

---

# Foreword

Once upon a time, there was a boy.

...with a computer

...and a phone.

This simple beginning begat much trouble!

It wasn't that long ago that telecommunications, both voice and data, as well as software, were all proprietary products and services, controlled by one select club of companies that created the technologies, and another select club of companies who used the products to provide services. By the late 1990s, data telecommunications had been opened by the expansion of the Internet. Prices plummeted. New and innovative technologies, services, and companies emerged. Meanwhile, the work of free software pioneers like Richard Stallman, Linus Torvalds, and countless others were culminating in the creation of a truly open software platform called Linux (or GNU/Linux). However, voice communications, ubiquitous as they were, remained proprietary. Why? Perhaps it was because voice on the old public telephone network lacked the glamor and promise of the shiny new World Wide Web. Or, perhaps it's because a telephone just isn't as effective at supplying adult entertainment. Whatever the reason, one thing was clear. Open source voice communications was about as widespread as open source copy protection software.

Necessity (and in some cases simply being cheap) is truly the mother of invention. In 1999, having started Linux Support Services to offer free and commercial technical support for Linux, I found myself in need (or at least in perceived need) of a phone system to assist me in providing 24-hour technical support. The idea was that people would be able to call in, enter their customer identity, and leave a message. The system would in turn page a technician to respond to the customer's request in short order. Since I had started the company with about \$4000 of capital, I was in no position to be able to afford a phone system of the sort that I needed to implement this scenario. Having already been a Linux user since 1994, and having already gotten my feet wet in Open Source software development by starting l2tpd, gaim, and cheops,

and in the complete absence of anyone having explained the complexity of such a task, I decided that I would simply make my own phone system using hardware borrowed from Adtran, where I had worked as a co-op student. Once I got a call into a PC, I fantasized, I could do *anything* with it. In fact, it is from this conjecture that the official Asterisk motto (which any sizable, effective project must have) is derived:

*It's only software!*

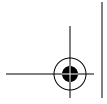
For better or worse, I rarely think small. Right from the start, it was my intent that Asterisk would do *everything* related to telephony. The name “Asterisk” was chosen because it was both a key on a standard telephone and also the wildcard symbol in Linux (e.g., `rm -rf *`).

So, in 1999, I have a free telephony platform I've put out on the web and I go about my business trying to eke out a living at providing Linux technical support. However, by 2001, as the economy was tanking, it became apparent that Linux Support Services might do better by pursuing Asterisk than general purpose Linux technical support. That year, we would make contact with Jim “Dude” Dixon of the Zapata Telephony project. Dude's exciting work was a fantastic companion to Asterisk, and provided a business model for us to start pursuing Asterisk with more focus. After creating our first PCI telephony interface card in conjunction with Dude, it became clear that “Linux Support Services” was not the best name for a telephony company, and so we changed the name to “Digium,” which is a whole other story that cannot be effectively conveyed in writing. Enter the expansion of Voice over IP (“VoIP”) with its disruptive transition of voice from the old, circuit-switched networks to new IP-based networks and things really started to take hold.

Now, as we've already covered, clearly most people don't get very excited about telephones. Certainly, few people could share my excitement the moment I heard dial-tone coming from a phone connected to my PC. However, those who *do* get excited about telephones get *really* excited about telephones. And facilitated by the Internet, this small group of people were now able to unite and apply our bizarre passions to a common, practical project for the betterment of many.

To say that telecom was ripe for an open source solution would be an immeasurable understatement. Telecom is an enormous market due to the ubiquity of telephones in work and personal life. The direct market for telecom products has a highly technical audience that is willing and able to contribute. People demand their telecom solutions be infinitely customizable. Proprietary telecom is very expensive. Creating Asterisk was simply the spark in this fuel rich backdrop.

Asterisk sits at the apex of a variety of transitions (Proprietary → Open Source, Circuit Switched → VoIP, Voice only → Voice, Video, and Data, Digital Signal Processing → Host Media Processing, Centralized Directory → Peer to Peer) while easing those transitions by providing bridges back to the older ways of doing things. Asterisk can talk to anything from a 1960s era pulse dial phone to the latest wireless VoIP



devices, and provide features from simple tandem switching all the way to bluetooth presence and DUNDi.

Most important of all, though, Asterisk demonstrates how a community of motivated people and companies can work together to create a project with a scope so significant that no one person or company could have possibly created it on its own. In making Asterisk possible, I particularly would like to thank Linus Torvalds, Richard Stallman, the entire Asterisk community and whoever invented Red Bull.

So where is Asterisk going from here? Think about the history of the PC. When it was first introduced in 1980, it had fairly limited capabilities. Maybe you could do a spreadsheet, maybe do some word processing, but in the end, not much. Over time, however, its open architecture led to price reductions and new products allowing it to slowly expand its applications, eventually displacing the mini computer, then the mainframe. Now, even Cray supercomputers are built using Linux-based x86 architectures. I anticipate that Asterisk's future will look very similar. Today, there is a large subset of telephony that is served by Asterisk. Tomorrow, who knows what the limit might be.

So, what are you waiting for? Read, learn, and participate in the future of open telecommunications by joining the Asterisk revolution!

—Mark Spencer

