Yocto's hidden gem: OTA and seamless updates with systemd-sysupdate

Emmanuele Bassi

2025-10-01

igalia

Who am I?

- Emmanuele
  - */em.ma.nu'ɛ.le/*
  - *Em·ma·nu·è·le*
- Core Team @ Igalia
- GTK, GNOME, occasional app developer

igalia

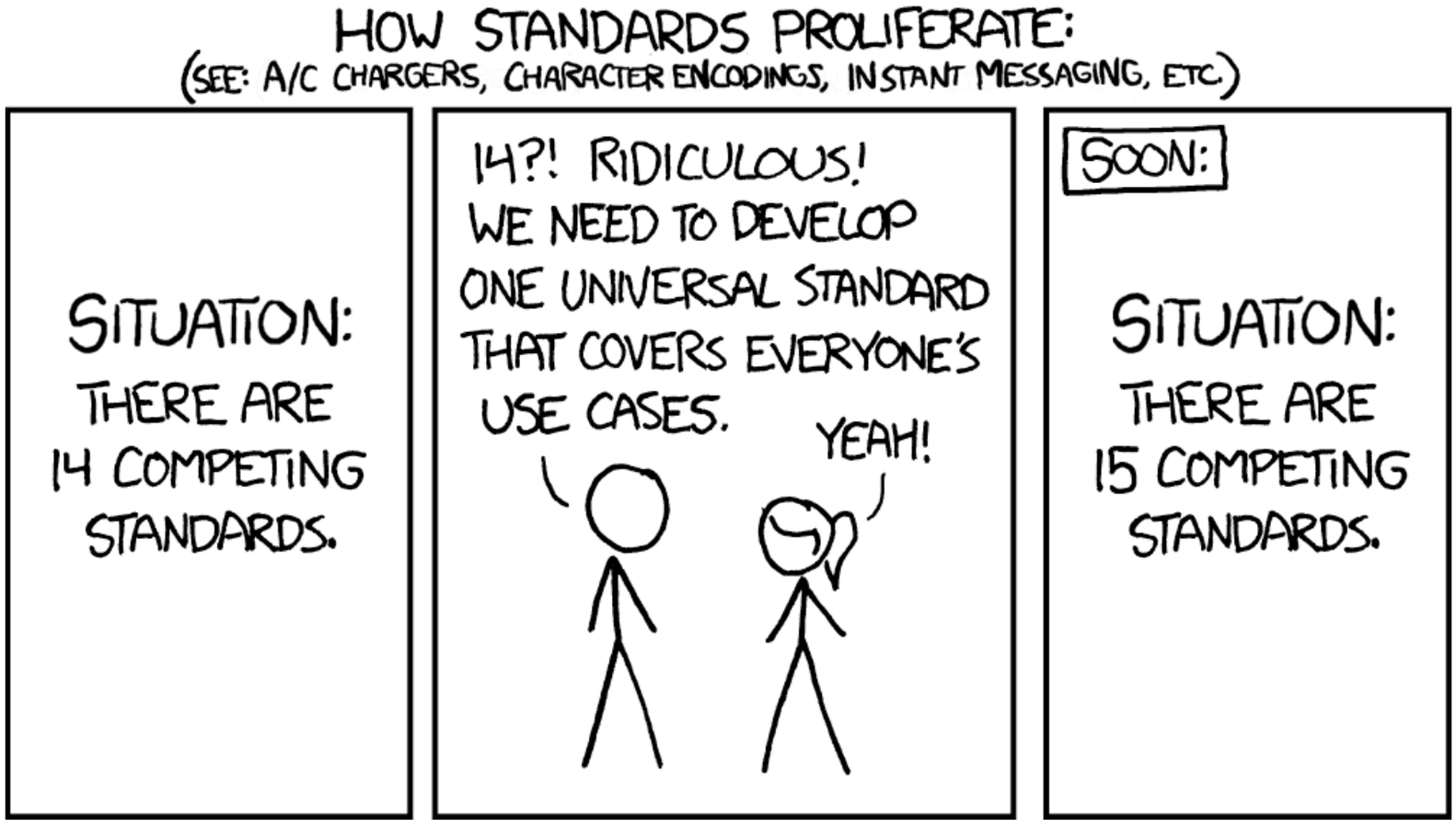
Yocto

- Create custom Linux-based systems
- Nightly and weekly builds, LTS releases
- Small, but extensible
- Pretty much the industry standard
- Swiss army bazooka of build systems

Updating an OS with Yocto

- **Lots of options**
  - swupd, swupdate, mender, ostree, RAUC, ...
- Need extra layers, extra dependencies, extra space

igalia


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)
SITUATION:
THERE ARE
14 COMPETING
STANDARDS.
14?! RIDICULOUS!
WE NEED TO DEVELOP
ONE UNIVERSAL STANDARD
THAT COVERS EVERYONE'S
USE CASES.
YEAH!
SOON:
SITUATION:
THERE ARE
15 COMPETING
STANDARDS.

igalia

`systemd-sysupdate`

- **Atomic updates** for different resources (files, directories or partitions)
- Updates from **remote and local sources** (HTTP/HTTPS and directories)
- Parallel installed versions **A/B/C/... style**
- Relative **small** footprint (~10 MiB or roughly 5% increase)
- Available since **systemd 251**

- Optional **built-in** services for updating and rebooting
- Optional D-Bus interface for **applications integration**
- Optional grouping of resources to be enabled together as **features**
- Leverages other mechanisms already baked-in (boot assessment, **systemd-boot**, etc)

https://www.freedesktop.org/software/systemd/man/latest/systemd-sysupdate.html

igalia

Yocto integration

- Set up a base OS image
- Add UKI to the layout
- Add versioning configuration

igalia

Step 0: a Poky image

`kas-poky-demo.yml`:

```
INIT_MANAGER = "systemd"
EFI_PROVIDER = "systemd-boot"
INITRAMFS_IMAGE = "core-image-minimal-initramfs"
QB_KERNEL_ROOT = ""
QB_DEFAULT_KERNEL = "none"
IMAGE_FSTYPES = "wic"
WKS_FILE = "core-image-demo.wks.in"
```

`recipes-core/images/core-image-demo.bb`:

```
# Metadata
SUMMARY = "A demo image with UKI support enabled"
LICENSE = "MIT"

# Configuration
UKI_CMDLINE = "rootwait root=PARTLABEL=root console=${KER

inherit core-image uki
```

https://github.com/uapi-group/specifications/blob/main/specs/unified_kernel_image.md

igalia

`wic/core-image-demo.wks.in`:

```
part /boot --ondisk sda --fstype vfat \
  --part-name ESP \
  --part-type c12a7328-f81f-11d2-ba4b-00a0c93ec93b \
  --source bootimg-efi \
  --sourceparams="loader=systemd-boot,install-kernel-into
  --align 1024 --active --fixed-size 100M

part / --ondisk sda --fstype=ext4 \
  --source rootfs \
  --part-name root \
  --part-type 4f68bce3-e8cd-4db1-96e7-fbcaf984b709 \
  --align 1024 --use-uuid --fixed-size 300M

bootloader --ptable gpt --timeout=5
```

https://uapi-group.org/specifications/specs/discoverable_partitions_specification/

Step 1: Adding `IMAGE_VERSION` to os-release

igalia

`recipes-core/os-release/os-release.bbappend`:

```
OS_RELEASE_FIELDS += " \
  IMAGE_VERSION \
"

OS_RELEASE_UNQUOTED_FIELDS += " \
  IMAGE_VERSION \
"
```

https://www.freedesktop.org/software/systemd/man/latest/os-release.html

igalia

- `recipes-core/images/core-image-demo.bb`:

```
UKI_FILENAME = "uki_${IMAGE_VERSION}.efi"
UKI_CMDLINE = "rootwait root=PARTLABEL=_${IMAGE_VERSION}
```

- `wic/core-image-demo.wks.in`:

```
part / --ondisk sda --fstype=ext4 \
  --source rootfs \
  --part-name "_${IMAGE_VERSION}" \
  --part-type 4f68bce3-e8cd-4db1-96e7-fbcaf984b709
```

Step 2: enabling and configuring sysupdate

igalia

`recipes-core/systemd/systemd_%.bbappend`:

```
EXTRA_OEMESON:append = " \
  -Dfdisk=enabled \
  -Dsysupdate=enabled \
  -Dsysupdated=enabled \
"
```


igalia

`recipes-core/systemd/systemd/60-rootfs.transfer`:

```
[Transfer]
ProtectVersion=%A
Verify=no
[Source]
Type=url-file
Path=http://10.0.2.2:3333/
MatchPattern=rootfs_@v.ext4
[Target]
Type=partition
Path=auto
MatchPattern=_@v
MatchPartitionType=root
InstancesMax=2
```

https://www.freedesktop.org/software/systemd/man/latest/sysupdate.d.html

`recipes-core/systemd/systemd/70-kernel.transfer`:

```
[Transfer]
ProtectVersion=%A
Verify=no
[Source]
Type=url-file
Path=http://10.0.2.2:3333/
MatchPattern=uki_@v.efi
[Target]
Type=regular-file
Path=/EFI/Linux
PathRelativeTo=boot
MatchPattern=uki_@v+@l-@d.efi uki_@v+@l.efi uki_@v.efi
Mode=0444
TriesLeft=3
TriesDone=0
InstancesMax=2
```

`recipes-core/systemd/systemd_%.bbappend`:

```
SRC_URI += " \
  file://60-rootfs.transfer \
  file://70-kernel.transfer \
"

do_install:append() {
  install -d ${D}${base_libdir}/sysupdate.d
  install -m 0644 ${UNPACKDIR}/60-rootfs.transfer ${D}${b
  install -m 0644 ${UNPACKDIR}/70-kernel.transfer ${D}${b
}
"
```

`wic/core-image-demo.wks.in`:

```
part / --ondisk sda --fstype=ext4 \
  --source rootfs \
  --part-name root \
  --part-type 4f68bce3-e8cd-4db1-96e7-fbcaf984b709 \
  --align 1024 --use-uuid --fixed-size 300M
part --ondisk sda --source empty \
  --part-name "_empty" \
  --part-type 4f68bce3-e8cd-4db1-96e7-fbcaf984b709 \
  --align 1024 --use-uuid --fixed-size 300M

bootloader --ptable gpt --timeout=5
```


igalia

Step 3: serving updates

igalia

A simple server:

```
$ ls -1 ./server/
rootfs_0.ext4
rootfs_1.ext4
SHA256SUMS
uki_0.efi
uki_1.efi
```

```
$ sha256sum * > SHA256SUMS
$ python3 -m http.server 3333
```

Demo

igalia

Limitations

- sysupdate-wise: missing delta updates
  - **https://github.com/systemd/systemd/issues/28227**
  - **https://github.com/systemd/systemd/issues/33351**
- Yocto-wise: nicer integration
  - dm-verity is still painful to set up
- Support-wise: requires UEFI firmware
  - yes, there are embedded systems that do not support UEFI

Alternatives

- RAUC
- Mender

igalia

Recommended readings

https://mabente.pages.igalia.com/blog/how-to-integrate-systemd-sysupdate-with-your-yocto-based-image/

https://linuxembedded.fr/2024/09/mise-a-jour-dun-systeme-embarque-la-voie-de-systemd

https://x86.lol/generic/2024/08/28/systemd-sysupdate.html

Source code

https://github.com/Igalia/yocto-sysupdate-demo

igalia

Thanks for your attention!

igalia

igalia