

Un pingüino en tu coche

Los retos que plantea crear una distribución de Linux para automóviles

Sergio López Pascual

Principal Software Engineer @ Red Hat

CentOS AutoSD

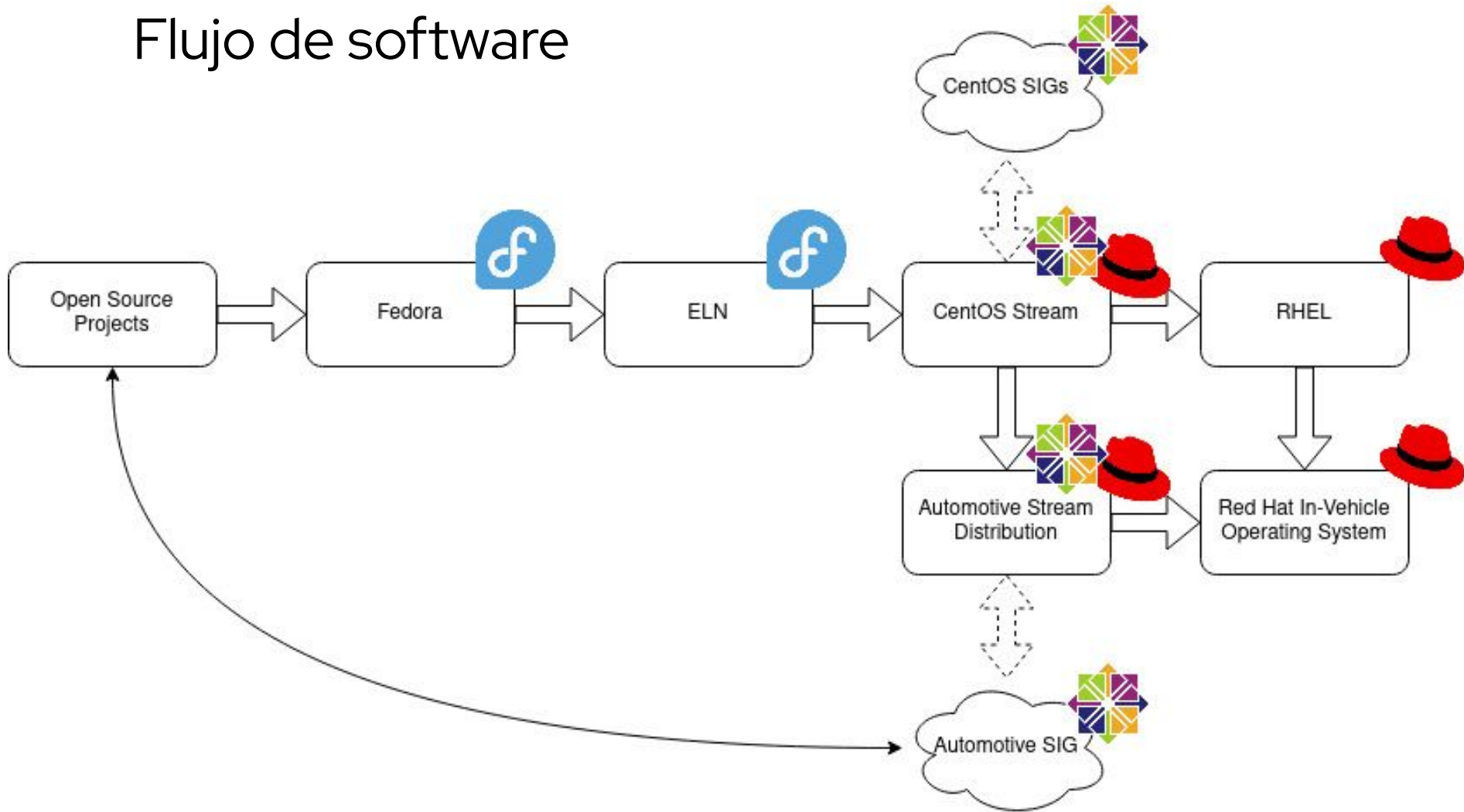
CentOS Automotive Stream Distribution



CentOS

**Automotive
Stream Distribution**

Flujo de software



CentOS AutoSD

¿Cómo puede servir como una base común?

1. Capacidad para influir en la plataforma

CentOS AutoSD

¿Cómo puede servir como una base común?

1. Capacidad para influir en la plataforma
2. Hardware que soporte ARM SystemReady

CentOS AutoSD

¿Cómo puede servir como una base común?

1. Capacidad para influir en la plataforma
2. Hardware que soporte ARM SystemReady
3. Drivers “upstream first”

CentOS AutoSD

¿Cómo puede servir como una base común?

1. Capacidad para influir en la plataforma
2. Hardware que soporte ARM SystemReady
3. Drivers “upstream first”
4. Containers sobre una imagen inmutable

Distros OSTree



SILVERBLUE



CentOS
AutoSD

Hirte

Coordinando servicios en múltiples nodos

- Un sistema capaz de coordinar servicios de systemd distribuidos a través de múltiples nodos
- Se comunica con systemd a mediante D-Bus
- Es capaz de manejar contenedores como servicios usando [podman](#) y [quadlet](#)
- <https://github.com/containers/hirte>

Optimizaciones en Podman

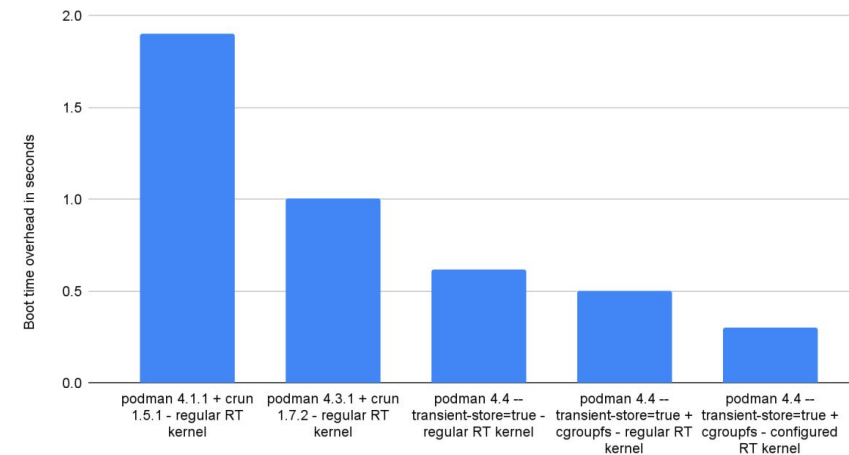
Arrancando containers en 300ms

How we achieved a 6-fold increase in Podman startup speed

By cutting unnecessary processes, you can realize near-real-time container startup, critical in cars and other time-sensitive applications.

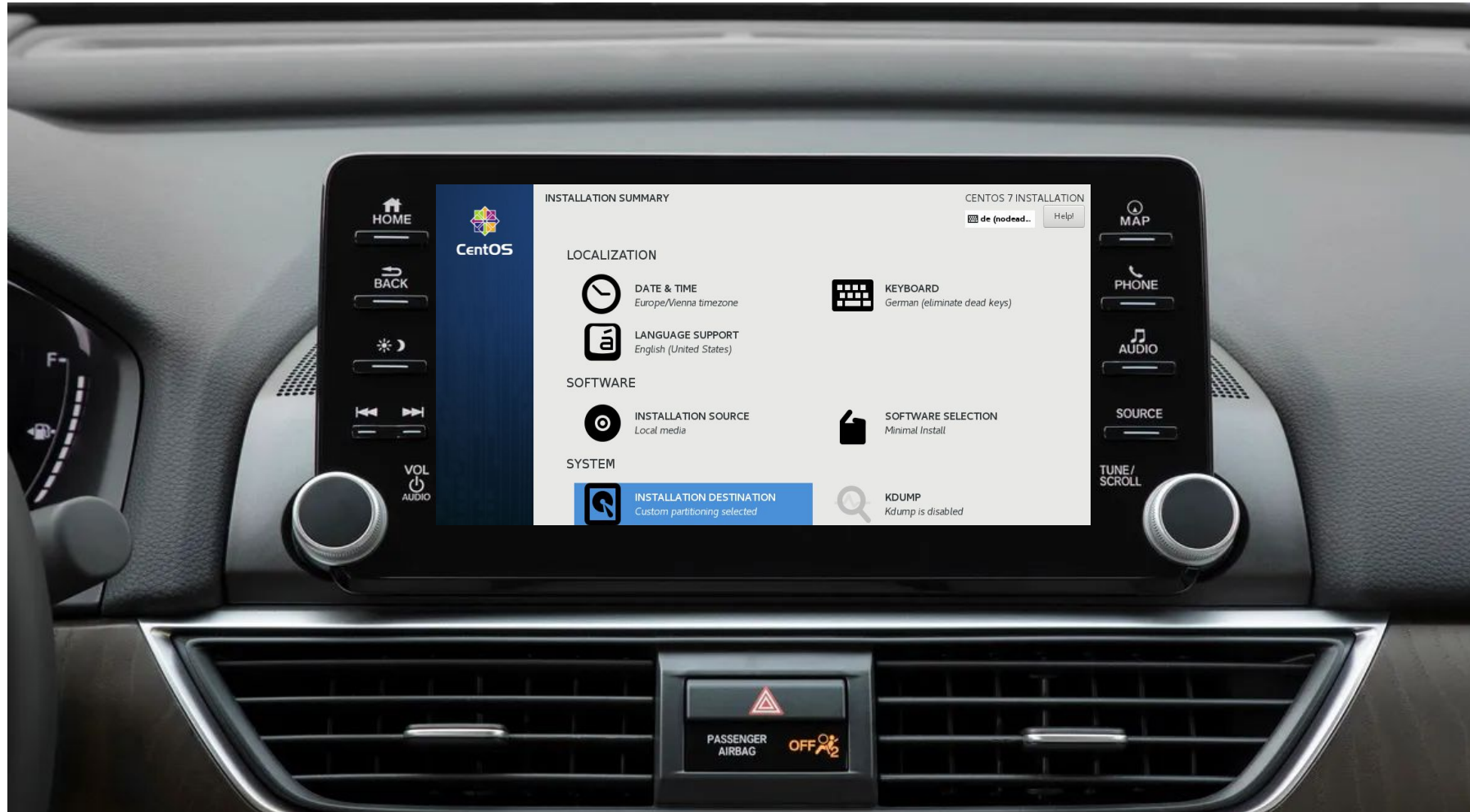
Posted: April 13, 2023 | 8 min read | [Dan Walsh \(Red Hat\)](#), [Alexander Larsson \(Red Hat\)](#), [Pierre-Yves Chibon \(Red Hat\)](#)

Evolution of podman boot time overhead (on a rpi4)



<https://www.redhat.com/sysadmin/speed-containers-podman-raspberry-pi>

Instalando AutoSD en un vehículo



Era broma... 🤪



Functional Safety

A ver cómo explico esto sin que os durmáis

- **ISO 26262**
 - Niveles de riesgo y de rigurosidad en el desarrollo de ASIL-A a ASIL-D, siendo este último el más alto.
 - Quality Management
- **Mixed criticality**
 - Correr, en un mismo sistema, elementos con distinta clasificación de riesgo (por ejemplo, ASIL-B y QM)

Mixed Criticality en AutoSD

Namespaces, cgroups y SELinux

- **Dos particiones**
 - Partición ASIL-X
 - Partición Quality Management
- **Creando una partición QM**
 - podman run --rootfs ...
 - <https://github.com/containers/qm>

Android como QM

Pues al final sí que hacía falta Virtualización

- Hay un interés creciente en correr Android directamente en el vehículo
 - Car Play/Android Auto es una fuente continua de problemas y una solución muy poco eficiente desde el punto de vista energético
- Correr Android en un cartucho independiente
 - Otro kernel, más hardware enablement, más CI.
 - Extensiones para coordinar distintos componentes del vehículo
 - Más coste en hardware
- Correr Android en una VM
 - Abstraer el soporte de HW en VirtIO
 - Coordinar entre componentes desde el VMM (Virtual Machine Monitor)
 - Consolidar en menos HW

Mejorar el soporte de Android en QEMU

Sonido, pantalla táctil, SCMI, aceleración gráfica...

- Mejoras en QEMU
 - vhost-user-video
 - vhost-user-snd
 - vhost-user-gpu con virgl
 - vhost-user-scmi
 - [virtio-input-multitouch](#)
- Herramientas para manejar imágenes de Android
 - [cvd2img](#)
 - [start-avm](#)
 - Imagen OCI

¿Preguntas?