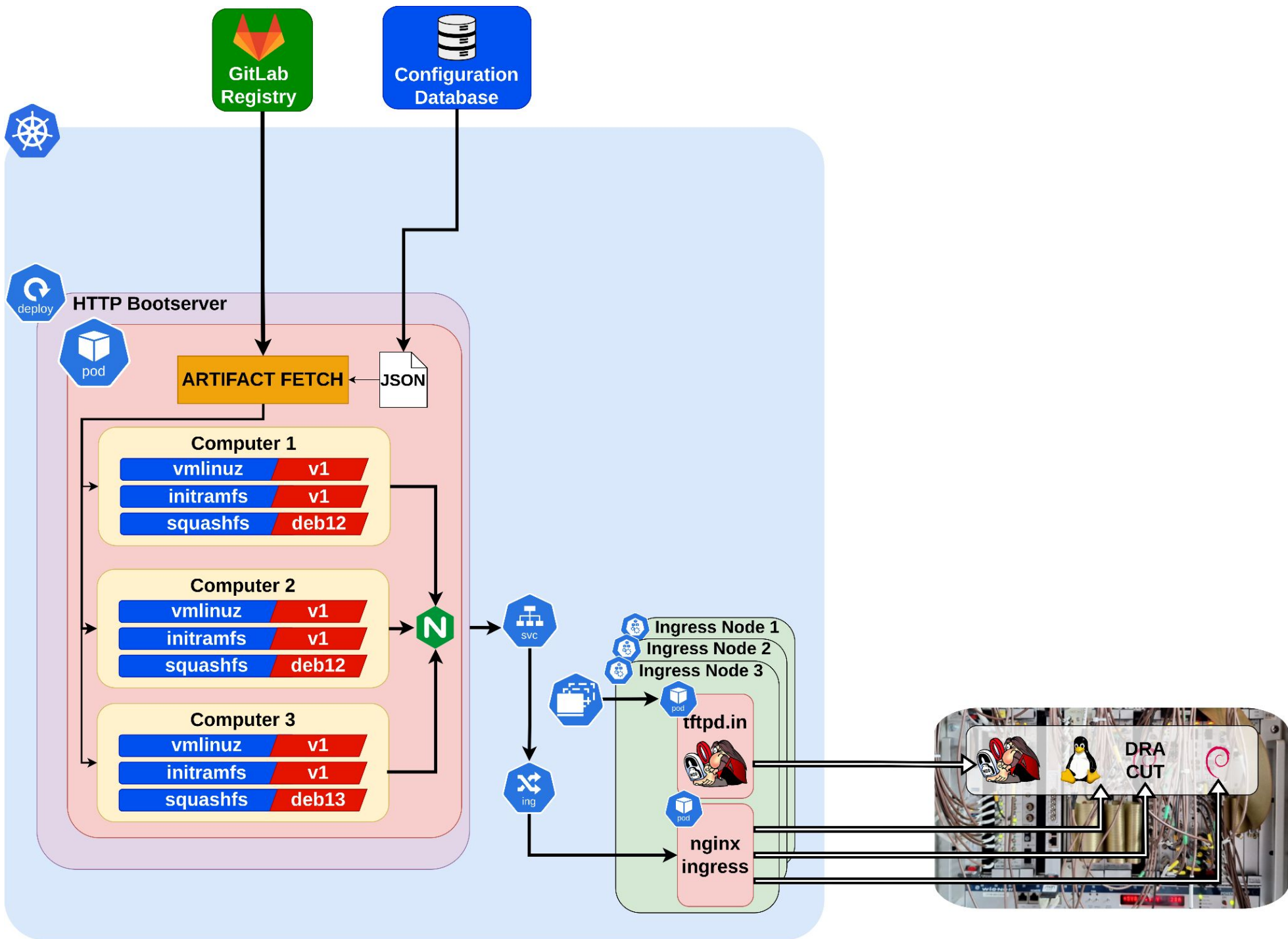


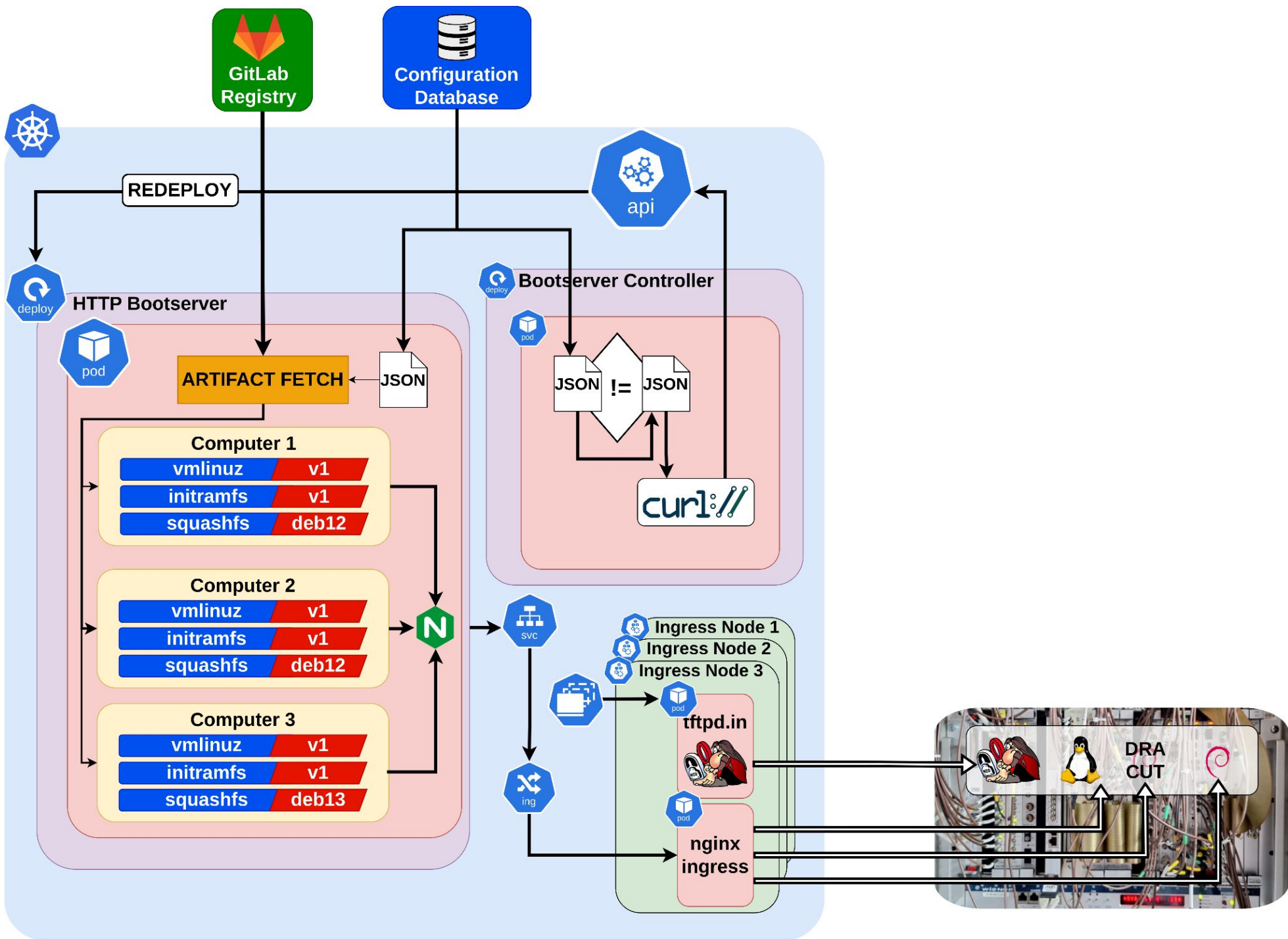
Becoming Cloud Native

From bare metal servers to an inhouse cloud native infrastructure





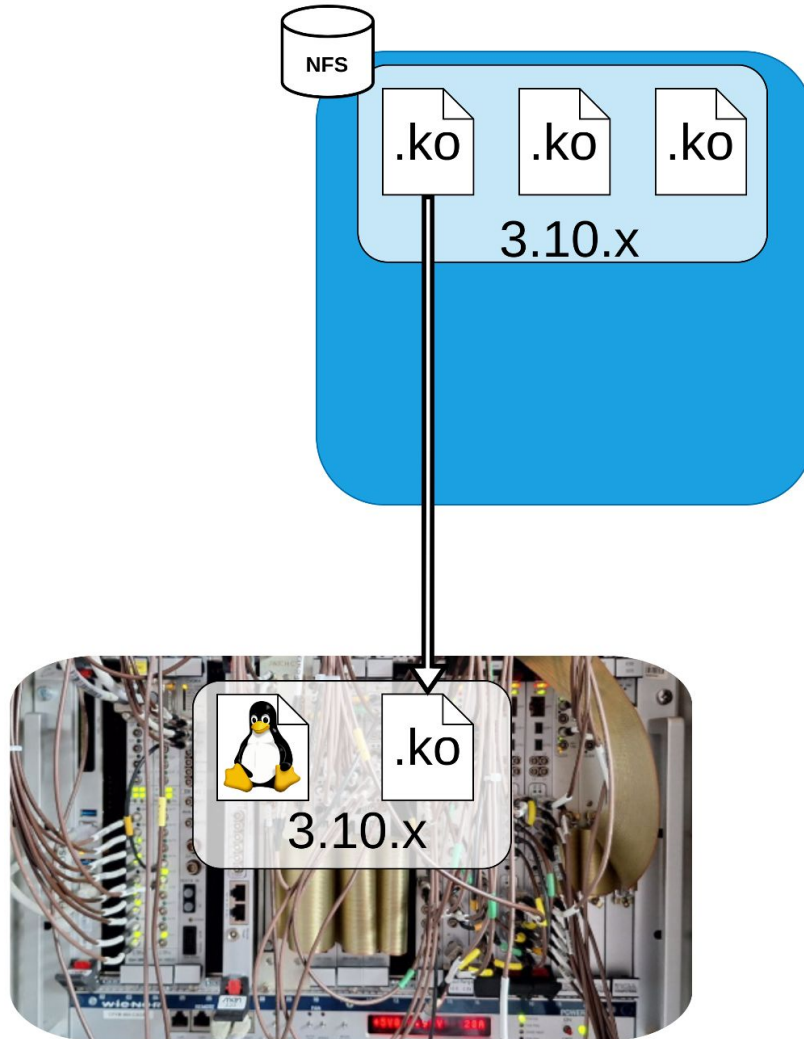




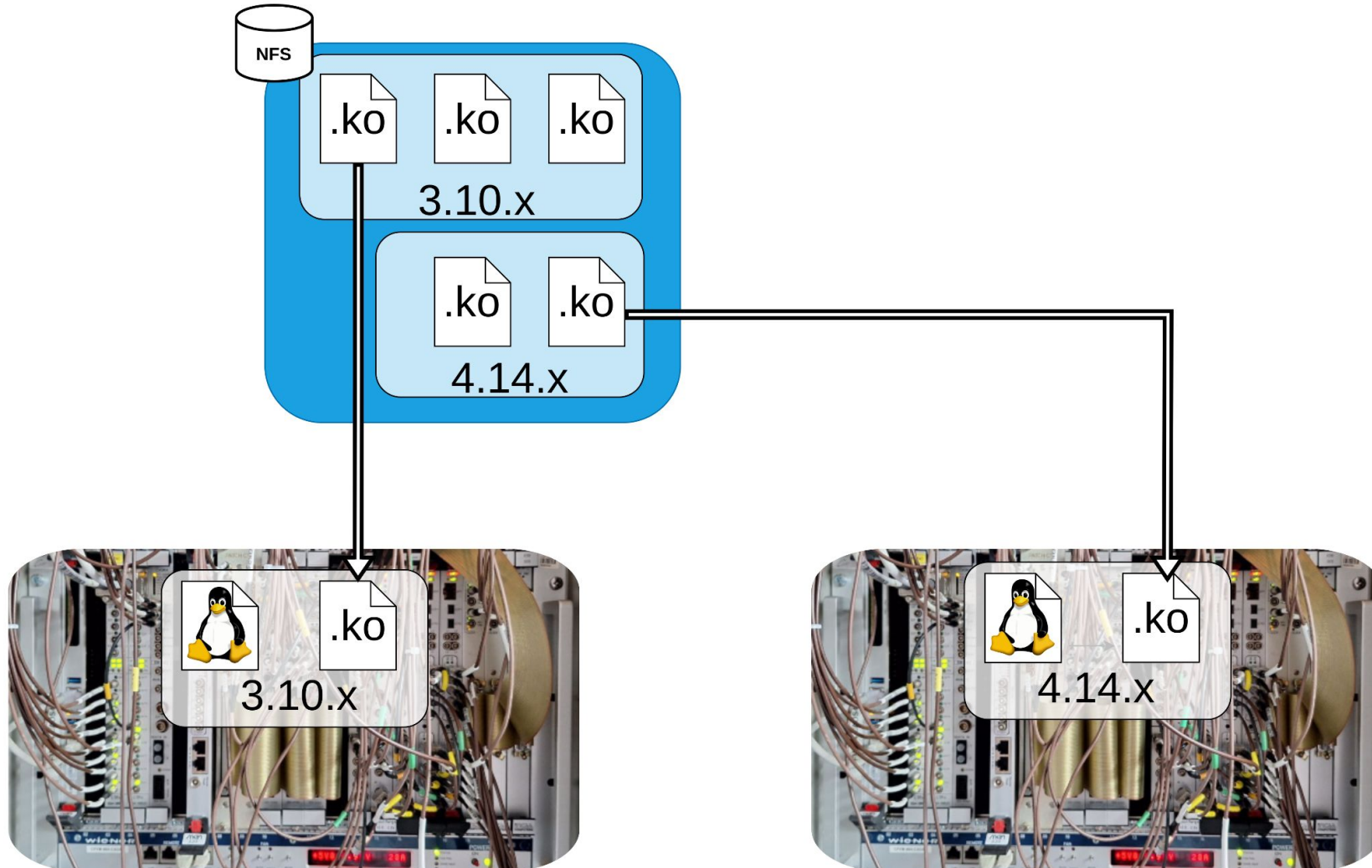
We have a working operating system...

... but an operating system doesn't do much without applications

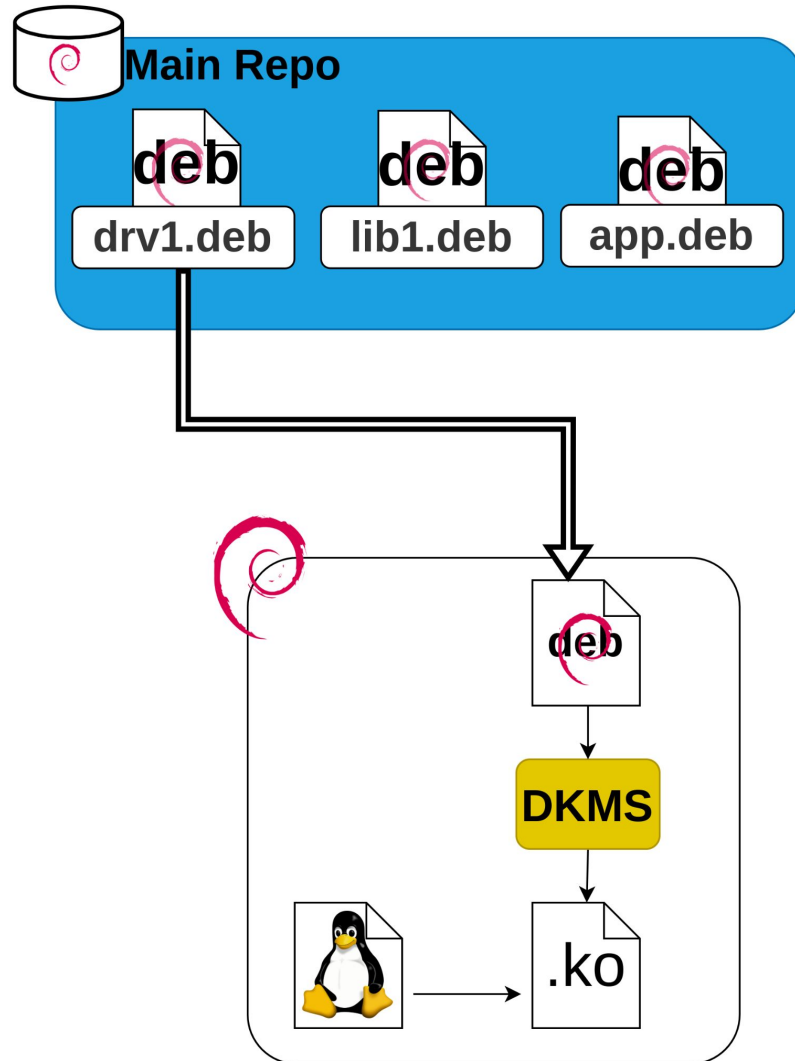
Device Drivers: Old CERN Architecture



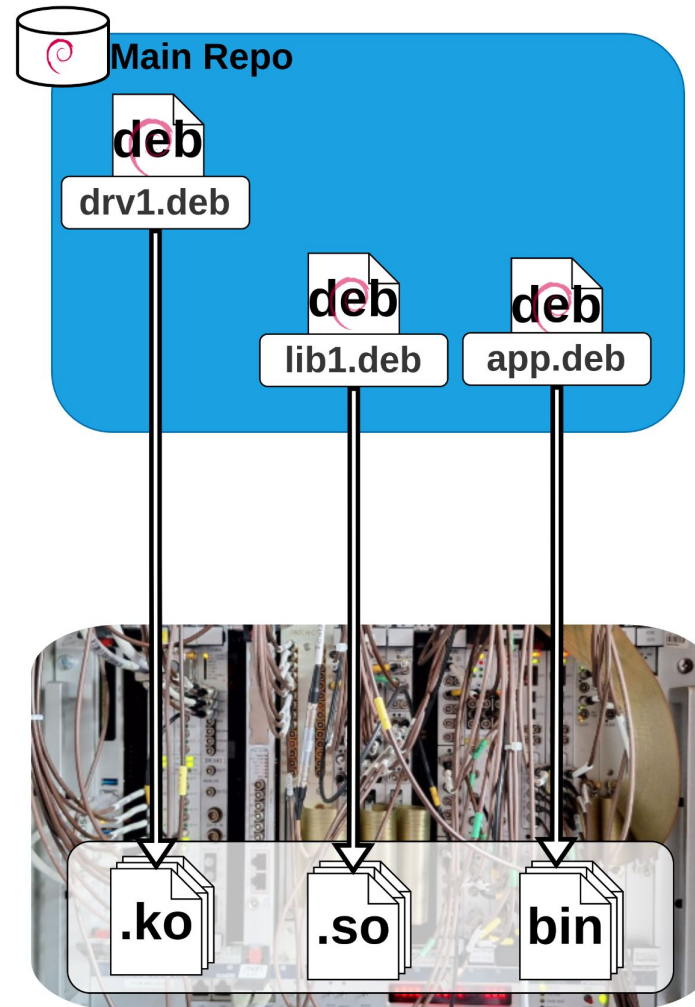
Device Drivers: Old CERN Architecture



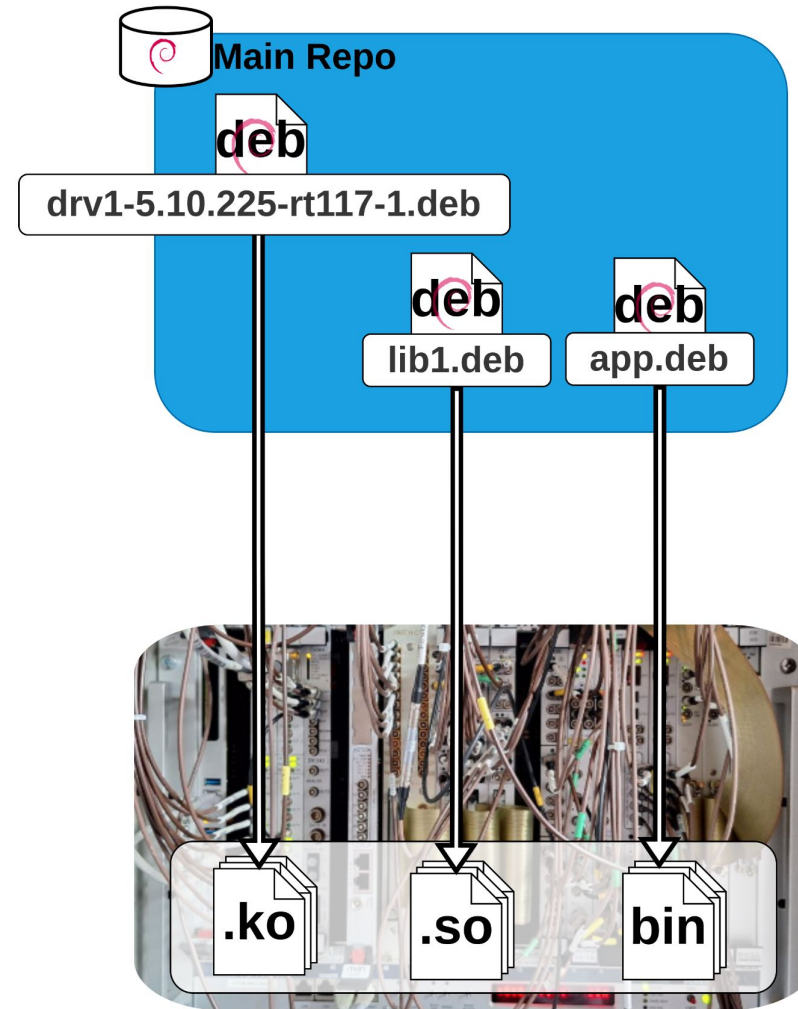
Device Drivers: Orthodox Debian



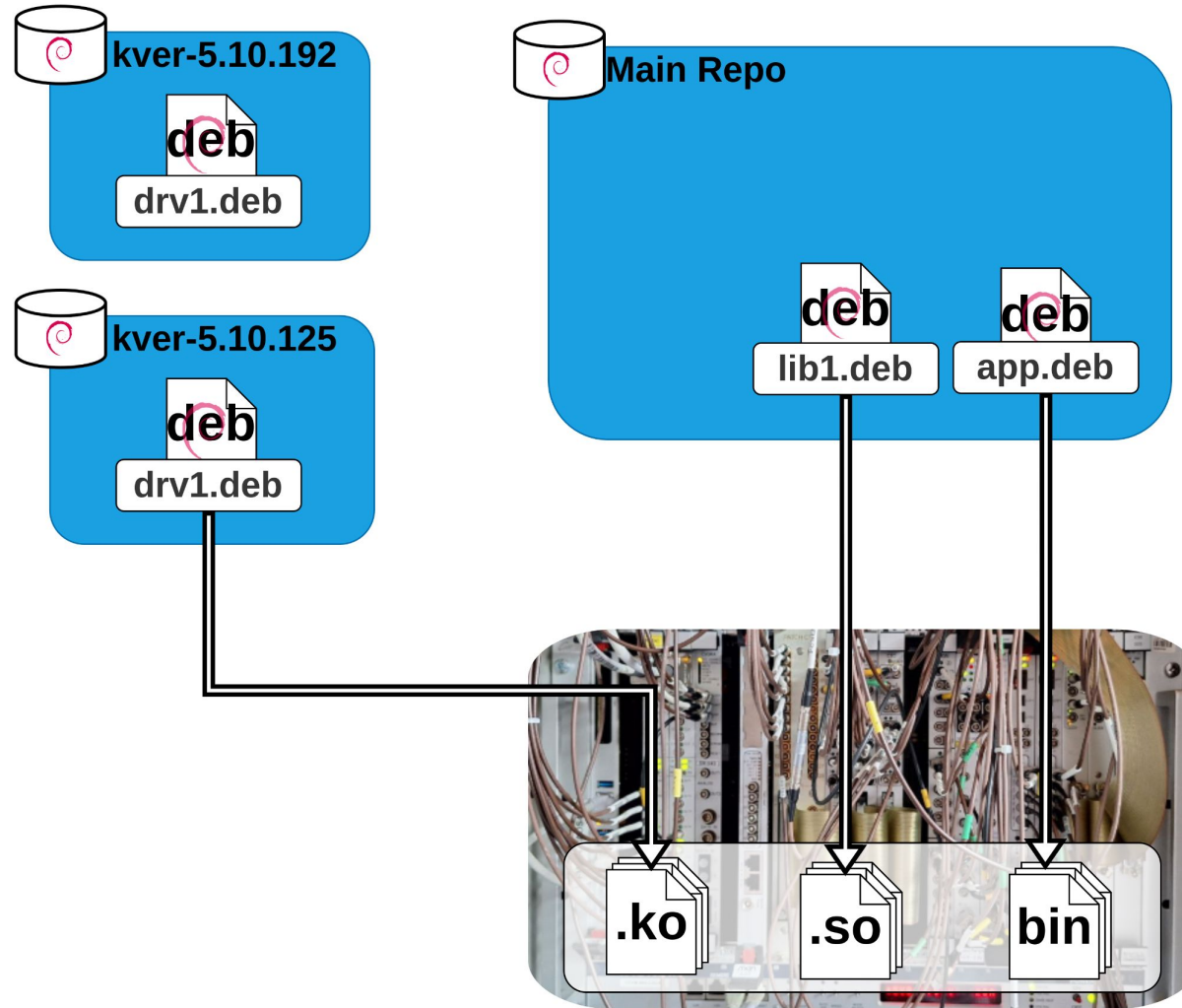
Device Drivers: Redesign



Device Drivers: Redesign



Device Drivers: Redesign



Conclusion(s)

Appreciations

The things we are grateful for

- To Debian's dedication in keeping the distribution working on older and less known architectures, a real “Universal Operating System”.
- To the Debian community that has managed to make a community-led distribution that can rival Red Hat.

Challenges, Uncertainties and Needed Adaptations

What we are still working on, and some we are struggling with

- **No standard tooling for automated building & publishing of packages.**
 - Official upstream tools builddd, dak etc are very specialized.
 - There is no upstream policy for binary kernel modules so we had to invent our own.
 - We are looking forward to the maturity of debusine.
- **A lot of tools not supporting multiple versions of the same package.**
 - Limits choice, makes managing repositories more difficult (e.g. not with reprepro).
 - Makes it necessary to have different repositories for each revision/patch.
- **We evaluated Debusine, OBS and other options for package distribution**
 - Limitations made both of them unsuitable for our use-case.
 - Developed a custom Koji plugin to add debian package support.
 - Reusing the existing CERN RPM building infrastructure.
 - Thanks to Erick Andrei Vilcica for hammering-in debian support in koji.

Final Takeaways

- Do not assume that a system that works with version X of a distribution will work on version X+1.
- Distribution portability is a very good thing.

There are dozens operational Debian machines are currently active in the accelerator complex.

By the time end of 2026 all 2200+ Industrial computers & embedded systems around CERN will be running Debian 13.



home.cern