

Status fwupd/LVFS support for Qubes OS

Qubes OS and 3mdeb minisummit 2020





Norbert Kamiński



- fwupd/LVFS - overall information
- Qubes OS support challenges and architecture solutions
- What is done
- Downloading firmware
- Updating firmware
- To Do's
- Q&A



Norbert Kamiński
Junior Embedded Systems Engineer

- open-source contributor:
 - meta-pcengines
 - meta-virtualization
 - scope of interests:
 - embedded Linux
 - virtualization
 - bootloaders
-  norbert.kaminski@3mdeb.com
 -  [linkedin.com/in/norbert-kami%C5%84ski/](https://www.linkedin.com/in/norbert-kami%C5%84ski/)
 -  [facebook.com/nkaminski3](https://www.facebook.com/nkaminski3)
 -  [@_@siderr](https://twitter.com/_@siderr)

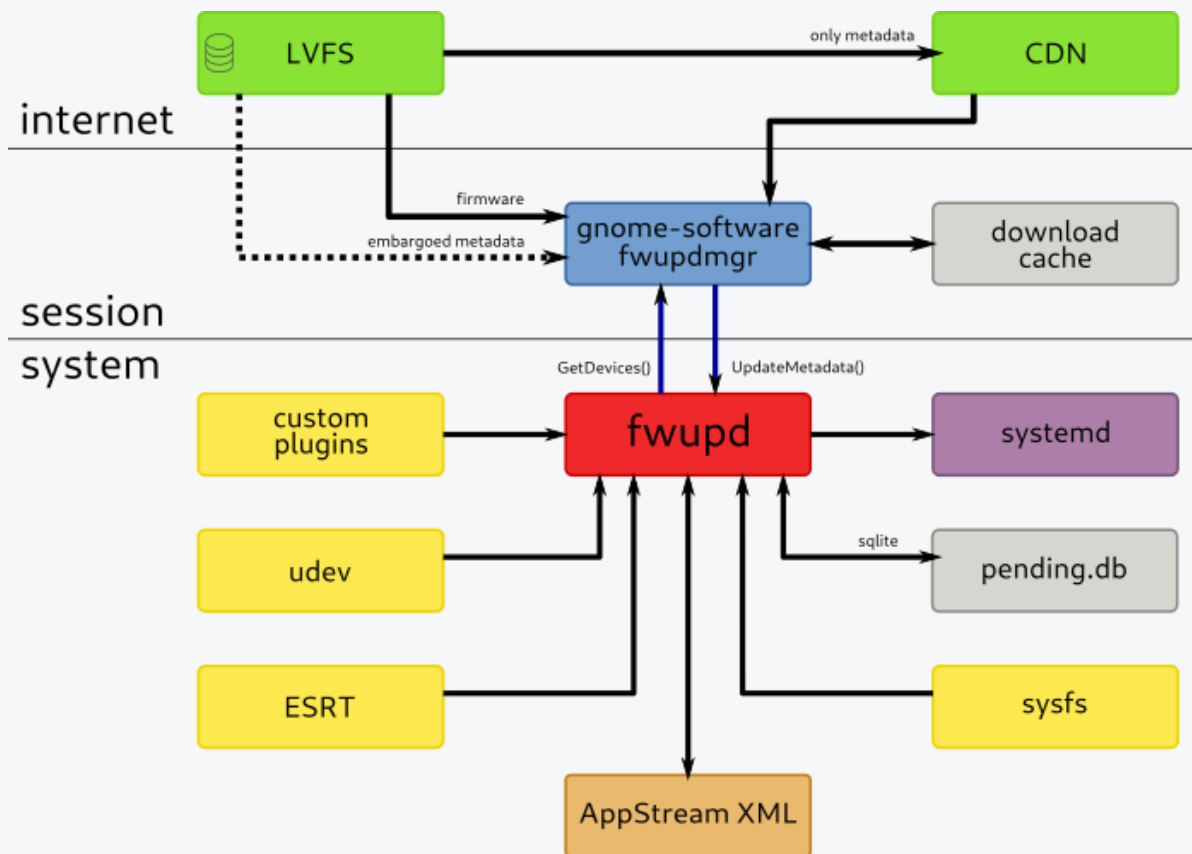
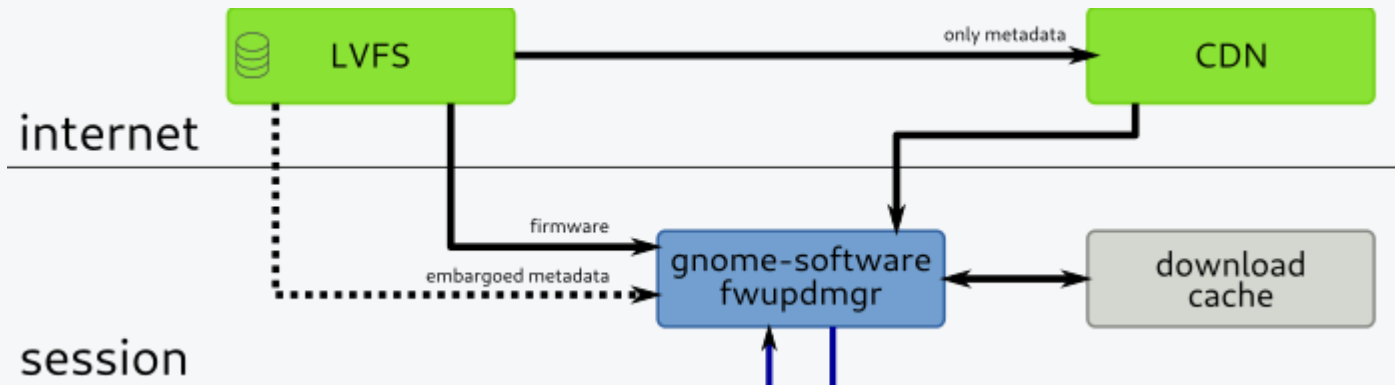
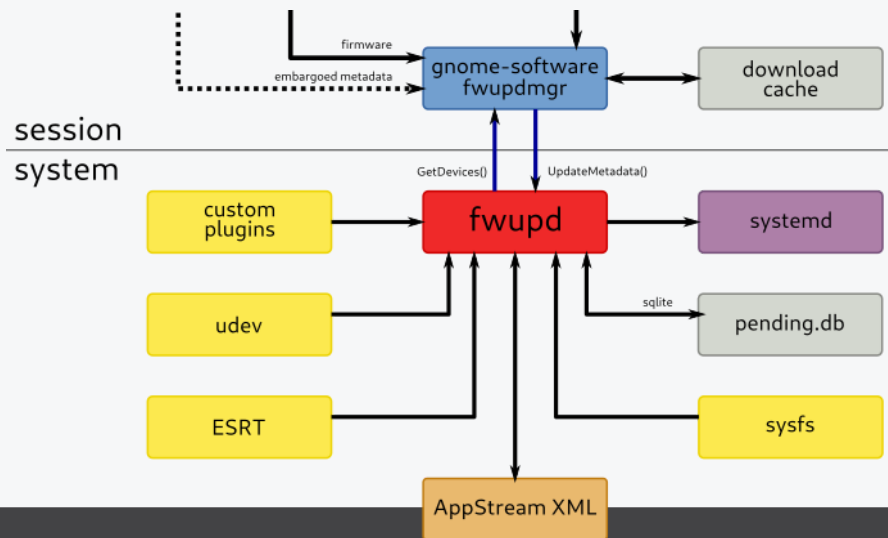


Image source: <https://lvfs.readthedocs.io/en/latest/intro.html>

- The LVFS is a secure web service that can be used by hardware vendors to upload firmware archives
- Customers can securely download metadata about the available updates.
- Firmware update files are stored in cabinet archives files, that contain firmware, metadata and detached signature

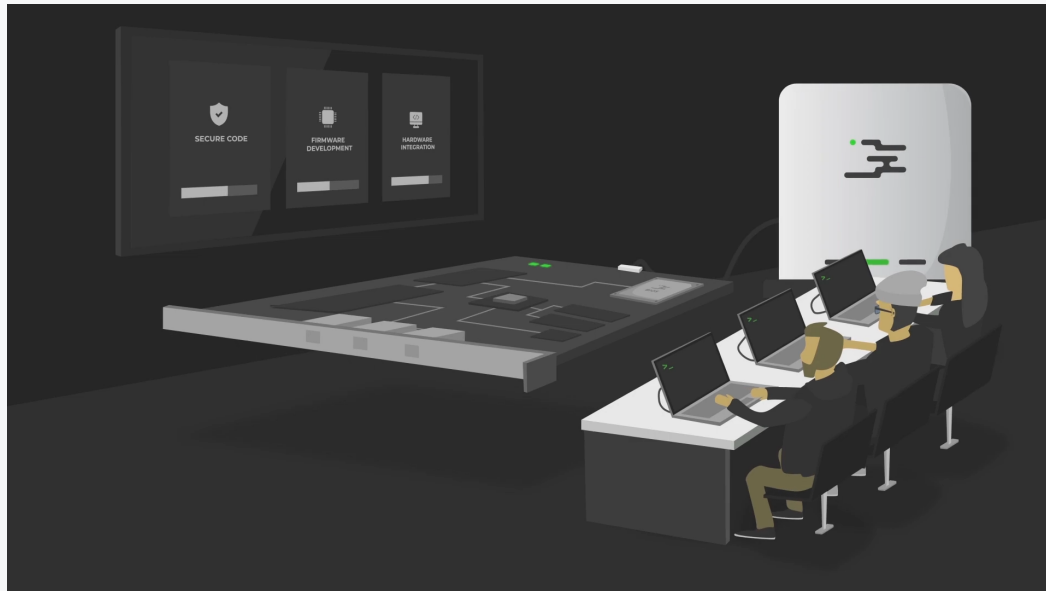


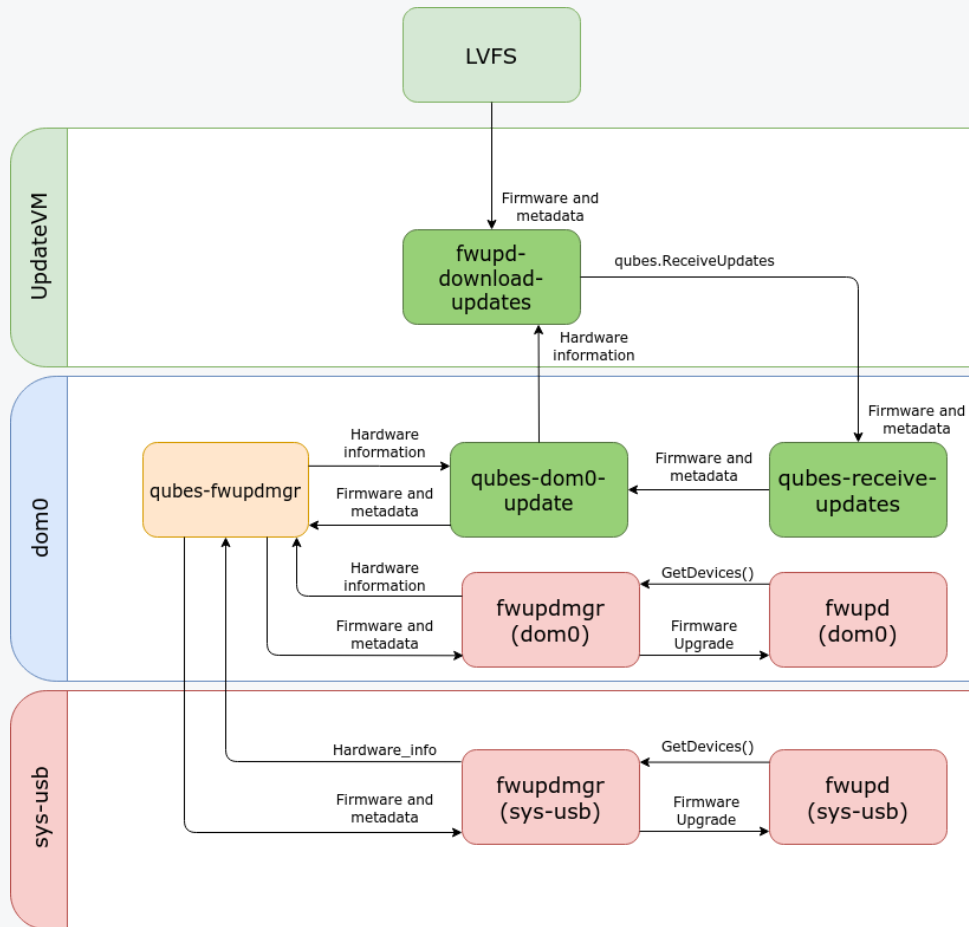
- The fwupdmgr is a CLI client tool, that allows user to preform the update process manually
- It takes the role of connector between LVFS database and the fwupd
- The fwupd is a system activated daemon with a D-Bus interface, that can be used to perform wide upgrades and downgrades according to security policy



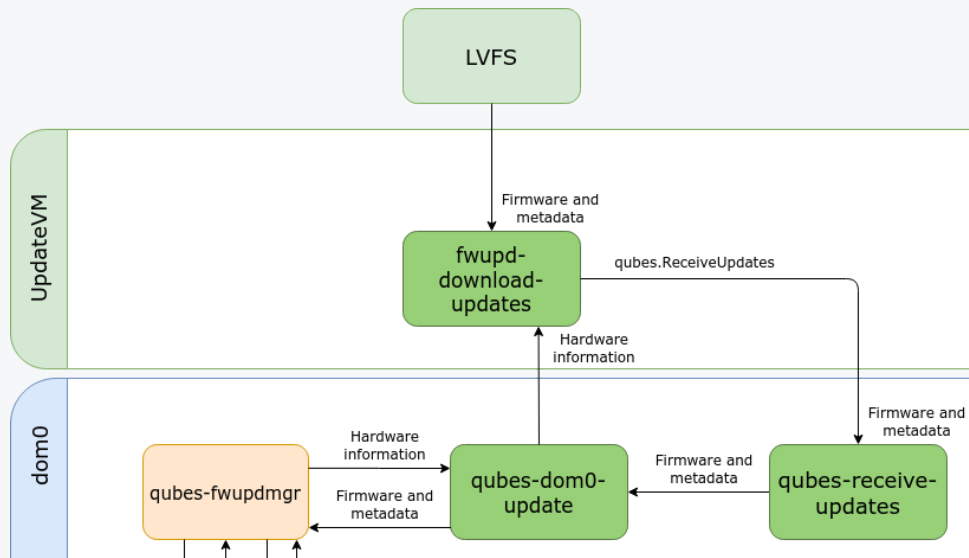
- Virtual machines (AdminVM (dom0), sys-usb) which handle devices to be flashed have no internet connection
- UpdateVM must check metadata and provide a update archive for a device
- Update files must be verified at all steps of the download process
- fwupd must support the firmware update process divided into three VMs (UpdateVM, AdminVM , sys-usb)

- Architecture plan of the fwupd/LVFS support for Qubes OS
- Frame of the update process
- Building the fwupd from the source at the AdminVM.

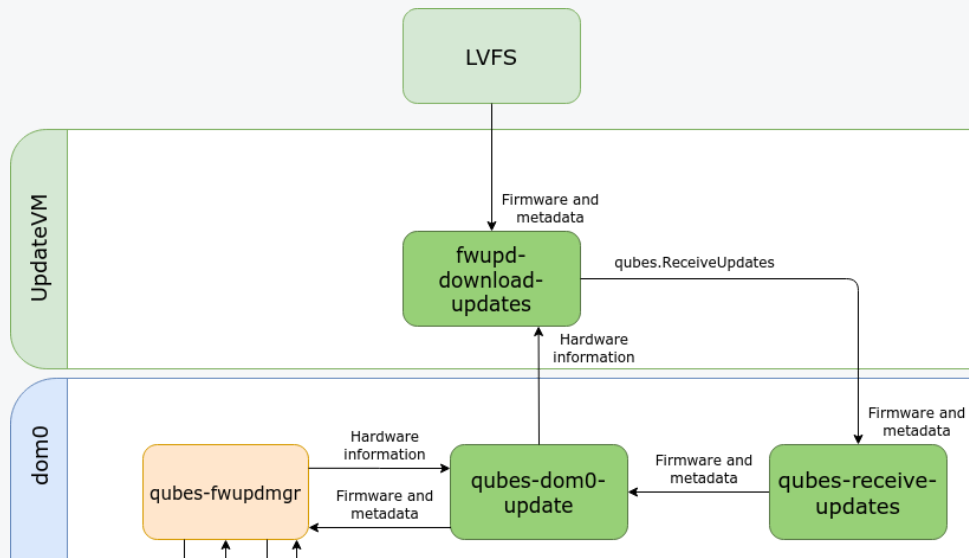




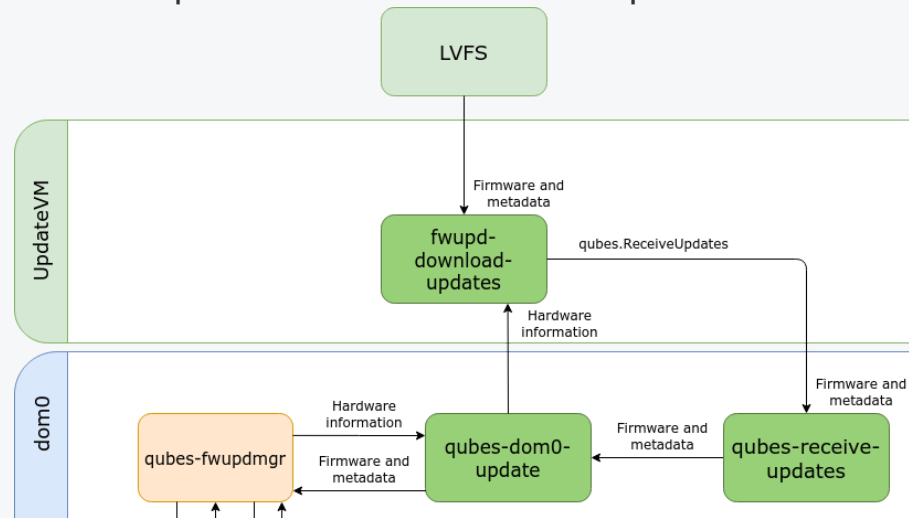
- dom0 and sys-usb are isolated from the network
- Download process is initiated via `qubes-dom0-update`
- `qubes-dom0-update` creates download directory in the UpdateVM
- Then it runs `fwupd-download-updates` in the UpdateVM



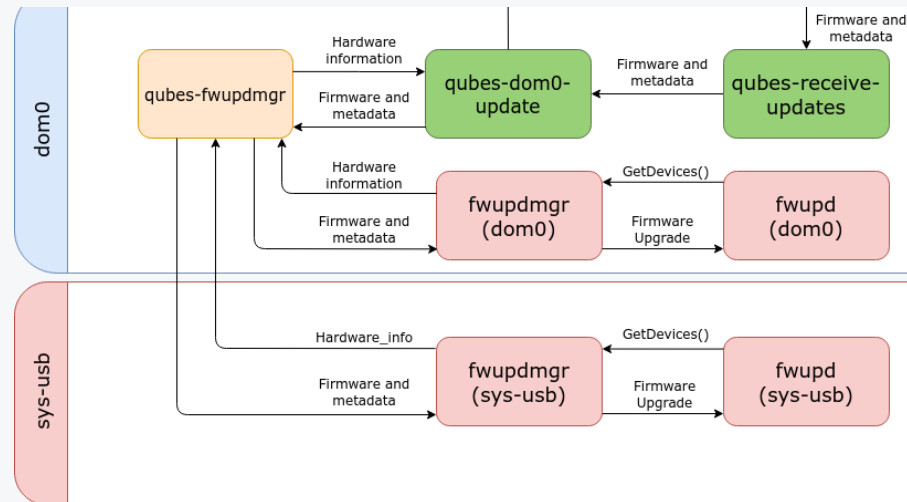
- fwupd-download-updates downloads metadata and firmware from the LVFS
- Script performs the first step of the validation
- If it is running with check-only, it sends only meta data
- Otherwise it download the .cab archive and it starts qubes.ReceiveUpdates



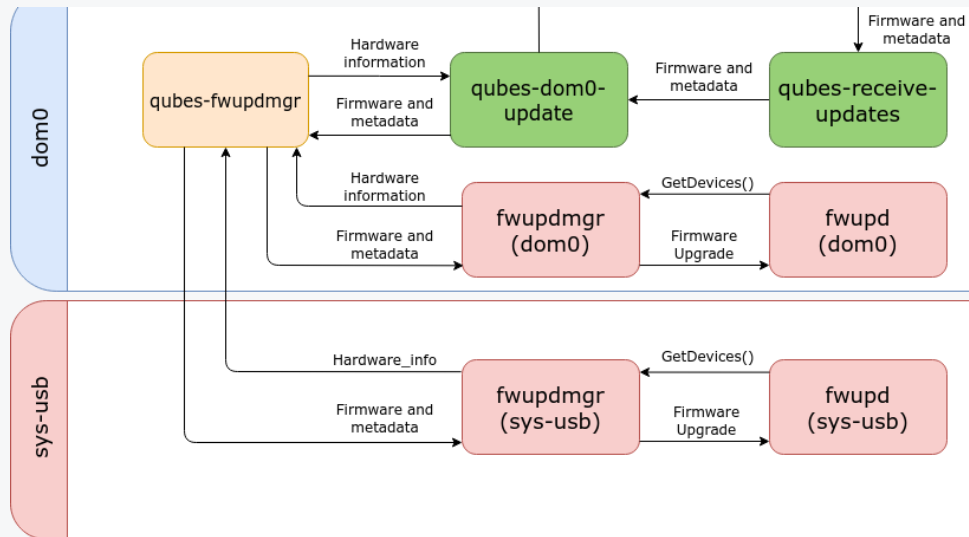
- `qubes.ReciveUpdates` is a symbolic link to the python script `qubes-receive-updates`
- The script `t` is responsible for receiving the update files from the UpdateVM
- `qubes-receive-updates` creates the update cache directory for `fwupd`, it copies the files and it performs the second step of the validation



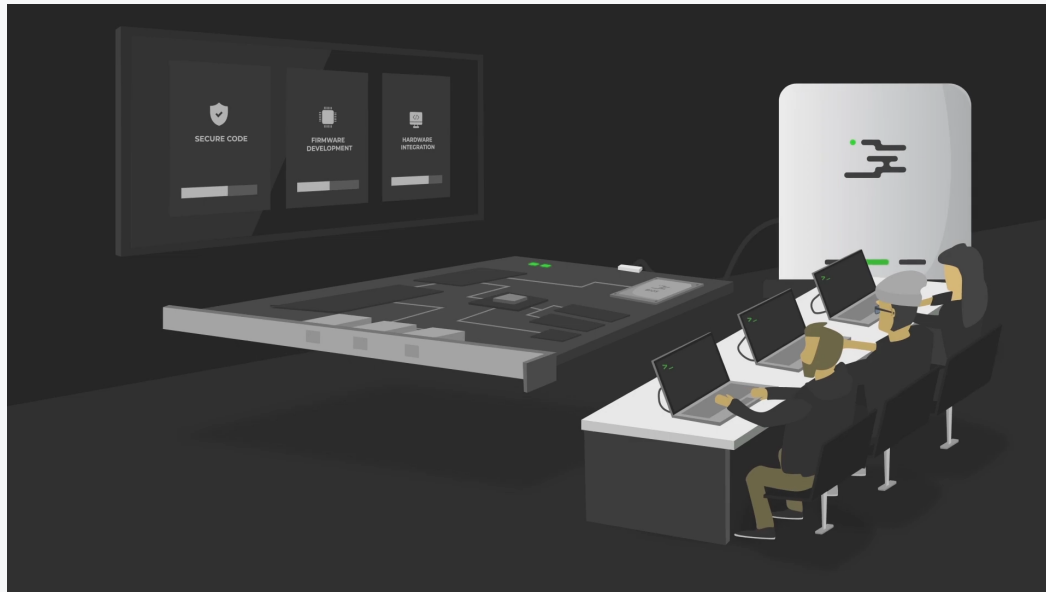
- We need two fwupd daemons to provide the updates to every type of device
- The first daemon is installed to AdminVM and It provides updates to non-USB devices
- The second daemon is placed on the sys-usb. It allows us to update the hardware connected via USB



- The update process is managed by qubes-fwupdmg
- The fwupdmg takes the hardware information from the daemon and pass it to qubes-fwupdmg
- If the update is available qubes-fwupdmg uses proper fwupdmg to perform the firmware update process



- Custom fwupd plugin that will use information from all VMs
- qubes-fwupdmgr script that will connect the downloading and updating firmware
- .cab archives validation that will ensure us about the safety of the files



Q&A