Sounding Blue

BossaConference 2007

Brad Midgley

slides at flamebot.com

Audio started out wireless

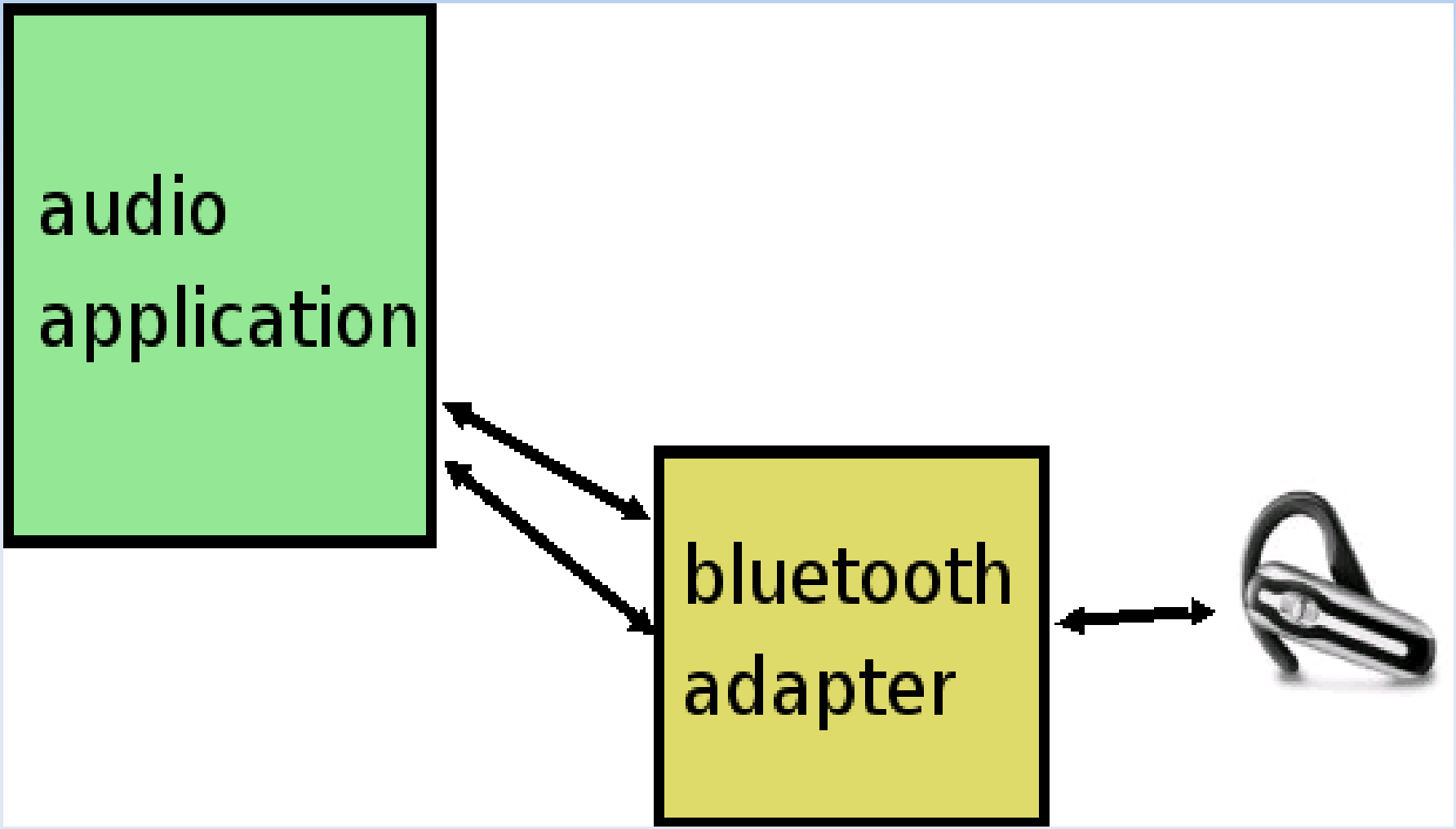
Losing it

The wires have been getting in the way ever since

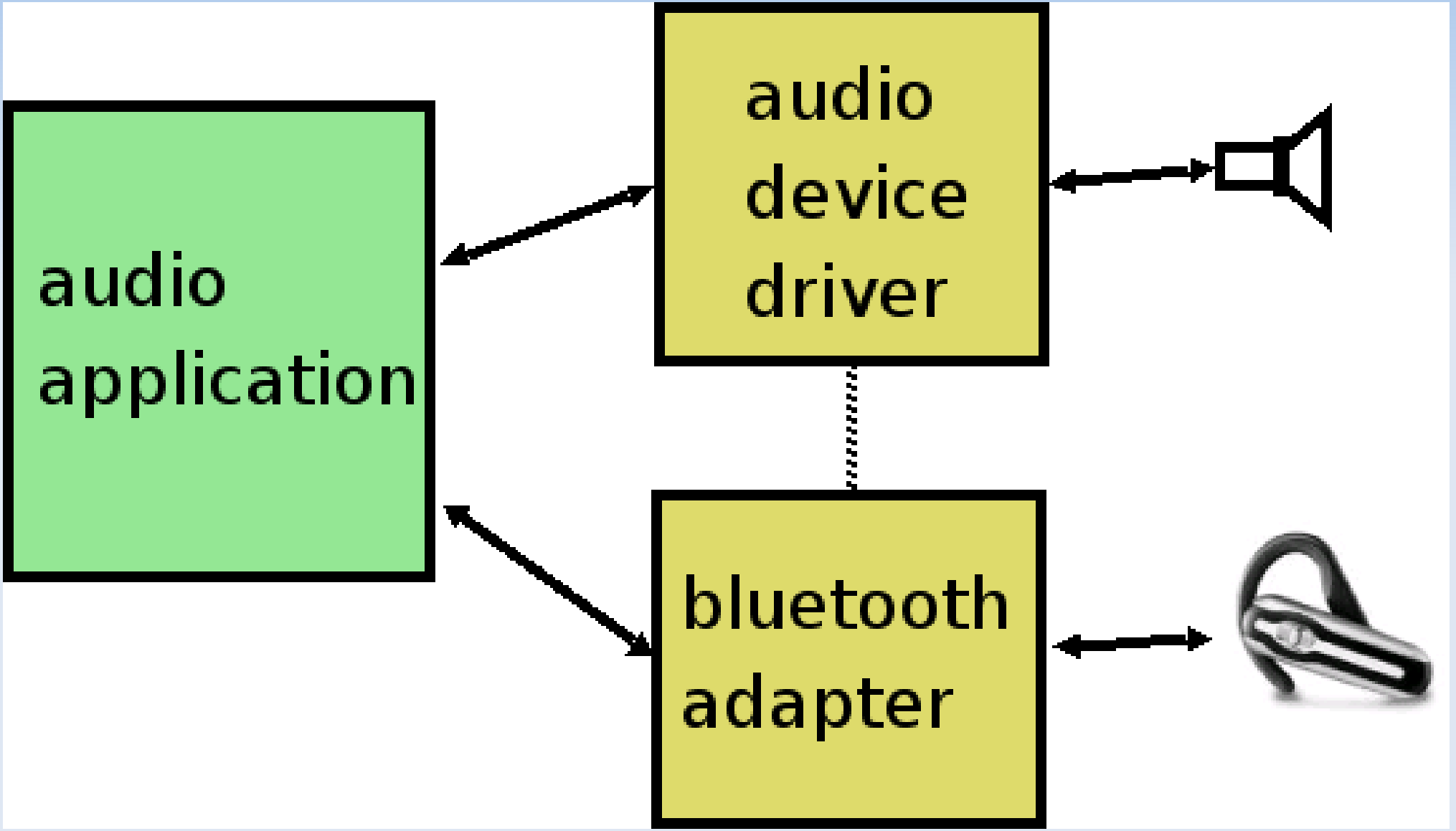
Overview of the Tech

- Voice (“SCO”) headsets
  - use with handset
  - low fidelity
  - two-way monophonic
- HiFi (“A2DP”) headphones
  - higher quality
  - lossy compression
  - higher latency
  - one-way

Basic flow



Hardware SCO optimization



SCO Optimization exceptions

- desktops (eg usb adapter)
- cellphone gateway device
- hacking around dsp

Project history

- btsco (kernel mod + daemon)
- a2play (early floating sbc, a2dp work)
- a2dp alsa plugin (no daemon)
- palmsource a2dp alsa plugin (daemon)
- Chevalier sco plugin (headsetd daemon)
- BlueZ audio service (pcm sco only)
- sbc encoder improvements

Where we are now

- alsa-bluetooth project (prototypes)
  - headsetd for sco (over hci)
  - a2dpd for a2dp
- sbc project
  - new, improved sbc encoder
- BlueZ project
  - headset service for sco (over pcm)

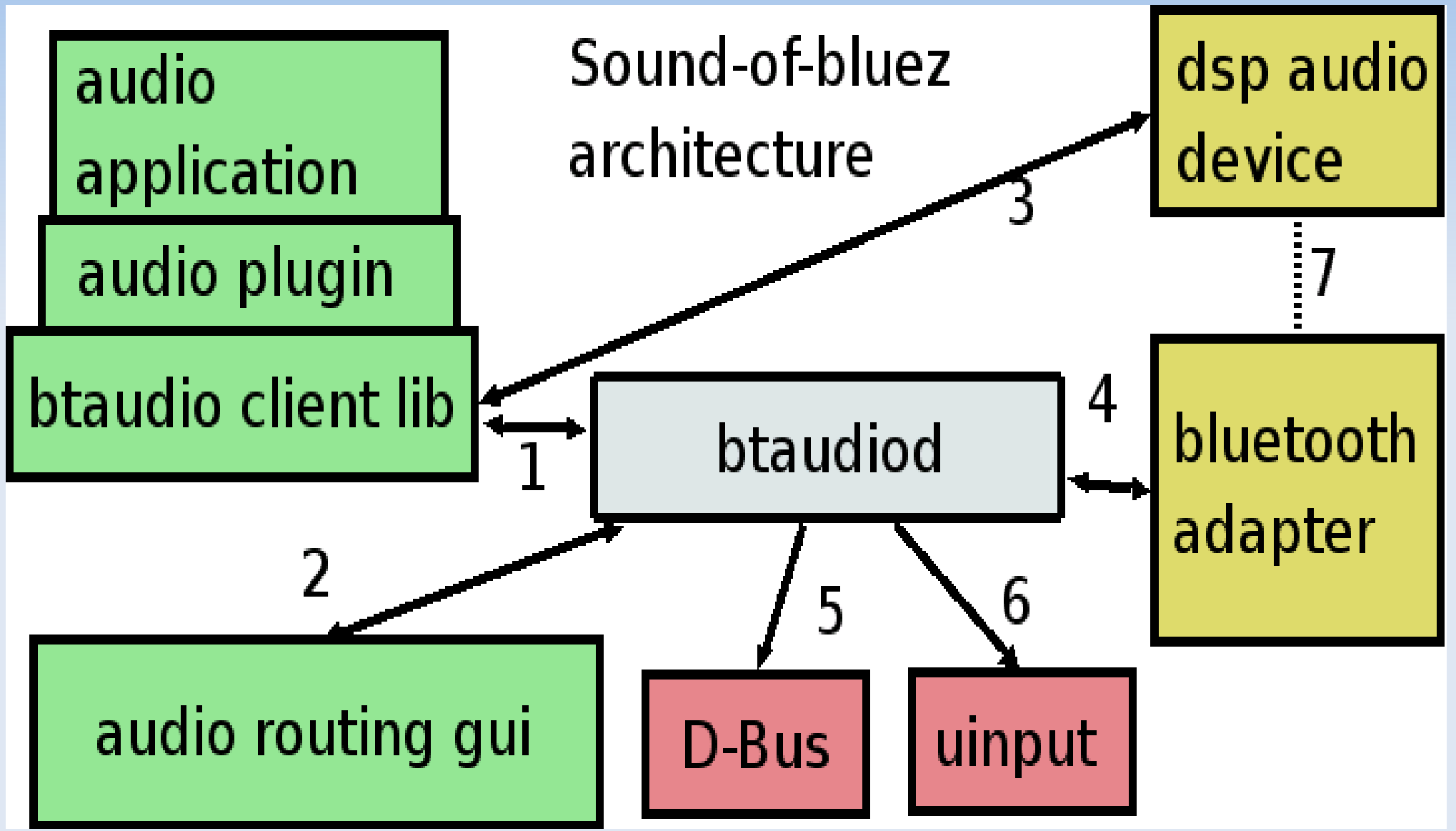
Limitations of the prototypes

- app must choose alsa device
- dynamic switching problems
- all audio types need to merge
- user keeps track of device capabilities
- user must manage device address

BlueZ implementation

- rework the prototypes and put in BlueZ
- single system-wide audio device
- user chooses system-wide audio routing
- client does not need to choose devices
- choose voice or hifi low in driver (audio device capabilities)

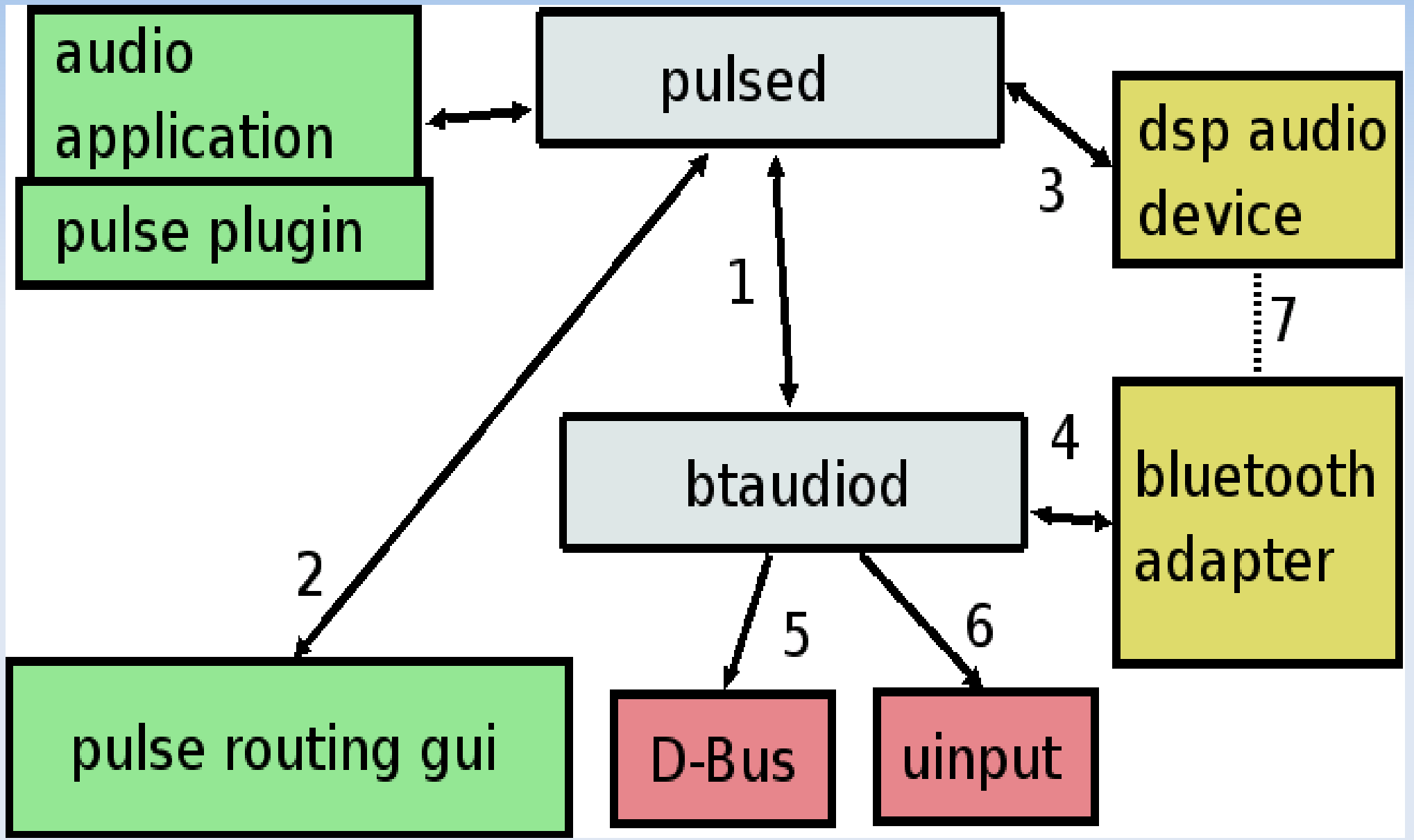
Big picture BlueZ design



What about audio servers?

- audio servers are trying to solve routing
- audio servers migrating from desktop
- pulse also works with dynamic devices
- devices appear/disappear
- let the user route audio system-wide

Audio server integration



Implementation details

- alsa, pulse, etc plugins must be in bluez
  - they will link a private bluez library
  - not like jack plugin, kept inside alsa
  - hopefully plugin APIs are stable
- writing strictly alsa plugins in first pass
- modular design that can be reworked
- audio service shouldn't care about client

Kernel issues

- SCO flow control over HCI
  - overflow/underflow can cause crash
  - quality issues
  - patch to be merged
- multiple simultaneous SCO connections
  - works for uart (embedded) adapters
  - won't work for USB-connected adapters

Optimizations

- do not pass audio through daemon
  - daemon manages file descriptor
  - daemon passes file descriptor to client
- combine with embedding pulse
  - maybe on “very” small devices
- move codec to dsp

Sounding off