

# Brief history of USB camera support in Qubes OS

Qubes OS mini-summit 2021

Piotr Król





Piotr Król  
*3mdeb Founder & CEO*

- coreboot contributor and maintainer
- Conference speaker and organizer
- Trainer for military, government and industrial organizations
- Former Intel BIOS SW Engineer
- 12yrs in business
- Qubes OS user since 2016
- C-level positions in





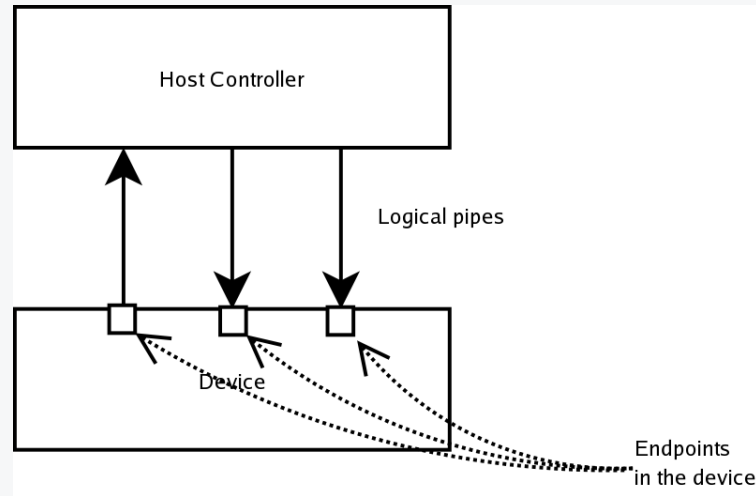
- coreboot licensed service providers since 2016 and leadership participants
- UEFI Adopters since 2018
- Yocto Participants and Embedded Linux experts since 2019
- Official consultants for Linux Foundation fwupd/LVFS project since 2020
- IBM OpenPOWER Foundation members since 2020
  - Our Firmware Engineer Michał is chair of SSWG since 2021

- Presentation goal
- Why USB is hard
- USB Endpoints
- USB device security and how Qubes OS deal with it
- Camera issues
- Qubes Video Companion
- Demo
- How you can help
- Q&A

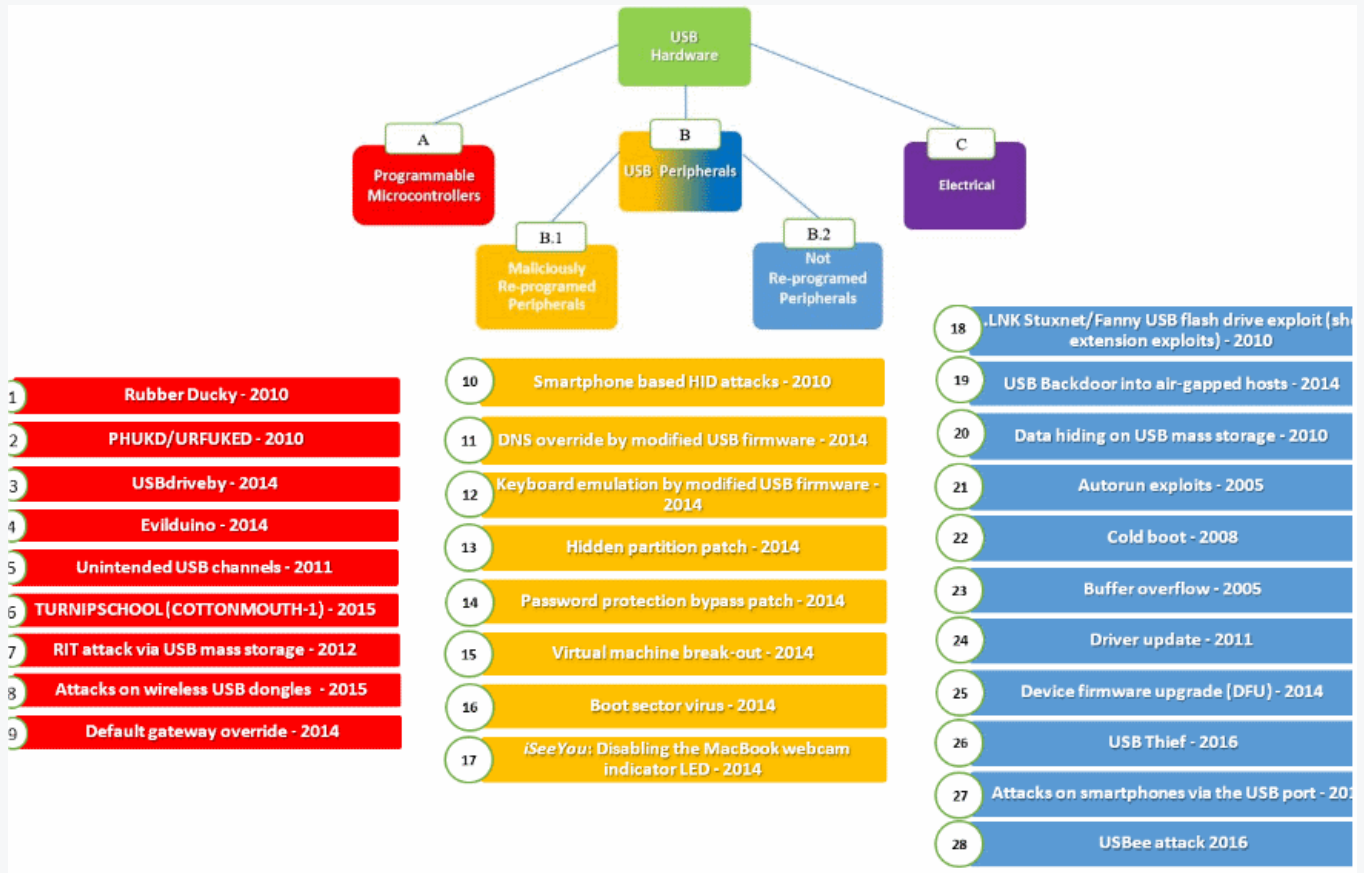
Discuss history of USB camera support in Qubes os and  
present demo of Qubes Video Companion project



- USB is relatively complex protocol
  - thousands pages of specs
  - multiple host controller implementations both hardware and software wise
  - massive amount of devices on the market, where some of them weakly comply to the spec
- OSFV perspective: USB is the area causing most problems even in stable projects
- There are multiple endpoints with various requirements based on transferred data types



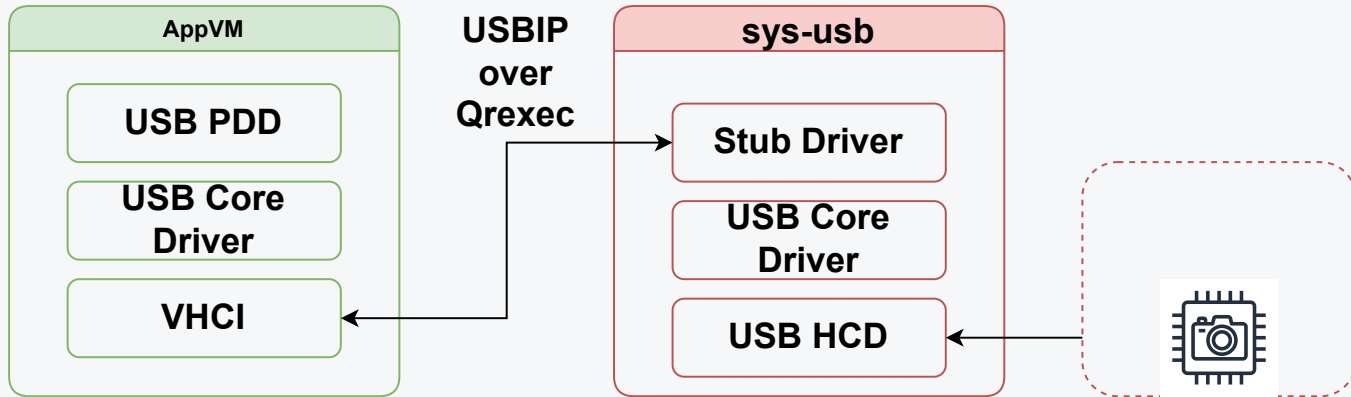
- One USB device can have even 32 endpoints
- There 3 types of transfers used in USB devices
  - **bulk** - large sporadic data transfers typical for storage devices
  - **interrupt** - devices requiring quick response e.g. mouse, keyboard
  - **isochronous** - devices requiring guaranteed data rate e.g. video, audio
- Some devices combine multiple endpoints e.g. camera with microphone



"USB-based attacks" Nissim,Yahalom,Elovici 2017



- Qubes 2.0 added experimental support for PV USB
  - it caused some problems from the beginning being unstable and unmaintained code
- Qubes 3.1 added out of the box setup for USB VM, mouse support was included
- Qubes 3.2 support for `qvm-usb` and with it USB pass through was added, what allowed to deliver better USB isolation through improved `sys-usb`
- Qubes 4.0 added system tray widget for handling USB devices

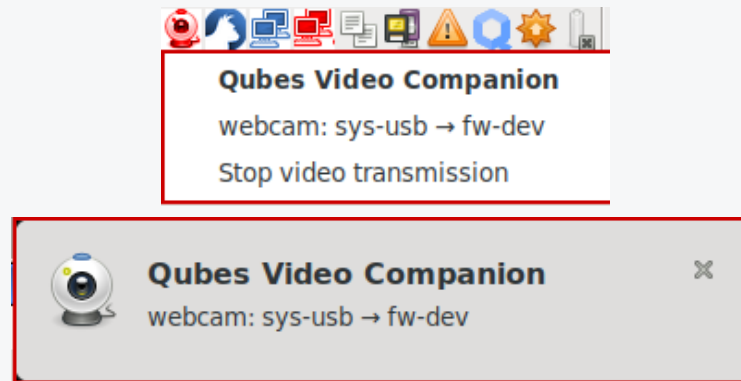


- Qubes OS use USB/IP project to provide devices from sys-usb to AppVMs
- This configuration has some limitations
  - USB/IP stack is primitive and despite handling bulk endpoints reliably, some isochronous endpoints cause problem to it
  - e.g. `vhci_get_frame_number` is not yet implemented
  - handling multi-endpoint devices rise the complexity
  - some users complain on performance
  - finally, there is security risk exposing USB device without modification to AppVM

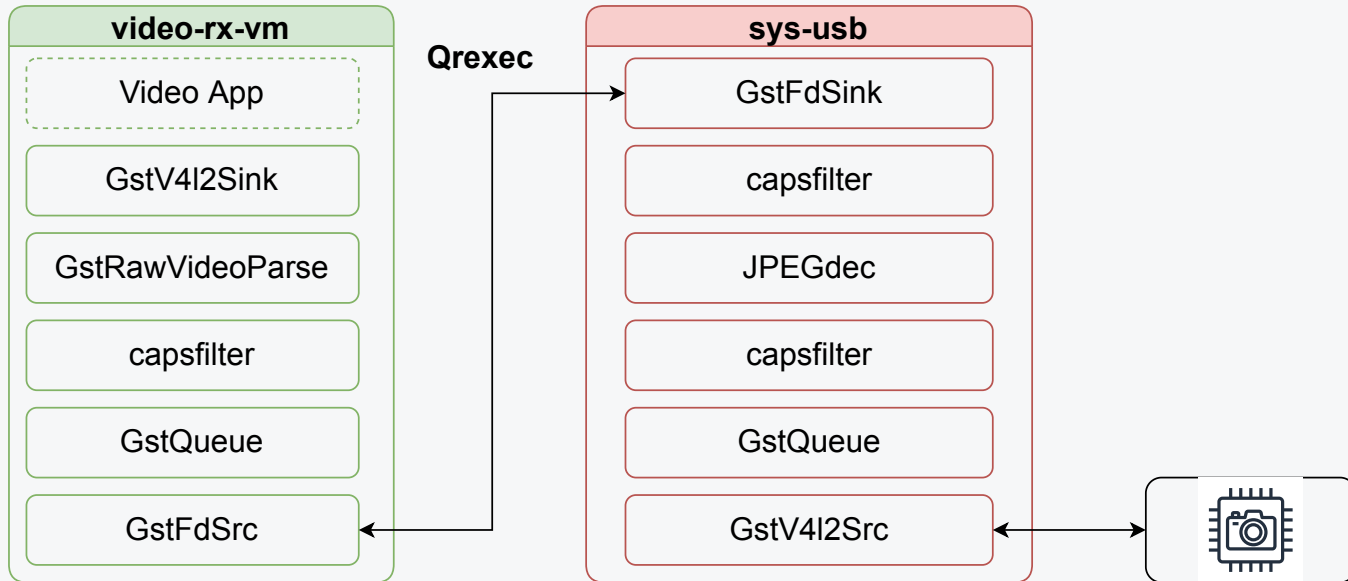
- Because of remote work priority of correct support for wide range of USB cameras in Qubes OS has grown
  - There are even some comments that people resigned from using Qubes because of issues with camera [1](#)
- Historical Qubes OS issues
  - [Cannot use a USB camera](#) - created in Jun 2018, 26 participants, 68 comments, initially created for Logitech C270
  - [Feature: Trusted stream for webcam input](#) - feature request by tasket submitted in 2016
  - [I cannot use my \(good\) Logitech 922 USB Webcam](#) - 48 replies, 1.1k views
- Security papers
  - [iSeeYou: Disabling the MacBook Webcam Indicator LED](#)
  - [Preventing Covert Webcam Hacking in the Civilian and Governmental Sectors](#)



- Project created by Elliot Killick in Sep 2020 with the goal of securely streaming webcams and sharing screens across VMs
- MIT-licensed source code available on Github:
  - <https://github.com/elliotkillick/qubes-video-companion>
- Initially written in BASH, now rewritten with support of Demi to Python
- Candidate for being part of core Qubes OS



- system tray icon and notifications while streaming or screen sharing
- guaranteed one-way communication from video sending domain to video receiving domain
- minimized attack surface of video receiving domain (video caps are sanitized, kernel modules reloading between session etc.)
- no direct access to hardware by video receiving domain
- any many others



- Qubes Video Companion create GStreamer pipeline which sanitize video stream in sys-usb and expose it as file descriptor through Qrexec
- video-rx-vm use provided file descriptor as source and through GStreamer pipeline expose v4l2sink in form of /dev/video0 device that can be consumed by Video Application

- Based on v2.0.0 tag

```
(debian-10)$ git clone https://github.com/elliottkillick/qubes-video-companion.git
(debian-10)$ cd qubes-video-companion
(debian-10)$ ./build/create-deb.sh
(debian-10)$ sudo apt install ../qubes-video-companion_1.0.0-1_all.deb
(debian-10)$ /usr/share/qubes-video-companion/scripts/v4l2loopback/install.sh
(debian-10)$ sudo poweroff
```

- Restart your debian-10 based video receiving and video sending (sys-usb) VMs
- Connect your camera (tested with Logitech C922)
- Run in video receiving VM

```
(video-rx-vm)$ ./qubes-video-companion webcam
```

- Select sys-usb as video sender

- Jitsi (in Chrome 91) and OBS v0.0.1 works like a charm in 4 vCPUs and 12GB RAM VM
- In case of performance issues
  - use minimal size preview window
  - in OBS preview can be disabled by unchecking Enable preview in context menu
  - do not play videos in other VMs at the same time
  - assign as many vCPUs as possible to receiving VM



## Qubes OS mini-summit 2021: Qubes Video Companion Demo



- test, test, test
  - we need test reports for various hardware
  - we need more information about users configurations in context of performance (what CPUs it worked fine)
- potential issues
  - wouldn't v4l2loopback compilation be needed every time debian-10 kernel will be updated?
- future ideas
  - integrate with sys-usb tray icon

# Q&A