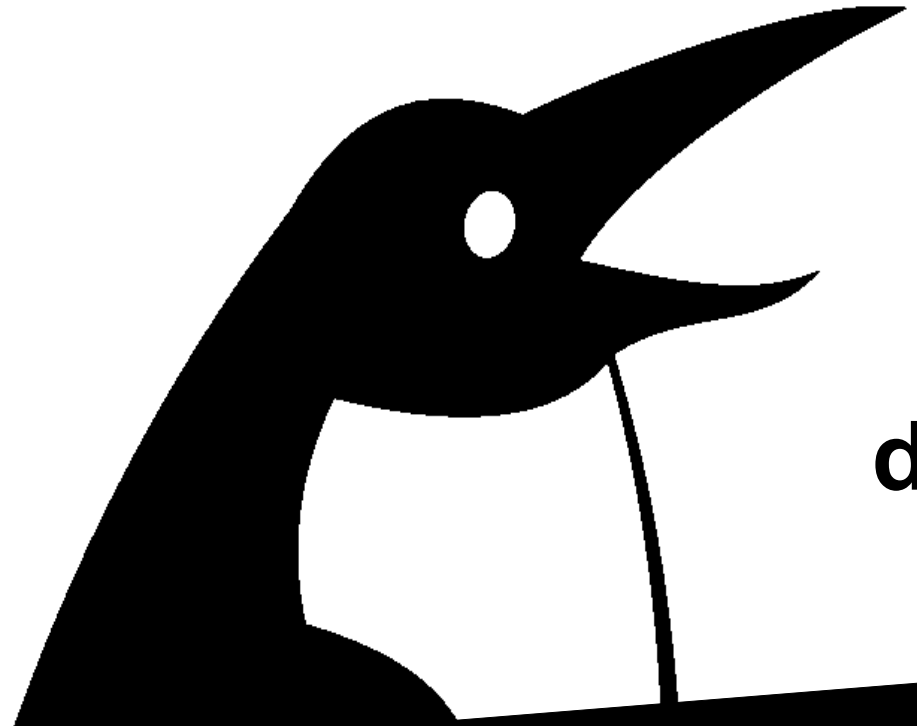
Scaling a Spam Filter



2017-05-04

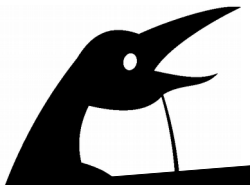
Dianne Skoll

Roaring Penguin Software Inc.

dfs@roaringpenguin.com

Outline

- Short History of Roaring Penguin
- CanIt Architecture and Capabilities
- Scaling a Spam Filter
- Q&A

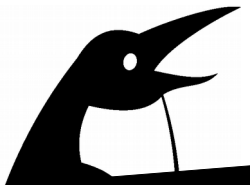


History of Roaring Penguin

- Started in 1999 by Dianne Skoll as Linux consultancy.

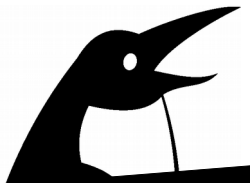
History of Roaring Penguin

- First release of MIMEDefang (GPL'd email filtering framework) in 2000.
- First commercial release of CanIt in 2002.
- Currently 12 employees; filtering mail for hundreds of thousands of people both with SAAS and on-premise appliances.



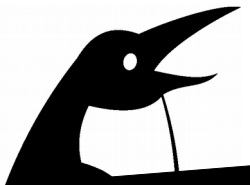
Our Toolkit

- Linux OS for desktop and server
    - Even non-technical employees
- CanIt is developed in:
    - Perl: Filtering framework
    - PHP: Web interface
    - C: Performance-critical components
- Revision control: git
- Continuous integration: buildbot



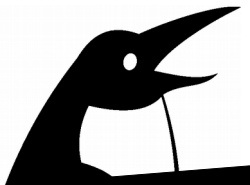
Our Toolkit - 2

- Ticket-tracking: RT (Request Tracker) from Best Practical
- Documentation production: LaTeX and htlatex
- Monitoring: Xymon (used to be “Hobbit”)
- Metric tracking: Munin
- All of our tools are free and open-source!

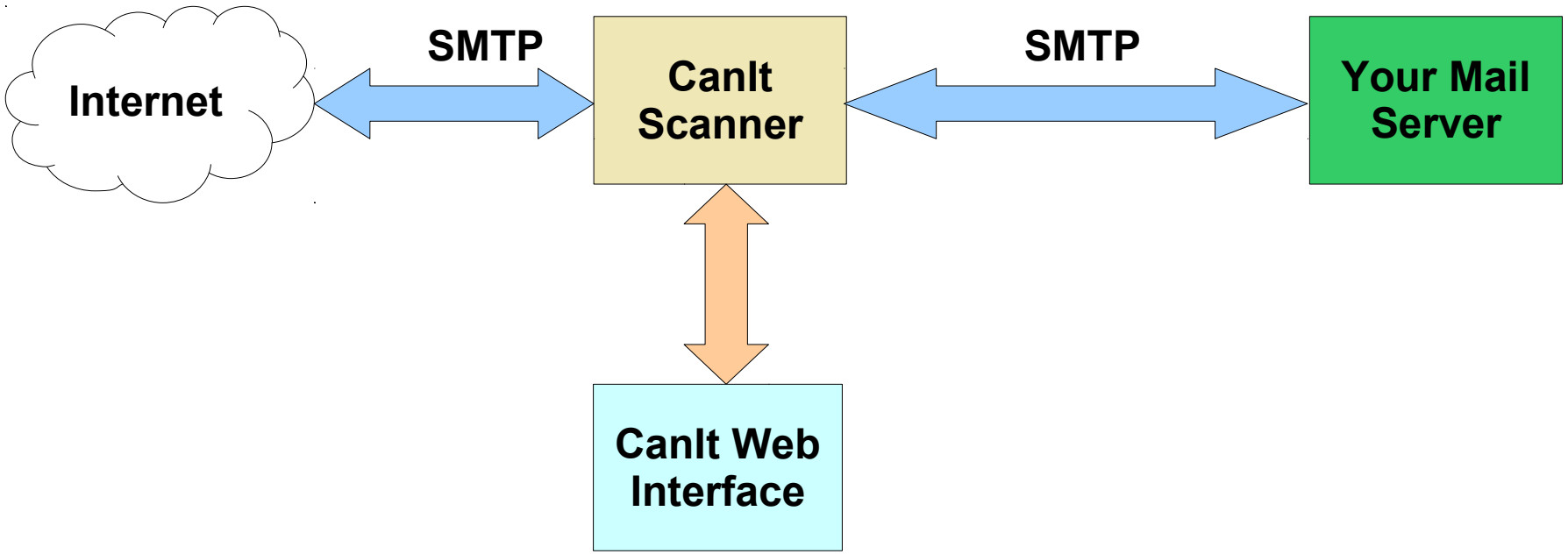


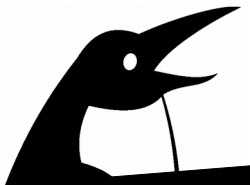
CanIt

- CanIt is a family of email security products:
  - CanIt-PRO: Suitable for one organization
  - CanIt-Domain-PRO: Multi-tenant version suitable for multiple organizations
  - Hosted CanIt: CanIt-Domain-PRO as a service.
  - Secure Messaging add-on
  - Email Archiving add-on



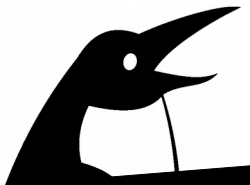
CanIt High-Level Architecture





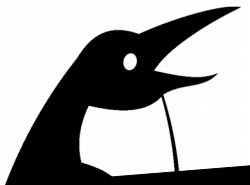
CanIt High-Level Architecture

- MTA is Sendmail
- Filtering is done via Sendmail's “Milter” API.
- Basic filtering framework is MIMEDefang, consisting of a C supervisor and Perl workers.
- Filtering code is Perl.
- UI is PHP.



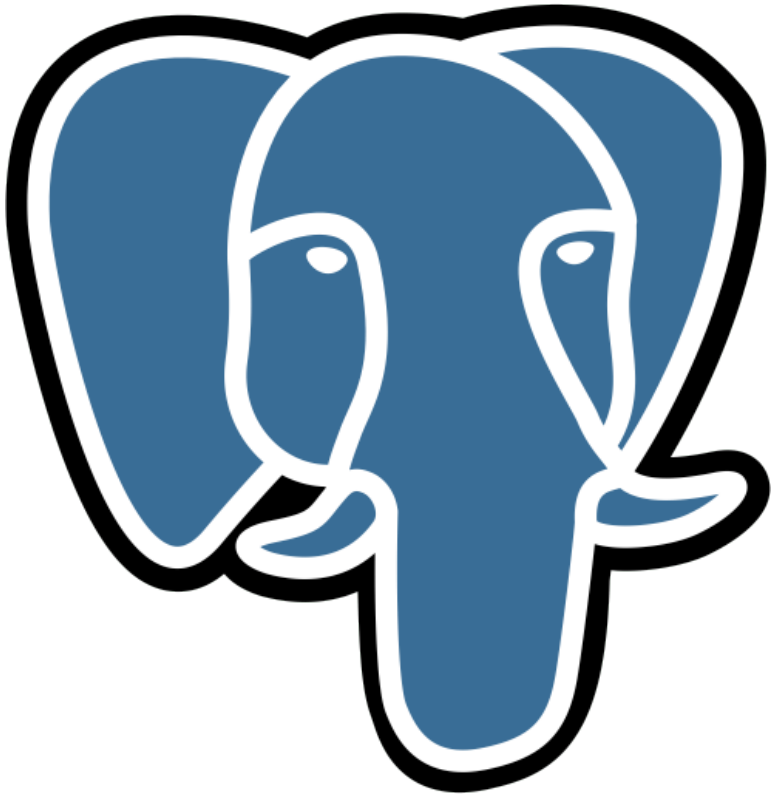
The Elephant in the Room

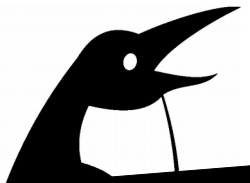
- A brief digression as I address the elephant in the room, especially for those of you who met me more than a couple of years ago...



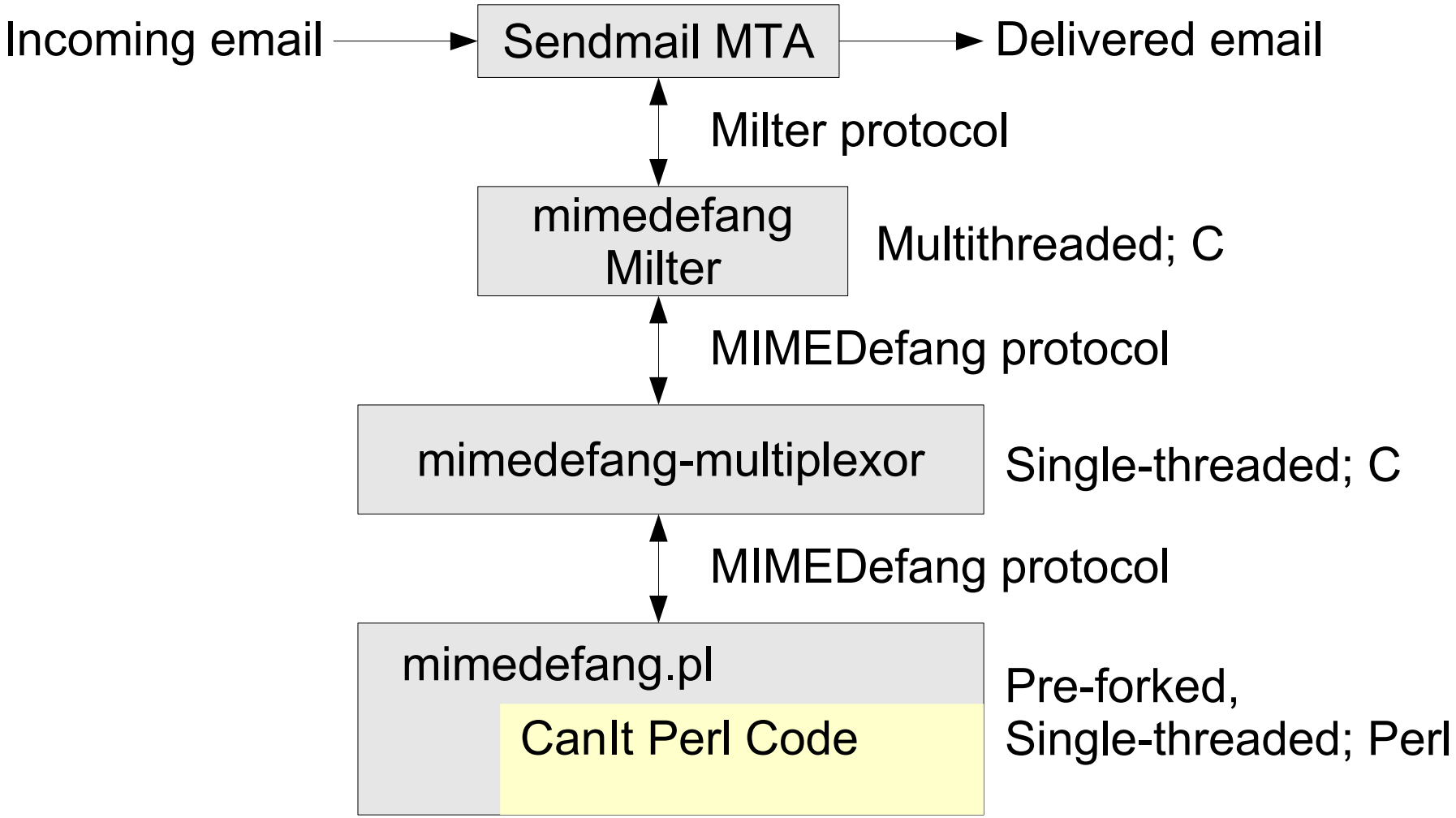
The Elephant in the Room

- Our database is PostgreSQL



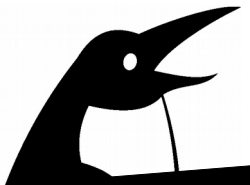


Details of a Scanner



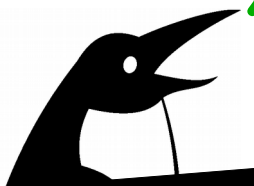
Typical SMTP Conversation

```
C: Connect to server
S: 220 server_hostname ESMTP Sendmail 8.14.4...
C: HELO client_hostname
S: 250 server_hostname Hello client_hostname, pleased...
C: MAIL FROM:<dfs@roaringpenguin.com>
S: 250 2.1.0 <dfs@roaringpenguin.com>... Sender ok
C: RCPT TO:<foo@roaringpenguin.com>
S: 250 2.1.5 <foo@roaringpenguin.com>... Recipient ok
C: RCPT TO:<bar@roaringpenguin.com>
S: 250 2.1.5 <bar@roaringpenguin.com>... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: (transmits message followed by dot)
S: 250 2.0.0 h0AJVcGM007686 Message accepted for delivery
C: QUIT
S: 221 2.0.0 server_hostname closing connection
```



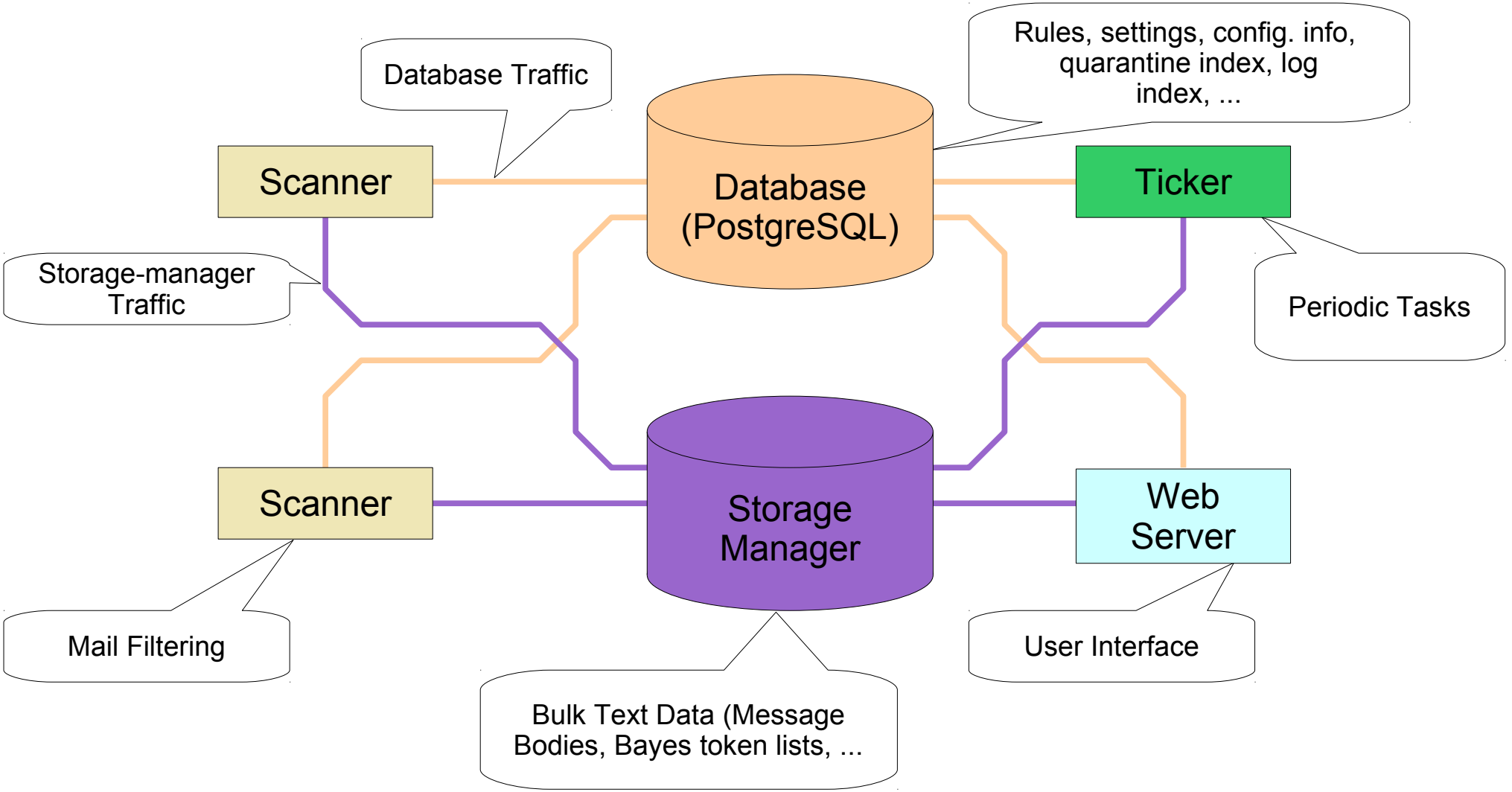
SMTP Conversation with Milter

```
C: Connect to server                                                    *
S: 220 server_hostname ESMTP Sendmail 8.14.4...
C: HELO client_hostname                                                 *
S: 250 server_hostname Hello client_hostname, pleased...
C: MAIL FROM:<dfs@roaringpenguin.com>                                   *
S: 250 2.1.0 <dfs@roaringpenguin.com>... Sender ok
C: RCPT TO:<foo@roaringpenguin.com>                                     *
S: 250 2.1.5 <foo@roaringpenguin.com>... Recipient ok
C: RCPT TO:<bar@roaringpenguin.com>                                     *
S: 250 2.1.5 <bar@roaringpenguin.com>... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: (transmits message followed by dot)                                  *
S: 250 2.0.0 h0AJVcGM007686 Message accepted ...
C: QUIT
S: 221 2.0.0 server_hostname closing connection
```



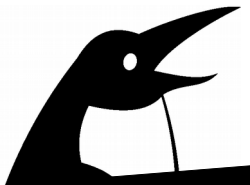
* = response-modification opportunity * = filtering opportunity

All the Moving Parts... AIEEE!



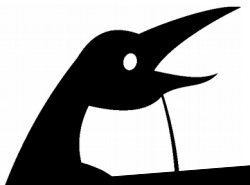
All the Moving Parts - 2

- On small installations: All pieces can run on the same machine (or even virtual machine).
    - A small installation is one that does < 50k messages per day.
- On larger installations, pieces may be split out over multiple hosts.
- You can have multiple scanners and Storage Manager nodes
- Still only one (active) database per cluster!



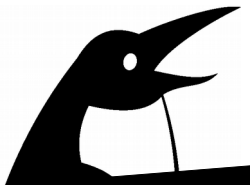
CanIt Features

- Hierarchical administration: Accounts can have sub-accounts and so forth.
    - Resellers can make containers for their clients and manage them all.
    - Each client can only see his or her container.
- Per-user settings, block/allow lists, rules, and preferences.
    - No arbitrary limit on the number of rules a user can make



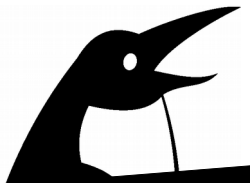
CanIt Features - 2

- Support for SPF, DMARC, DKIM, name-your-email-buzzword.
- Fully IPv6 compliant.
- Novel Bayes implementation that shares votes across our entire user base while still allowing per-user statistics.
- Outbound rate-limiting to catch internal spammers or compromised users.

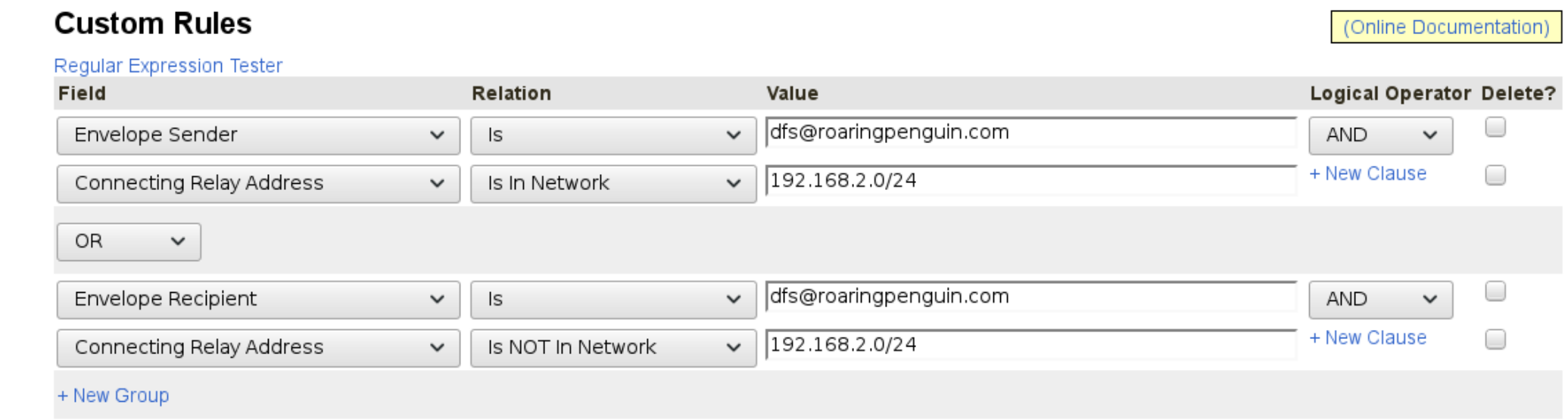


CanIt Super-Duper Features - 3

- Custom Rules that support logical operators (AND/OR) and two-level grouping.
    - The rule is entered with a GUI and then we compile it down to Perl for speed.
- Admins only: Log-indexing and searching.
    - You have full visibility into your mail logs via the Web interface.
- Integration with LDAP/IMAP/POP3 for authentication and alias resolution (LDAP only)



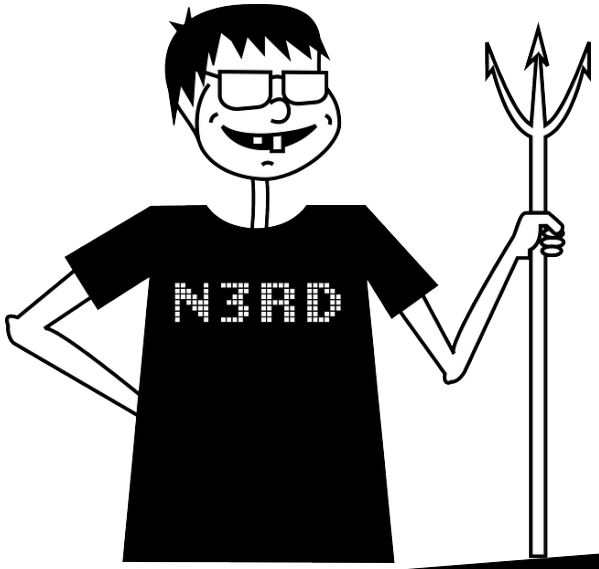
Custom Rules – Graphical Input



Custom Rules – Compiled Output

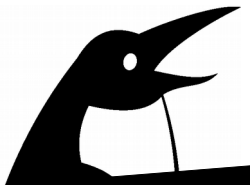
```
my $real_relay_address = $message->get_real_relay_address() || '';
my $envelope_sender = lc($message->get_envelope_sender());
my $envelope_recipients = [ map { lc } @{$message->get_envelope_recipients()} ];


if ((($envelope_sender eq "dfs\x{40}roaringpenguin.com") &&
      (CanIt::CompoundRuleUtils::is_in_network($real_relay_address, "192.168.2.0/24") )) ||
     ((grep { $_ eq "dfs\x{40}roaringpenguin.com" } @{$envelope_recipients}) &&
      (!CanIt::CompoundRuleUtils::is_in_network($real_relay_address, "192.168.2.0/24") ))) {
         return 1;
} else {
         return 0;
}
```



Enough with the Features!

- OK. Enough. All those features are great. We understand that CanIt is insanely flexible and powerful.
- But can it handle load?
- Is it scalable?
- What has Roaring Penguin learned about scalability in the last 15 years?



Email Filtering Hates You

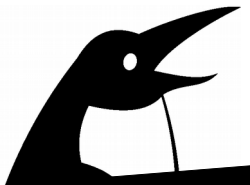
- Email filtering is just about the **worst** use-case for stressing out a computer.
- The filtering process is typically CPU-intensive.
- High-volume mail delivery hammers your disks.

Email Filtering - 2

- Tracking everything in a database **really** hammers your disks.
    - There are way more write transactions than read transactions! People don't bother checking their quarantines often.
    - This is precisely the ***opposite*** pattern for which most database systems are optimized.

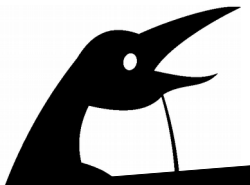
Email Filtering - 3

- People expect a responsive GUI.
- Rendering a page may take from 10 to 500 database queries, depending on the page.
- Some things just cannot be made fast enough.
    - So we cheat.



CPU Bottlenecks

- Modern CPUs are crazy-fast.
- We do try to make our code efficient, but we don’t go nuts optimizing it to death. Efficient algorithms are more important than micro-optimizations.

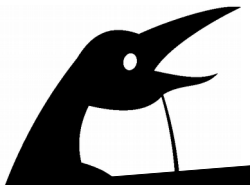


CPU Bottlenecks

- Lesson 1: ***Don't cheap out on hardware.***
- Lesson 2: ***If you do cheap out on hardware, you will have angry customers and lose far more money than you save.***

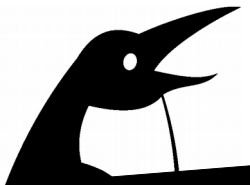
RAM Bottlenecks

- RAM can be a limiting resource.
- Our Perl scanning processes are memory-hungry! Plan on allocating 100MB for each scanning process.
- If you do start swapping, it's game over. The system will spiral down to a horrible death.
- So *don't ever let your systems swap!!!* It will give you bad memories.



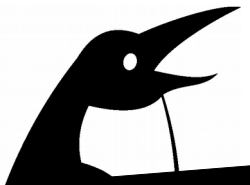
Network Bottlenecks

- Network bottlenecks are not typically a problem unless you cheap out on hardware or connectivity (see Lessons 1 and 2).
- However, if you design your cluster badly, network bottlenecks ***can*** bite you.
    - It might seem like a good idea to operate two data centres 200km apart linked over the public Internet.
    - We did that for a while. The intra-cluster latency was killing us.



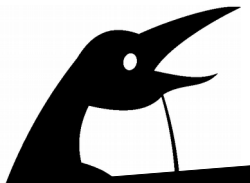
Network Bottlenecks - 2

- Although geographically-dispersed clusters are appealing from the point of view of robustness, unless you have a *reliable, low-latency* link between the sites, it's not worth the trouble.
- We had two Hosted CanIt data centres (Rogers in Kanata and iWeb in Montreal.)
- Closed the iWeb one and consolidated everything in Rogers.



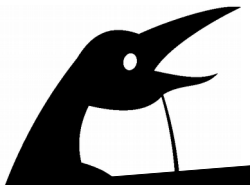
Disk Bottlenecks – The #1 Killer

- Disk I/O bandwidth is typically the most important performance-killer.
- Mail queues are on disk and files must be sync'd before an SMTP transaction can be ack'd.
    - Thou Shalt Lose No Mail
    - Though Spam it May Be
    - And Thou Shalt Not take the Name of thy Postmaster in vain



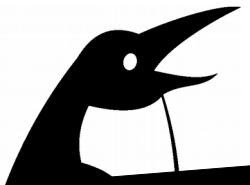
Disk Bottlenecks - 2

- A database transaction *cannot* complete until it's safely stored on disk.
    - Thou Shalt Lose No Transactions
    - Though Pointless they May Be
    - And Thou Shalt Not take the Name of thy DBA in vain.



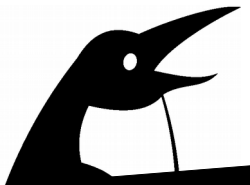
Disk Bottlenecks - 3

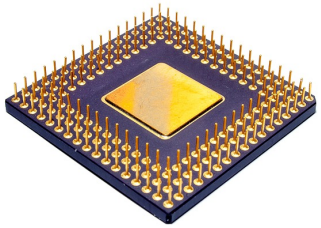
- Message bodies are stored in files by the Storage Manager servers.
- A message cannot be considered safely in the quarantine until it is sync'd to disk.
    - Thou Shalt Lose No Quarantined Message
    - Spam Though it Almost Certainly Is
    - Thou Shalt Not take the Name of Prince Abdiki Mumbassa of Nigeria with his 5 Million United States Dollars in vain.



Scaling in Real Life (so-called)

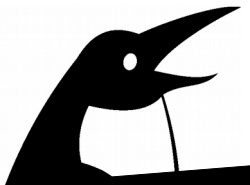
- So... how do we scale up?
- Hosted CanIt filters email for about 160,000 users.
    - That makes it 160 micro-Googles.
- We peak at about 116 messages/second or 10 million messages per day.
- Average weekday traffic is 4.5 million messages.





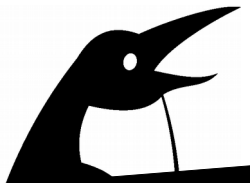
CPU and RAM

- We buy good servers!
  - SuperMicro Rackmount servers.
  - Our newest servers have 512GB RAM.
  - Total of 16 CPU cores per server (Intel Xeon E5-2623 at 3GHz.)
  - We do not configure any swap space.
    - If we run out of RAM at 512GB, we have problems that swap won't fix.



Disk

- Good disk performance is ***the key*** to making everything work.
- We still use traditional spinning hard drives.
- Linux Software RAID-10 FTW!!!
- We keep two copies of each chunk.
- Given n drives, capacity = n/2 * single drive.
- Read performance ~ n * single drive.
- Write performance ~ n/2 * single drive.

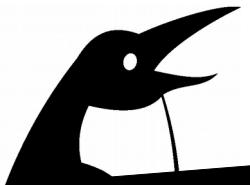


Disk - 2

- Newest-generation servers: 16 x 2TB SATA drives.
- Storage capacity: 16TB
- hdparm -t output:

```
Timing buffered disk reads: 900 MB in  3.00
seconds = 299.94 MB/sec
```

- We have a hardware RAID controller, but use Linux Software RAID for ease of management. Performance is decent.



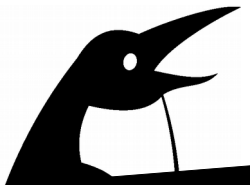
Disk - 3

- RAID controller has a BBU that lets it flush data to disks safely even in the event of a power failure.
- Greatly improves write performance as the RAID controller can tell “white lies” to the OS.

Flush Data. Hahaha...
(Don’t give up your day job, Dianne.)

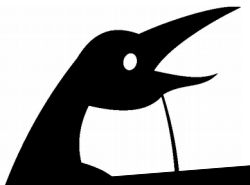
Database Failover

- The central PostgreSQL database is a single point of failure.
- We use PostgreSQL's built-in streaming replication to run a second database server in hot-standby mode.
- Cron jobs automatically check the health of the primary and fail over to the secondary if necessary.



Database Failover - 2

- We could balance read-transactions over the master and the hot-standby database.
- But we don’t (yet) need to do this.
    - When we were using two data centres and had the primary / secondary DB machines in different locations, it helped a lot.
    - But this was because of the high-latency link between the sites, not because we were stressing the primary DB server.

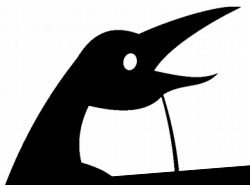


Cheating for Fun and Profit

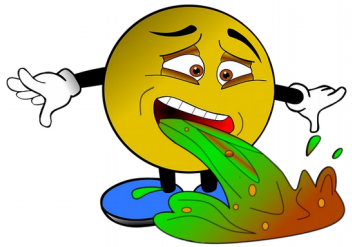
- Some operations are inherently slow.
    - Updating Bayes statistics
    - Remailing messages out of quarantine
    - Indexing archived mail
- We can’t make them fast. But we can make them ***seem*** fast.

Cheating for Fun and Profit - 2

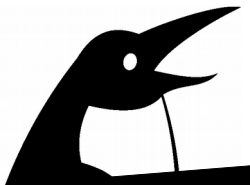
- When a user initiates a slow operation in the Web interface, or when something really slow has to happen in the delivery path...
- ... we just make a note to get around to it later.
- A background task periodically runs the work queue and does all the actual work.
- Note that you have to monitor this to make sure the system can keep up on average.



Other Systems Suck



- Hosted CanIt connects to thousands of back-end SMTP, IMAP, POP3, and LDAP servers.
- Not all of them are as (*ahem*) well-managed and reliable as Hosted CanIt.
- We used the ***circuit-breaker*** design pattern to avoid getting bogged down by unresponsive servers.
    - If a back-end server is unresponsive after X tries, we mark it down and fail fast for the next Y minutes, after which we test again.

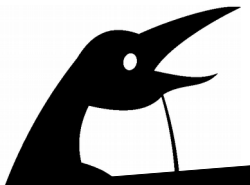


RTFM

- In order to reduce our support load, CanIt performs many diagnostics on the back-end servers and reports them in a helpful format to system administrators.
- This has not reduced our support load.
- Nobody reads. Nobody cares. I am bitter.

Management

- Our Hosted CanIt cluster consists of 17 servers (mostly real; a few VMs.)
- They are all identical Debian 8 machines.
    - Having identical boxes greatly eases administration!
- We use **ClusterSSH** to open terminals on all of them if we need to run the same command on all.

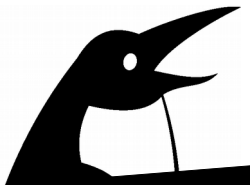


Monitoring

- Xymon monitors the health of the cluster
  - Are SSH/SMTP/HTTPS listening?
  - Are queue sizes, free disk space, load OK?
  - Are various CanIt scanning statistics OK?
  - How about round-trip email time?
  - And web response time?
- Sends text message to cell phone if problems are noticed

DEMO

- Live demo of CanIt if Internet access permits...



Questions?

Links

- Roaring Penguin: https://www.roaringpenguin.com/
- RT: https://bestpractical.com/
- MIMEDefang: http://www.mimedefang.org/
- Milter: https://en.wikipedia.org/wiki/Milter
- Xymon: https://www.xymon.org/
- Munin: http://munin-monitoring.org/
- PostgreSQL: https://www.postgresql.org/
- SW RAID 10: http://tinyurl.com/lx6mx5p
- ClusterSSH: https://github.com/duncs/clusterssh/wiki
- Shameless Self-Promotion: https://dianne.skoll.ca/comedy/