Linux and USB Audio Class 3

Ruslan Bilovol

FOSDEM 2019
Brussels, Belgium

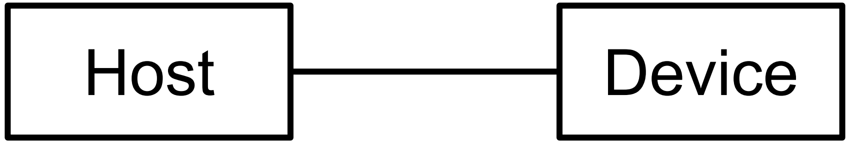
Agenda

- USB and Audio
- Audio Class 1.0, 2.0
- USB Audio Class 3.0
- Current status in Linux mainline
- Challenges
- Q&A

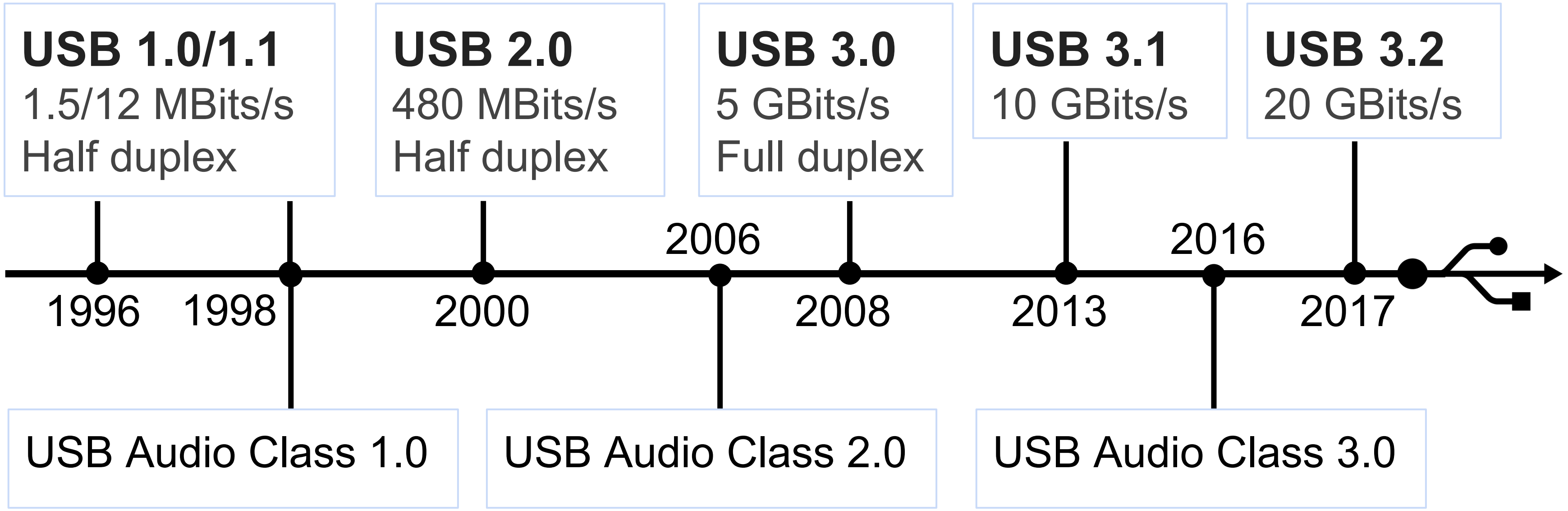
USB and Audio Class 1.0, 2.0

Universal Serial Bus

- USB = Universal Serial Bus
- Designed to standardize connection of peripherals to PC
- First release: 1996 (23 years old!)
- Nowadays: up to 20 Gbits/s
- Simple for end-user (“plug & play”)



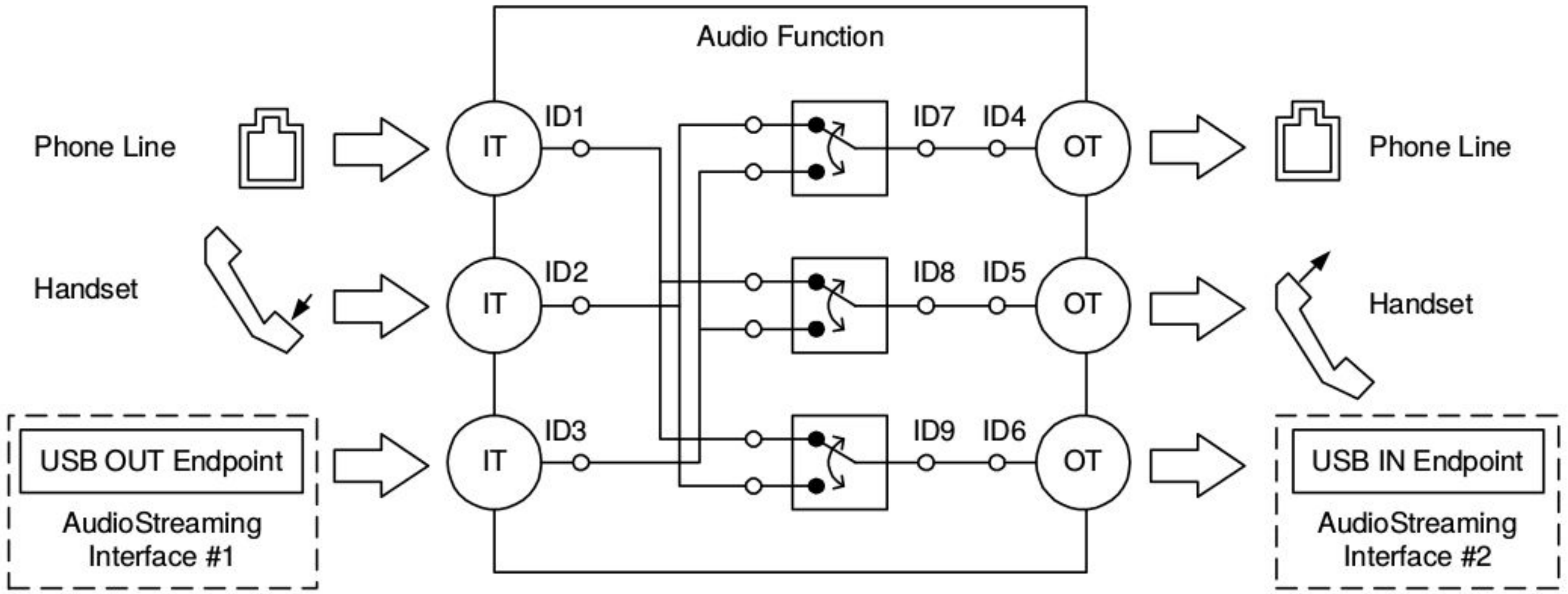
History: USB and Audio



USB Audio Class 1.0

- First USB audio spec released
- Full Speed (12 MBits/s) limited
- Simple (relatively)
- Has Basic Audio Device specification
  - optional to implement
- MIDI
- ALSA support in 2002

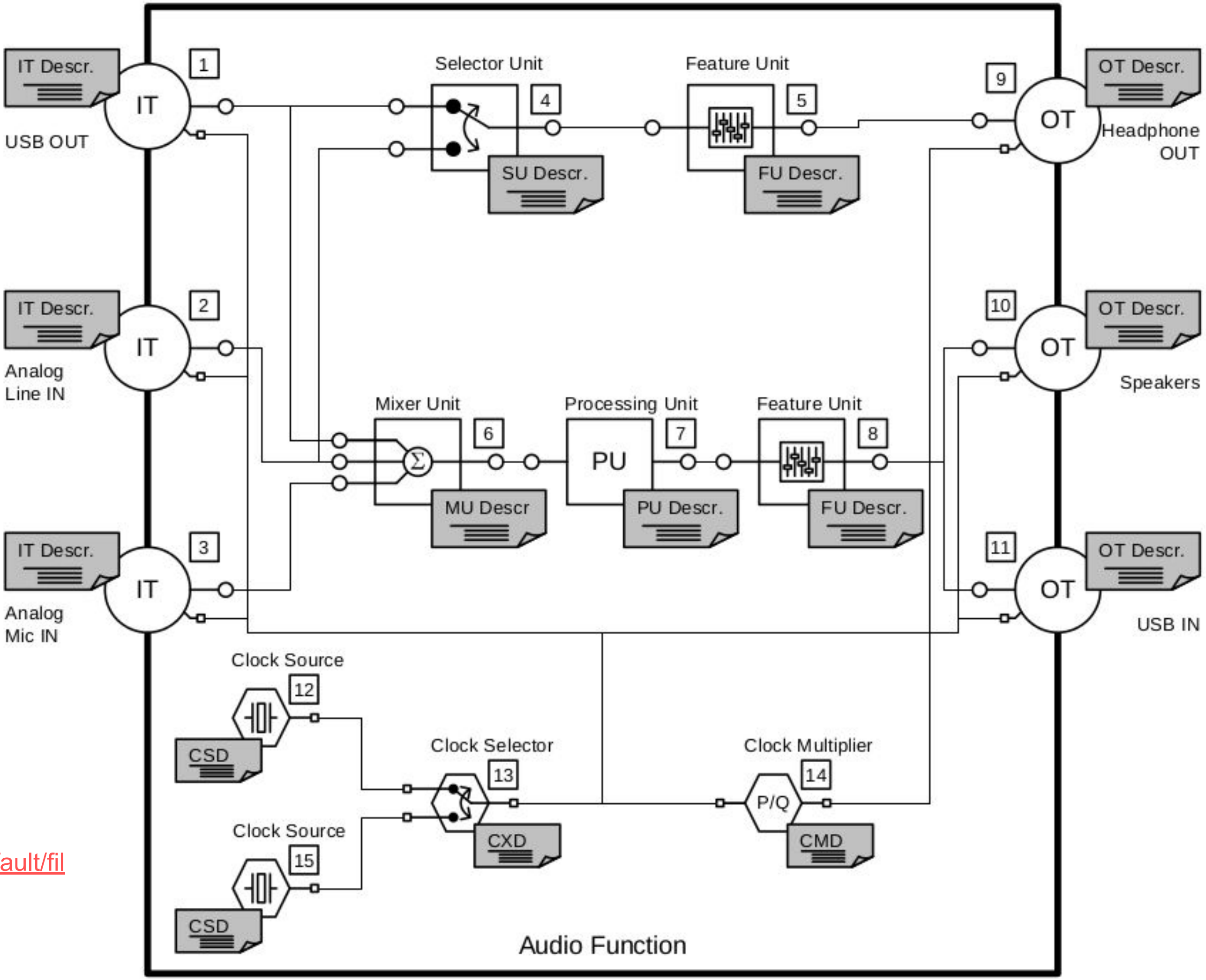
USB Audio Class 1.0



https://www.usb.org/sites/default/files/audio10.pdf

USB Audio Class 2.0

- Incorporates USB High Speed support
- Clocks improvements
- More audio controls
- Extensive support for interrupts from internal Entities
- Incompatible with UAC1
- ALSA support: 2010



https://usb.org/sites/default/files/Audio2.0_final.zip
(Audio20 final.pdf)

USB Audio Class 1 and 2 limitations

- Power management
- Incompatibility between versions

USB Audio Class 3.0

USB Audio Class 3.0

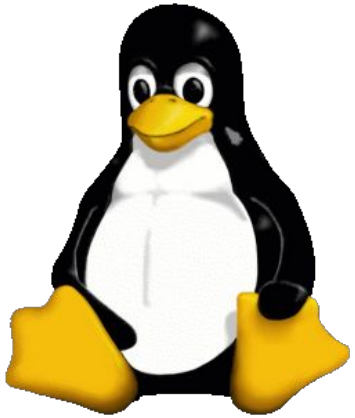
- Support of USB Super Speed
- Mandatory BADD profiles support
- Power Domains
- Connectors descriptor

USB Audio Class 3.0

- Support of USB Super Speed
- Mandatory BADD profiles support
- Power Domains
- Connectors descriptor
- New class-specific String descriptors
- Support of LPM (Link Power Management)
- Burst modes
- Incompatible with UAC1 and UAC2
    - But has backward-compatible implementation for old hosts

USB Audio Class 3.0

Supported in **Linux** before a real HW available on the market!



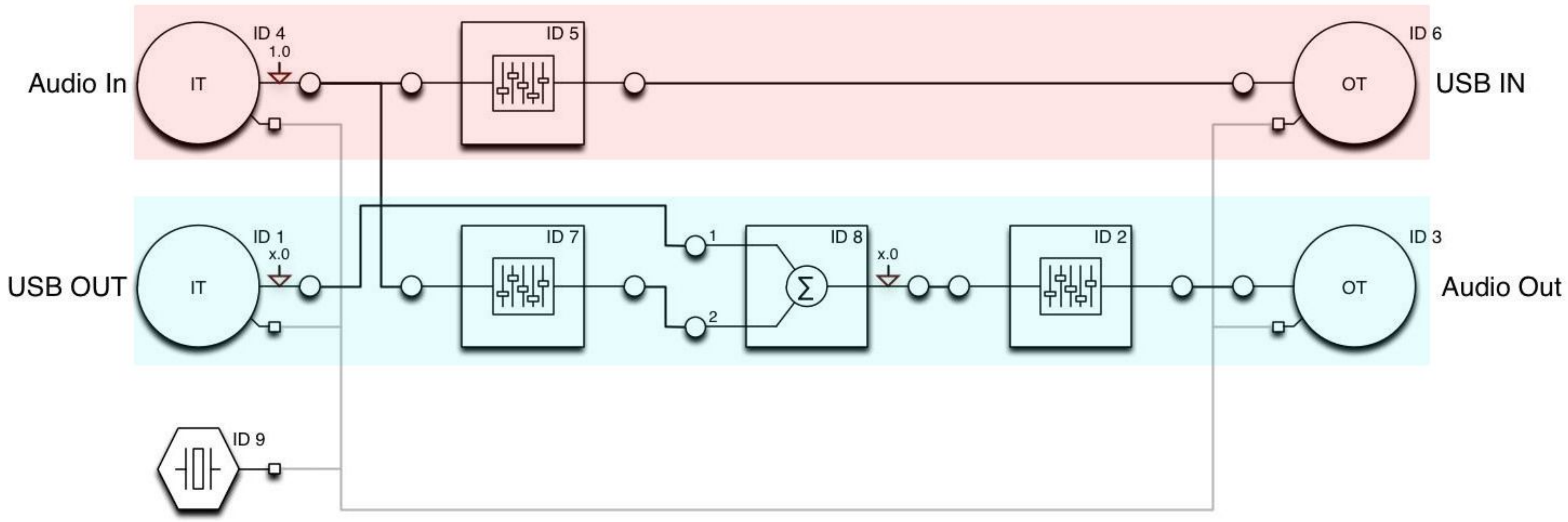
UAC3 BADD specification

- BADD = Basic Audio Device Definition
- Defines simple UAC3 devices like headset or speaker
- Mandatory to implement
- Important: **has no** Audio Class-specific USB descriptors
    - Guess it!
    - A headache for popular OS with full USB stack
- Useful for hosts with limited USB stack support
    - like microcontrollers

UAC3 BADD profiles

- 3 topologies are defined
    - BAIF (Basic Audio Input Function)
    - BAOF (Basic Audio Output Function)
    - BAIOF (Basic Audio Input/Output Function)
- mono or stereo only
- 48 kHz sampling rate only
- 16 bits and 24 bits samples
- Burst Mode support
- LPM/L1 power state support

UAC3 BADD profile example



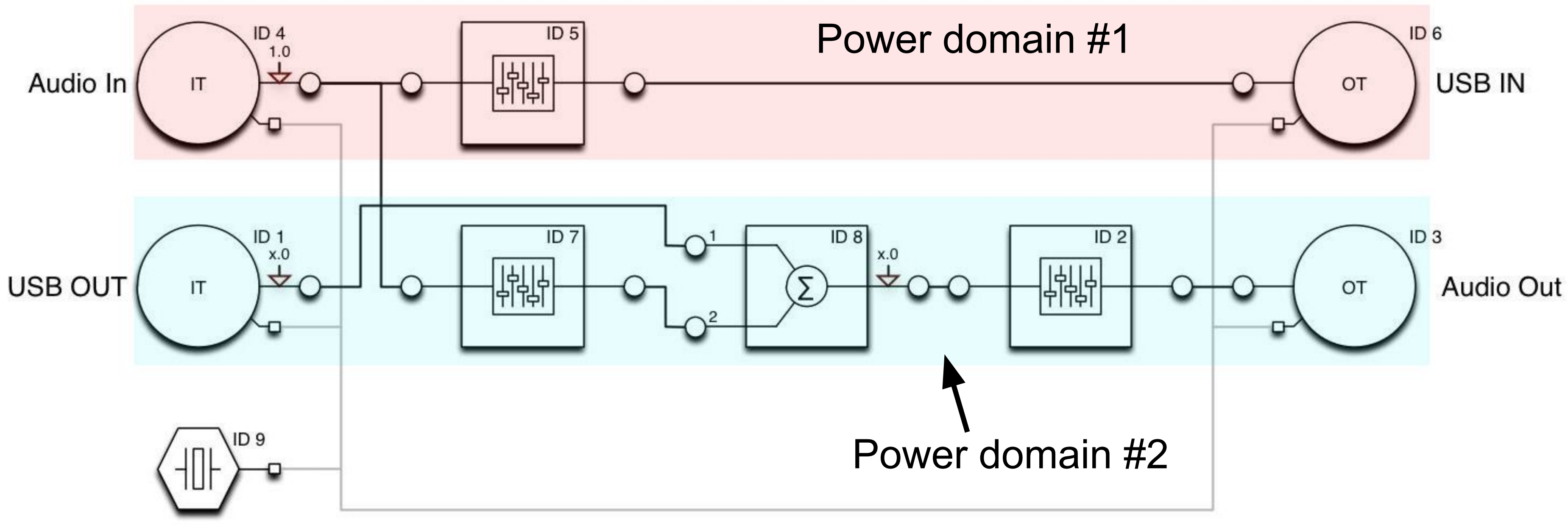
BAIOF topology

https://www.usb.org/sites/default/files/USB_Audio_v3.0.zip (BasicAudioDevice30.pdf)

UAC3 Power Domains (PD)

- PD is a zone within the Audio Function
  - can group multiple elements
- Host can control power consumption levels
- PD may be switched to a low power state
  - e.g. if not used, parts of UAC3 device can consume less power

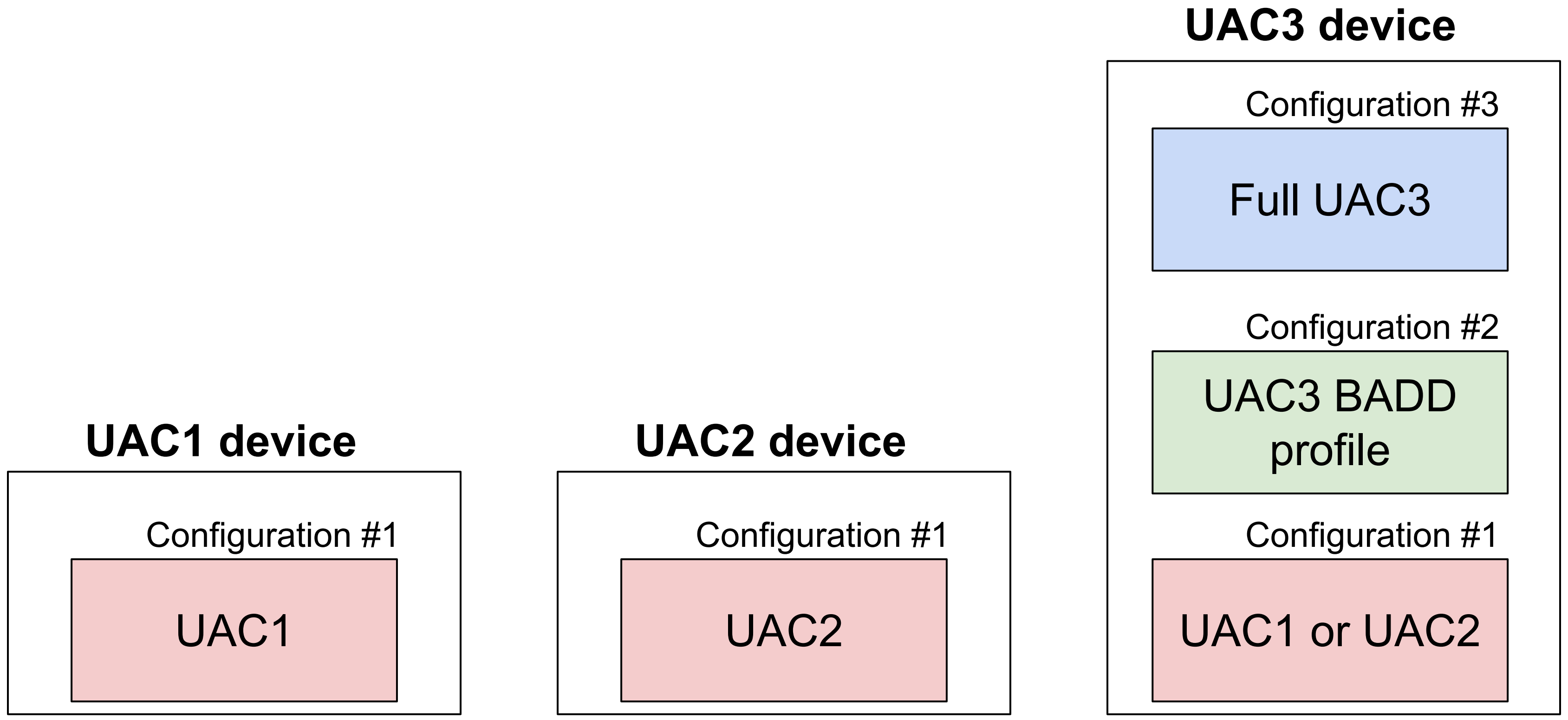
UAC3 Power Domains



BAIOF topology

https://www.usb.org/sites/default/files/USB_Audio_v3.0.zip (BasicAudioDevice30.pdf)

UAC3 compatibility with old hosts



Current UAC3 status in Linux mainline

Initial UAC3 support	v4.17
BADD profiles support	v4.18
Connector insertion	v4.18
Power Domains	v4.19
USB Configuration switching	v4.20
UAC3 Gadget driver	in progress (v5.xx?)
Strings parsing	-

UAC3 spec summary

- Can work on old Hosts
    - with no UAC3 driver
- Has mandatory BADD profiles support
    - Basic support is easy to implement in low-end equipment

UAC3 spec summary (cont.)

- Significant power saving improvements
  - Parts of device may be put in low-power state
  - Burst modes in data transfers
  - As per Synopsys' PoC, power consumption is comparable to 3.5mm jack analog headset solutions

UAC3 spec summary (cont.)

- A real alternative to 3.5mm analog jack solutions
- Already supported by Linux!

Challenges

Challenge: UAC3 documentation quality

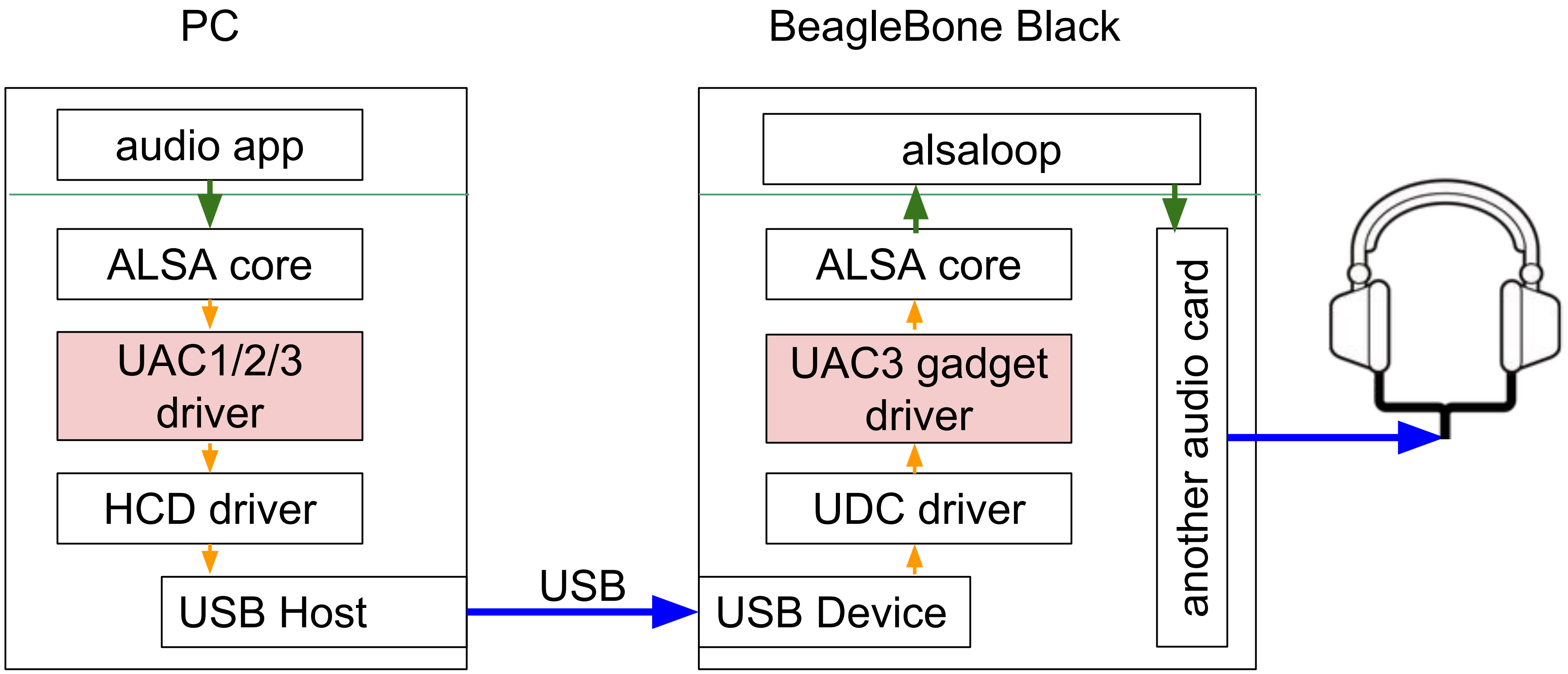
- UAC3 spec was released by USB-IF
- has mistakes and missing parts
  - contacted usb.org over email with no answer
  - thanks to Pierre-Louis Bossart for info from unreleased doc

Challenge: write driver without a real UAC3 hardware

How it’s done?

- use QEMU and dummy_hcd.ko
- write UAC3 gadget driver first
- then write UAC3 host (ALSA) driver
- test both on PC (host) <-> BeagleBoneBlack (gadget)
  - how to deal with bugs on both sides?

Challenge: write driver without a real UAC3 hardware



Challenge: lack of reviews in community

Lack of patches review and testing at the beginning

- UAC3 initial support patch is about 1500 LoC
- was not tested on a real hardware
    - tested on custom UAC3 gadget implementation only
- got only few reviews

Challenge: competing implementations of BADD

How to deal with missing descriptors? Two approaches proposed

- #1: build missing descriptors during enumeration
  - more generic, but more difficult
  - covered only few profiles

Challenge: competing implementations of BADD

How to deal with missing descriptors? Two approaches proposed

- #1: build missing descriptors during enumeration
  - more generic, but more difficult
  - covered only few profiles
- #2: initialize ALSA structures in-place
  - simpler implementation
  - no need to keep whole descriptors in the driver
  - covers all profiles

Challenges

Switching between UAC1/2, BADD and Full UAC3 configurations

- Who should select and switch configuration
  - kernel, userspace or both?
  - need a tool similar to usb_modeswitch?
- In-kernel switching accepted by USB maintainers

Q&A

Thank you

Links

https://www.usb.org/sites/default/files/USB_Audio_v3.0.zip

https://www.usb.org/sites/default/files/audio10.pdf

https://usb.org/sites/default/files/Audio2.0_final.zip