Changes on the audio framework

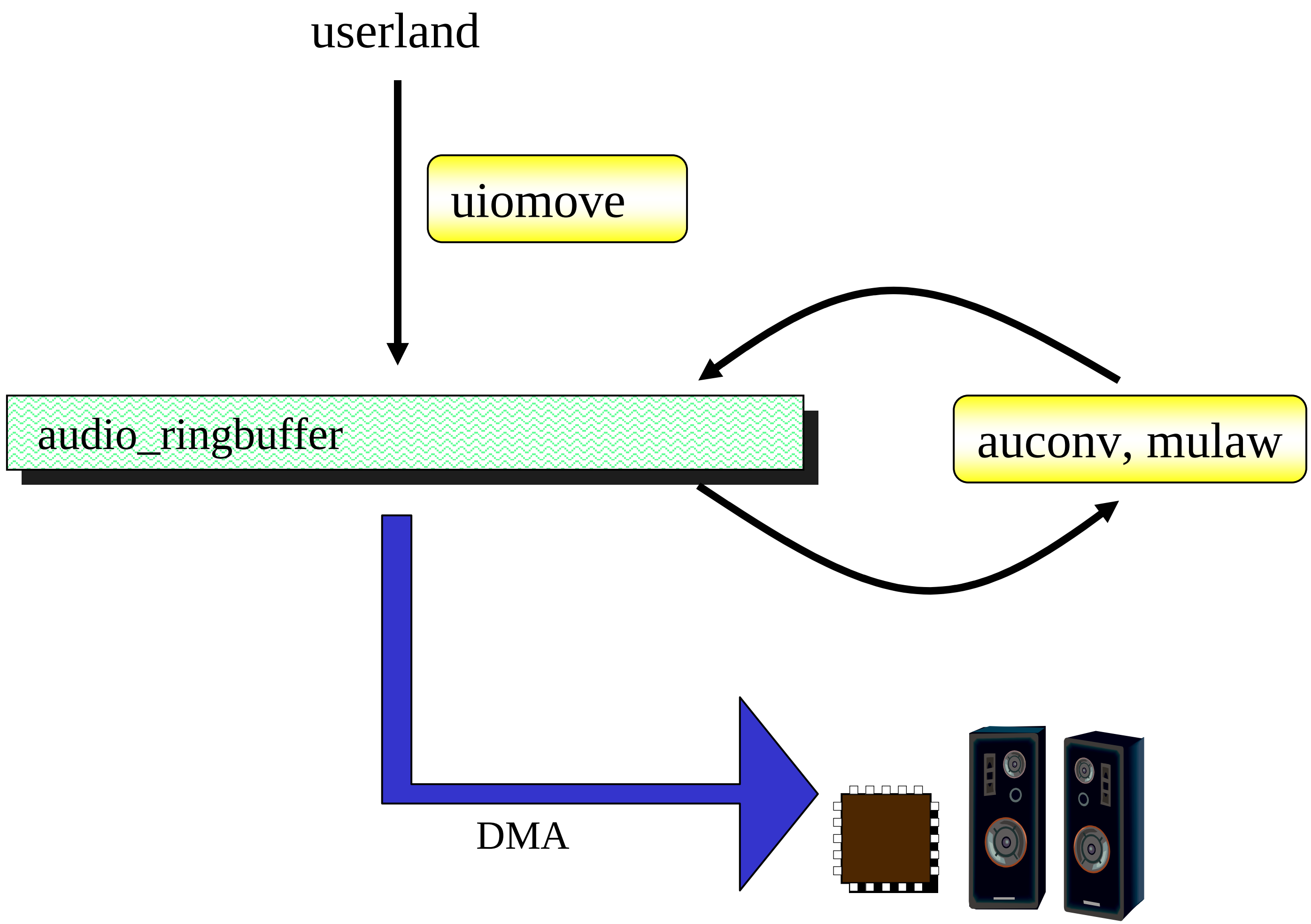
TAMURA Kent

<kent@NetBSD.org>

Agenda

- Sample rate conversion (aurateconv)
- struct audio_format (auconv)
- Audio filter pipeline (kent-audio1)
- High Definition Audio (azalia)
- In-kernel audio mixing (kent-audio2)

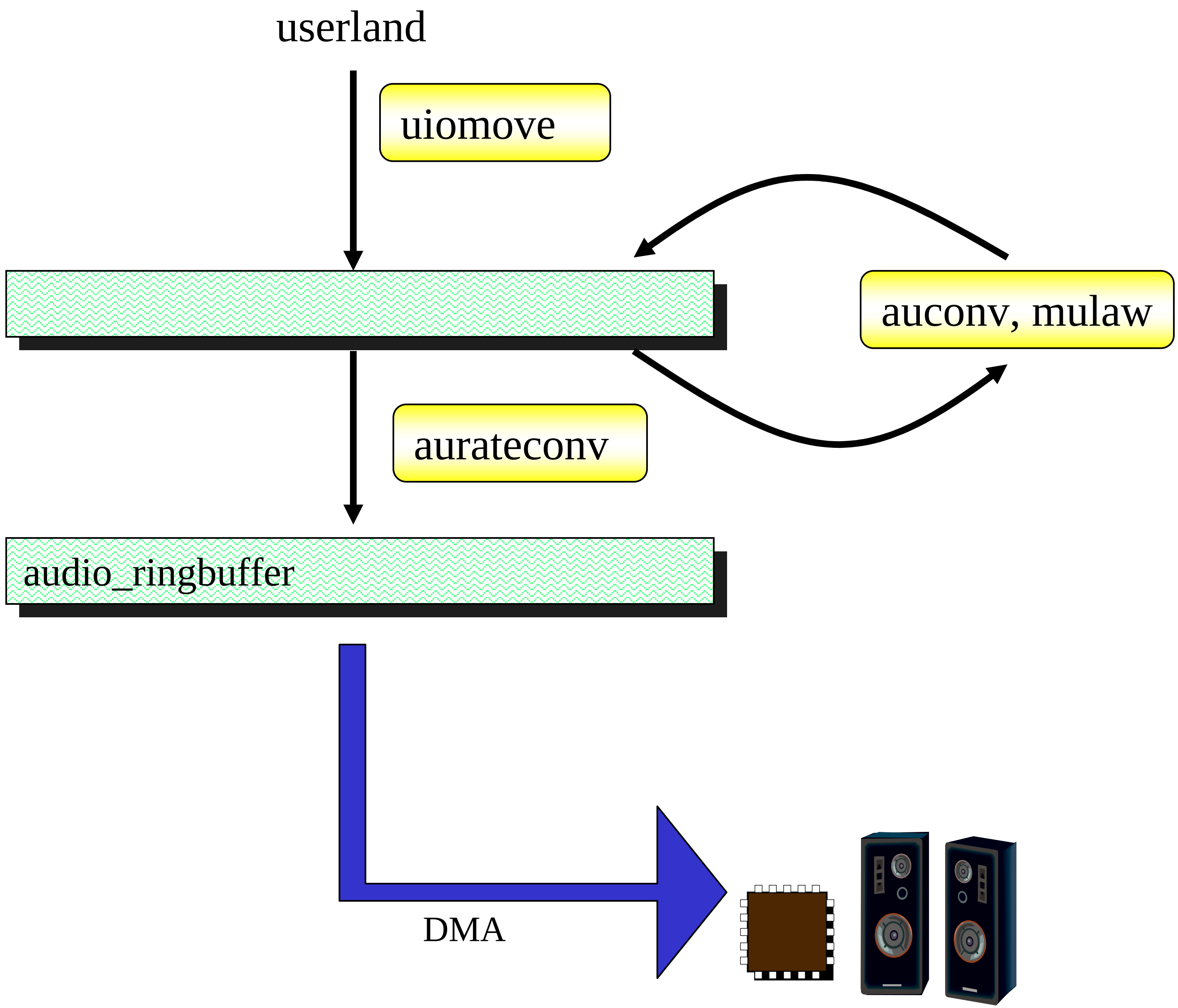
Audio architecture of NetBSD 1.5 or prior



Sample rate conversion

- Mar. 2002, NetBSD 1.6 or later
- Motivation
  - Couldn’t open /dev/audio though auich was recognized on my PC.
  - USB audio devices, too
- Problem
  - NetBSD requires support for mulaw 8kHz monaural format, and these devices did not support 8kHz.
- Solution
  - Implement sample rate conversion and monaural-stereo conversion
    This was a remedy for devices which supported for no 8kHz.

Audio architecture of NetBSD 1.6 and 2.x



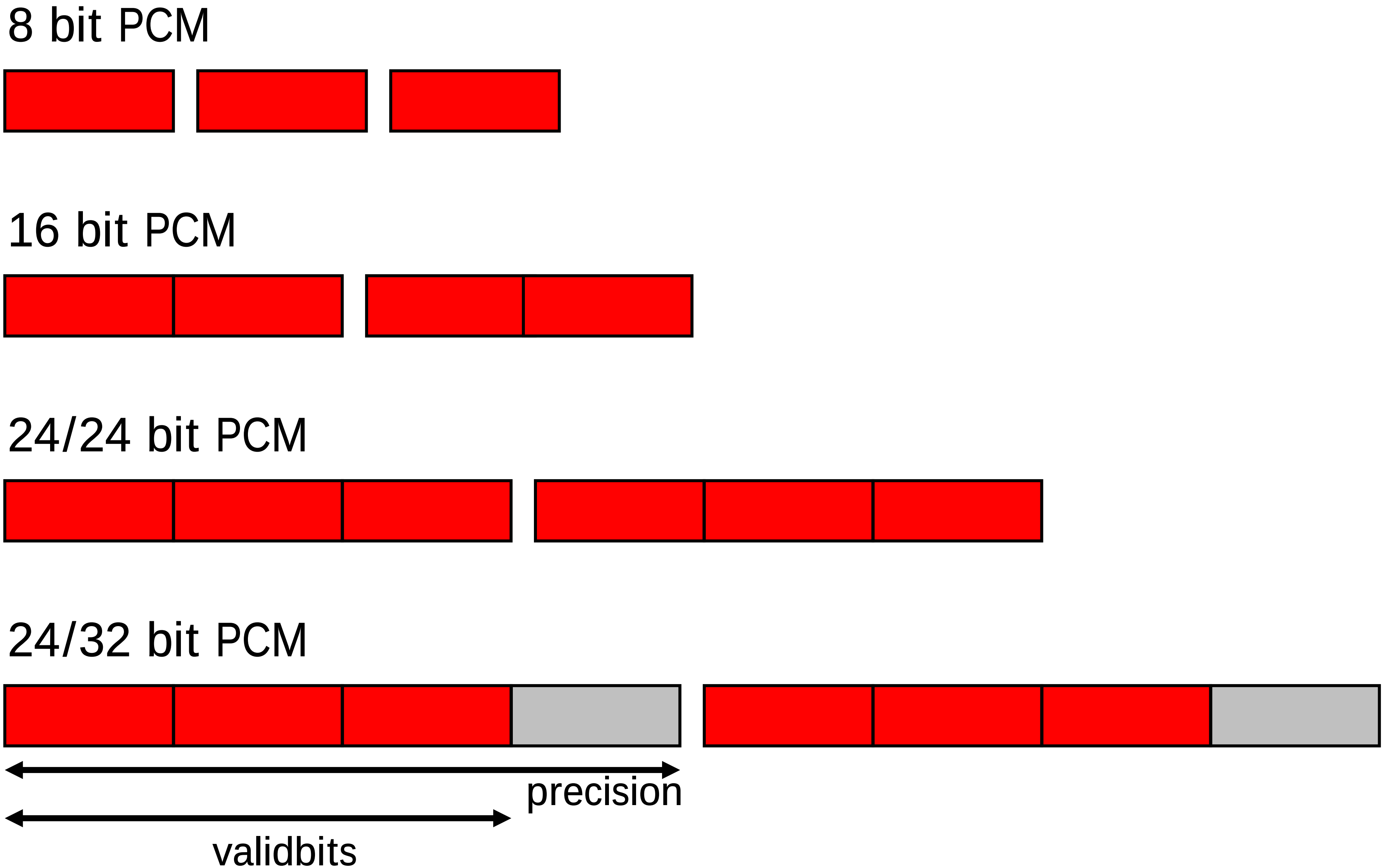
struct audio_format

- Nov. 2004, NetBSD 3.0 or later
- Problems
  - Hardware drivers had similar code in each of *_set_params() functions
  - If a hardware driver missed to set a converter in *_set_params(), an encoding would not be supported though it can be supported.
- Solution
  - Introduced `struct audio_format,' which represents capacities of a hardware, and utility functions to set an appropriate converter automatically

struct audio_format

```
struct audio_format {
  void *driver_data;
  int32_t mode;
  u_int encoding;
  u_int validbits;
  u_int precision;
  u_int channels;
  u_int channel_mask;
  u_int frequency_type;
  u_int frequency[AUFMT_MAX_FREQUENCIES];
};
```

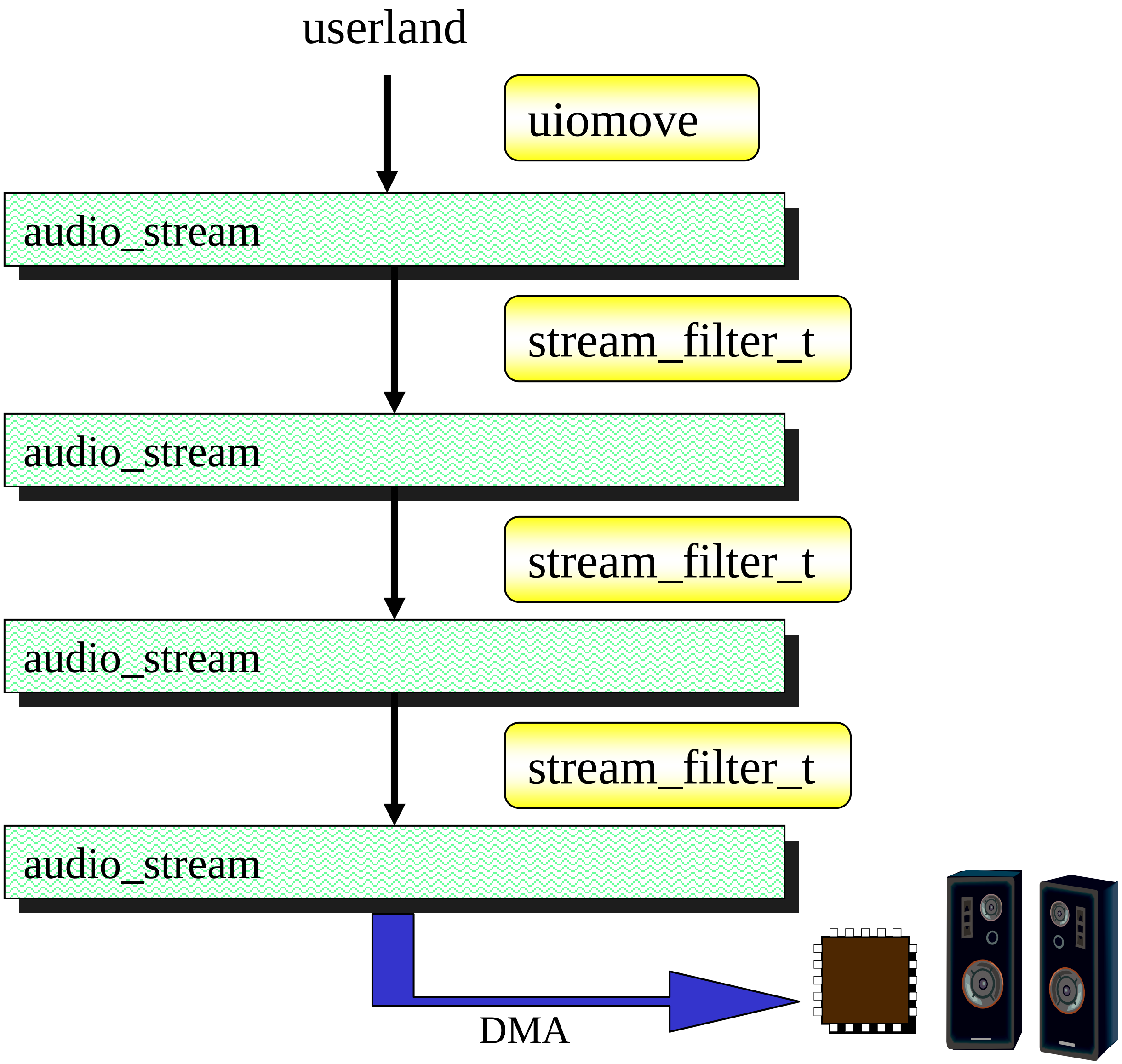
- A problem of distinction between 24/24bit PCM and 24/32bit PCM → validbits, precision
- A problem of channel-speaker mapping → channel_mask
- The `struct audio_format’ represents them, but there are no way to specify them from userland


8 bit PCM
16 bit PCM
24/24 bit PCM
24/32 bit PCM
precision
validbits

Audio filter pipeline

- Dec. 2004 - Jan. 2005, NetBSD 3.0 or later
- Problem
  - Inconsistent interfaces of audio converters in auconv.c, mulaw.c, and aurateconv.c
    - Difference in playback and recording
    - Just one converter can be applied
- Solution
  - Define new interface to represent an audio converter
    - Unify playback and recording
    - Enable multiple converters
    - Represent aurateconv as the new interface
    - Possibility of software reverb, software volume, sample rate conversion against non-PCM, and so on

Audio architecture of NetBSD 3.x

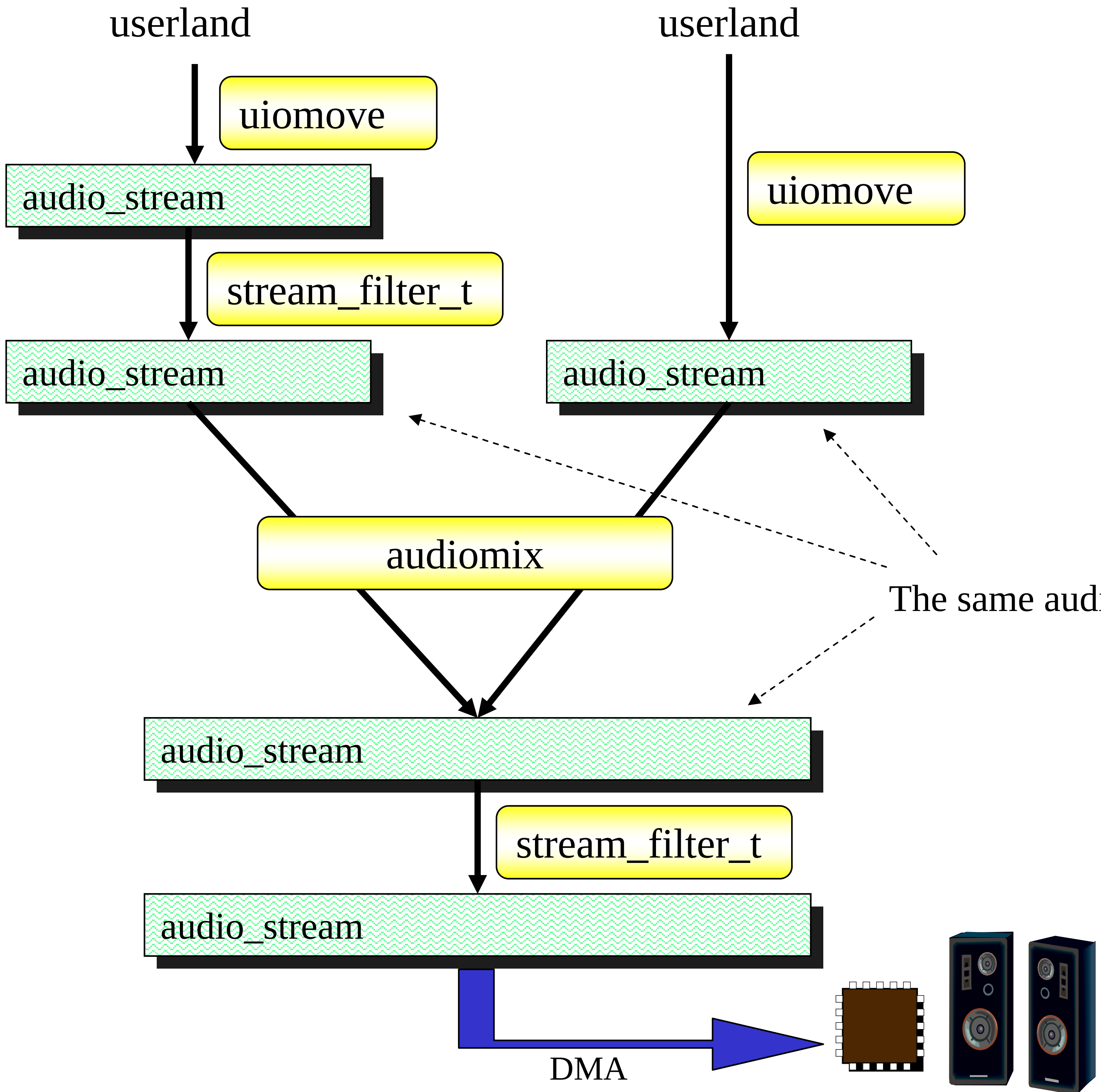


High Definition Audio

- Jun. 2005, NetBSD 3.0 or later
- Motivation
  - A rest
- Problem
  - No driver for High Definition Audio
- Solution
  - Write it
- Unresolved problem
  - No way to specify 24/32bit PCM which is a special feature of High Definition Audio because of a MI audio restriction

In-kernel audio mixing

- Under development, NetBSD 4.0? 5.0?
- Problems to be solved
  - Want to play multiple audio streams on a single audio device
  - Userland solutions such as EsounD and artsd don’t satisfy us because of long delay. In-kernel mixing like Windows or FreeBSD would be better.
  - Need DirectSound API for PEACE


userland
userland
uiomove
uiomove
audio_stream
stream_filter_t
audio_stream
audio_stream
audiomix
The same audio_params
audio_stream
stream_filter_t
audio_stream
DMA

Requirements on the in-kernel audio mixing

- Multiple playback, and exclusive recording
- Don’t change existing userland API (But the behavior might be changed a little)
- Don’t change existing hardware drivers as possible

Issues of implementing the in-kernel audio mixing

- audio.c is mess
- Special handling for some encodings such as AUDIO_ENCODING_MPEG_*, which can not be converted to PCM easily.
- How to support mmap(), or no support?

Comments by the BOF participants

- Support mmap() only for non-mixed non-converted mode
- Need a userland API to get details of a hardware capacity
- mmap() is needed to achieve low-latency like ASIO

- Expose DirectSound-like API to non-PEACE