

# Asterisk has helped by showing us what not to do

By Anthony Minessale II

After considerable contribution to Asterisk I have learned a great deal about telephony here is a list of my personal contributions to Asterisk:

<http://www.cluecon.com/anthm.html>

The biggest lesson I have learned is that the fundamentals of Asterisk are built on assumptions and hard coded limitations. The flow chart for its code will make you dizzy:

[http://www.freeswitch.org/astdoc/structast\\_channel\\_coll\\_graph.jpg](http://www.freeswitch.org/astdoc/structast_channel_coll_graph.jpg)

[http://www.freeswitch.org/astdoc/pbx\\_8c\\_incl.jpg](http://www.freeswitch.org/astdoc/pbx_8c_incl.jpg)

People who use asterisk from the outside wouldn't know there is absolutely no structure or discipline in the code and may not care. But once they invest a ton of time trying to make their dream Telco or whatever their dreams may be, the truth is all too obvious. Spoken from experience, only a seasoned technical wizard with years of computer skills to boast will ever be able to successfully implement Asterisk beyond a modest implementation. To truly understand how Asterisk works holds only a slightly smaller prerequisite. To those who find this unimportant, I understand your point, but be aware that Asterisk, being an open source project, needs to have a somewhat easy learning curve to attract new developers especially considering the developer turnover they suffer due to the maddening politics their community has to offer. The development is focused on owning all the code even if it means re-inventing things that already exist just to maintain the right to sell the code. This practice is fine with me though I am less than pleased by the end result when the home-rolled version is a poor contender with several existing solutions. The modular intentions of Asterisk are great though there is no structure there either. Any module can dig its way into nearly all of the code of the core and often, inexperienced module programmers will re-implement existing functionality to the extent that even inside the same C source file, you may find multiple versions of the same functions with different names. The other problem with Asterisk modules are that many of the in-tree modules carry cross dependencies that make it impossible for the core to function without them. Some modules even depend on each other. This practice limits the portability since many operating systems will not tolerate one dynamic object from using symbols from another without hard linking them together. This is not the worst offense as far as portability; there are dozens more with many being accredited to Linux-specific assumptions. Apart from the technology problems the biggest remaining problem to consider is the community. The first experience for most Asterisk newcomers is an IRC channel where people fight for supremacy like information hungry pirates hoarding what they know and then sticking it to people for being so "stupid". (In other words, in the same boat they were in a few months back.) For those of us who are experienced developers, we are used to the l33t thing. The deal breaker is the issue management process. Submissions will generally be ignored for months then a one sentence overview will command the developer to fix minor issues and resubmit. This is

almost tolerable if the submitted code was a new feature but more times than not it also happens with meaningful clean-up and repair of broken core functionality. I have heard this same complaint from countless ex-asterisk contributors over the past year and I am sure it is the number one cause of their ex status.

In conclusion, I actively develop Asterisk code but now I only do it as a consultant. I am quite good at it and I know what I am talking about and I feel that the issues with Asterisk will never be addressed because there may be more Asterisk users every day but there are also less developers every day too and soon all the developers will be nothing but users who have no other choice but to try and be developers. I could go on for ages documenting more issues but they tend to fall on deaf ears. To avoid this problem I have decided to design my own open source telephony application from scratch called FreeSwitch <http://www.freeswitch.org>. It is only 5 months old but I aim to address many of the design flaws I learned from studying Asterisk code. Why don't I just push Asterisk to address them you ask? Well, I did in fact and nobody was interested so I felt I may as well see what I can do. So my biggest thanks to Asterisk is for showing me what not to do!

More Reading:

<http://www.sofaswitch.org>