Manuel BACHMANN
 **<*manuel.bachmann@iot.bzh*>**

GENIVI AudioManager

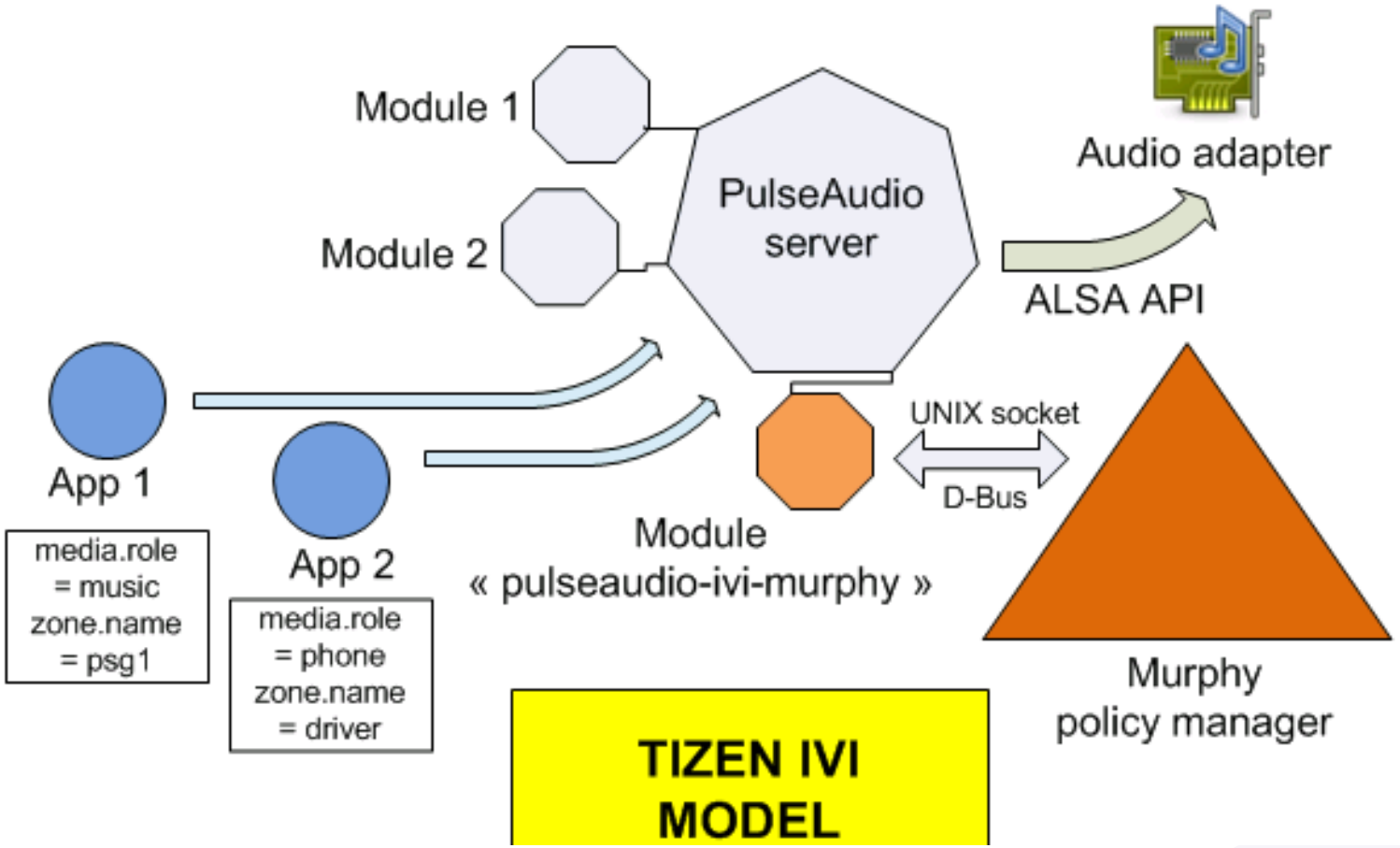
- ***GENIVI AudioManager***
  - Dependencies : 7 not in AGL
    (CommonAPI, DLT -Diagnostic Log and Trace, Persistence Manager, Node-State-Manager…)
  - Code line count (all dependencies) : **144,168**
  - Code line count (AudioManager) : **67,027**
  - Out-of-tree patches needed : 7
  - Code nature : C/C++

Tizen IVI PulseAudio Module (Murphy)

- ***Tizen IVI PulseAudio Module (Murphy)***
  - Dependencies : 1 not in AGL
    (Murphy)
  - Code line count (dependencies : Murphy) : **100,866**
  - Code line count (Tizen IVI PulseAudio module) : **23,499**
  - Out-of-tree patches needed : 17 (mainly for PulseAudio)
  - Code nature : C
    (PS : Tizen IVI puts part of the logic in PulseAudio itself, probably for performance reasons)

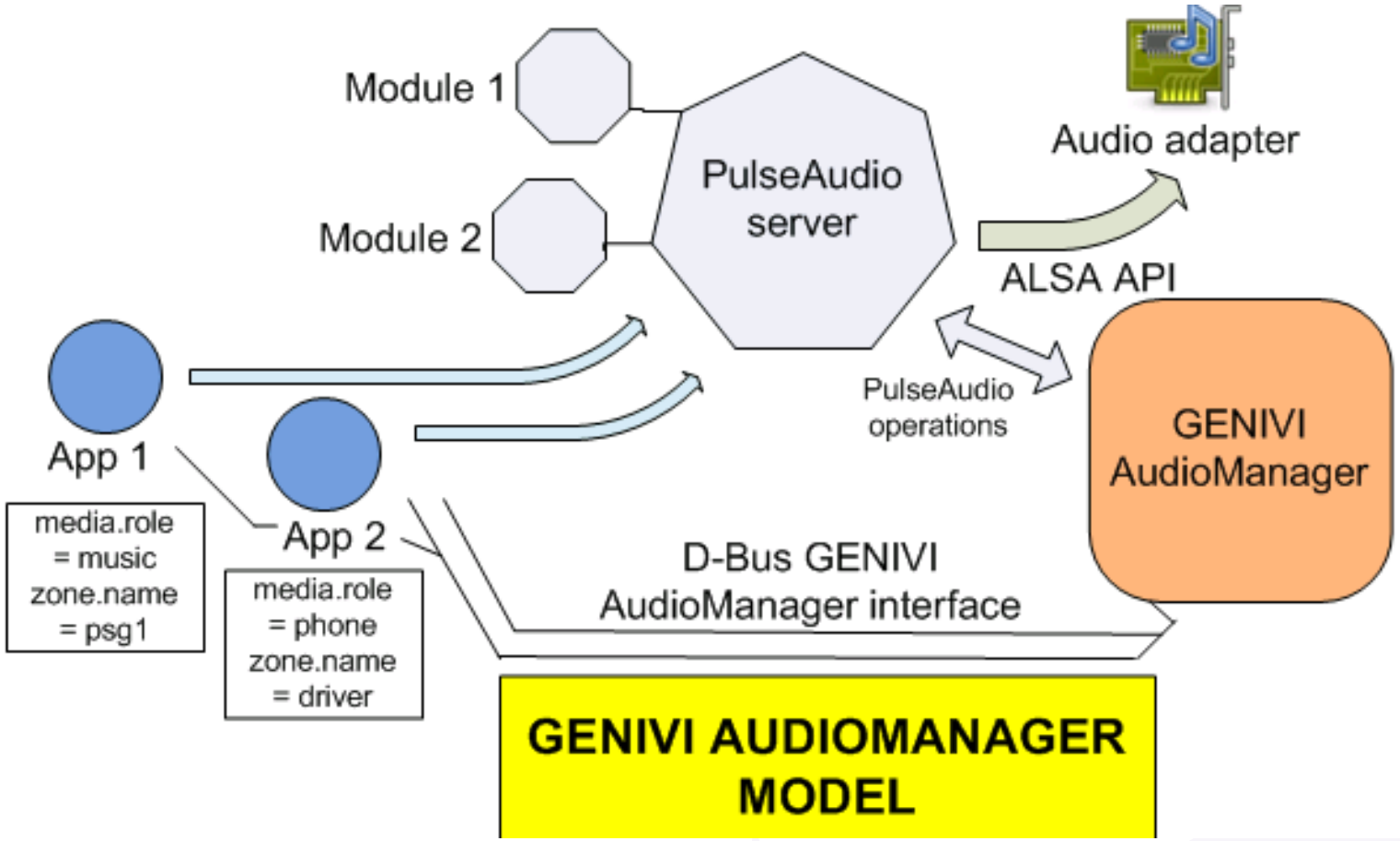
(PS2 : we want to remove/cut down Murphy, thus drastically reducing code line count)

Tizen IVI audio model



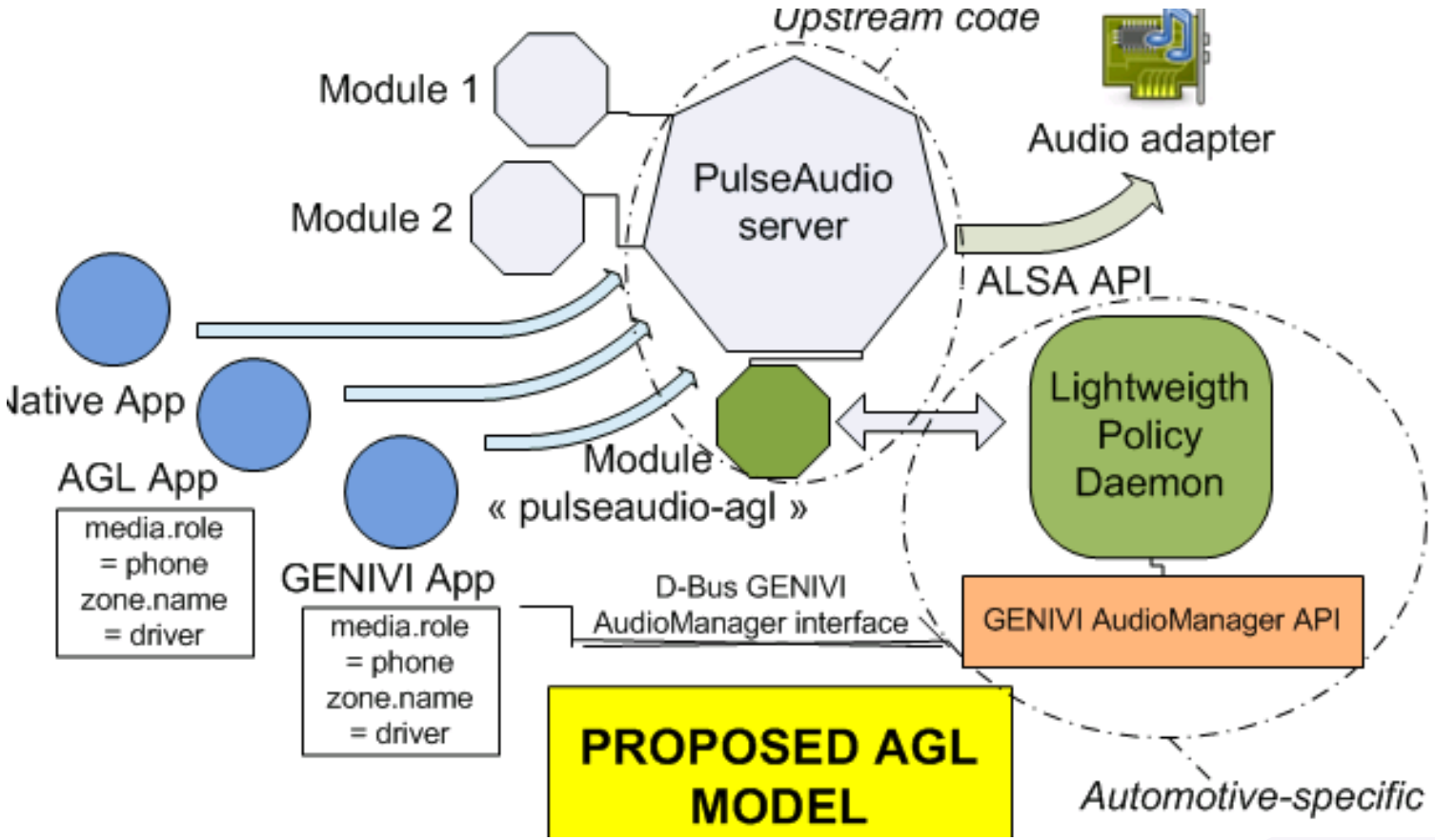
- *Applications use PulseAudio directly*
- *An internal PulseAudio module requests the Policy Manager*
- *Policy Manager (Murphy) uses rules written in LUA langage*
- *PulseAudio module decides*

GENIVI AudioManager audio model



- *Applications use PulseAudio and the GENIVI AudioManager D-Bus interface*
- *GENIVI AudioManager controls PulseAudio (via a plugin)*
- *rules are written in XML files*

Proposed audio model



- *Native applications use PulseAudio directly*
- *AGL applications use PulseAudio with media roles, and the AGL PulseAudio modules applies policy (provided by a stripped-down daemon)*
- *GENIVI applications use the GENIVI AudioManager D-Bus interface provided by a compatibility layer*

Upstreamable PulseAudio code ?

- GENIVI AudioManager PulseAudio Routing Plugin :
  - routing_sender_move_sink_input,
  - routing_sender_move_source_output
  - routing_sender_sink_input_volume_ramp
  - routing_sender_sink_input_volume
  - routing_sender_sink_volume

- Tizen PulseAudio IVI Plugin :
  - mir_router_make_routing
  - mir_volume_make_limiting
  - mir_router_phone_compare
  - mir_volume_suppress
  - mir_volume_correction

(also has a "main-volume-policy" module in PulseAudio, providing a "volume_ramp" API)

Common code

- these APIs live in both codebases :
  - volume_ramp APIs
  - volume_mute APIs
  - move_sink/move_source/routing APIs

Annex

Questions & Answers

Q&A

That's All Folks !