

# *Evolving the Linux-DVB API*

- Steven Toth (stoth@linuxtv.org)
  - Software Engineer, Hauppauge NY.
  - Hauppauge \$100M turn-over, 130 employees.
  - Selling 2 Million TV Tuner products per year.
  - By day, Windows Drivers and Middleware
  - By night, Hack on Linux Drivers for fun! :)
  - High volume of sales = leverage with our silicon suppliers = access to datasheets for Linux drivers.
- 
-

# *Evolving the Linux-DVB API*

- GREAT NEWS! In kernel support for 4 digital television standards:
- ATSC, DVB-C, DVB-T, DVB-S



# *Evolving the Linux-DVB API*

- GREAT NEWS! In kernel support for 4 digital television standards:
  - ATSC, DVB-C, DVB-T, DVB-S
  - 
  - What are we missing? 10+ emerging standards
  - DVB-S2, ISDB-T, DVB-H, DAB2, ATSC-M/H, CMMB, DVB-T2, DVB-C2, DVB-SH, DMB-T/H
- 
-

# *Two proposals for the API*

- Multiproto – In development for 2.5 years.
- S2API – In development for 4 weeks.



# *Multiproto Overview*

- Multiproto – 2.5 years – too little too late?
  - Changes the user-kernel interface, significant internal DVB core structure changes, Tuner and demodulator changes.
  - 10 new function pointers spanning three critical internal interfaces. New enums and structures exposed from the kernel, 5 new IOCTLs.
  - Current API passes fixed structures into the kernel. Multiproto adds more structures to solve the problem.
- 
-

# *S2API Overview*

- S2API – Rebellion – Can we do it better / smaller?
  - Changes the user-kernel interface, minimal internal DVB core structure changes, no tuner changes. 1 Demodulator change.
  - 4 new function pointers spanning 1 critical internal interface. New enums and structures exposed from the kernel, 2 new IOCTLs.
  - S2API uses a simpler structure to pass arbitrary arrays of commands to the tuner hardware, easier to add new TV standards on the fly.
- 
-

- Multiproto - Pros
  - Backwards compatible
  - Fixed length structures which extend the current regime further.
  - Simple tuning API, fill the new structure and issue the IOCTL.
- S2API - Pros
  - Backwards compatible
  - Variable length arrays and command chaining form tuning operations for new and future standards, without having to revisit or extend structures.
  - Simple tuning API, fill the struct then IOCTL
  - Fine-grain tuner control with comands.
- 
-

- Multiproto - Cons
  - Fixed length structures are rigid, padding the union = future proofing?
  - No ability to control board specific features in a clean generic way so it's a tuning only solution.
  - Larger internal changes to critical code and structures.
- S2API - Cons
  - Variable length arrays are different to the current API. Application developers with more tuning control = more room for error.
  - Are board specific feature important? Do they belong in the API? Use cases?
- 
-



# *Adapting the API, new standards...*

- Multiproto
  - Define new structures for any new TV standards, add them to the union and adjust the padding.
  - Application developers implement new structures.
  - S2API
  - Define new commands in the enum for each new modulation field, in any new TV standard.
  - Application developers use the existing structures with these new commands.
- 
-

## *What Haven't I discussed?*

- Application support – MythTV, VDR, Kaffeine
  - DVB-apps toolset integration, SZAP, TZAP
  - Device support – Exactly how TV tuner cards will these patches be enabling?
  - Multi-Frontend patch support
  - What are your issues?
  - Stop and talk with me, or attend the BOF!
- 
-

## *Closing thoughts...*

- Multiproto was designed 2 years ago largely by a single developer. The patchset has a troubled history, mass arguments, in fighting and severe flame mail. Bad for the community.
  - S2API is a direct response to this. It's achieving in 4 week what previous took years.
  - Both API's are now near completion, > 90%.
  - Which API is the right approach?
- 
-