

## **David Muir Sharnoff**

resume addendum: Idiom ISP details

<http://dave.sharnoff.org/resume>

### *Details of technical work at Idiom*

**Billing system:** I built a billing system to handle the retail customers. Initially I concentrated on clear presentation of the items charged. At first it billed in arrears with anniversary-day monthly billing and could prorate partial months for new/expiring services. Later, as I purchased other ISPs and incorporated their billing procedures, I expanded it to handle billing in advance with automatic adjustments for changes; non-monthly billing cycles; and generating pretty PDF bills (scaling the fonts and formatting to try to keep the bill on a single page). The billing system consistently presented accurate and clear bills. It did a very effective job of nagging laggards to catch up. It kept credit card numbers on a separate secure server.

**Customer care:** I built a customer self-service web site where customers could see their previous invoices; preview their next (tentative) invoice; change their payment options; pay their bill; forward their email; set an auto-responder; route email for their domain; choose spam filtering options; review their services; and access sub-accounts.

**Inbound spam filtering:** Using a combination of Spam Assassin, Clam AV, Bogofilter, public blacklists, selective grey-listing, postfix rules, procmail rules, log-generated blacklists, and custom perl scripts I built an effective spam filtering system. Based on the number of complaints from my users, I beat the spammers: about %0.25% false negative rate with only rare complaints about a false positive. Some of the bits I built for this ended up on CPAN: Daemon::Generic, Plugins, SyslogScan::Daemon, and SyslogScan::Daemon::SpamDector.

**Outbound spam filtering:** I allowed my users to forward their email off-site. This created a problem because users could choose to forward all their email (including spam) to services that would blacklist spam senders. To handle this, I wrote Qpsmtpd::Plugin::Quarantine and SyslogScan::Daemon::BlacklistDetector. I quarantined outbound email that looked like it was spammy, letting both the sender and the recipient choose to release the mail from quarantine. I noticed when I was blacklisted and automatically routed around the problem while I tried to get the blacklist removed.

**DSL Customer Migrations:** In the process of providing wholesale DSL services, I sometimes found myself needing to make bulk changes to Cisco router configurations. I wrote a whole bunch of perl scripts and one CPAN module, Cisco::Reconfig, to help with this. I was able to use my programs clean up the complex hand-generated configurations that grew from years of work by my provisioning team.

**Systems administration:** I was the main systems administrator for the 20-ish servers that supported my various brands and customers. Some of the issues I resolved included: preventing permission bleed-over in CGIs before suexec existed; upgrading FreeBSD systems (Debian makes it very easy, but older FreeBSD versions do not); upgrading perl without breaking customer scripts and CGIs; automatic fail-over of critical services (authoritative & lookup DNS; radius); mysql performance tuning for SpamBeyes; providing netnews service without using much bandwidth; and backups.

**Network management:** An ISP requires reliable networking and I made things work using various protocols (BGP, OSPF, HSRP, RIP, tunnels) on various equipment (Cisco, Livingston, BSD with gated). Eventually, I settled on a nearly 100% Cisco configuration but it was a long evolutionary process to get there.

**imail migration:** One of the ISPs I bought used an email system that required a significant effort to convert away from Windows. It supported IP-based virtual hosting so that you could pick up your email with POP or IMAP logging in with just "user" rather than "user@host" by using "mail.host" for the server. You could also control mail routing via the web at "mail.host". I put together a service with these features using postfix, dovecot, tpop3d, postfix.admin, and squirrelmail. I imported the metadata from in the old system: It was stored in the Windows registry, Microsoft SQL Server, LDAP, and in files on a Windows server. The mailbox format was proprietary. By using my entire network of systems to do the conversion work in parallel, I was able to do a hard cut-over to the new system with only two hours of email downtime.

**High Availability Virtual Servers:** Using drbd to mirror the disk across the network, openvz for virtual servers, and custom scripts on top of CARP for fail-over; I put together a highly-available virtual server setup. On the failure of one of the host systems, all the virtual servers automatically migrate and restart.

**OOPS:** I built a object-oriented persistent store for perl objects. It stores nearly any perl object into a relation database (mysql, PostgreSQL, or sqlite) in a way that is theoretically querryable (not BLOBs). Access to persistent objects is fully transaction safe. Persistent objects are reference counted and I built a background garbage collector for persistent data to persistent leaked objects (with circular references). This module is used by Qpsmtpd::Plugin::Quarantine.

**VOIP Configuration:** Idiom operated as four separate brands. Customers from the various brands where not aware of the other brands. All Idiom employees worked from home. I set up Asterisk to rewrite the caller id of the incoming calls to display which brand was called and to remember which numbers called which brand so that outbound calls could appear from the right brand.