Selection Of Linux Distribution For Front-End Computers and System-on-Chip in CERN-ATS

Federico Vaga

2023-05-24

Who am I?

- **I work in CERN BE-CEM-EDL**
- **I'm a Software Engineer, specialised in:**
  - Low-Level Software
  - Linux kernel development
  - Linux systems

Disclaimer

This is an on-going project

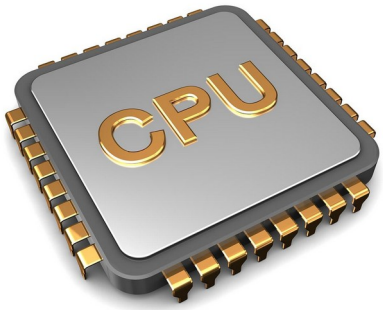
From Last SoC Working Group Meeting

How Does The Next OS Boot?

DRA
CUT

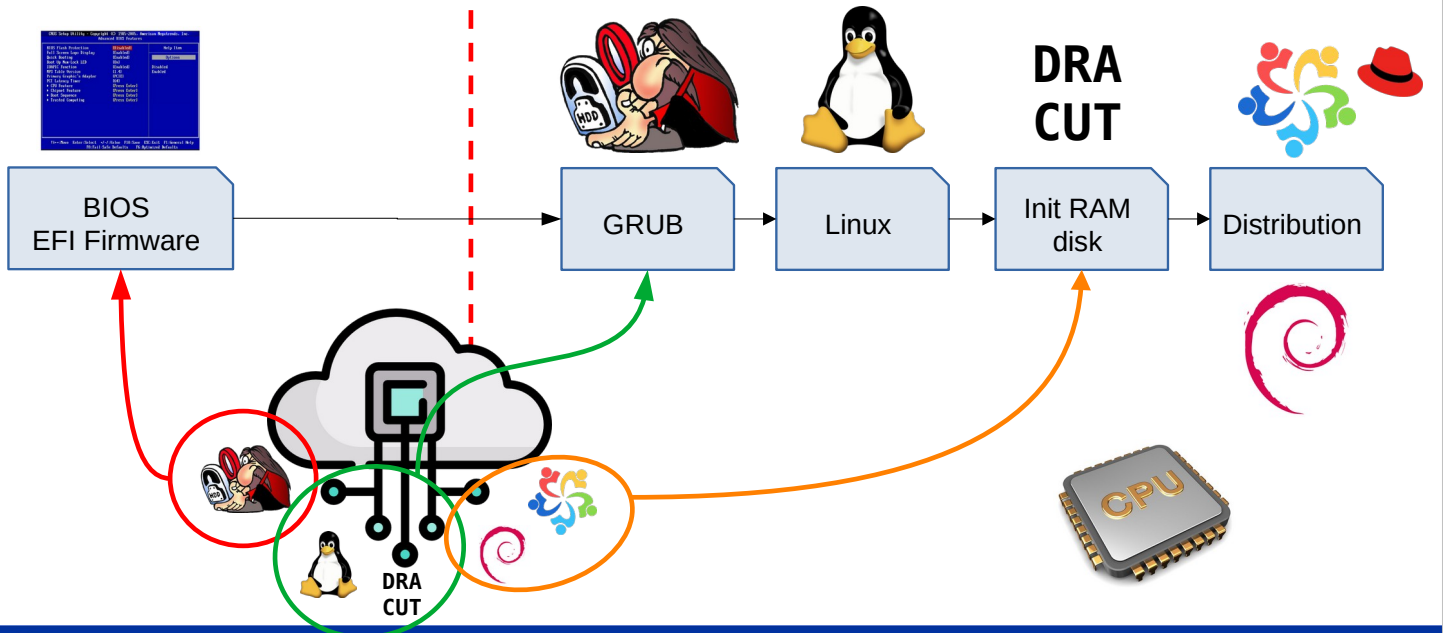
GRUB → Linux → Init RAM disk → Distribution

www.gnu.org/software/grub
www.kernel.org
github.com/dracutdevs/dracut
almalinux.org
www.debian.org



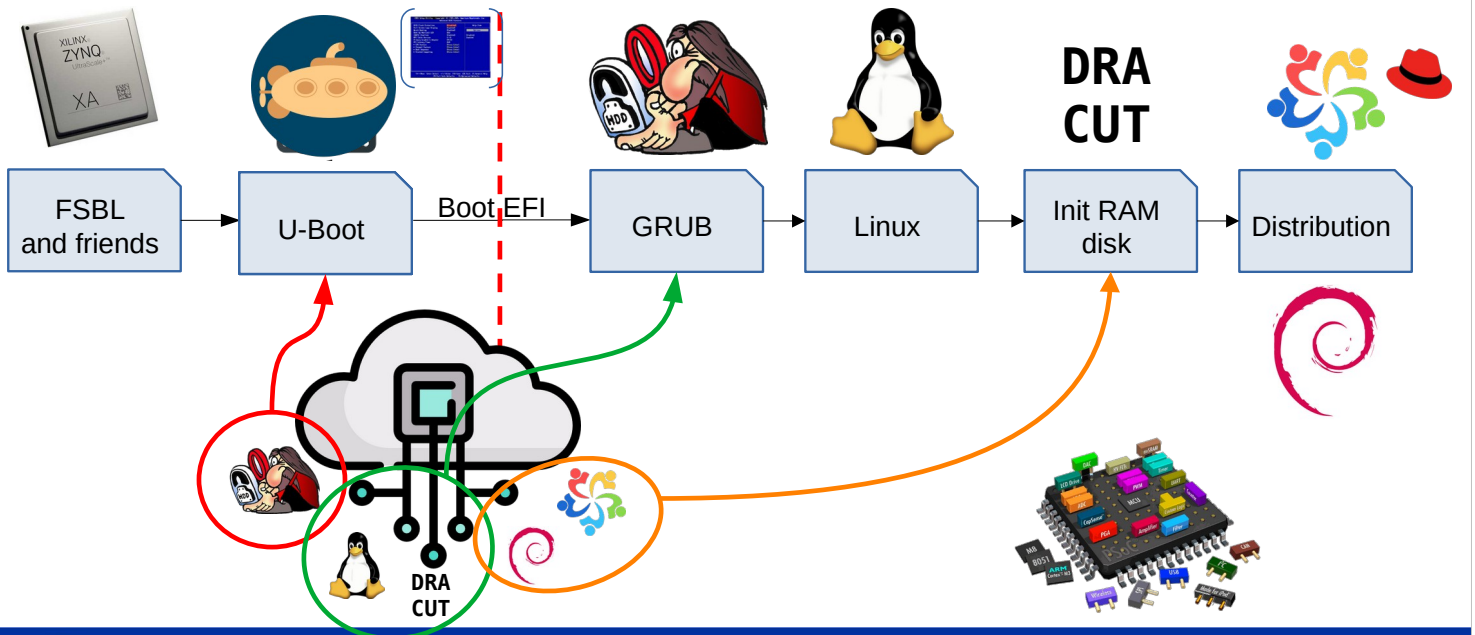
This is the booting process for the next Operating System for FEC and SoC-based systems in ATS.

Network RAM Booting: Industrial PC



On an Industrial PC (IPC), the UEFI/BIOS will download and execute grub, and the booting process follows.

Network RAM Booting: System-on-Chip

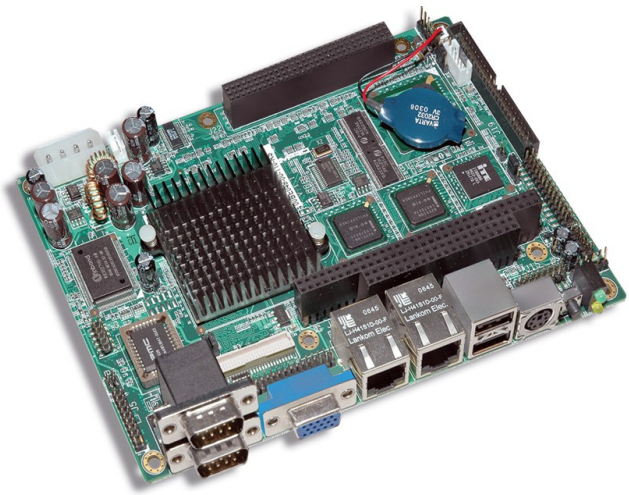


On an SoC-based systems we want the same booting process as on IPCs.

In this case a bootloader (u-boot/barebox/coreboot) can play the role of UEFI/BIOS. Hence, it will download and execute grub, and the booting process follows.

Back On Topic: OS Selection Process

Embedded Systems



yocto PROJECT

By looking at them we might immediately say: “this is an embedded system and I will put Yocto on it”.

Definition

embedded system

A combination of computer hardware and software, and perhaps additional mechanical or other parts, designed to perform a dedicated function. In some cases, embedded systems are part of a larger system or product.

A Front End Computer (FEC) is usually an industrial PC (IPC) that connects equipment to the technical network (TN).

A FEC is made of a CPU on a Single Board Computer (SBC), plus a number of cards used as interface to the equipment.

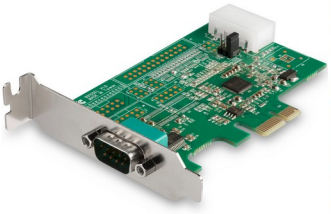
A FEC can be used to control private networks like FIP, or White-Rabbit. Over these private networks there are nodes in charge of controlling other equipment.

System-on-Chip are getting more popular and in some cases they want to be considered as a FEC, and being able to be connected on the TN.

Equipment is controlled by software running on the FEC. Thus, the need for an OS to manage the applications.

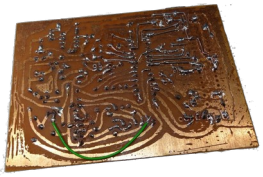
Embedded Systems ?

CAN Bus

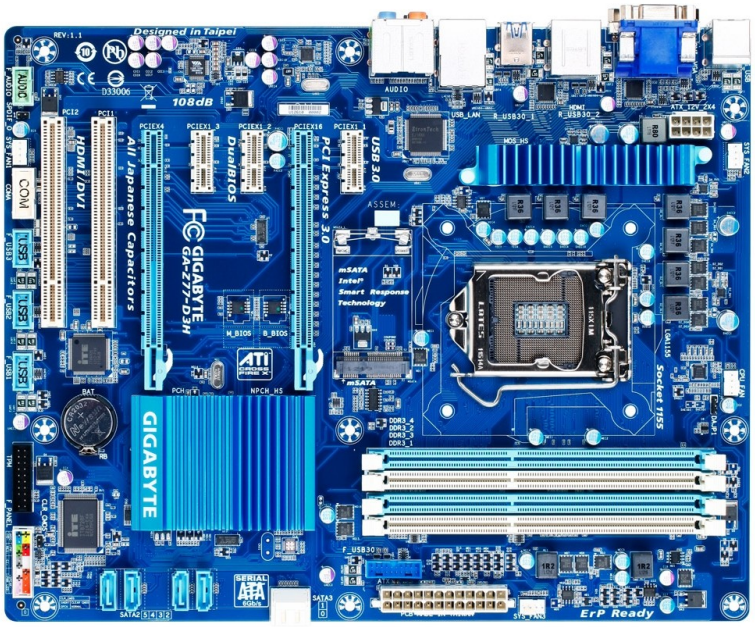


Serial Bus

Digital IO



Custom Card

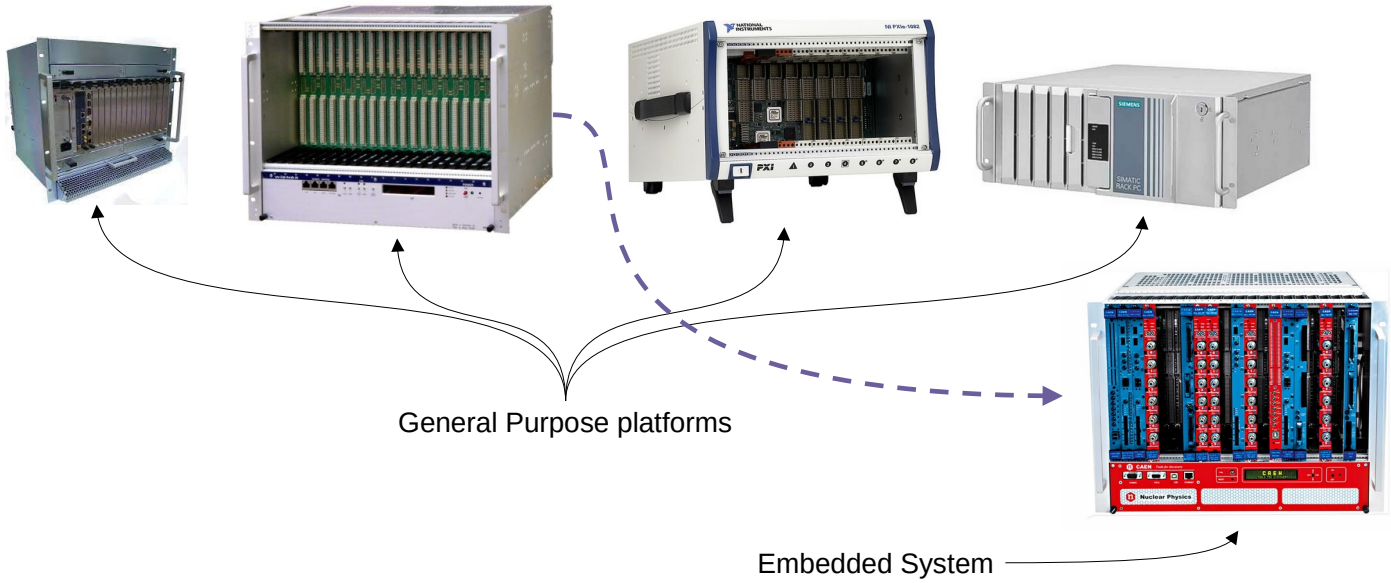


On the other hand, when we look at a motherboard ATX form factor, we might immediately say: “this is a desktop PC, I will install Ubuntu”.

But what if we fill this desktop with I/O cards, field bus controllers, and some custom cards. Then, we place this next to a piece of equipment that we want to driver. Isn’t this an embedded system?

Would you still use Ubuntu? Or you might consider yocto?

Embedded Systems ⁉



Perhaps it is easy to get fooled by the system modularity (e.g. VME in the picture) and think that because it is modular it is general purpose.

It is a general purpose platform, but once it is configured it can't be used as a general purpose machine. Instead, it will be used for one or few purposes "forever".

Then, when we speak about embedded systems we usually want more "control" on what runs, compiler flags, hardware setup/features/support. And for this reason we tend to select tools like yocto or buildroot. They give us the control we need over the system.

The Next Operating System

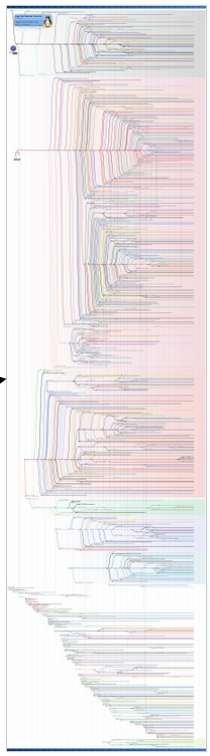
For FEC and SoC

Linux Distributions Panorama

- **There are hundreds of them.**
- **On top of Linux, they distribute open-source software**
  - Technically speaking any Linux distribution would do
  - Each software follows its own governance and development life-cycle
  - May add patches
- **Development could be driven by:**
  - a company,
  - a community.
- **Governance could be driven by:**
  - a company,
  - a community.
- **Each has a mission for its own existence.**

DistroWatch.com
Put the fun back into computing. Use Linux, BSD.

Wikimedia picture



Does this matter? YES!

The slide is self-explanatory

It Matters Because It Affects:

- governance priorities;
- community or company size;
- software availability, version, building, and configuration;
- supported architectures;
- the development life-cycle;
- the upgrade policy;
- the delivered support level;
- its maintenance;
- its long-term existence.

The slide is self-explanatory

A Few Examples

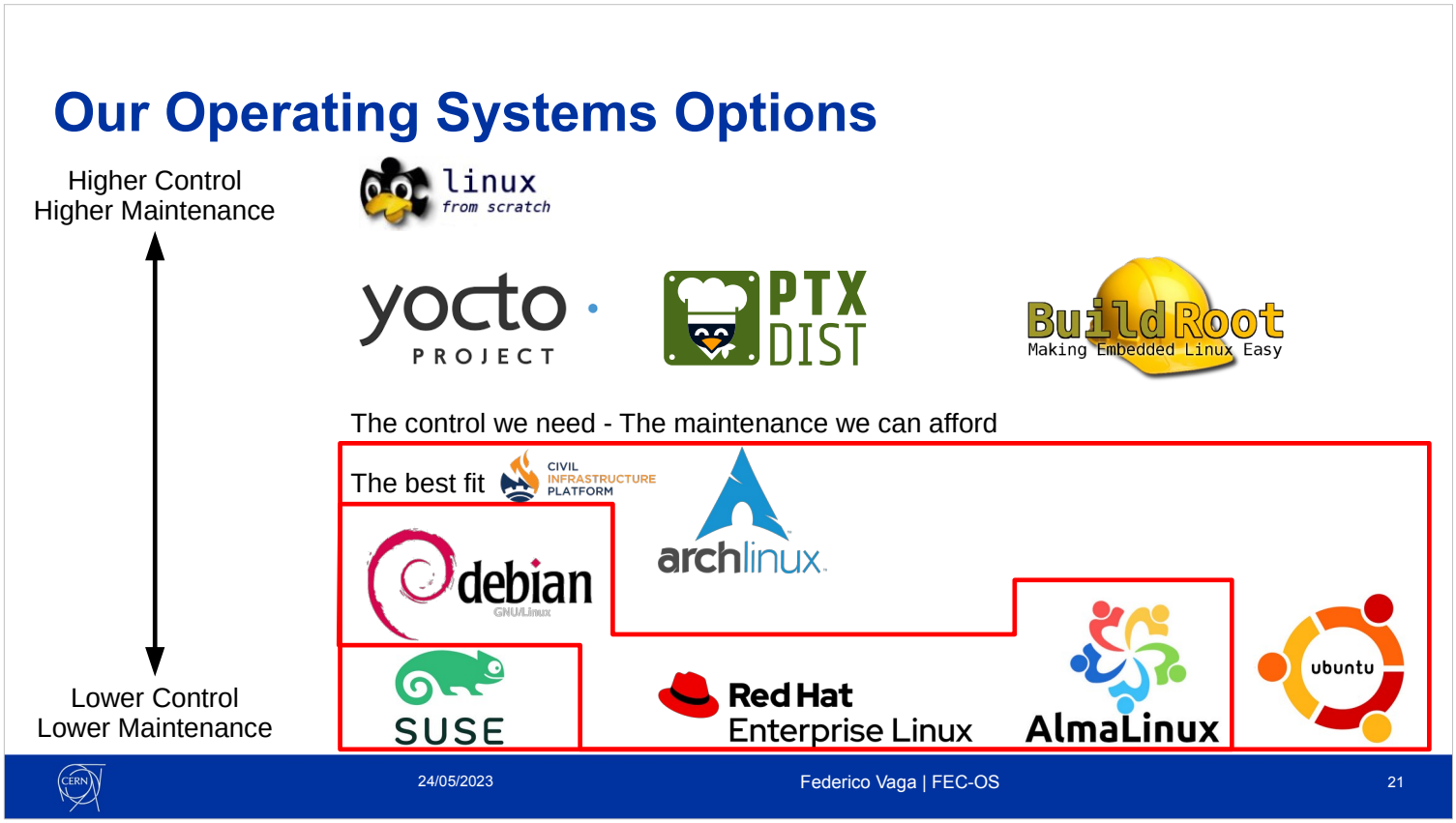
I’m not responsible for your choice

What would you install in a data centre where you want to deliver cloud services?

What would you install on a gaming PC?

What would you install on industrial systems?

What would you install when you have both a data centre for data crunching and an industrial system to maintain?



Many options on the table to build the next Linux OS.

The level of control and the required maintenance change among Linux distributions.

Linux images from source: high control, high maintenance. Users are free to tune all options, at the price of having a team able to cope with the maintenance in the long term.

Binary distributions: low control, low maintenance. Today's requirements drove us within this family of distributions.

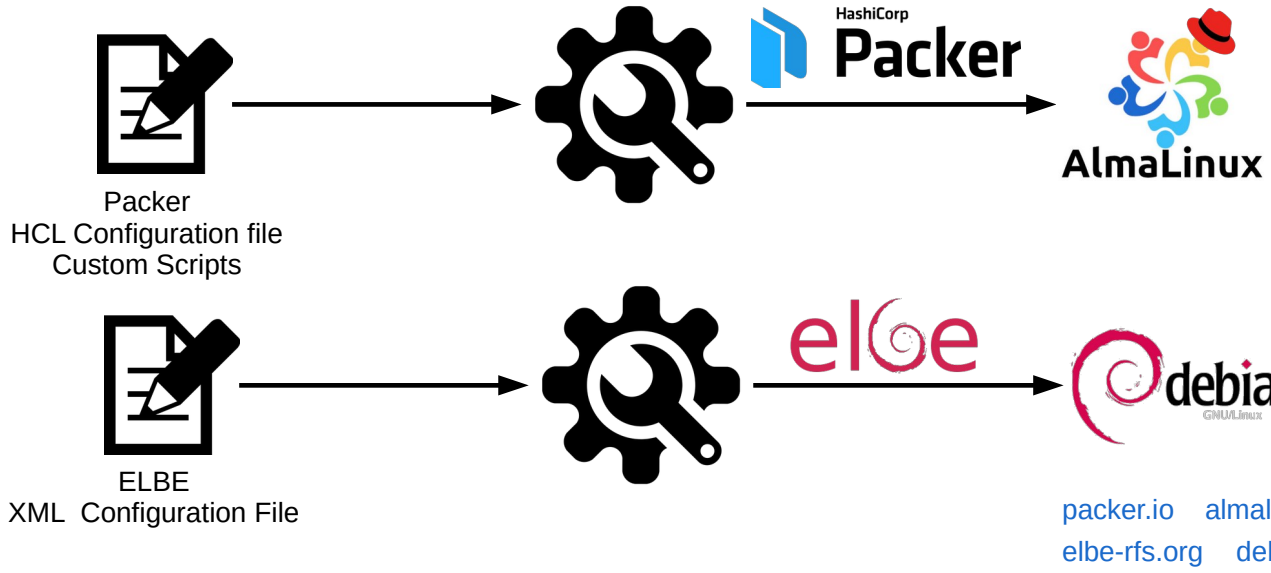
In particular, our best fit is with the RHEL family and Debian.

Why Two Linux Distributions?

- **It enforces software portability**
  - Everything must work on different Linux environments
- **It reduces the Total Cost of Ownership (TCO)**
  - It reduces the exit costs and prevents lock-in
- **It is relatively cheap to *maintain* both solutions**
  - After a first investment in portability, it is easy to maintain
- **It allows us to have an exit strategy**
  - We can always easily move to the second distribution if needed
- **We will never support two distributions at the same time**
  - Only one distribution will be officially supported (primary)
  - The second distribution will be our exit strategy (secondary)

The slide is self-explanatory

Building Processes



We build Debian 11 and CentOS Stream 8 using two different building processes. In both case, a recipe describes the result that we want to obtain.

Packer is used to build RHEL like systems. It is a tool developed and maintained by hashicorp

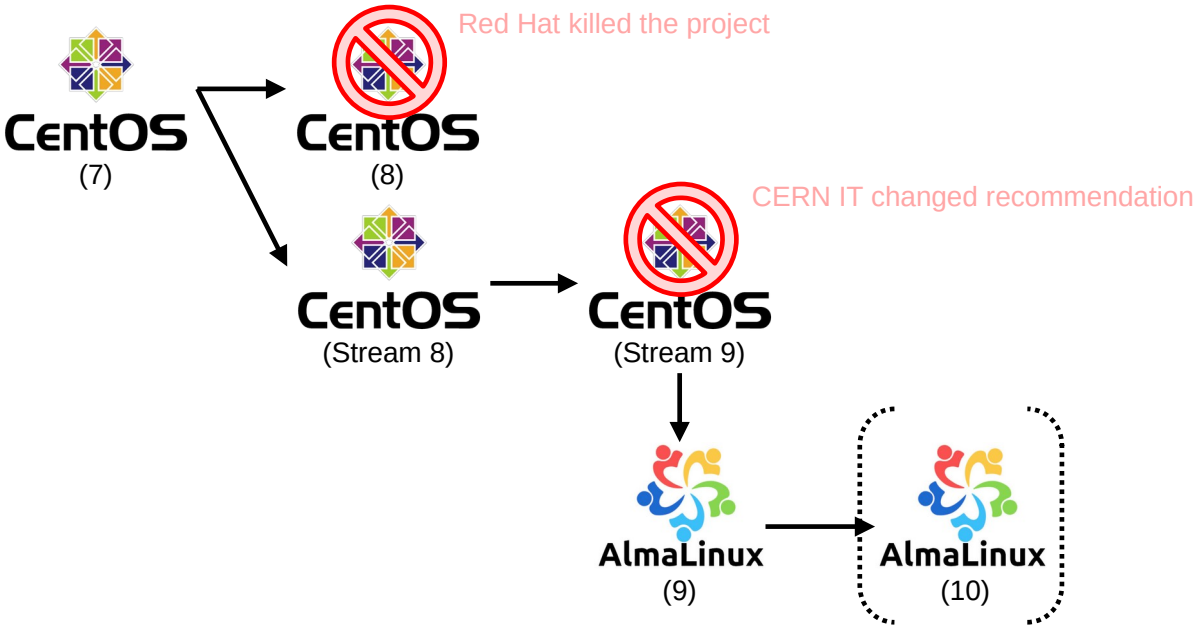
ELBE is used to build Debian. It is a tool developed and maintained by Linutronix (Real-Time Patches), recently bought by Intel.

First Choice: RHEL Family

It Is Good To Follow Recommendations

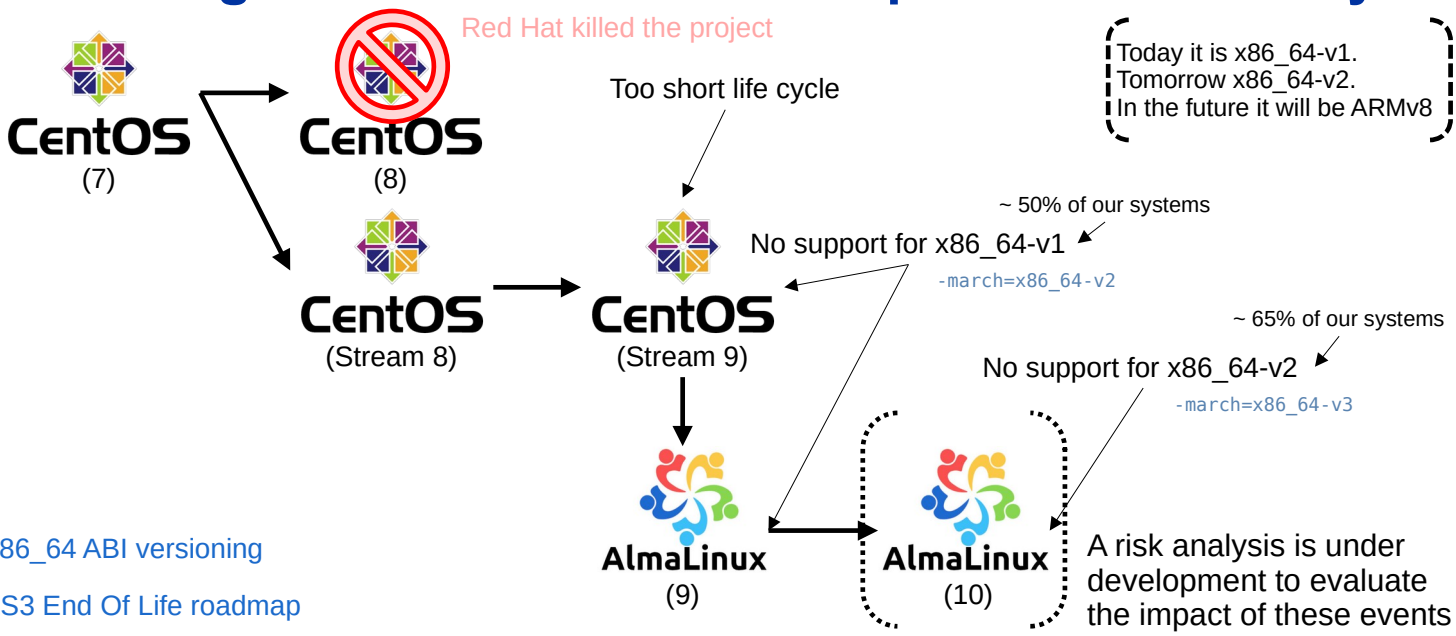
Someone else is offering good services to you

Evolution Of Our First Choice Preference



The slide is self-explanatory

Challenges Within Red Hat Enterprise Linux Family

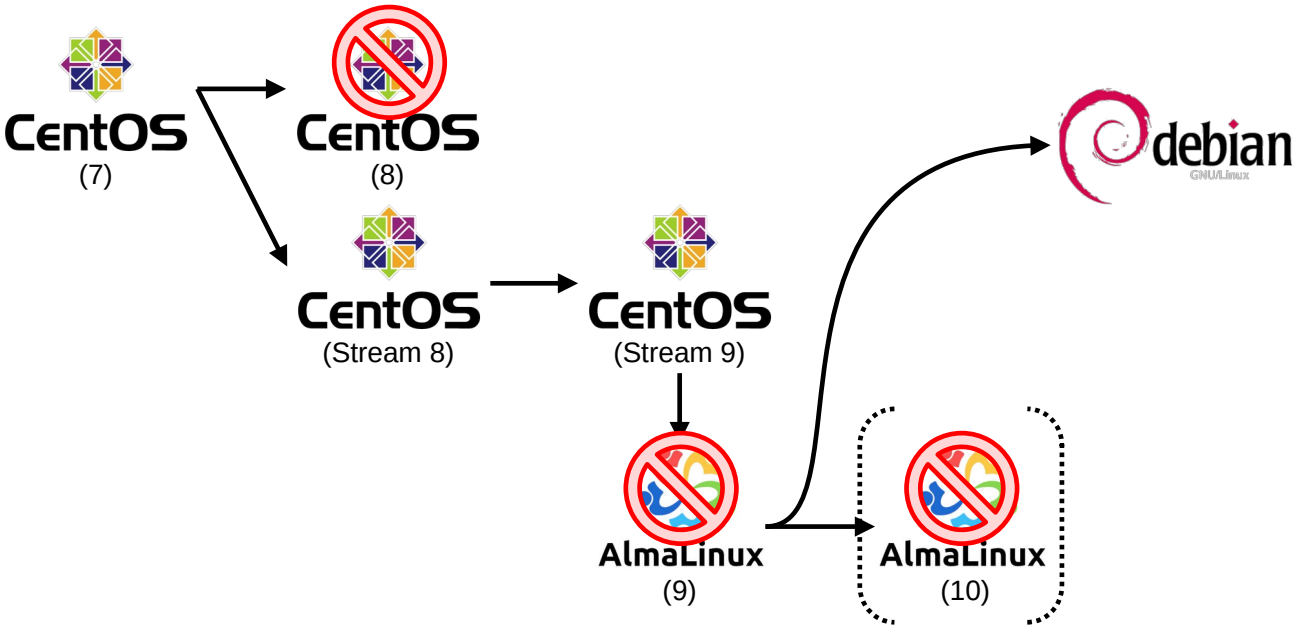


x86_64 ABI versioning

LS3 End Of Life roadmap

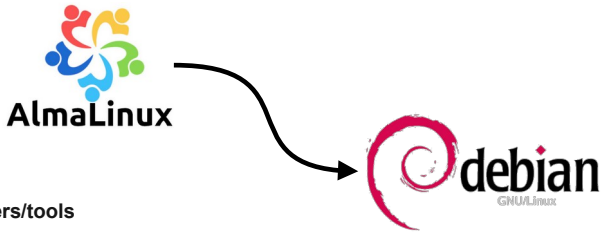
The slide is self-explanatory

A Possible Scenario: Promote The Second Choice



The slide is self-explanatory

Conclusions

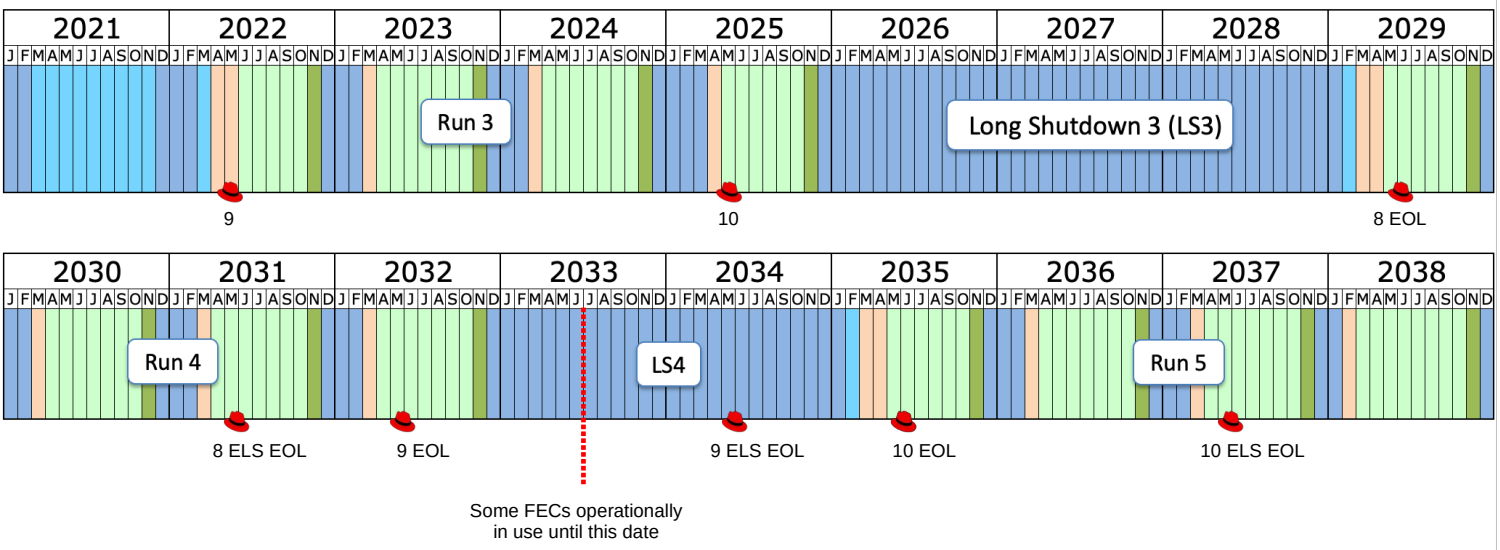


- **Because we speak about embedded systems**
  - Not servers, nor service oriented applications.
- **Because we develop custom hardware and dedicated drivers/tools**
  - Hence, we want finer control over these systems.
- **Because RHEL Family of distributions aims at “server farms” and workstations**
  - aarch64 support is not there because of embedded systems, but because aarch64 is entering in “server farms”
- **Because historically we could never benefit from Red Hat support line**
  - At least not directly, we benefit from Red Hat work in terms of distribution maintenance
- **Because we can’t rely on Red Hat support if we modify packages**
  - We want control over the Linux kernel, therefore we will provide our own build from vanilla sources
- **Because Red Hat dropped support for x86_64-v1 and x86_64-v2 CPUs**
  - Officially confirmed by Red Hat that RHEL 10 will not support x86_64-v2
- **Because using RHEL or derivatives requires the renovation of almost 65% of our systems**
  - Money and people must be put at work to compensate the compiler flag `-march=x86_64-v2`
- **Because we have an exit strategy**
  - it is easy for us to switch between RHEL Family and Debian
- **We are considering selecting Debian as our first choice, we would like to start a collaboration with CERN IT**
  - For both Industrial PCs and SoC systems (today Zynq UltraScale+)

The slide is self-explanatory

BACKUP SLIDES

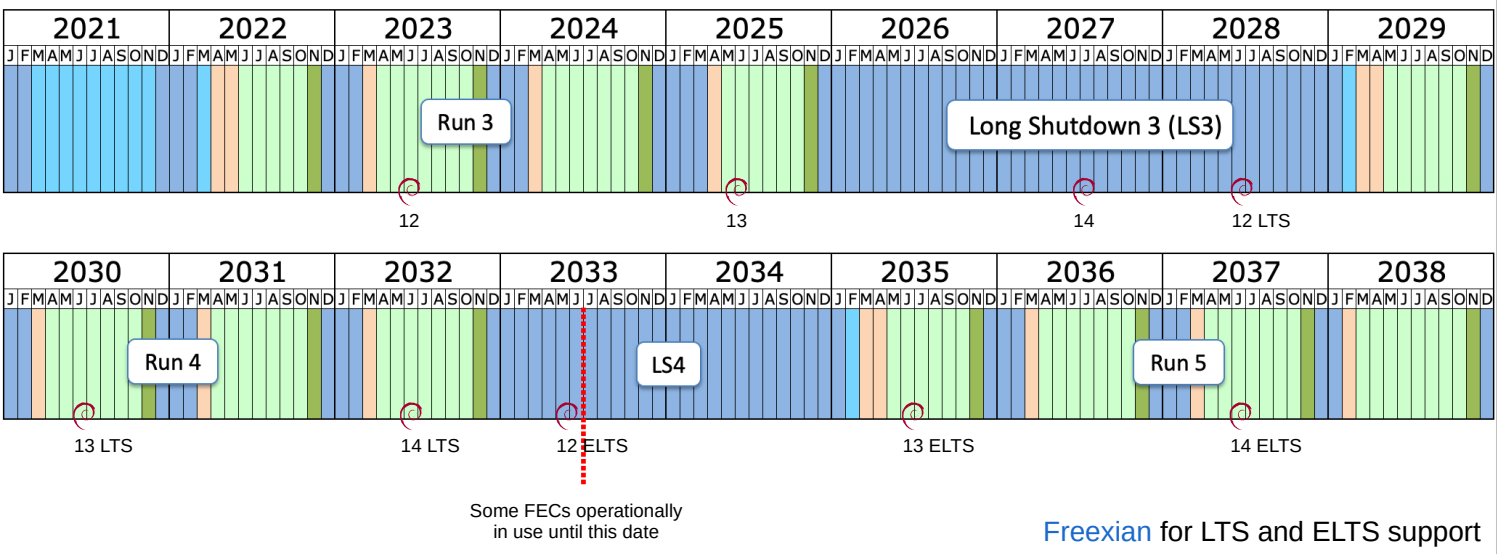
FEC OS Renovation Schedule: Red Hat Enterprise Linux



An Linux distribution available today will not survive LS3 + RUN4.

To avoid repeating the CentOS 7 problem we should select the final OS during in 2024/2025. Red Hat Enterprise Linux 10 might be available and it will cover LS3 + RUN4

FEC OS Renovation Schedule: Debian



Freexian for LTS and ELTS support

An Linux distribution available today will not survive LS3 + RUN4.

To avoid repeating the CentOS 7 problem we should select the final OS during in 2024/2025. Red Hat Enterprise Linux 10 might be available and it will cover LS3 + RUN4