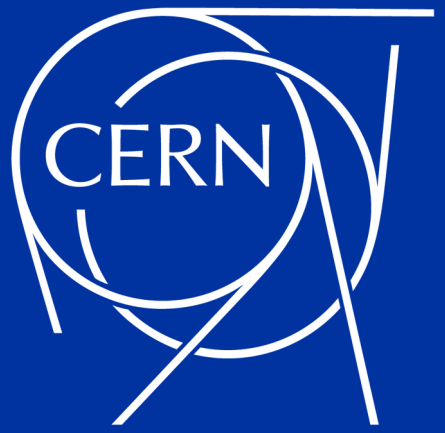


Selection Of Linux Distribution For Front-End Computers and System-on-Chip in CERN-ATS

Federico Vaga

2023-05-24

Who am I?

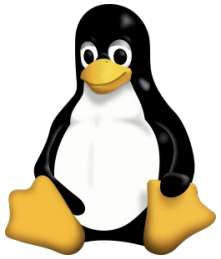
- **I work in CERN BE-CEM-EDL**
- **I’m a Software Engineer, specialised in:**
  - Low-Level Software
  - Linux kernel development
  - Linux systems

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

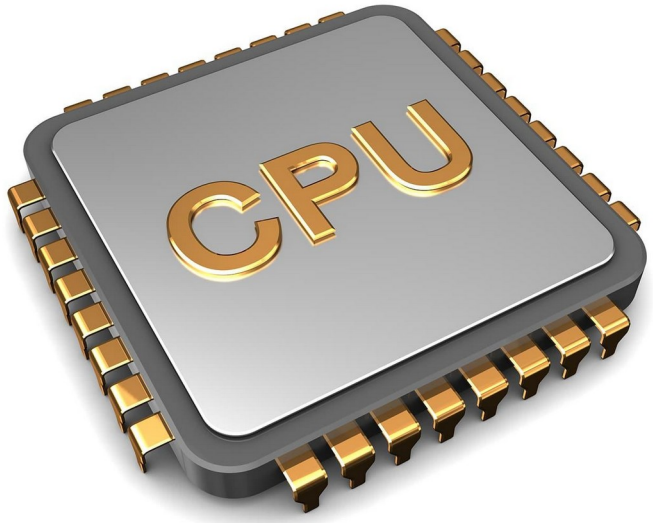
www.gnu.org/software/grub

www.kernel.org

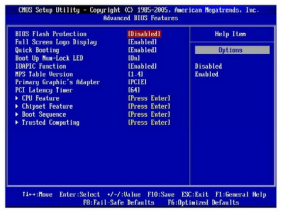
github.com/dracutdevs/dracut

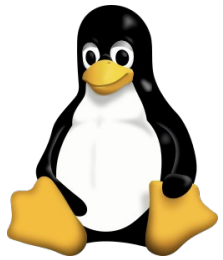
almalinux.org

www.debian.org



Network RAM Booting: Industrial PC





DRA
CUT

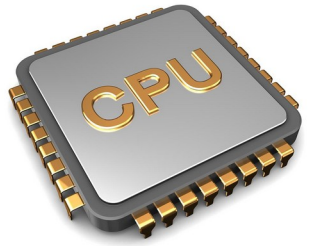
BIOS
EFI Firmware

GRUB

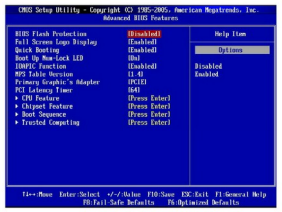
Linux

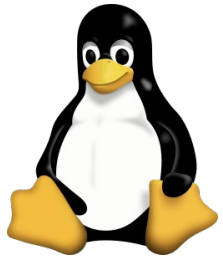
Init RAM
disk

Distribution



Network RAM Booting: Industrial PC





DRA
CUT

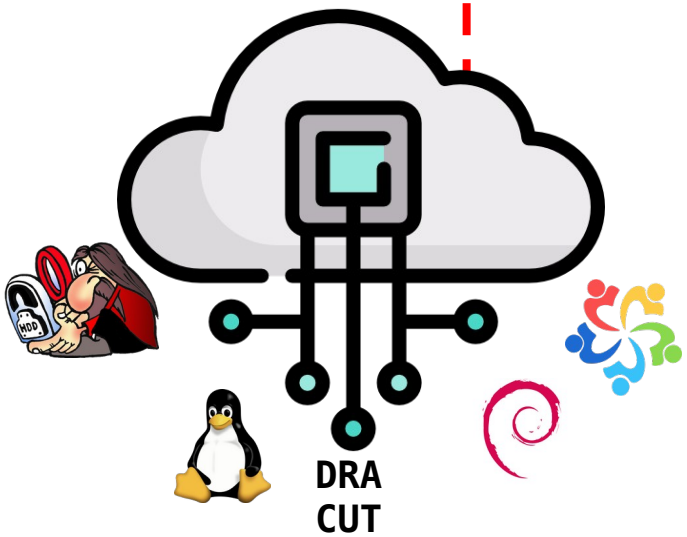
BIOS
EFI Firmware

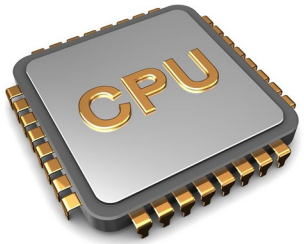
GRUB

Linux

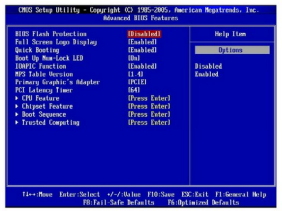
Init RAM
disk

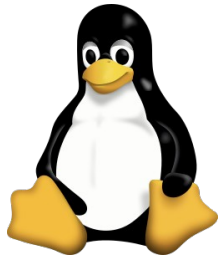
Distribution





Network RAM Booting: Industrial PC





DRA
CUT

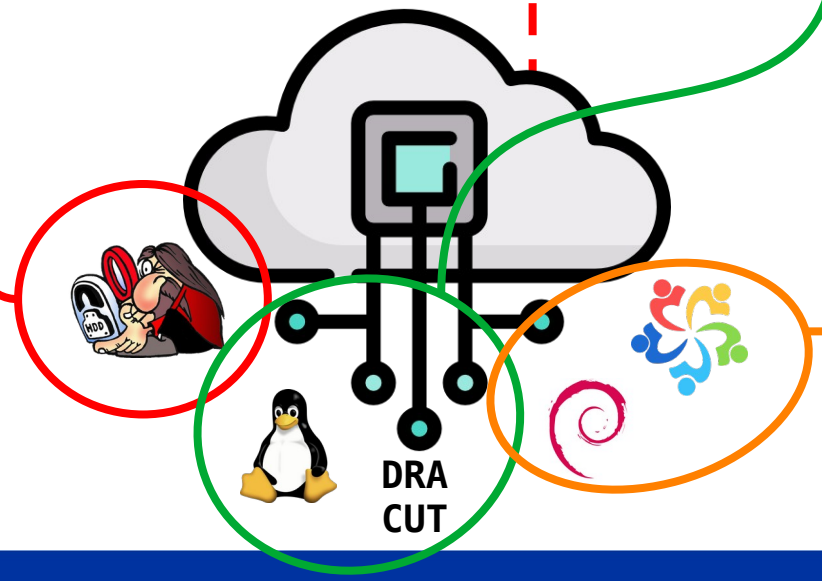
BIOS
EFI Firmware

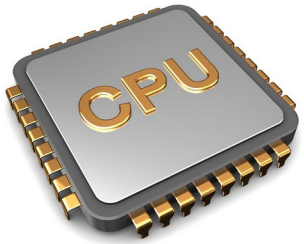
GRUB

Linux

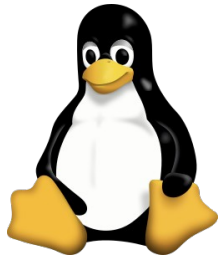
Init RAM
disk

Distribution



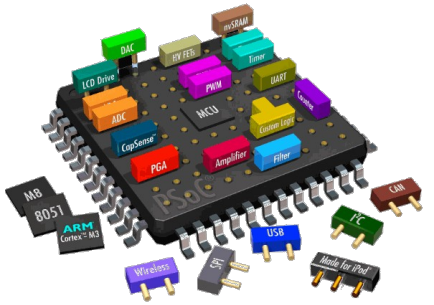


Network RAM Booting: System-on-Chip

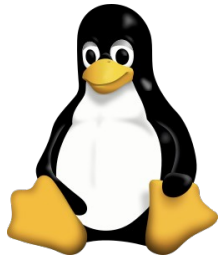


DRA
CUT

FSBL and friends → U-Boot → GRUB → Linux → Init RAM disk → Distribution

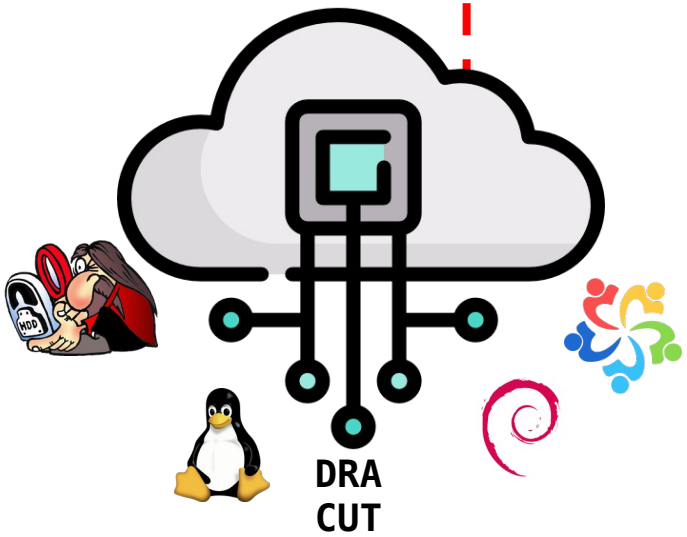


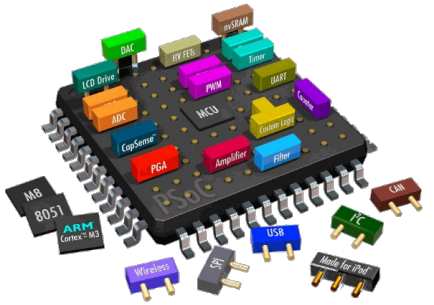
Network RAM Booting: System-on-Chip



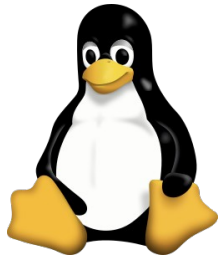
DRA
CUT

FSBL and friends → U-Boot → GRUB → Linux → Init RAM disk → Distribution



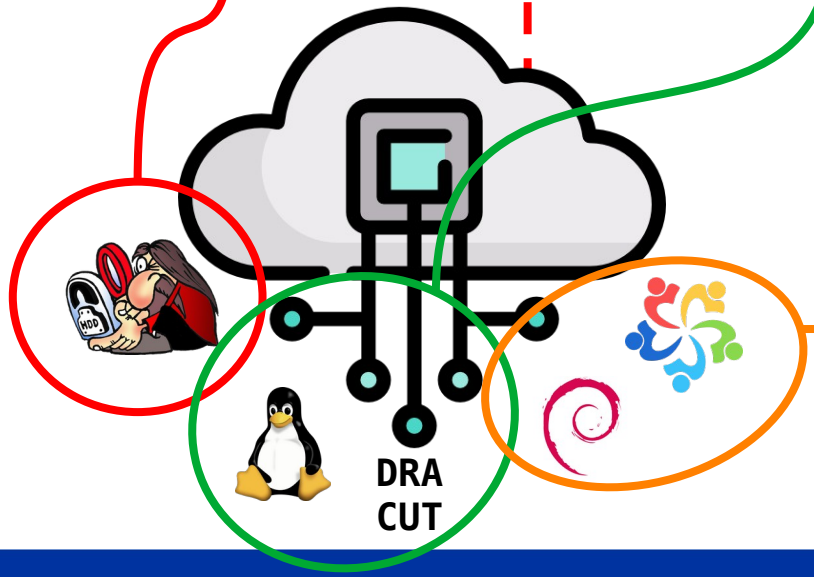


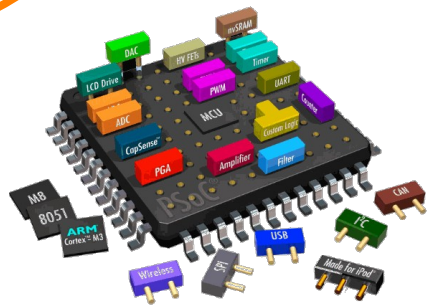
Network RAM Booting: System-on-Chip



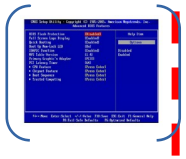
DRA
CUT

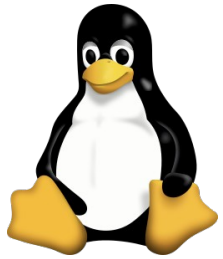
FSBL and friends → U-Boot → GRUB → Linux → Init RAM disk → Distribution





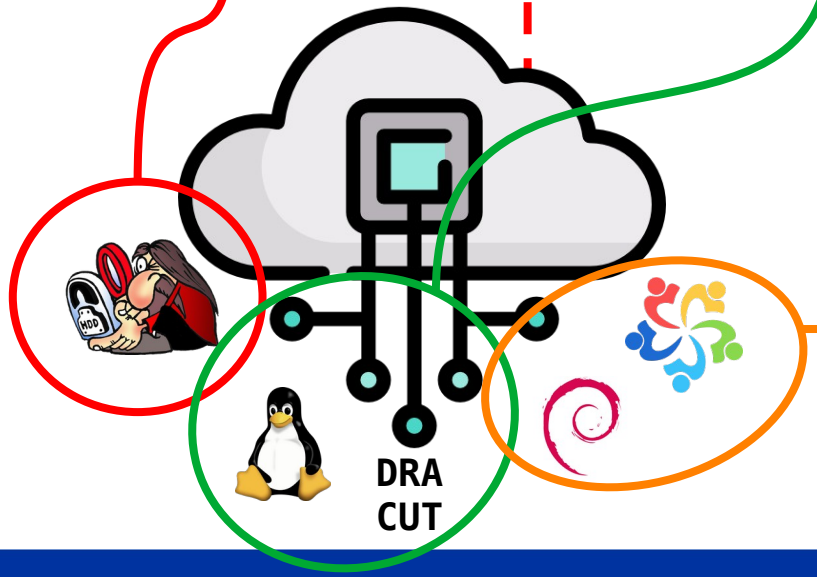
Network RAM Booting: System-on-Chip

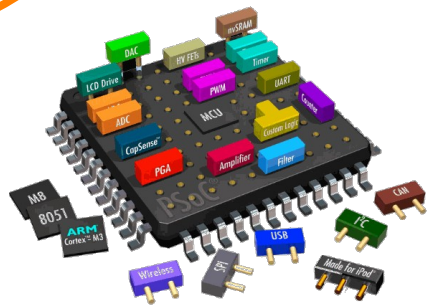




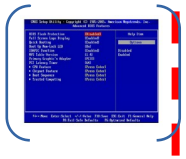
DRA
CUT

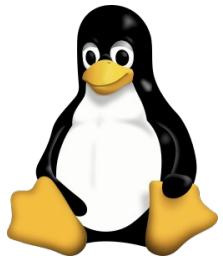
FSBL and friends → U-Boot → GRUB → Linux → Init RAM disk → Distribution





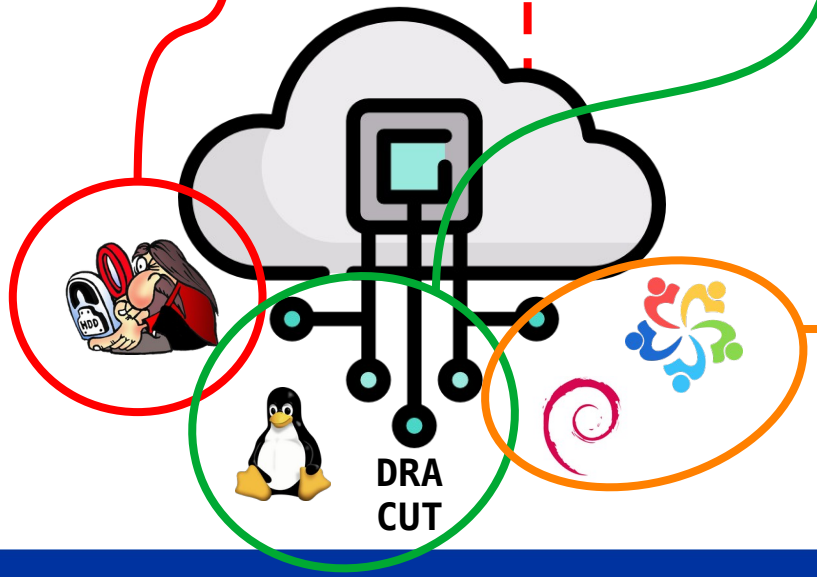
Network RAM Booting: System-on-Chip

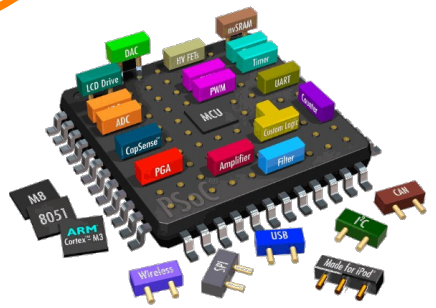




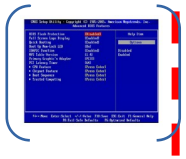
DRA
CUT

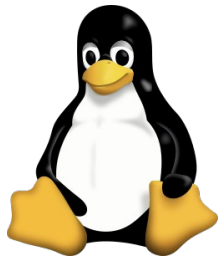
FSBL and friends → Barebox → GRUB → Linux → Init RAM disk → Distribution





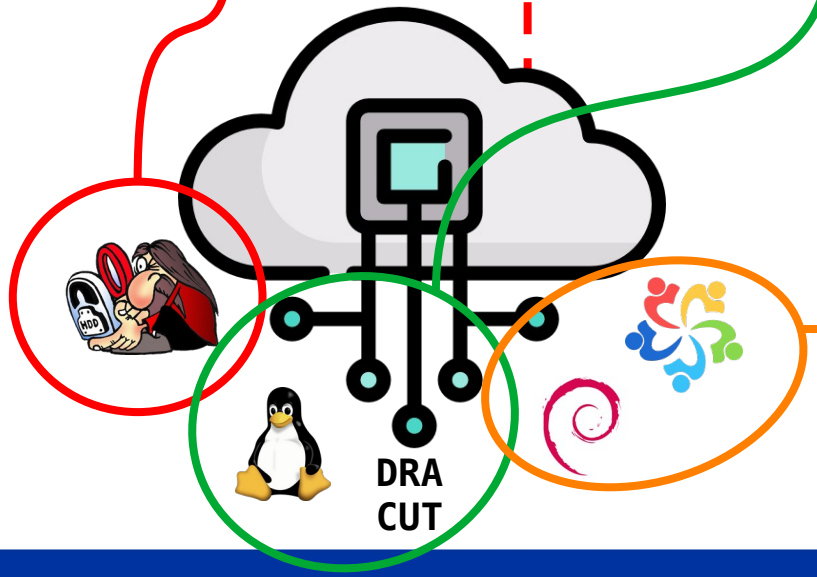
Network RAM Booting: System-on-Chip

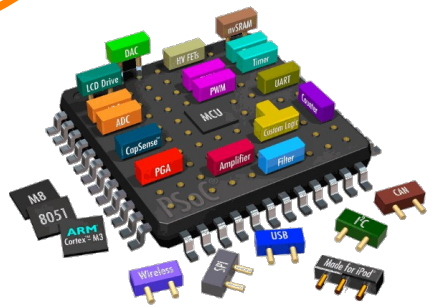




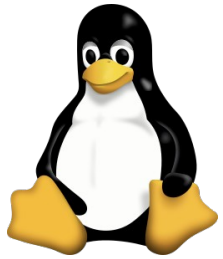
DRA
CUT

FSBL and friends → coreboot → GRUB → Linux → Init RAM disk → Distribution



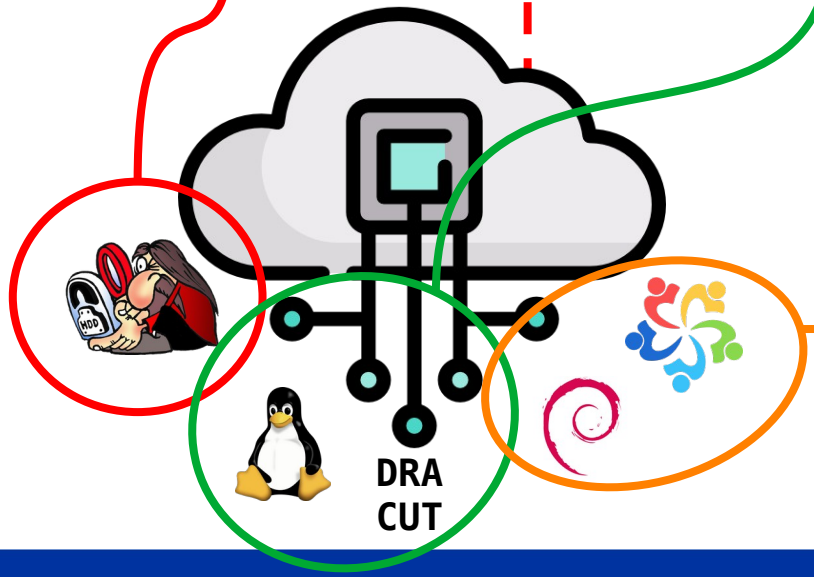


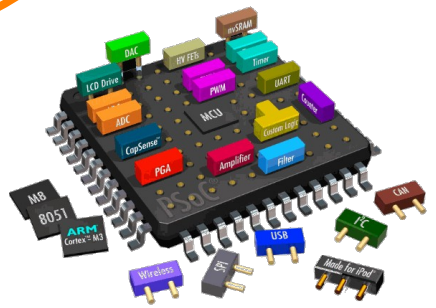
Network RAM Booting: System-on-Chip



DRA
CUT

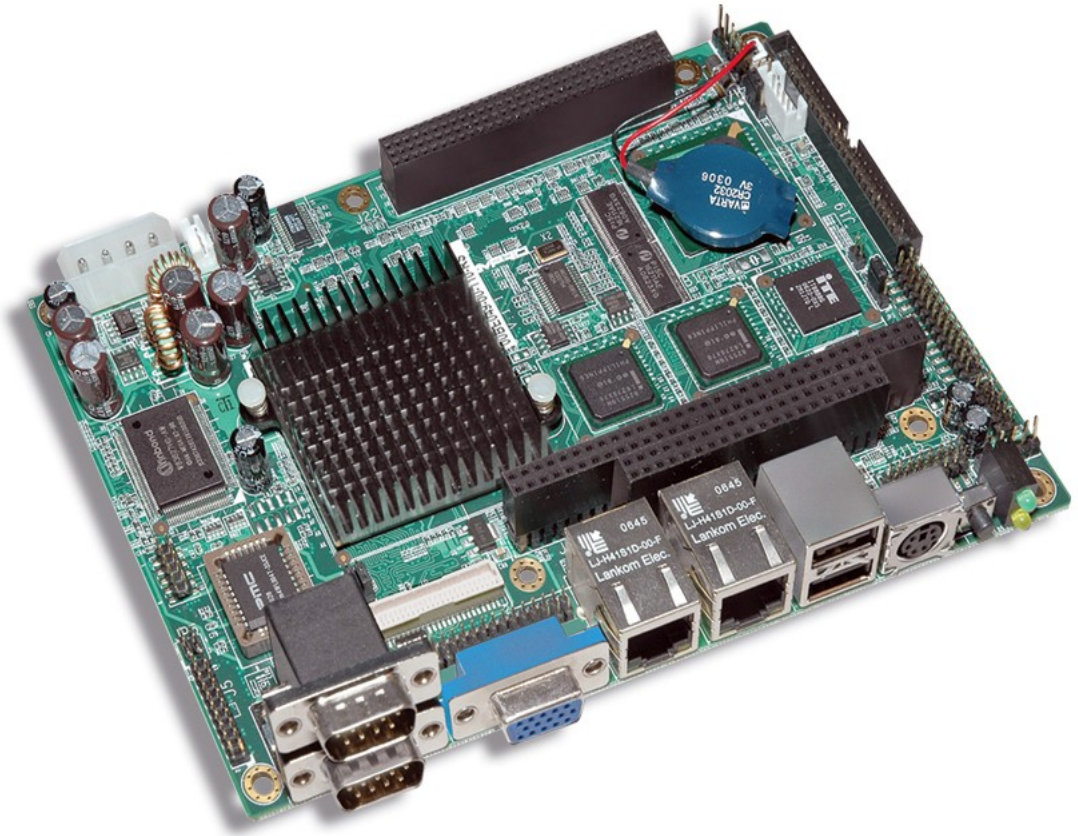
FSBL and friends → coreboot → Boot EFI → GRUB → Linux → Init RAM disk → Distribution



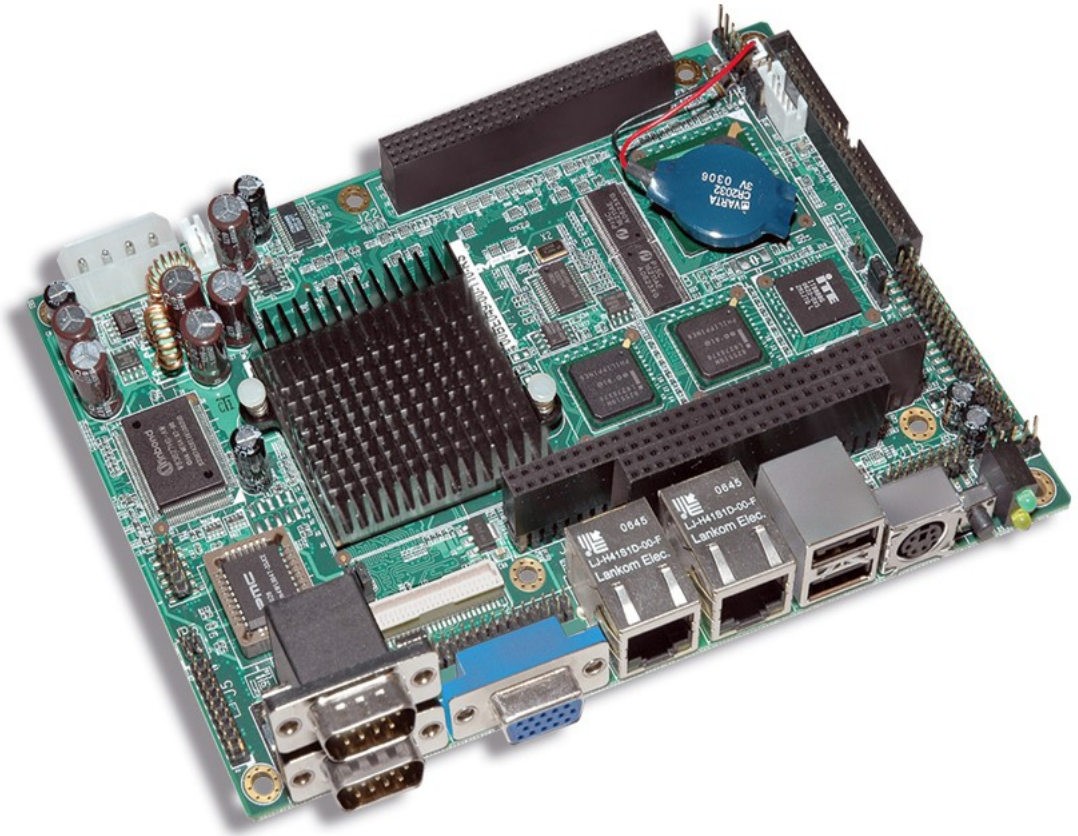


Back On Topic: OS Selection Process

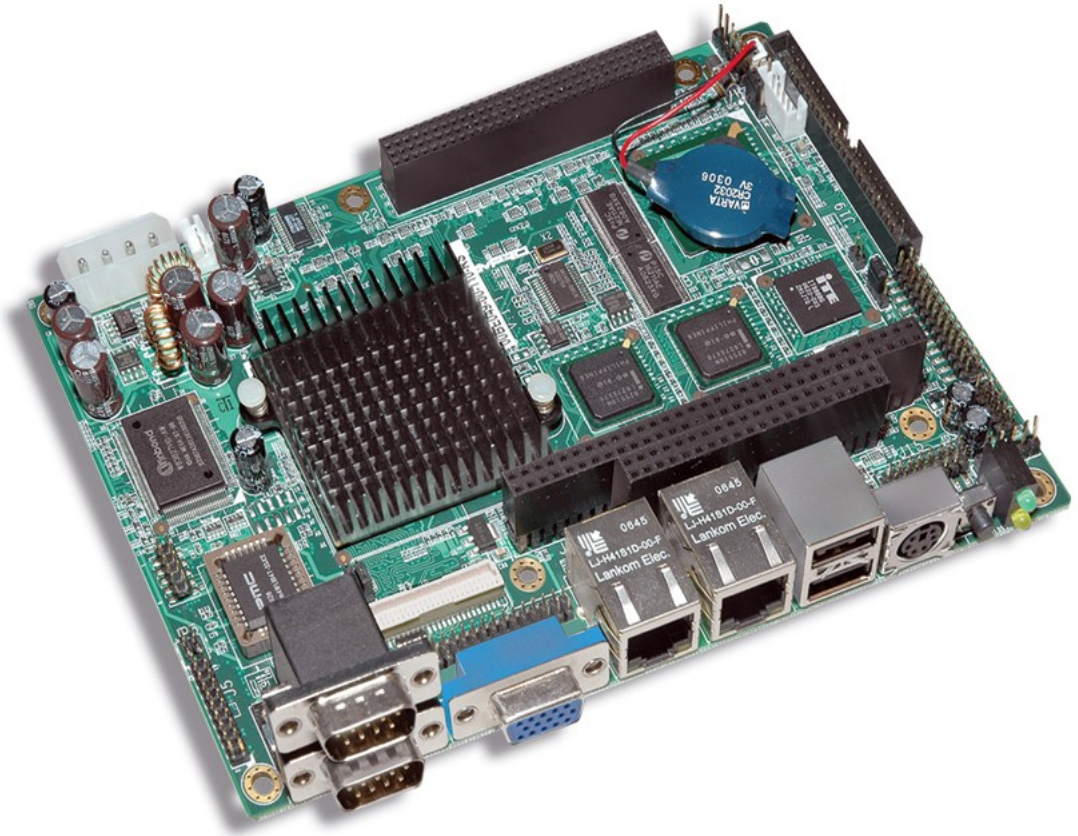
Embedded Systems



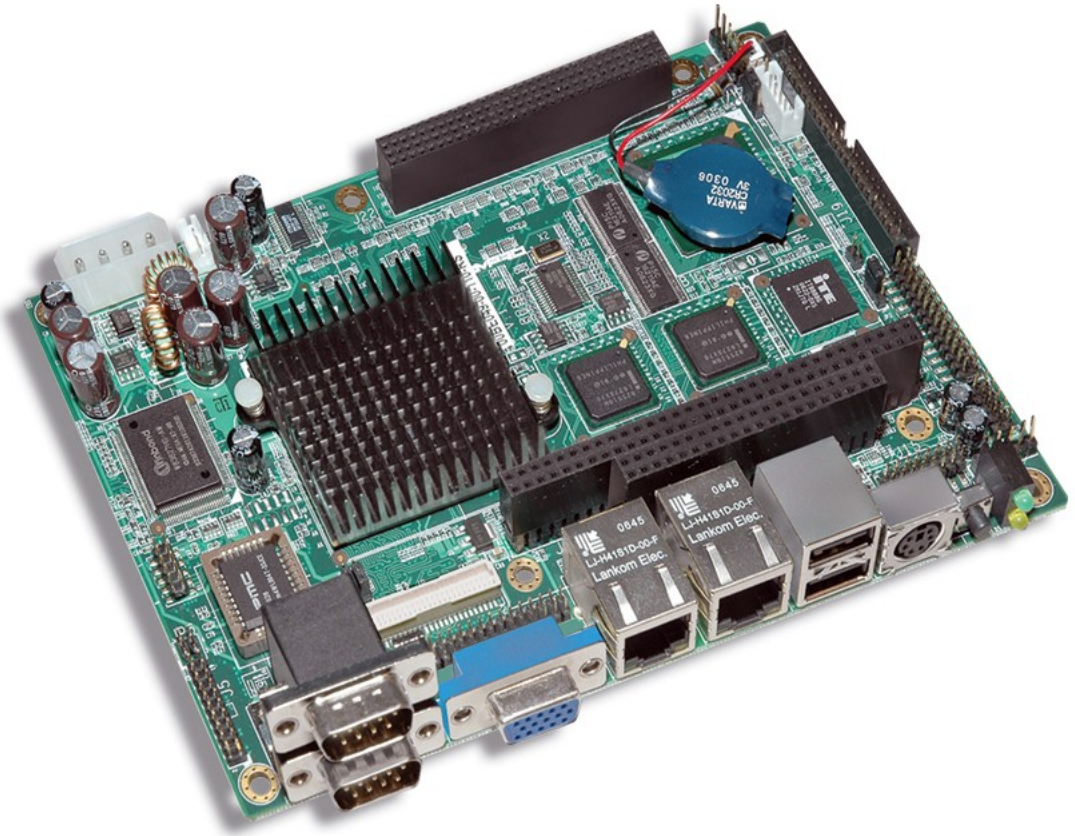
Embedded Systems



Embedded Systems



Embedded Systems



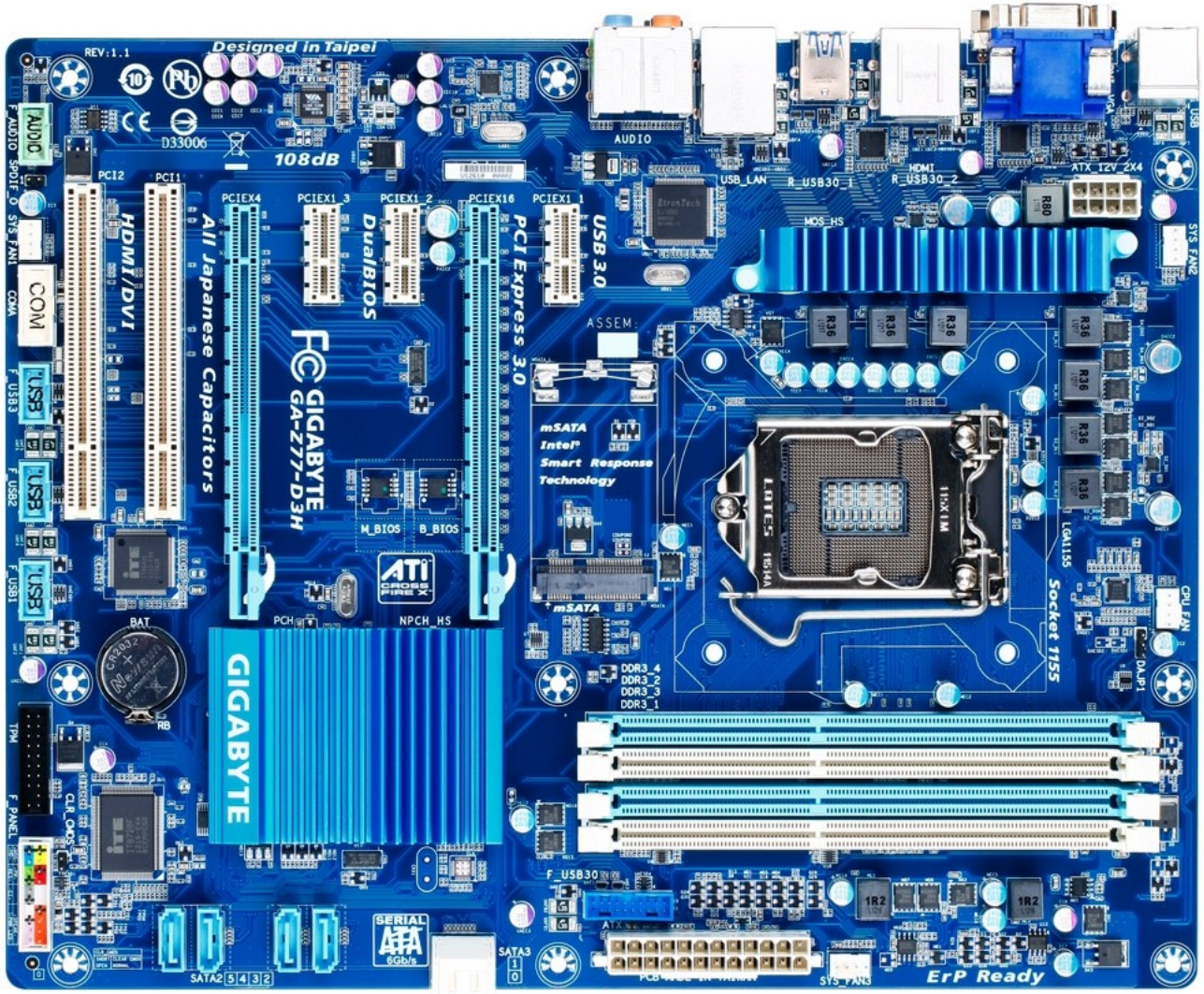
yocto PROJECT

Definition

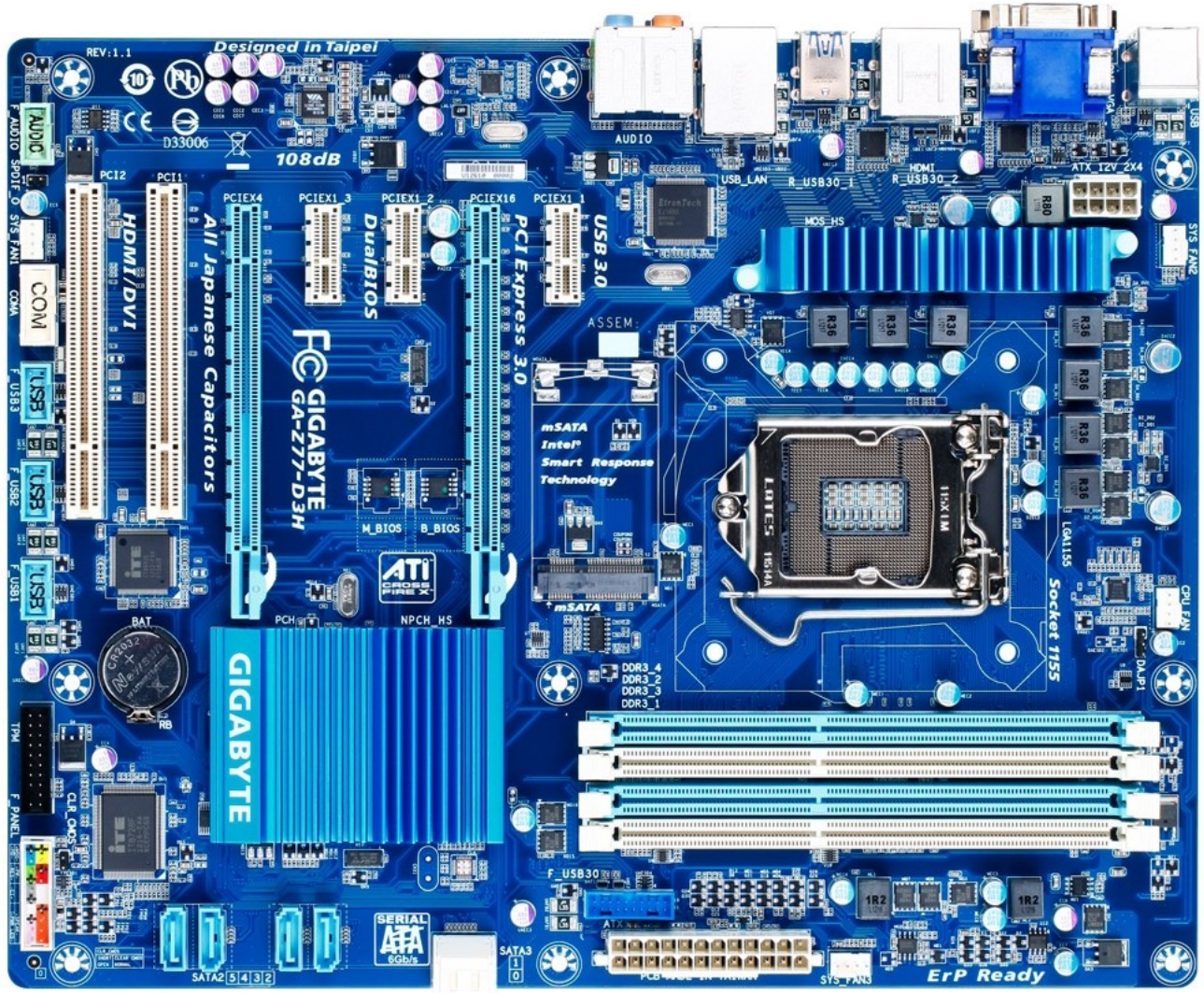
embedded system

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.

Embedded Systems ?

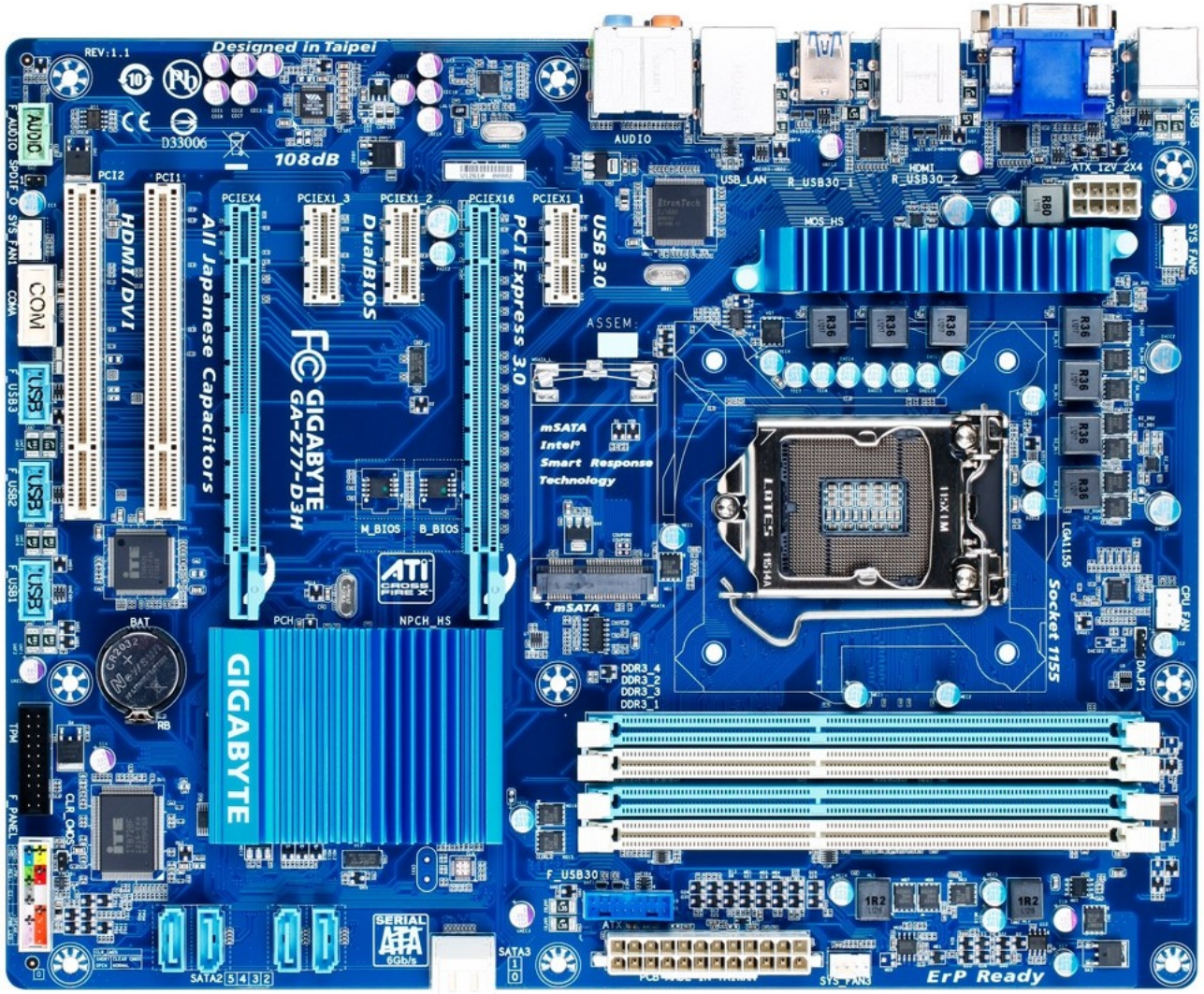


Embedded Systems ?



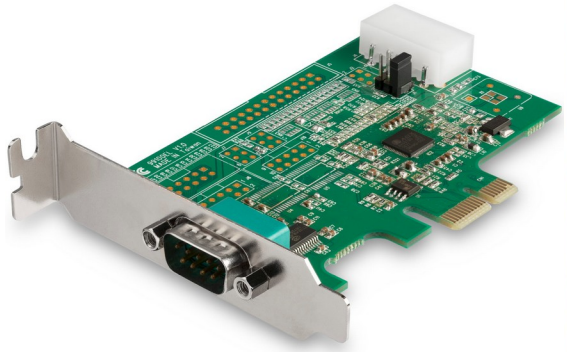
Embedded Systems ?

CAN Bus

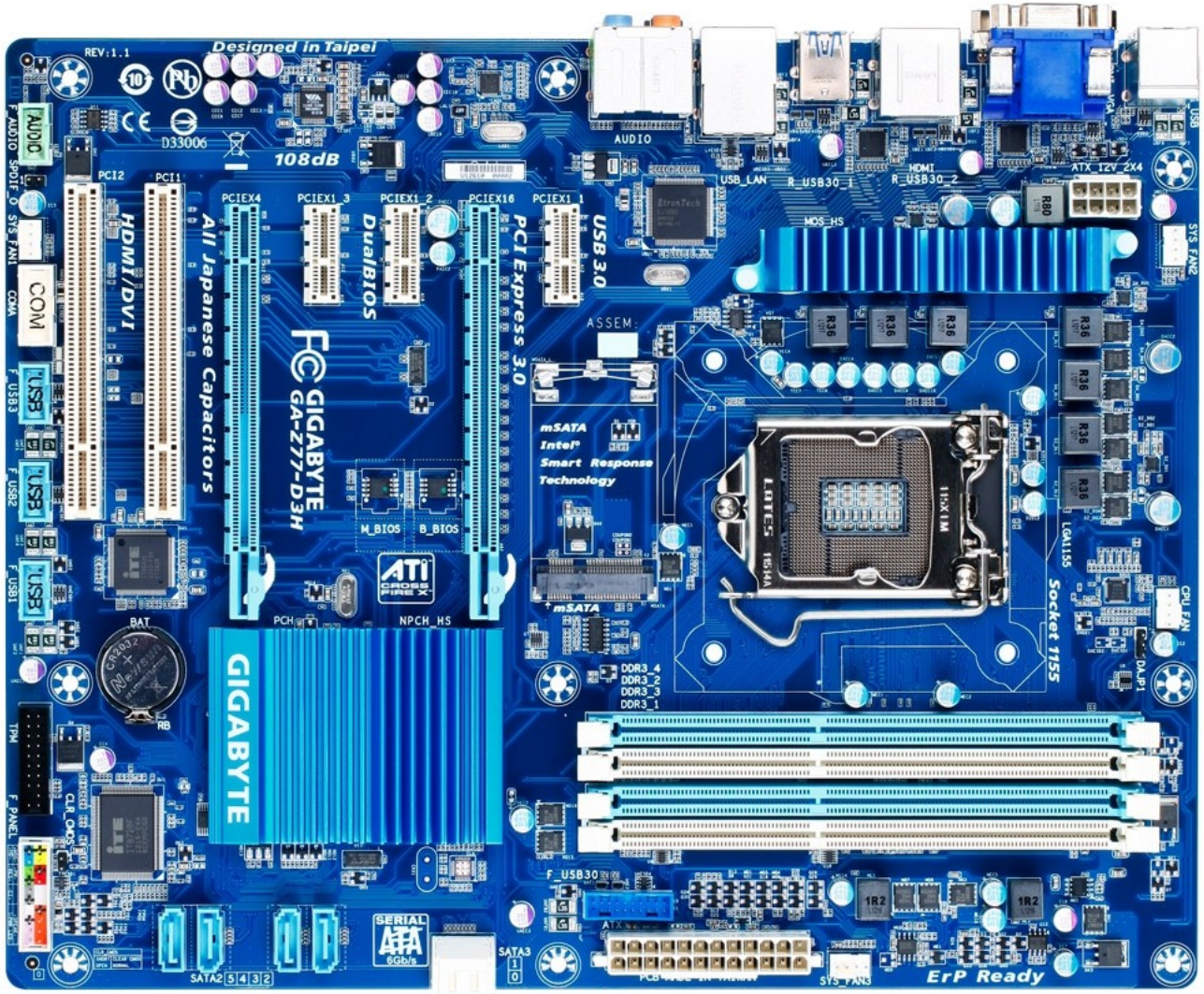


Embedded Systems ?

CAN Bus

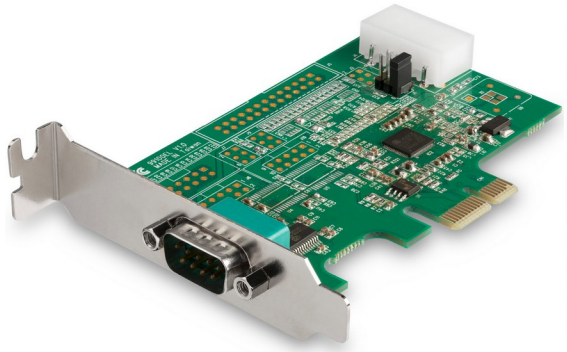


Serial Bus



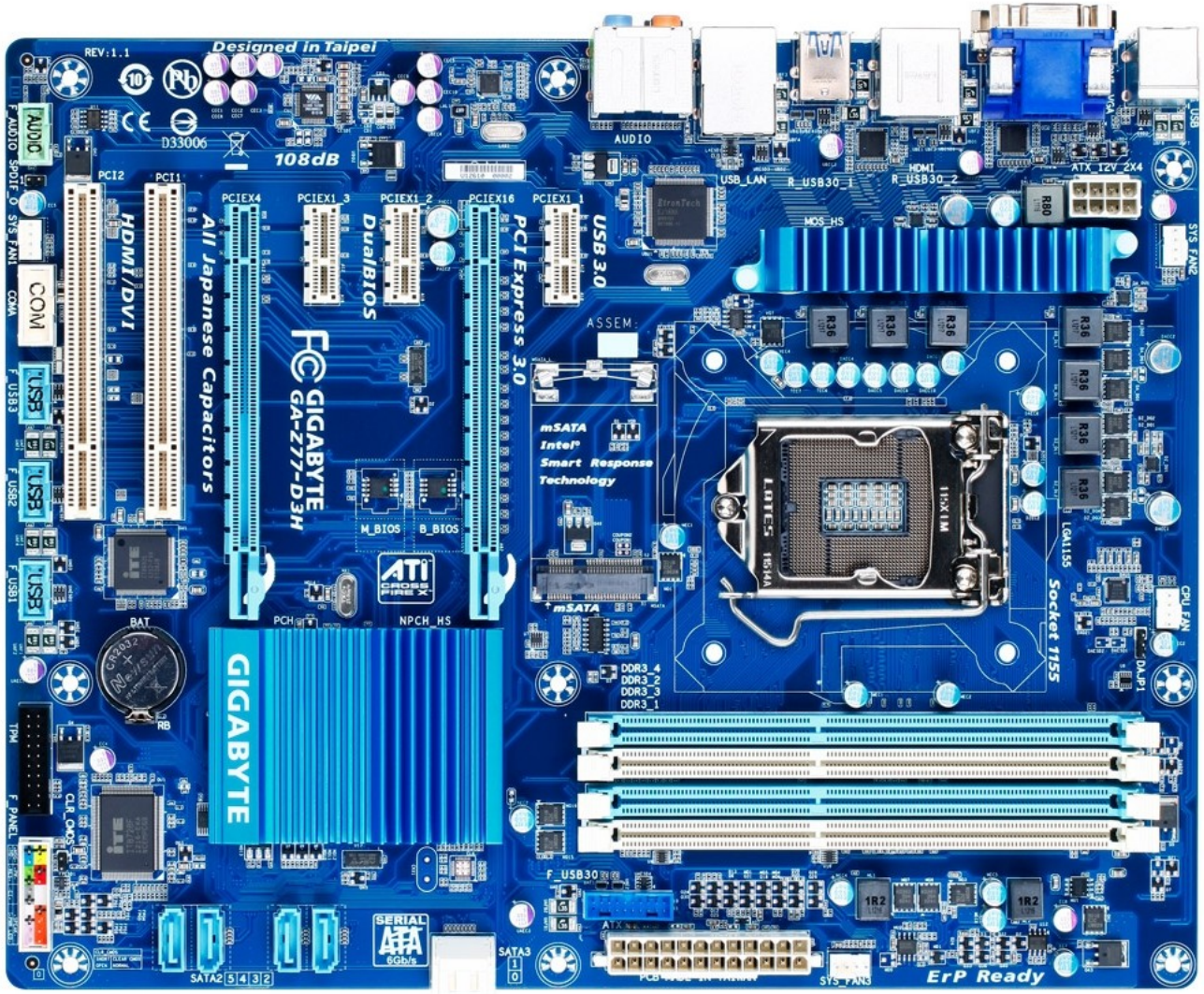
Embedded Systems ?

CAN Bus



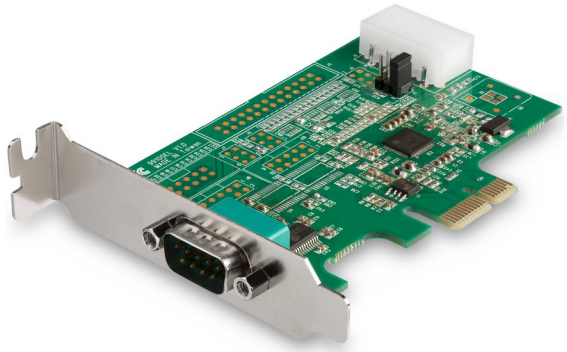
Serial Bus

Digital IO



Embedded Systems ?

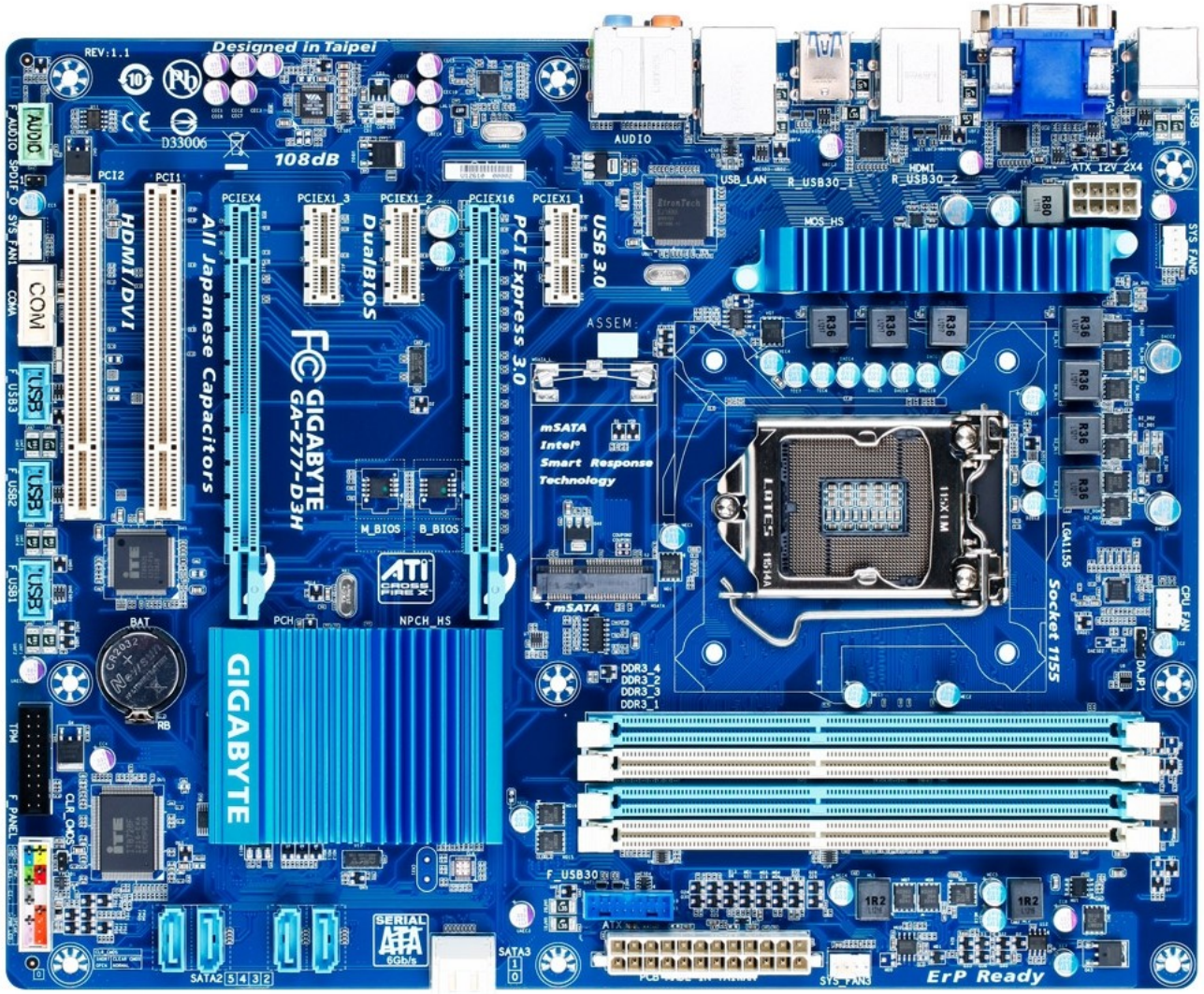
CAN Bus



Serial Bus

Digital IO

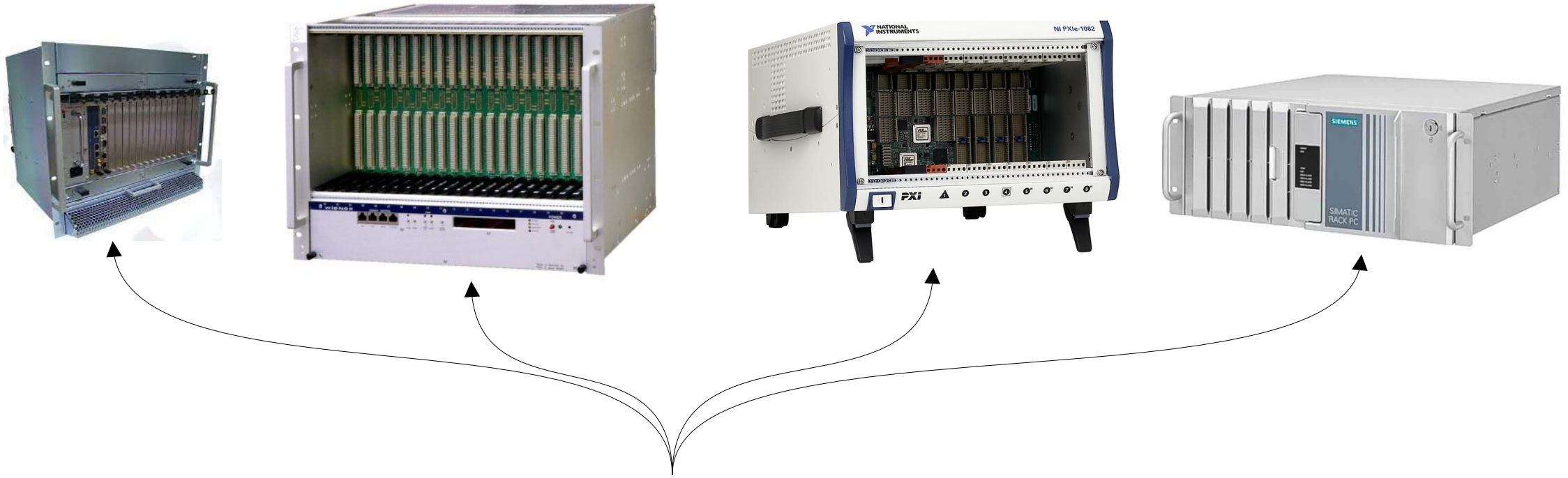
Custom Card



Embedded Systems ?

Embedded Systems ?

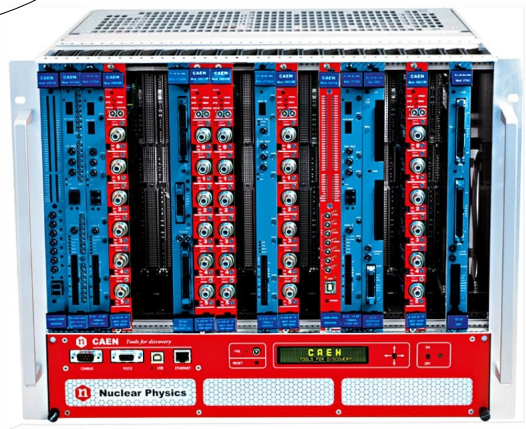
Embedded Systems ?



General Purpose platforms

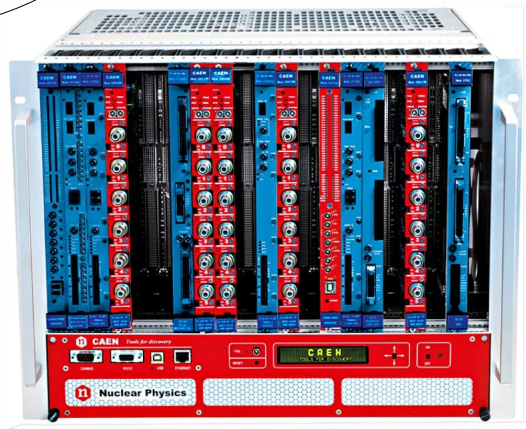
Embedded Systems ?

General Purpose platforms



Embedded Systems !

General Purpose platforms



Embedded System

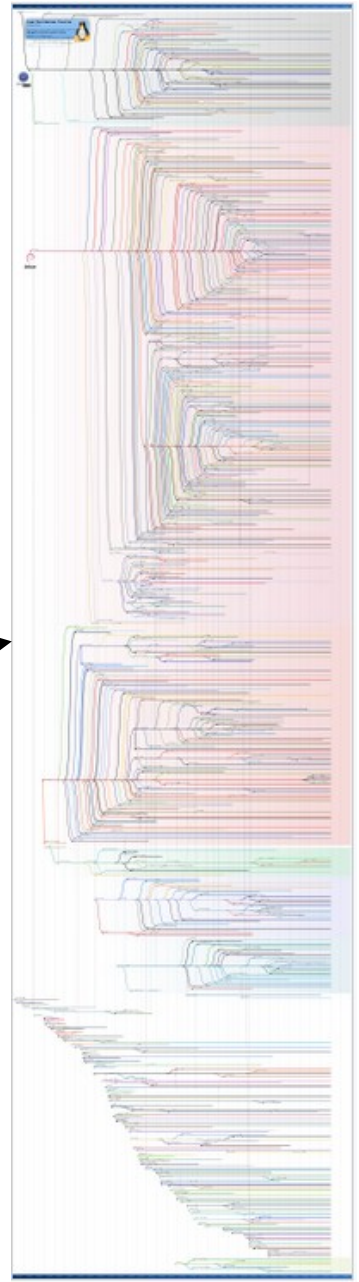
The Next Operating System

For FEC and SoC

Linux Distributions Panorama

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

Wikimedia picture

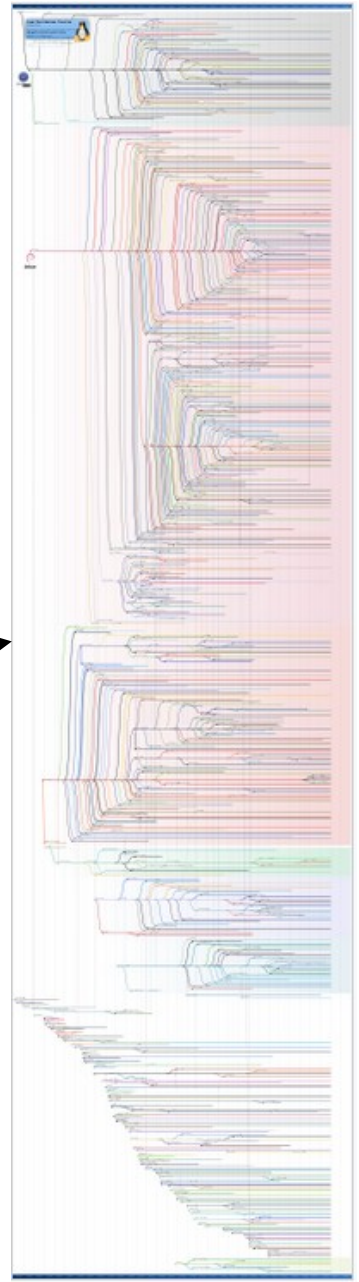


Linux Distributions Panorama

- **There are hundreds of them.**

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

Wikimedia picture

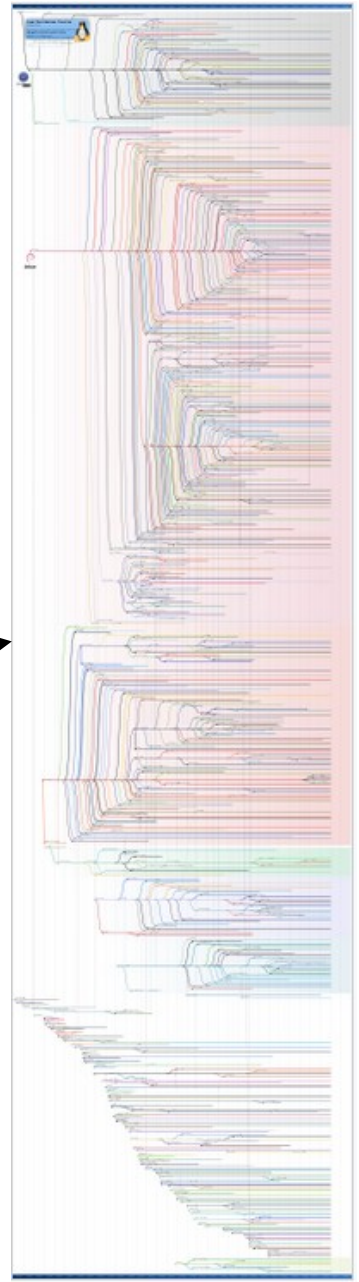


Linux Distributions Panorama

- **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

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

Wikimedia picture

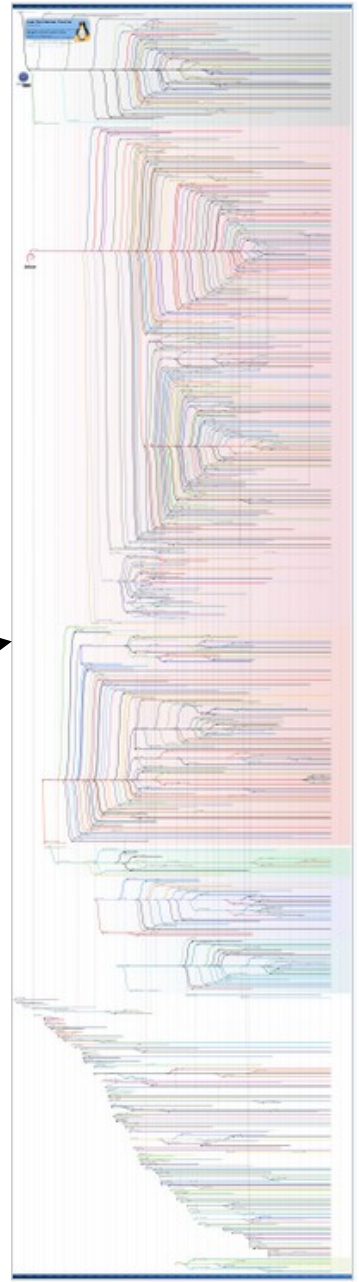


Linux Distributions Panorama

- **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.

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

Wikimedia picture

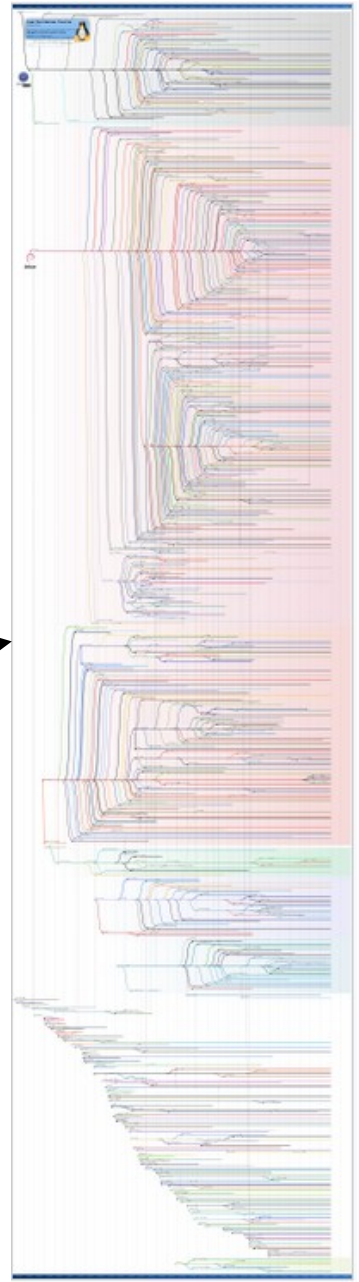


Linux Distributions Panorama

- **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.

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

Wikimedia picture

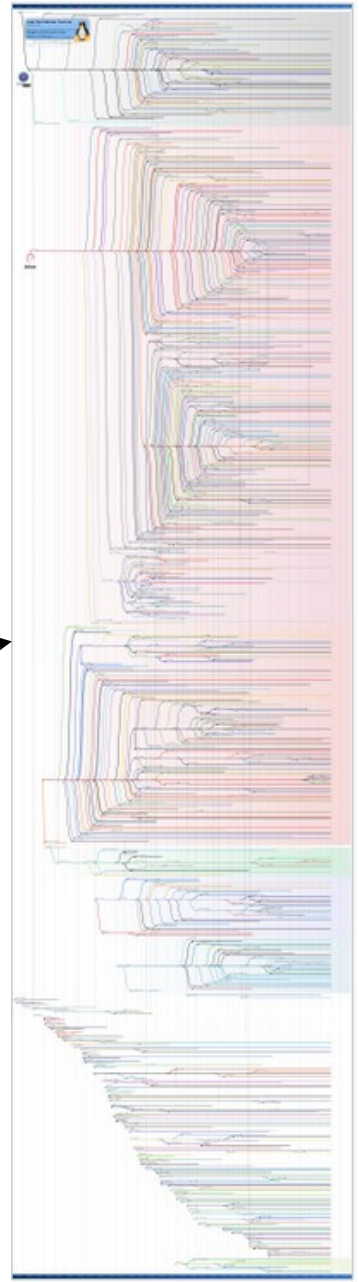


Linux Distributions Panorama

- **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

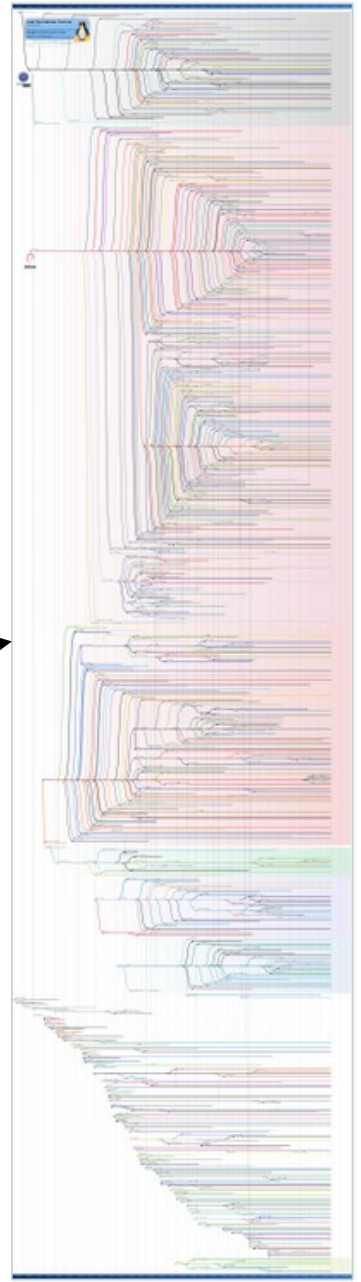


Linux Distributions Panorama

- **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



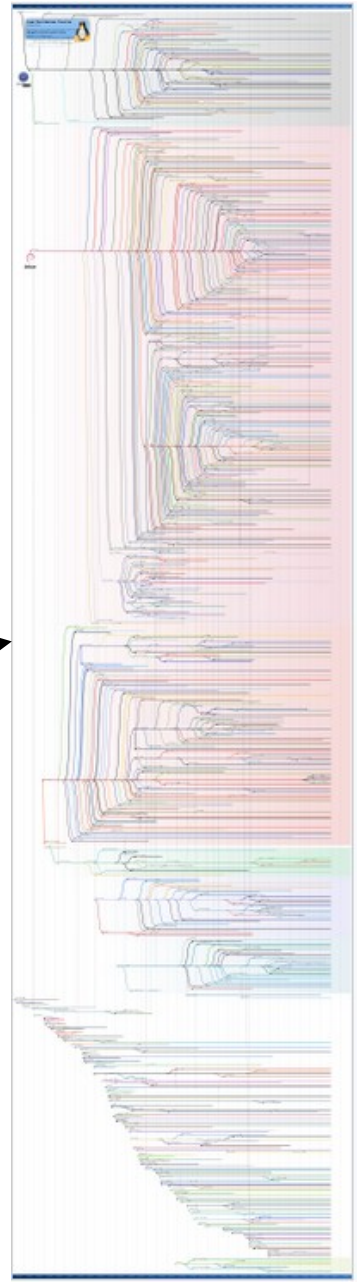
Does this matter?

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!

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.

A Few Examples

A Few Examples

I’m not responsible for your choice


Cloud
Computing

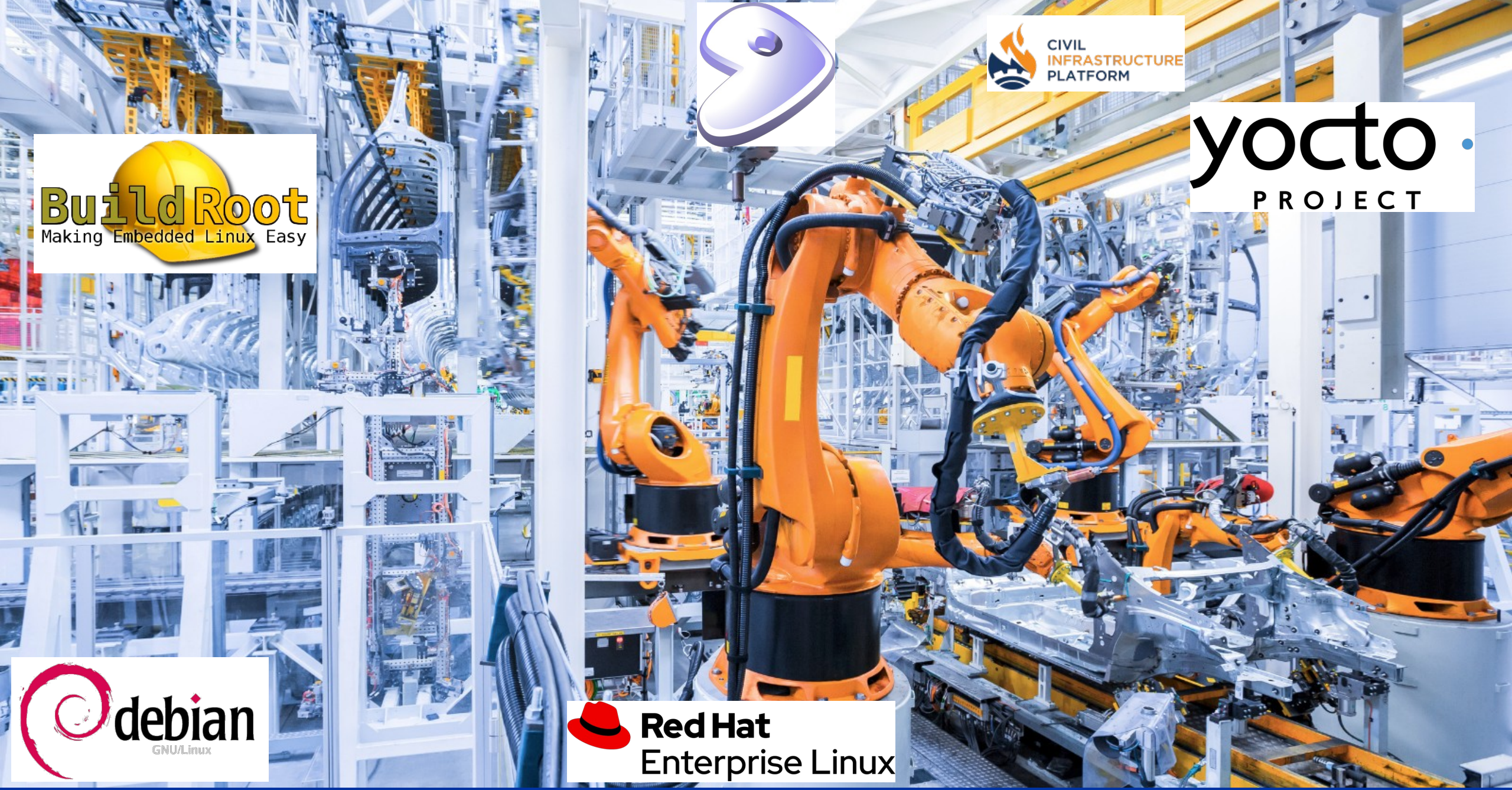

BuildRoot
Making Embedded Linux Easy
CIVIL
INFRASTRUCTURE
PLATFORM
Cloud
Computing
yocto
PROJECT
Red Hat
Enterprise Linux
debian
GNU/Linux

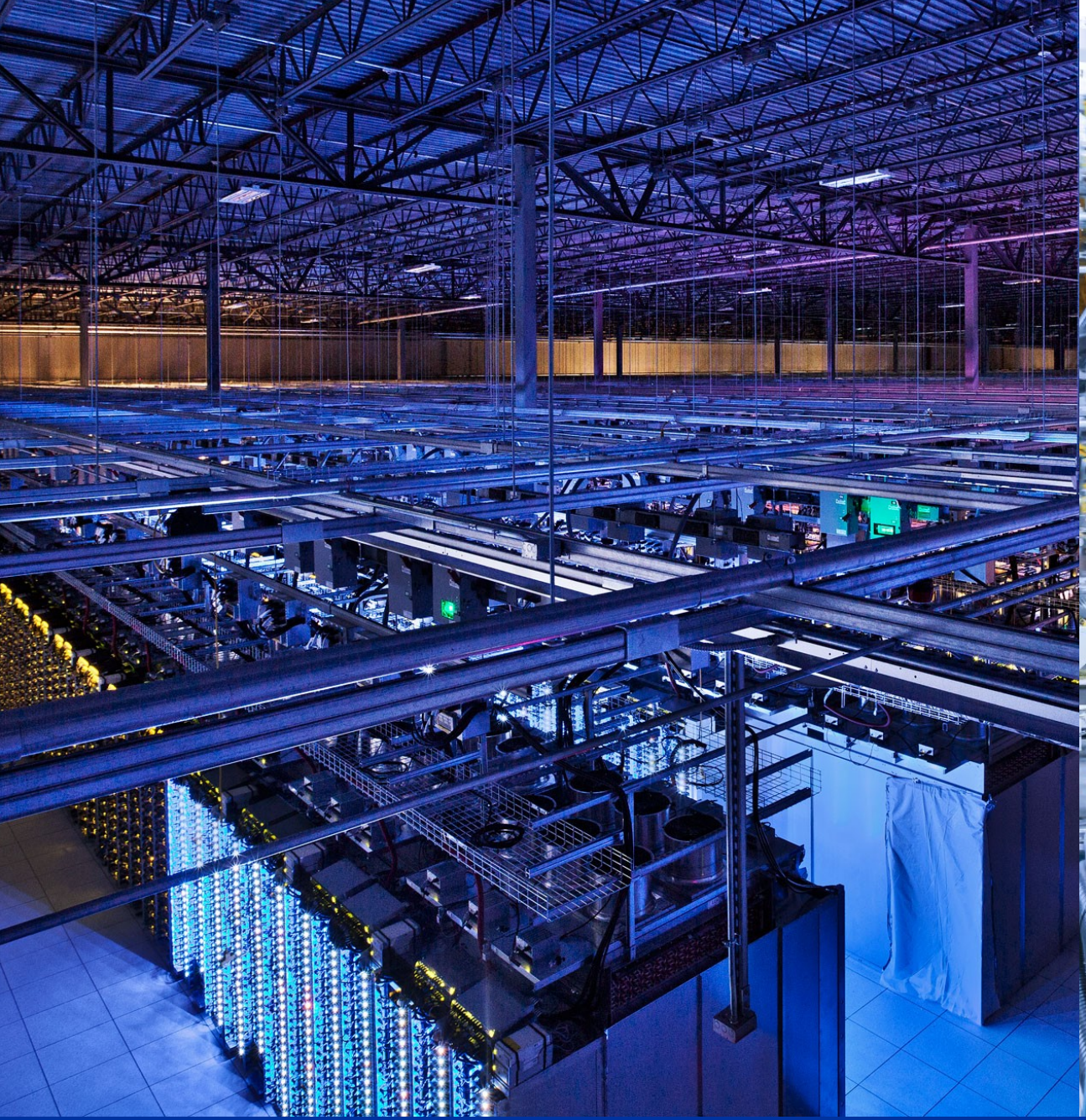

PCSPECIALIST
XFX
PCSPECIALIST
PCSPECIALIST


PCSPECIALIST
XFX
COOLER MASTER


BuildRoot
Making Embedded Linux Easy
CIVIL
INFRASTRUCTURE
PLATFORM
yocto
PROJECT
debian
GNU/Linux
Red Hat
Enterprise Linux
SteamOS


Windows 11
BuildRoot
Making Embedded Linux Easy
CIVIL INFRASTRUCTURE PLATFORM
yocto PROJECT
debian
GNU/Linux
Red Hat Enterprise Linux
SteamOS


BuildRoot
Making Embedded Linux Easy
CIVIL
INFRASTRUCTURE
PLATFORM
yocto
PROJECT
debian
GNU/Linux
Red Hat
Enterprise Linux




Build Root
Making Embedded Linux Easy
CIVIL
INFRASTRUCTURE
PLATFORM
yocto
PROJECT
Red Hat
Enterprise Linux
debian
GNU/Linux

Our Operating Systems Options

Our Operating Systems Options

Higher Control
Higher Maintenance

Lower Control
Lower Maintenance

Our Operating Systems Options

Higher Control
Higher Maintenance

Lower Control
Lower Maintenance

Our Operating Systems Options

Higher Control
Higher Maintenance

Lower Control
Lower Maintenance

linux from scratch

yocto PROJECT

PTX DIST

BuildRoot
Making Embedded Linux Easy

Our Operating Systems Options

Higher Control
Higher Maintenance

Lower Control
Lower Maintenance

linux from scratch

yocto PROJECT

PTX DIST

BuildRoot
Making Embedded Linux Easy

debian GNU/Linux

archlinux

SUSE

Red Hat Enterprise Linux

AlmaLinux

ubuntu

Our Operating Systems Options

Higher Control
Higher Maintenance

Lower Control
Lower Maintenance

linux from scratch

yocto PROJECT

PTX DIST

BuildRoot
Making Embedded Linux Easy

CIVIL INFRASTRUCTURE PLATFORM

archlinux

debian GNU/Linux

SUSE

Red Hat Enterprise Linux

AlmaLinux

ubuntu

Our Operating Systems Options

Higher Control
Higher Maintenance

Lower Control
Lower Maintenance

linux from scratch

yocto PROJECT

PTX DIST

BuildRoot
Making Embedded Linux Easy

The control we need - The maintenance we can afford

CIVIL INFRASTRUCTURE PLATFORM

archlinux

debian GNU/Linux

SUSE

Red Hat Enterprise Linux

AlmaLinux

ubuntu

Our Operating Systems Options

Higher Control
Higher Maintenance

Lower Control
Lower Maintenance

linux from scratch

yocto PROJECT

PTX DIST

BuildRoot
Making Embedded Linux Easy

The best fit CIVIL INFRASTRUCTURE PLATFORM

debian GNU/Linux

archlinux

SUSE

Red Hat Enterprise Linux

AlmaLinux

ubuntu

Why Two Linux Distributions?

Why Two Linux Distributions?

- **It enforces software portability**
  - Everything must work on different Linux environments

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

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

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

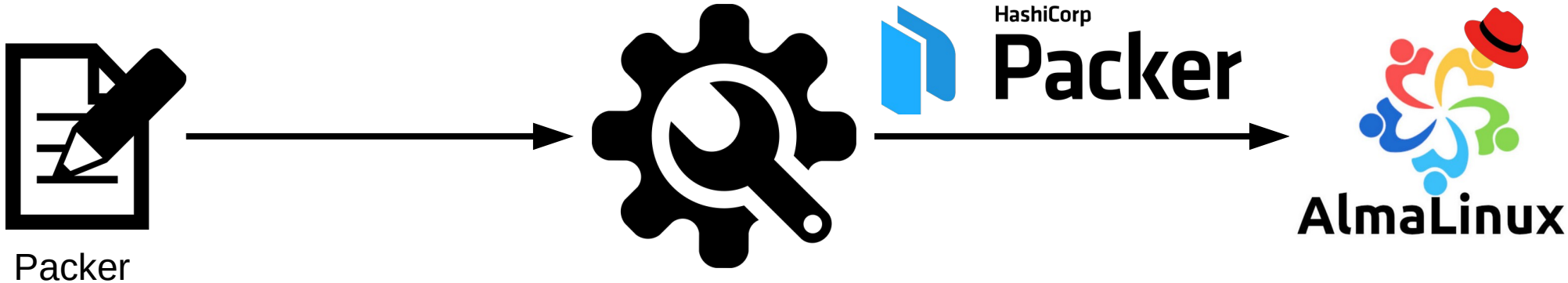
Why Two Linux Distributions?

- **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)

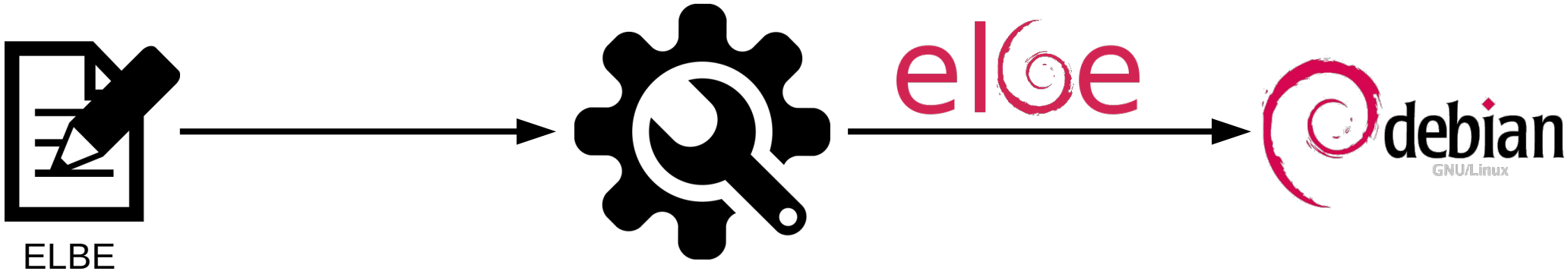
Why Two Linux Distributions?

- **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)

Building Processes



Packer
HCL Configuration file
Custom Scripts



ELBE
XML Configuration File

packer.io almalinux.org
elbe-rfs.org debian.org

First Choice: RHEL Family

First Choice: RHEL Family

It Is Good To Follow Recommendations

First Choice: RHEL Family

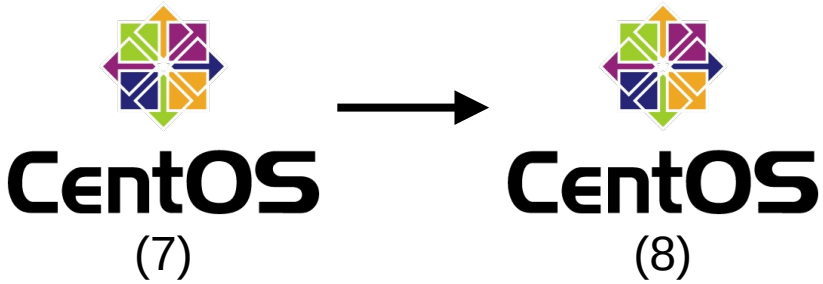
It Is Good To Follow Recommendations

Someone else is offering good services to you

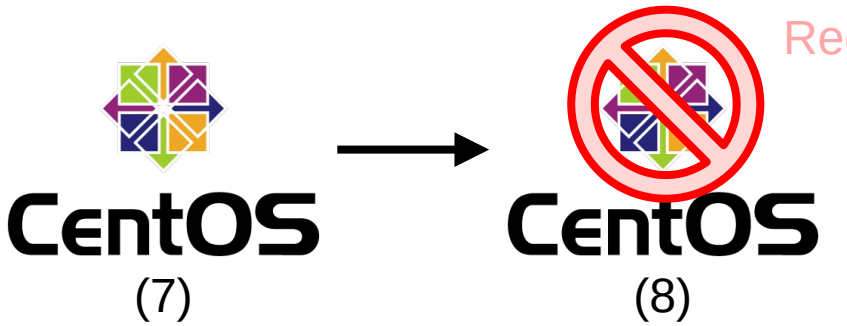
Evolution Of Our First Choice Preference

(7)

Evolution Of Our First Choice Preference

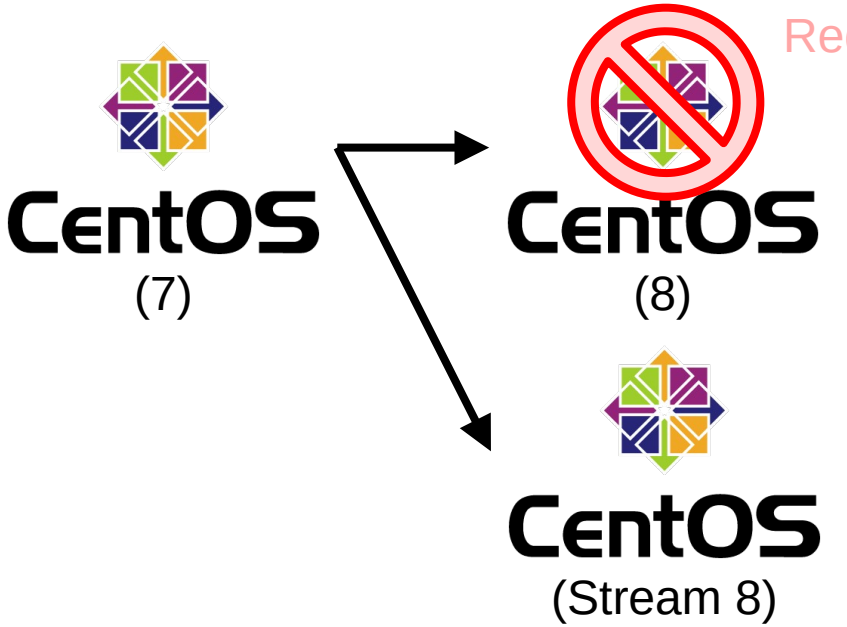


Evolution Of Our First Choice Preference

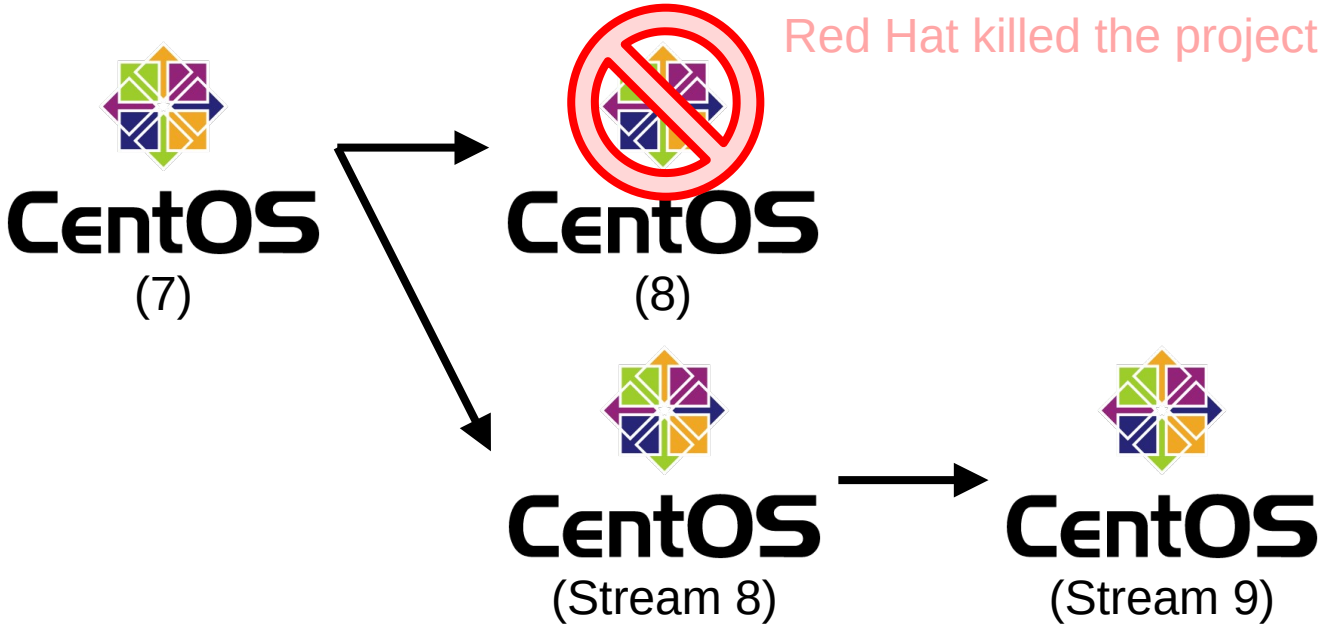


Red Hat killed the project

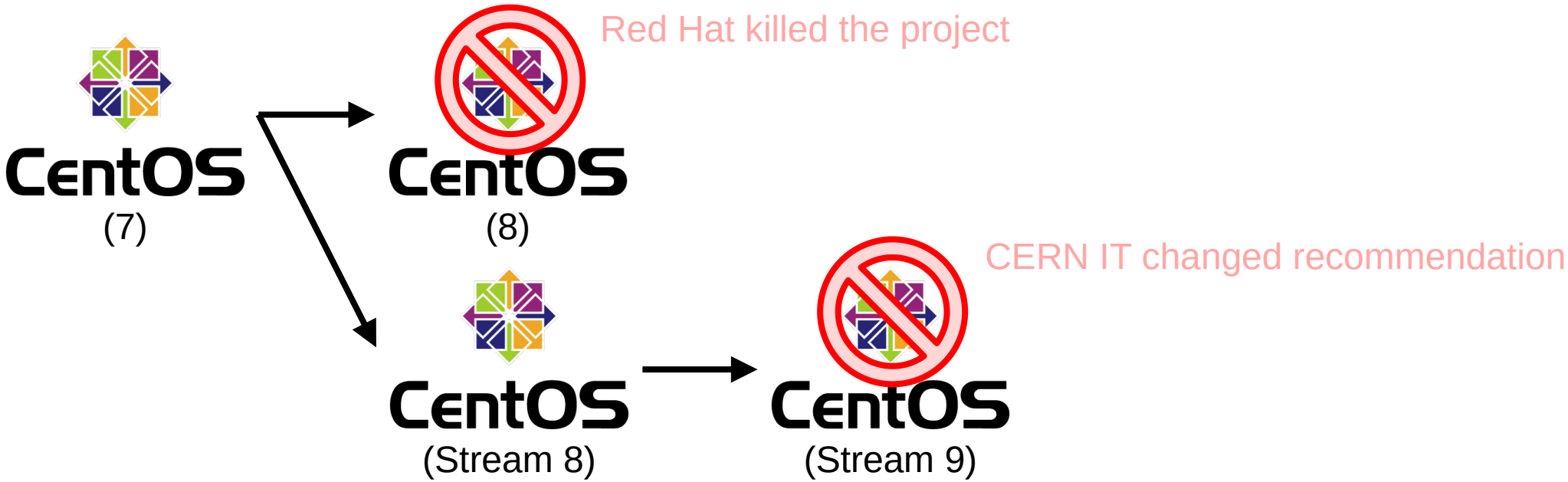
Evolution Of Our First Choice Preference



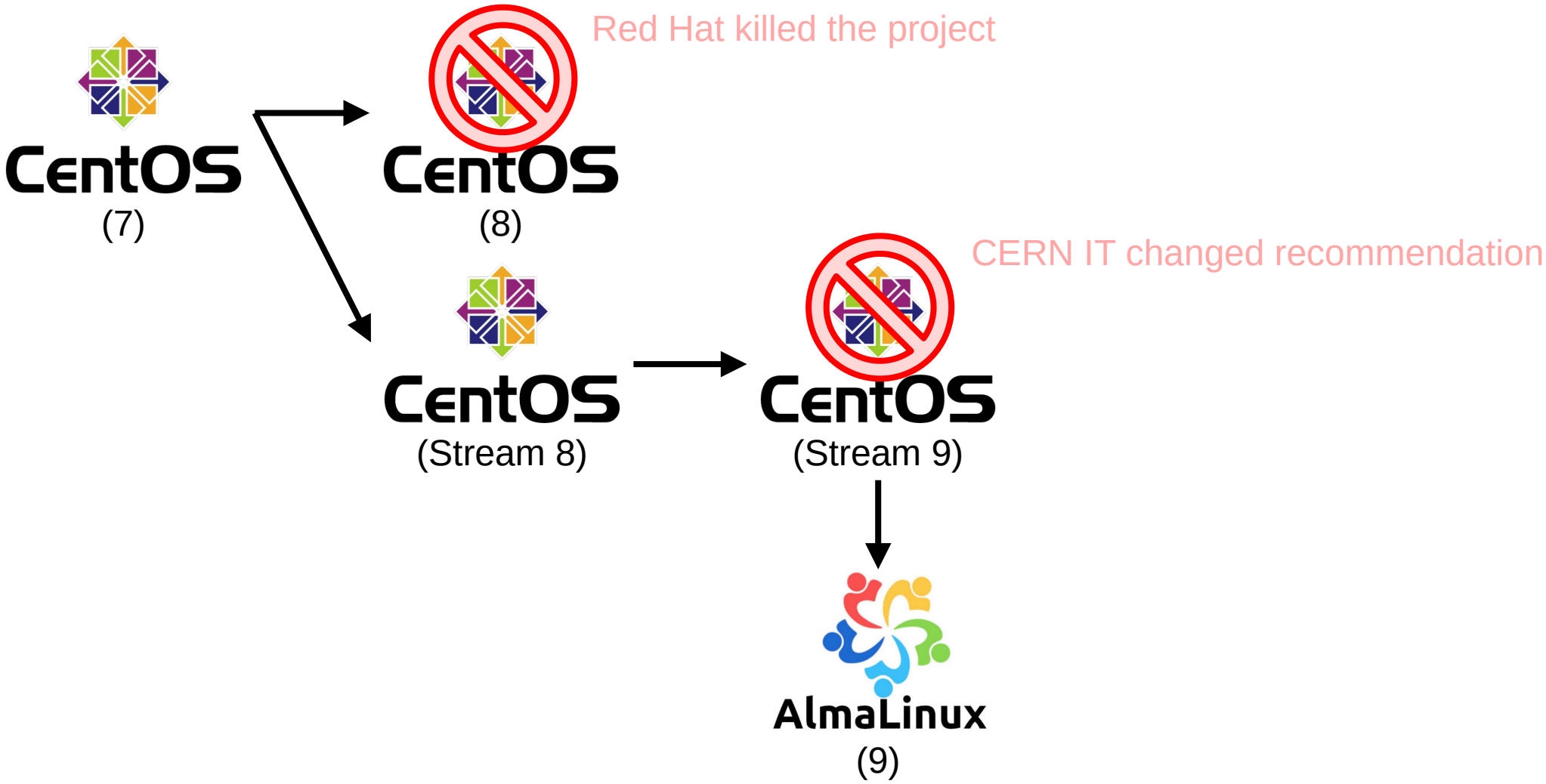
Evolution Of Our First Choice Preference



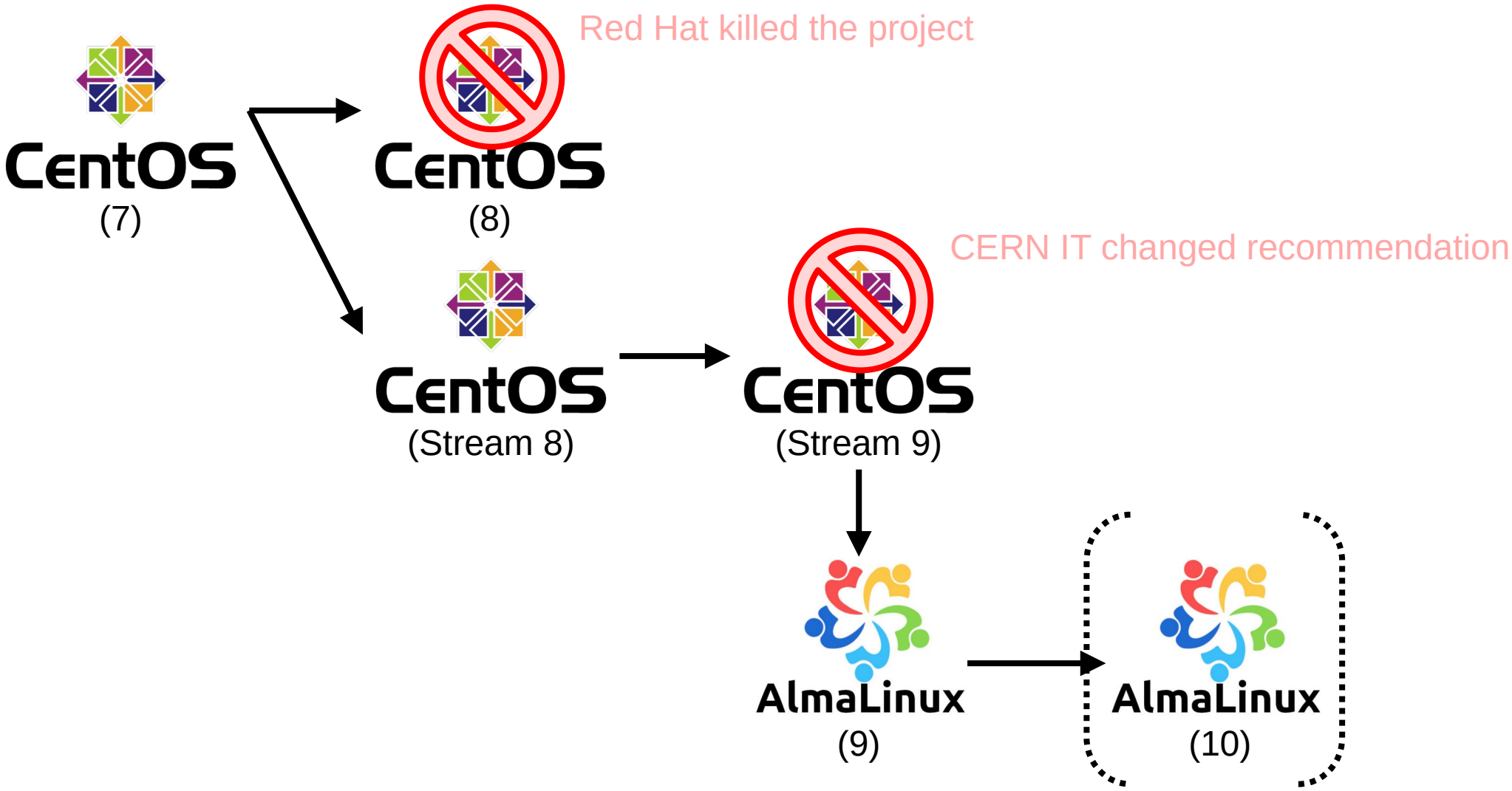
Evolution Of Our First Choice Preference



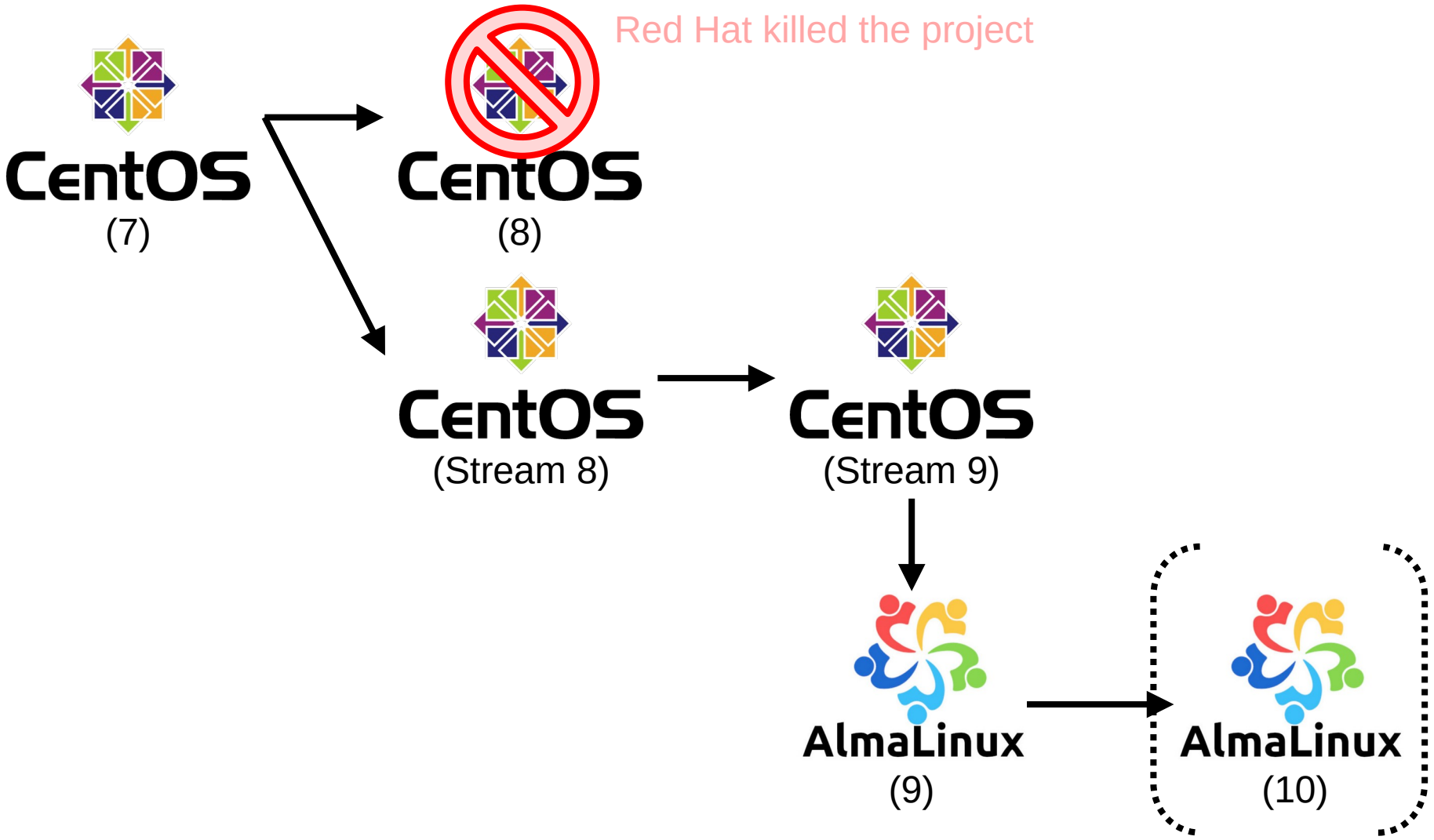
Evolution Of Our First Choice Preference



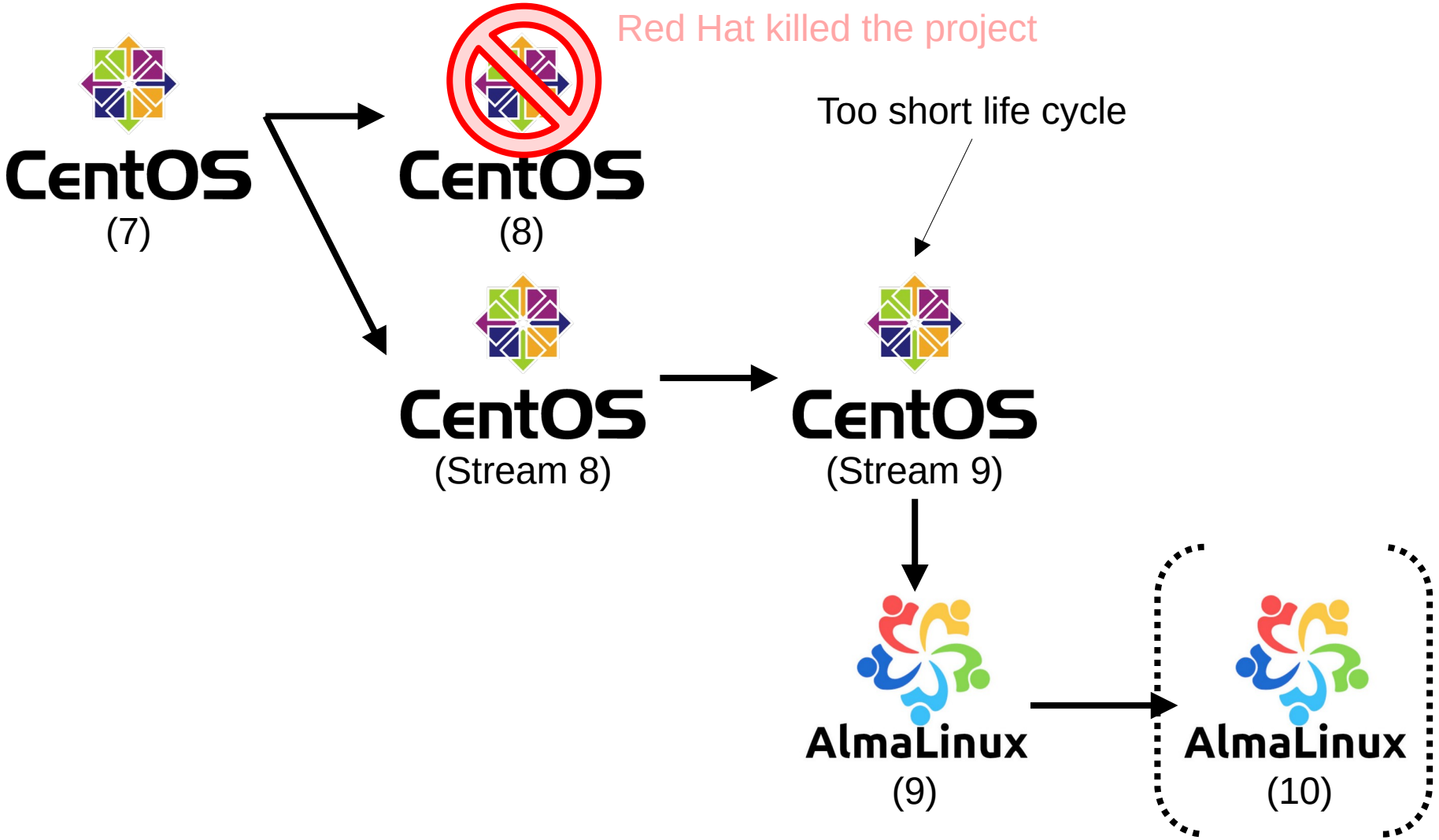
Evolution Of Our First Choice Preference



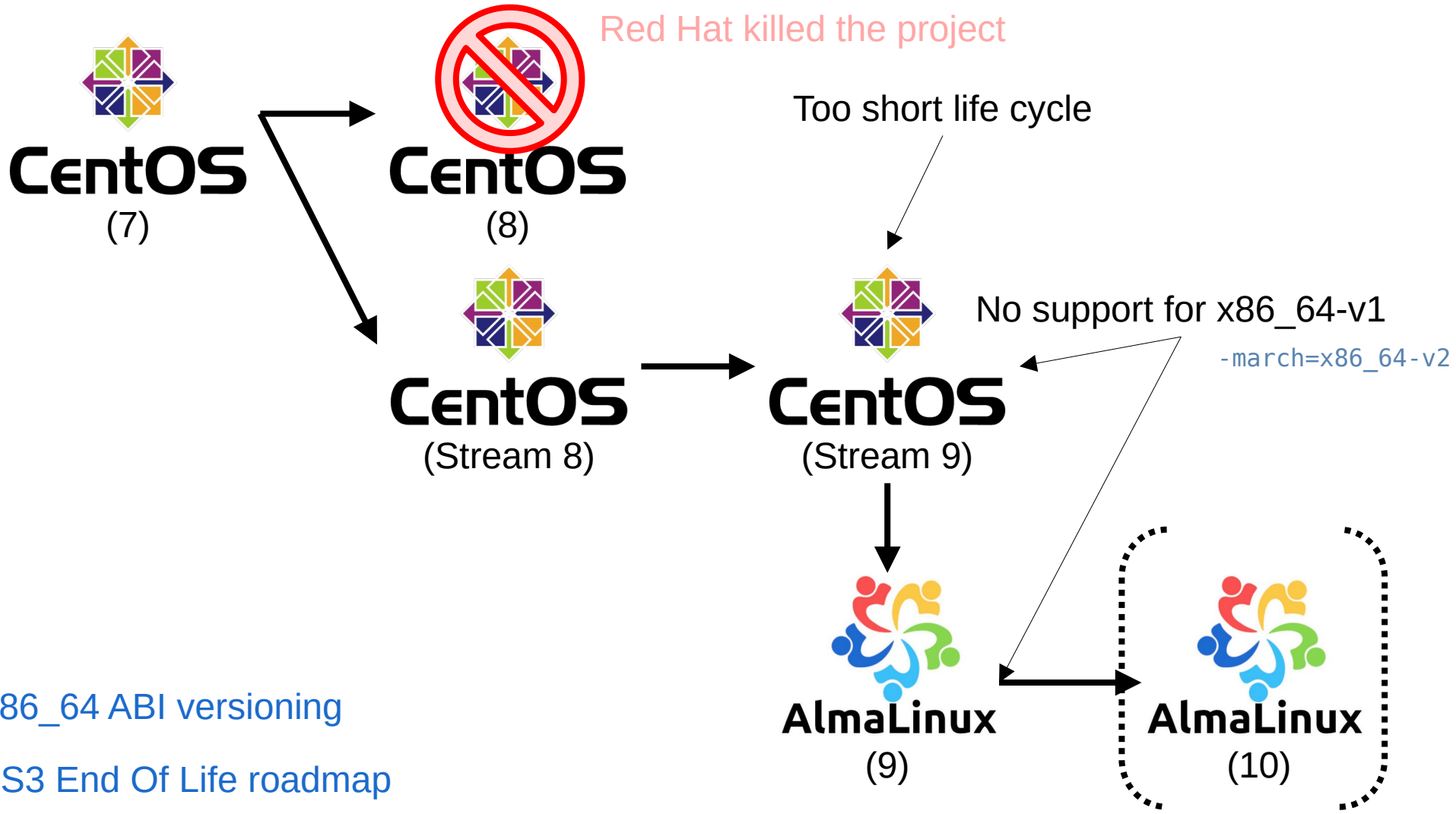
Challenges Within Red Hat Enterprise Linux Family



Challenges Within Red Hat Enterprise Linux Family



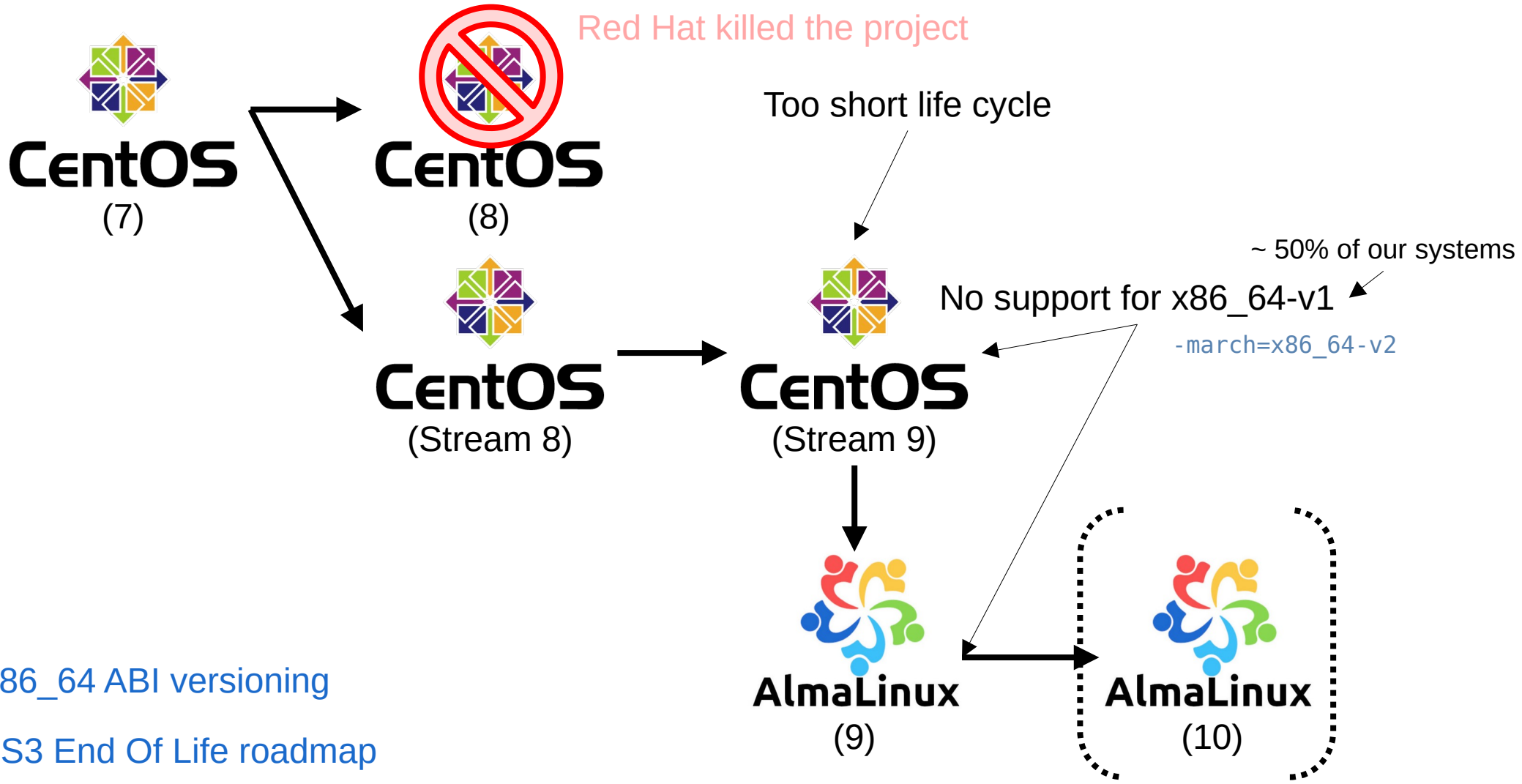
Challenges Within Red Hat Enterprise Linux Family



x86_64 ABI versioning

LS3 End Of Life roadmap

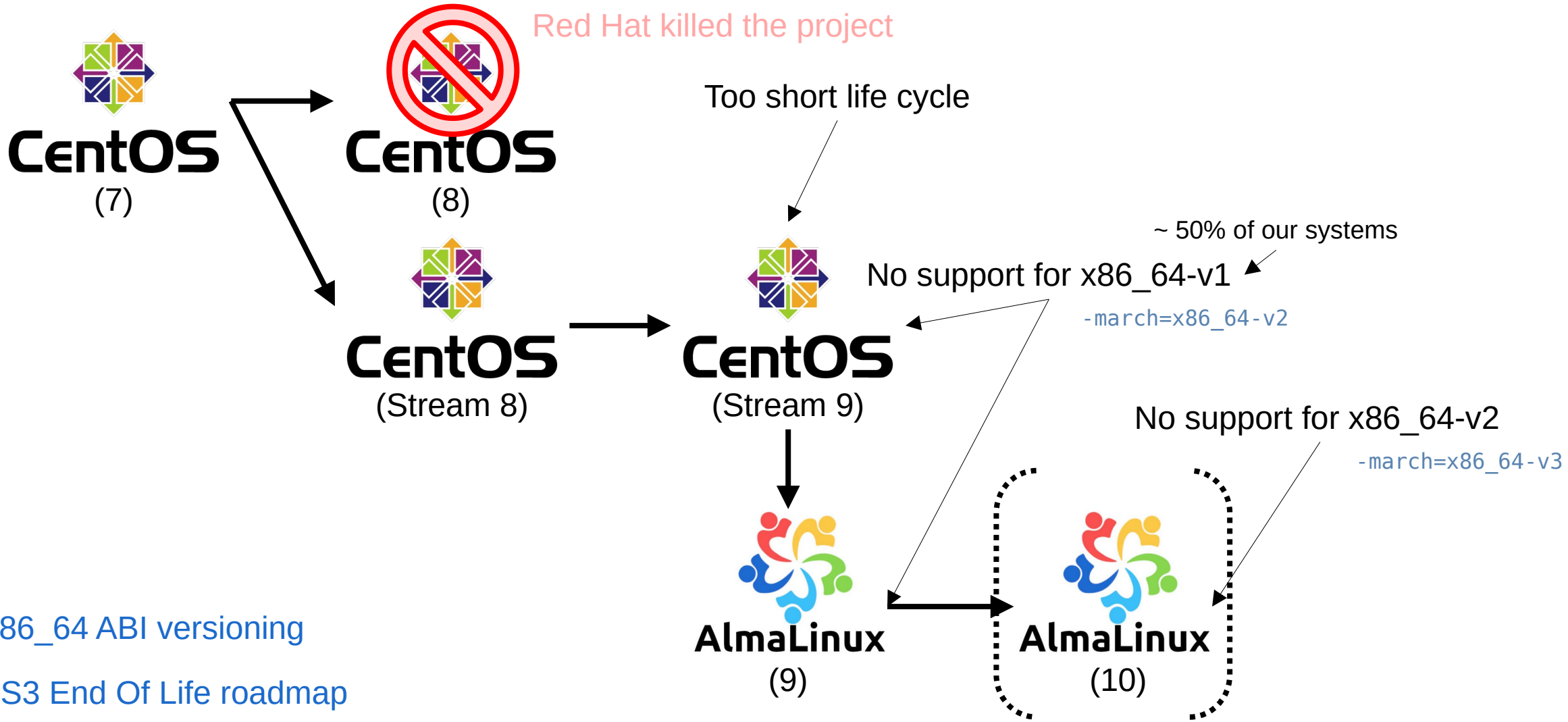
Challenges Within Red Hat Enterprise Linux Family



x86_64 ABI versioning

LS3 End Of Life roadmap

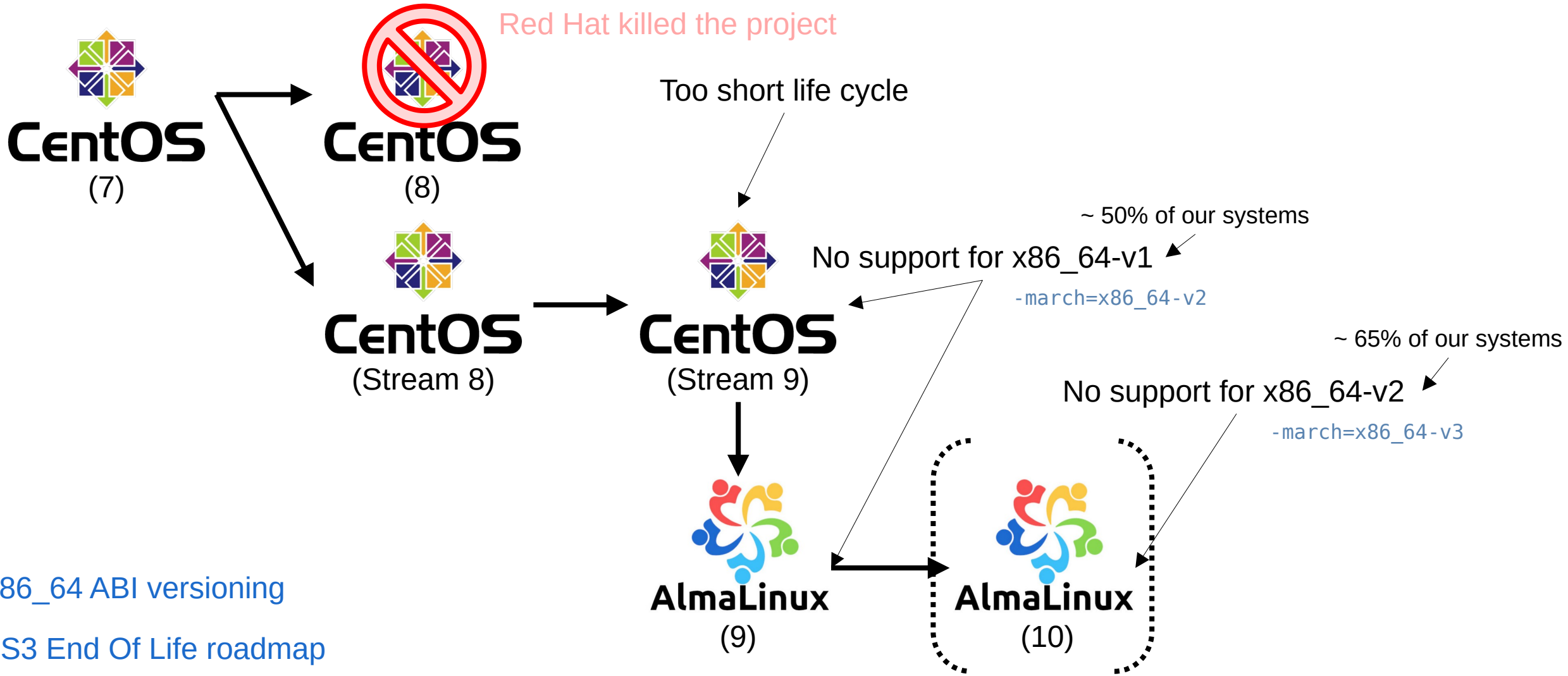
Challenges Within Red Hat Enterprise Linux Family



x86_64 ABI versioning

LS3 End Of Life roadmap

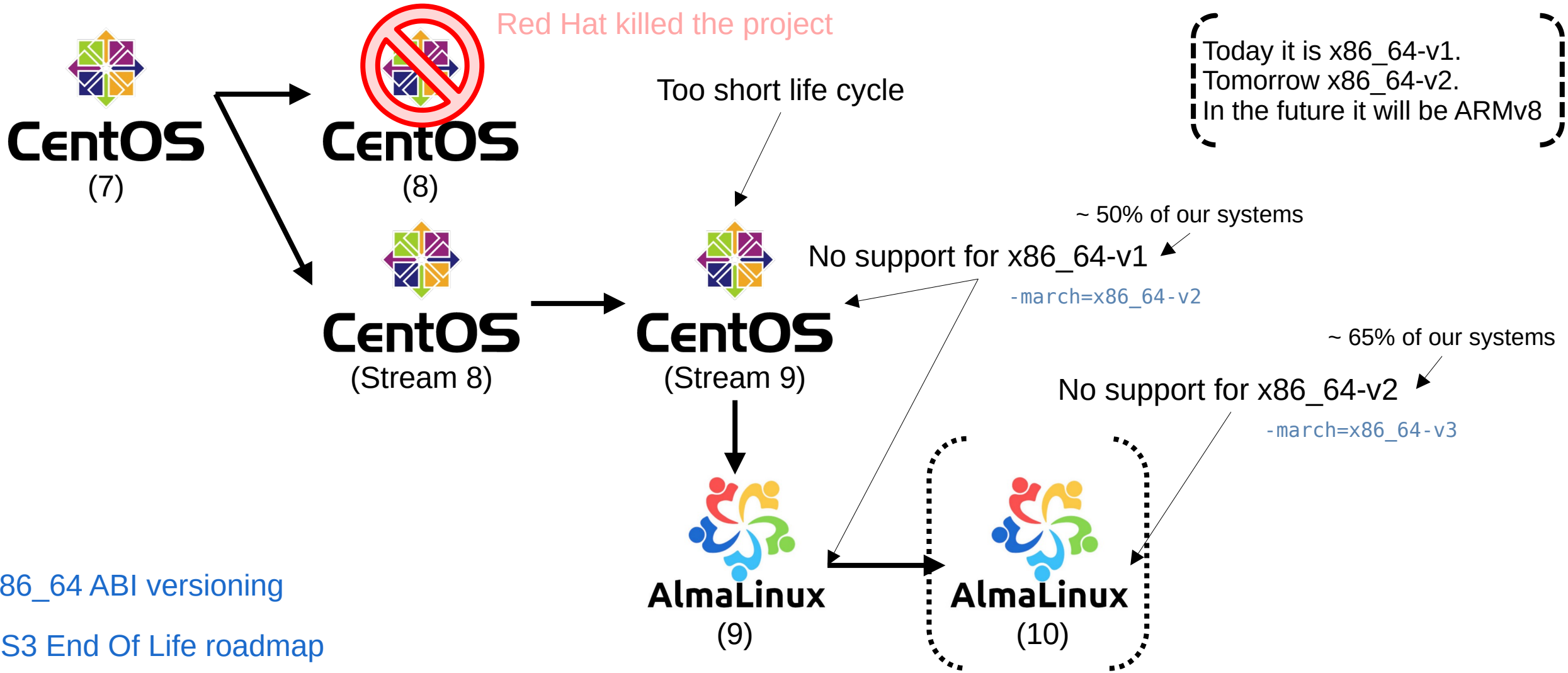
Challenges Within Red Hat Enterprise Linux Family



x86_64 ABI versioning

LS3 End Of Life roadmap

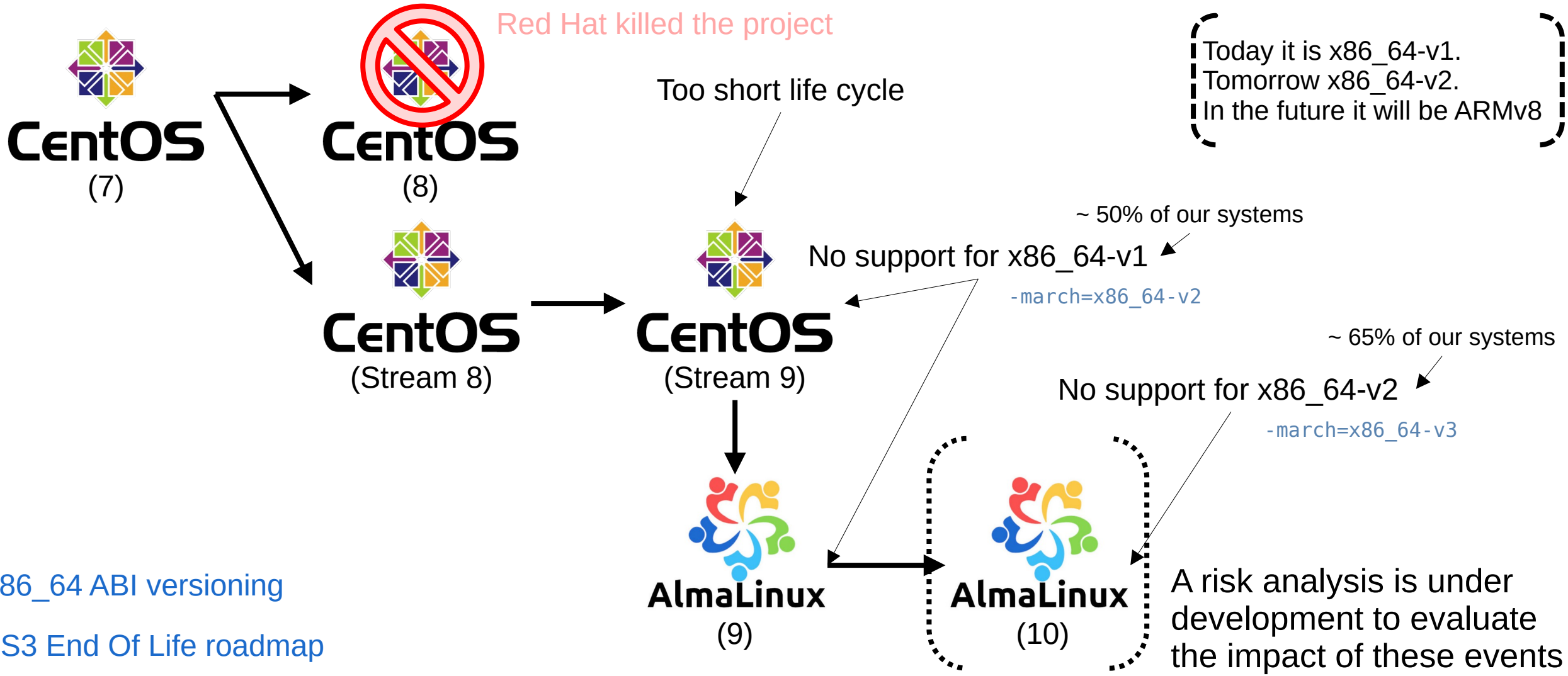
Challenges Within Red Hat Enterprise Linux Family



x86_64 ABI versioning

LS3 End Of Life roadmap

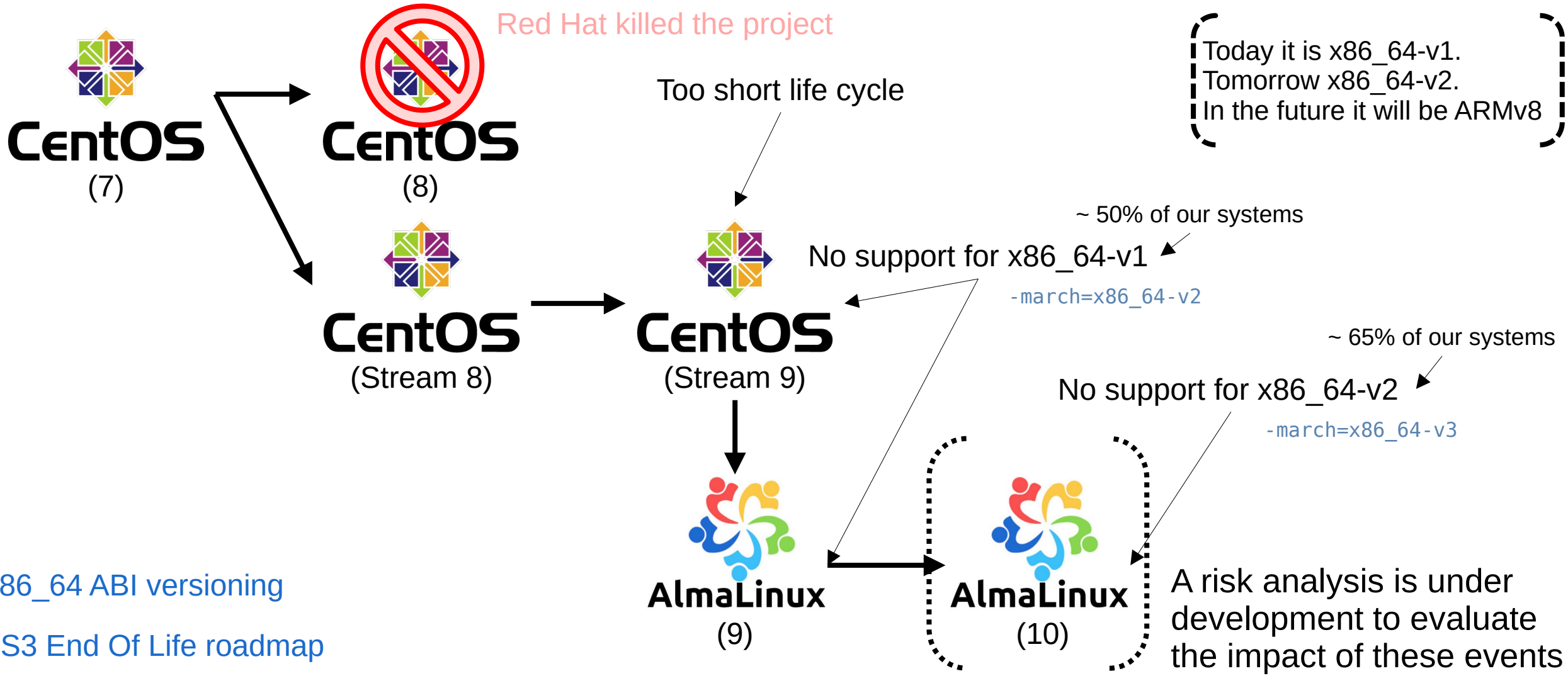
Challenges Within Red Hat Enterprise Linux Family



x86_64 ABI versioning

LS3 End Of Life roadmap

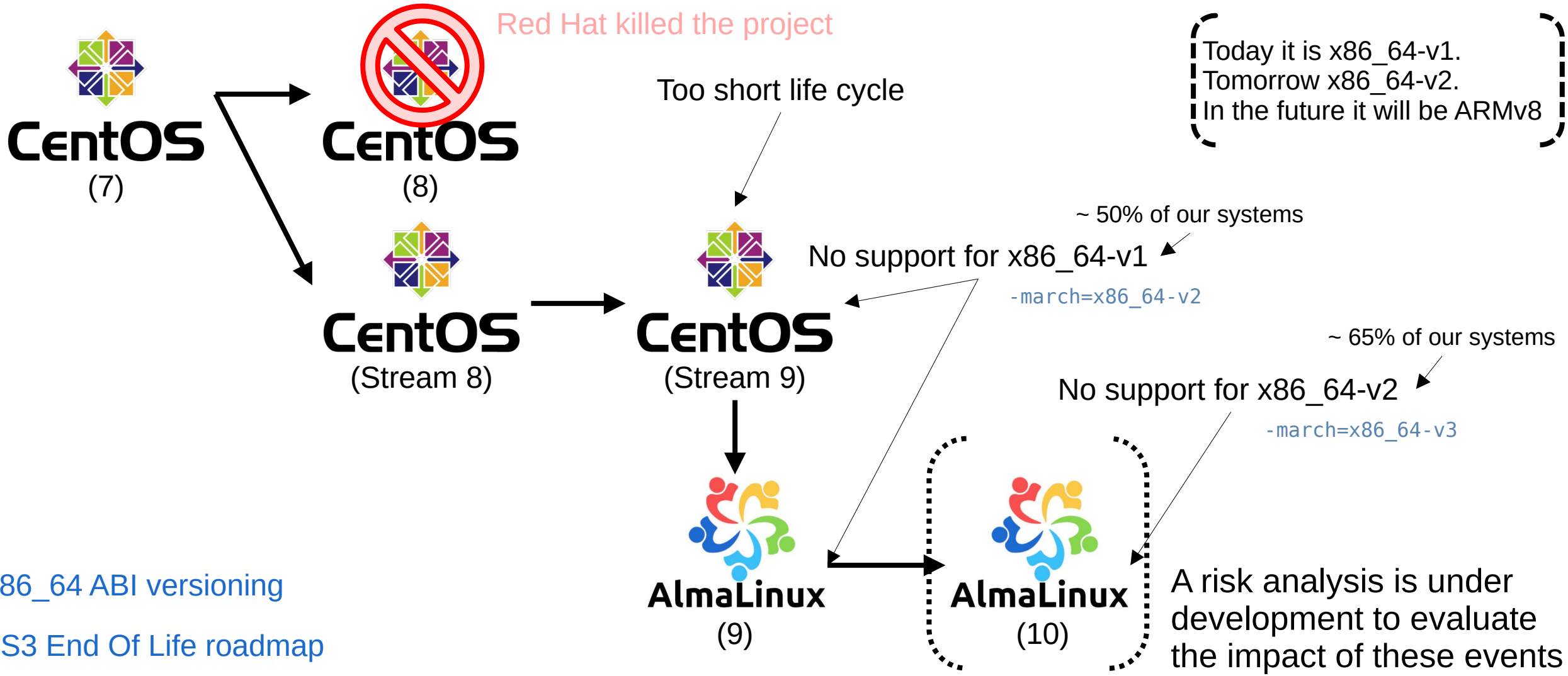
Challenges Within Red Hat Enterprise Linux Family



x86_64 ABI versioning

LS3 End Of Life roadmap

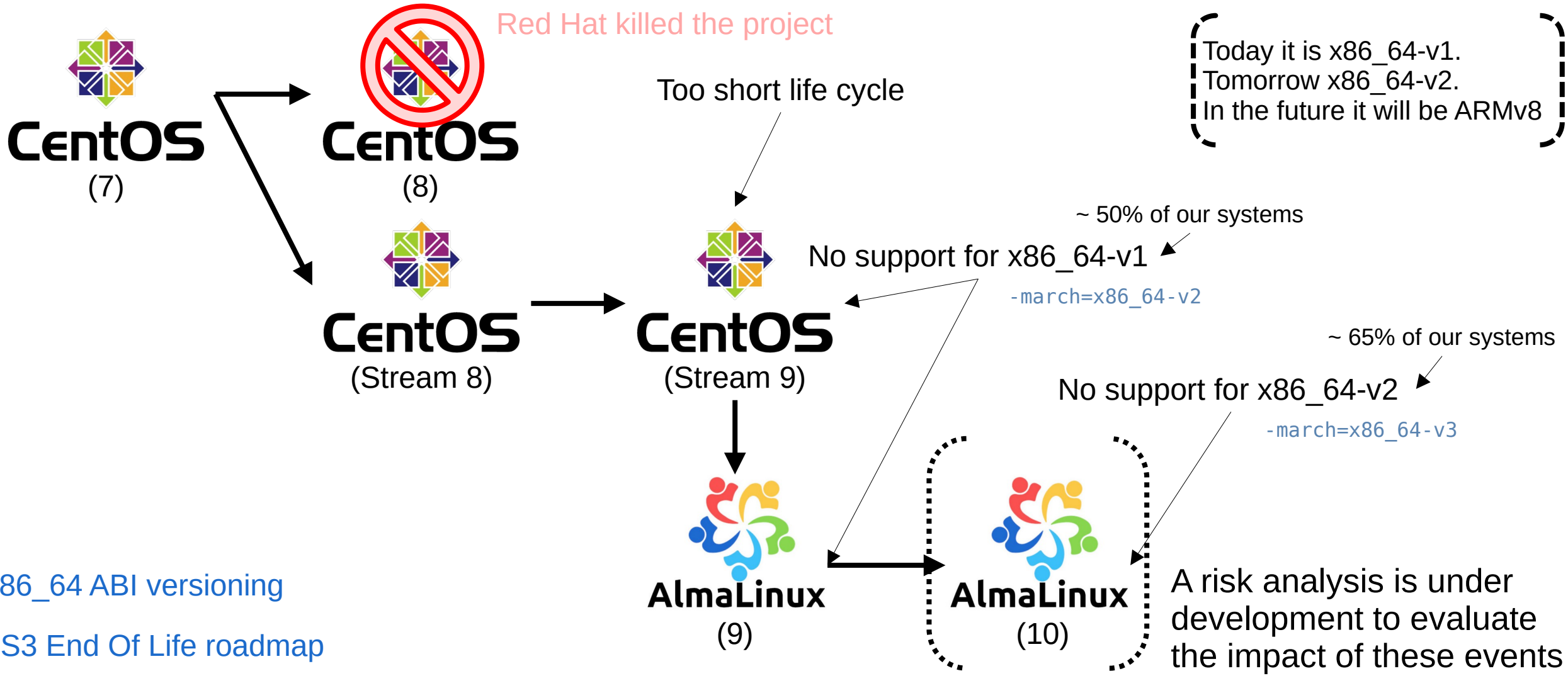
Challenges Within Red Hat Enterprise Linux Family



x86_64 ABI versioning

LS3 End Of Life roadmap

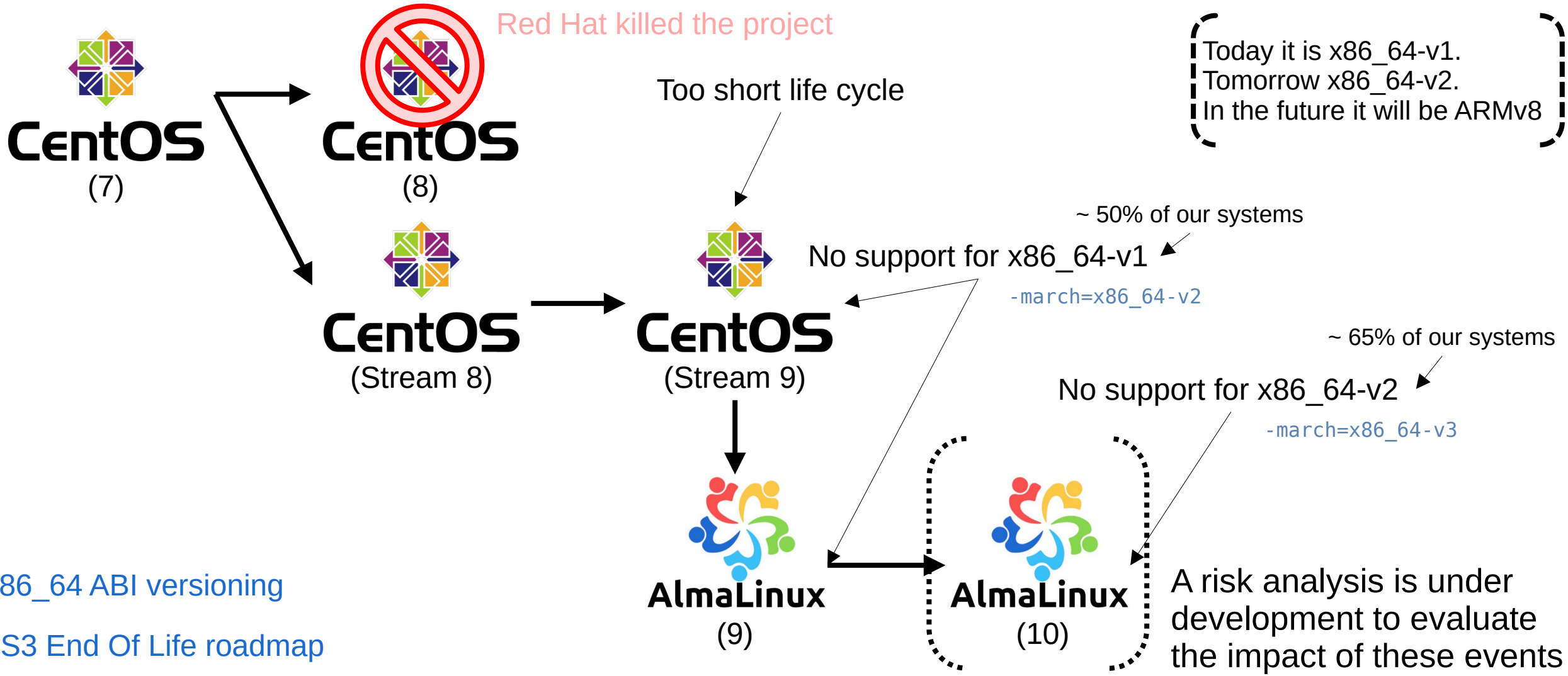
Challenges Within Red Hat Enterprise Linux Family



x86_64 ABI versioning

LS3 End Of Life roadmap

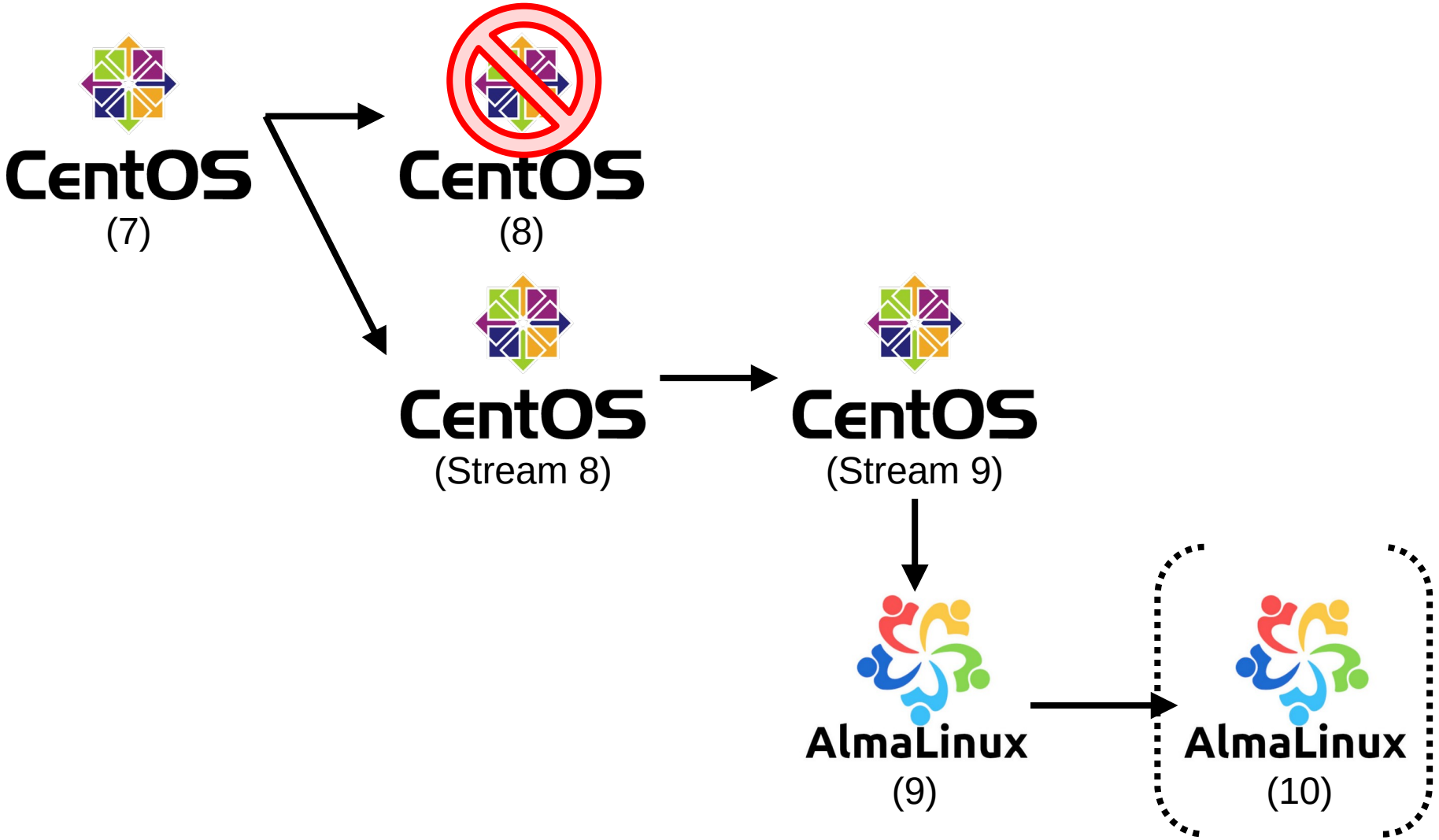
Challenges Within Red Hat Enterprise Linux Family



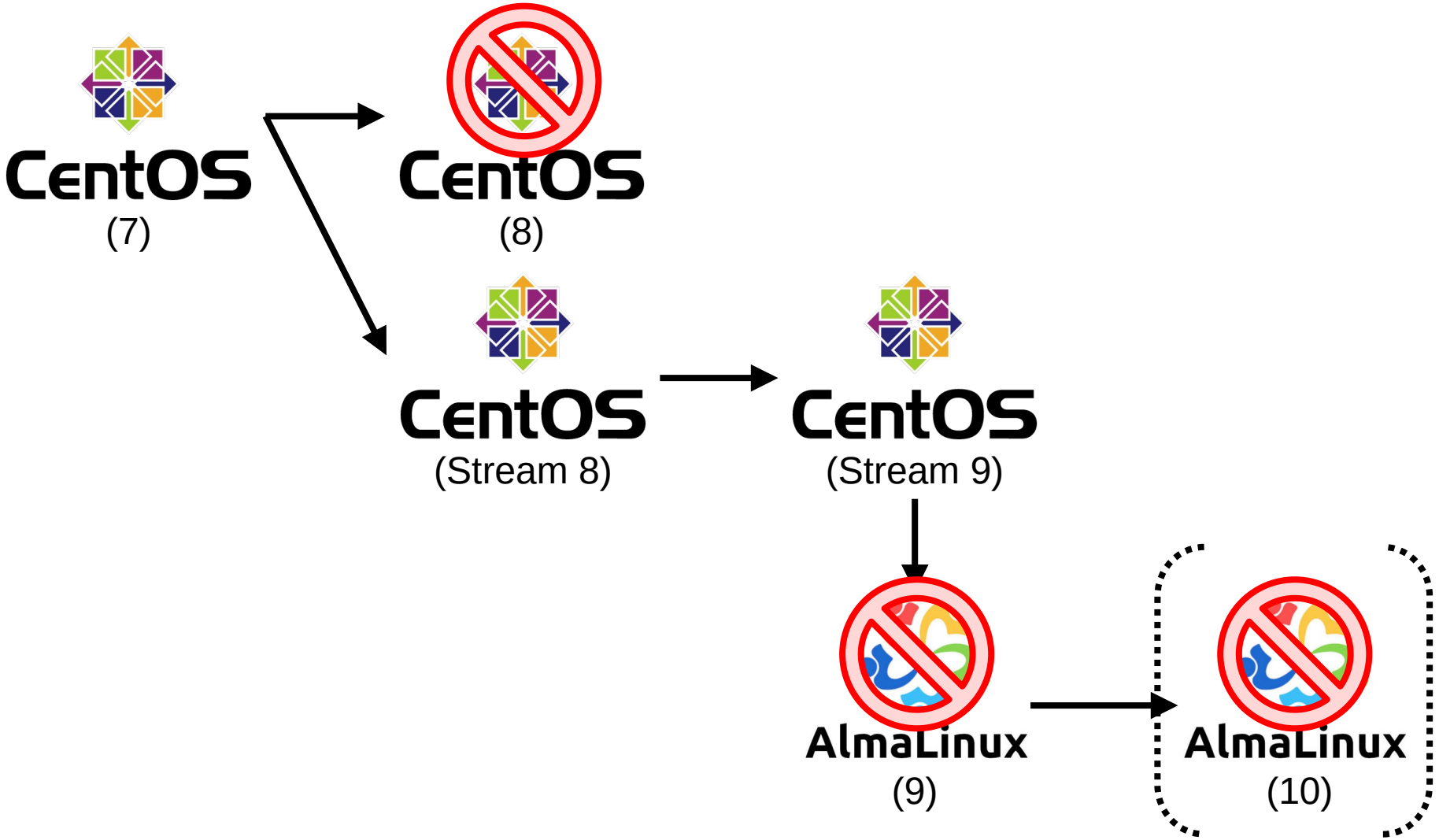
x86_64 ABI versioning

LS3 End Of Life roadmap

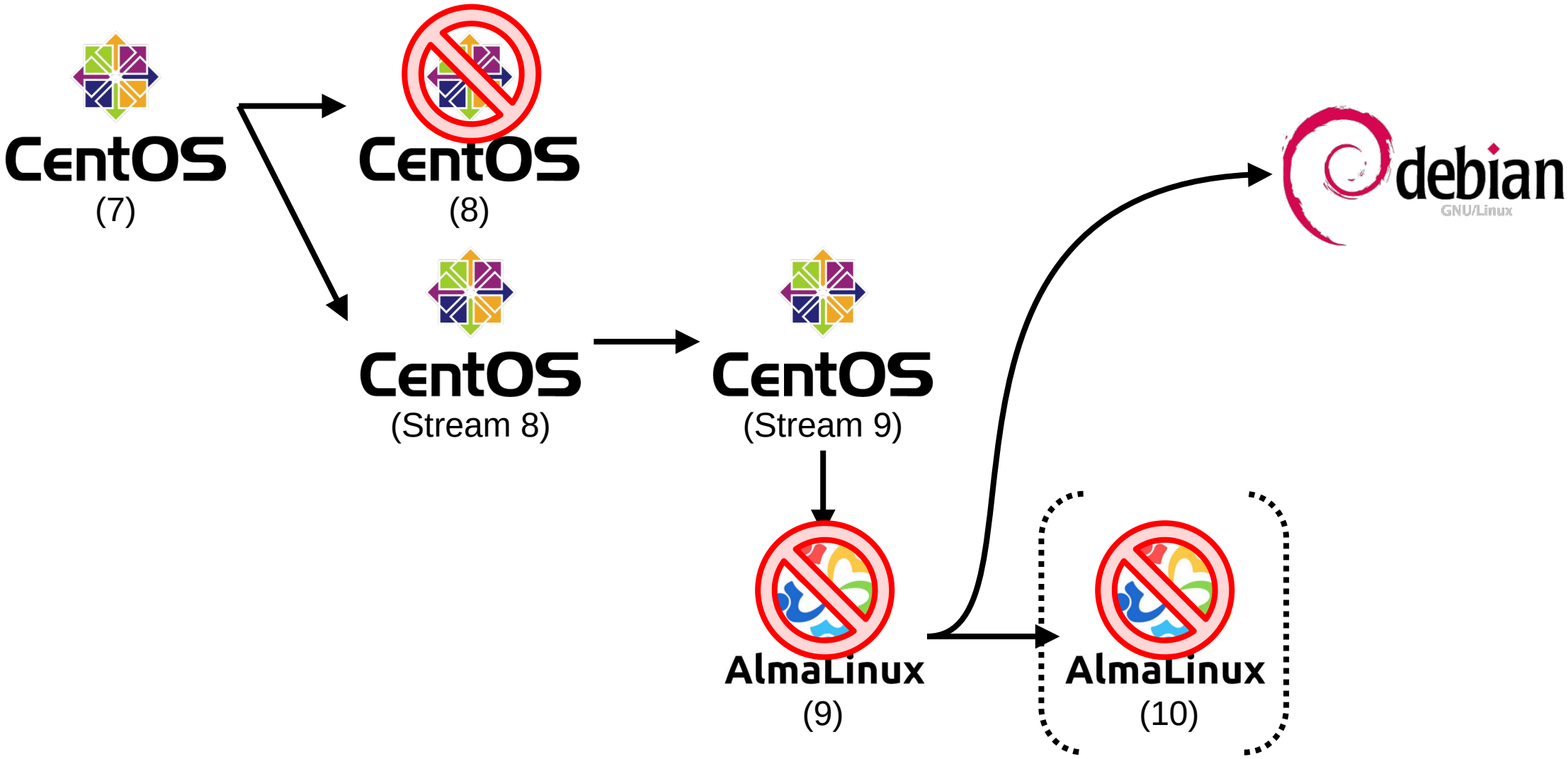
A Possible Scenario: Promote The Second Choice



A Possible Scenario: Promote The Second Choice



A Possible Scenario: Promote The Second Choice



Conclusions

Conclusions

- **Because we speak about embedded systems**
  - Not servers, nor service oriented applications.

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.

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”

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

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

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

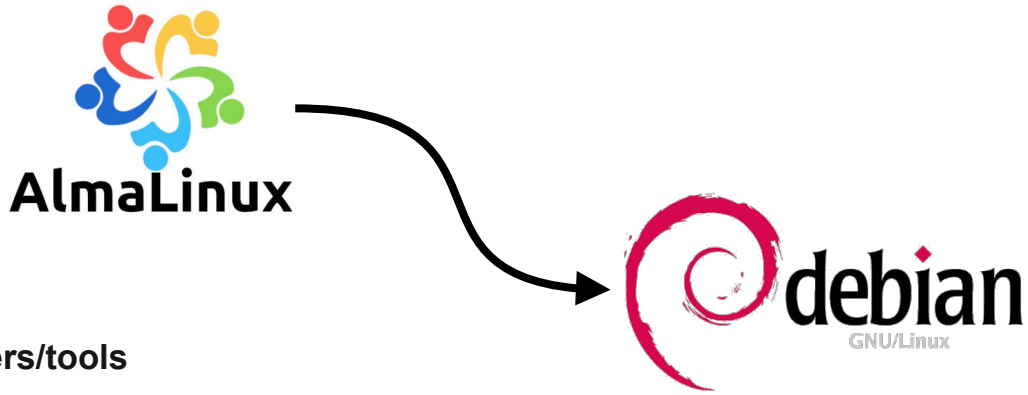
Conclusions

- **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`

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

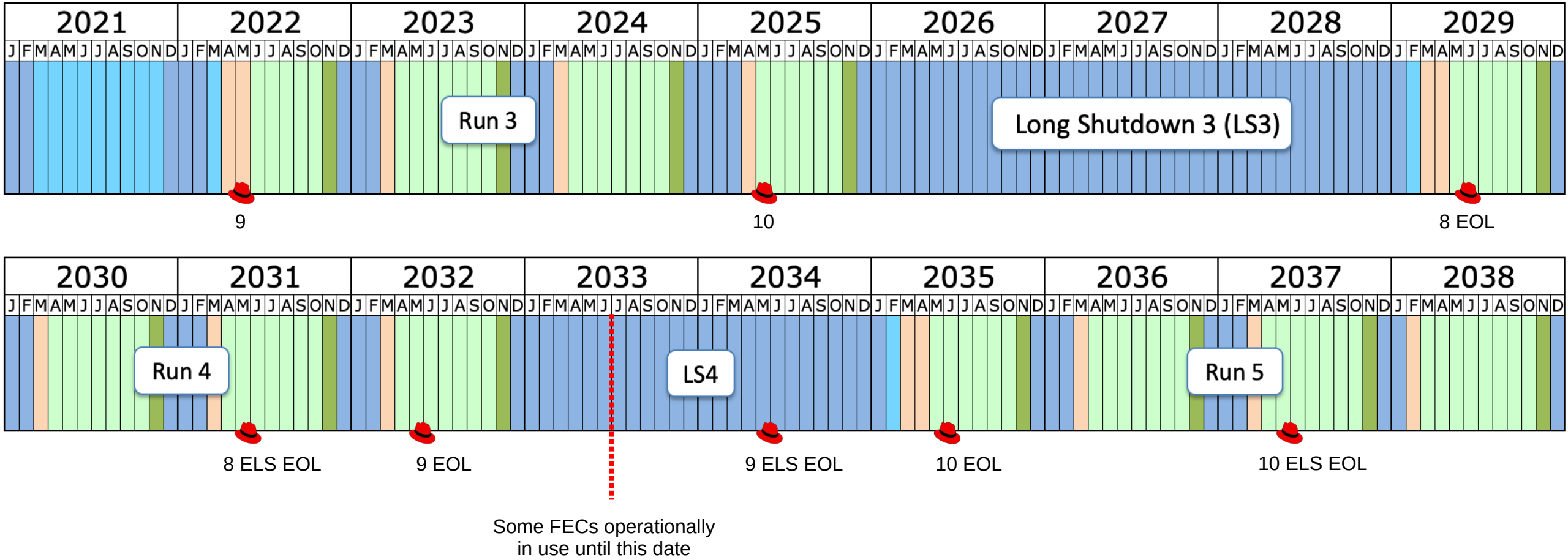
Conclusions



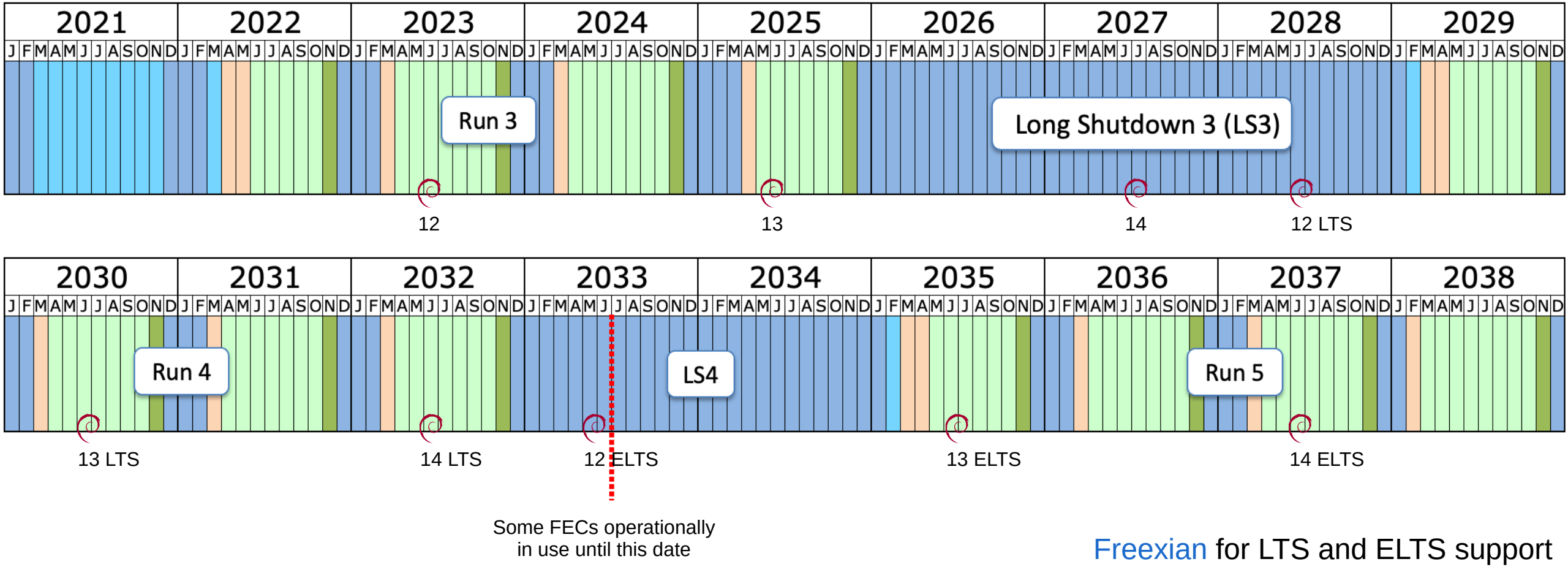
- **Because we speak about embedded systems**
  - Not servers, nor service oriented applications.
- **Because we develop custom hardware and dedicated drivers/tools**
  - Hence, we want finer control over these systems.
- **Because RHEL Family of distributions aims at “server farms” and workstations**
  - 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+)

BACKUP SLIDES

FEC OS Renovation Schedule: Red Hat Enterprise Linux



FEC OS Renovation Schedule: Debian



Freexian for LTS and ELTS support