You, Me, PuppetDB, and

Immutable Infrastructure

Code Mesh 2013

deepak giridharagopal

deepak@puppetlabs.com
@grim_radical [github twitter freenode]

Let’s talk about…

Sorry

NO

CHANGE

Immutability is great!

Immutability allows for invariants, which help you reason about correctness

Immutability prevents spooky action at a distance

Immutability fosters modular, composable abstractions

(not a tough sell to functional programmers)

That's great for develoment, but how about operations?

Immutability for infrastructure?

Because operations is in the same boat as development

Everyone who's got their app running on a fleet of servers has experienced spooky action at a distance

Known, good state is critical for reliable upgrades

A lack of predictability in your systems ruins automation and abstraction

Chance

Poor invariants!

Lose $172,222

per second


VNTV
Vantiv Inc
3.9695
-2.97 -42.80%
WCC
Wesco International
54.7200
-0.27 -0.49%
MWW
Monster Worldwide Inc
6.2500
-0.86 -12.10%
LAZ
Lazard Ltd
26.7150
0.01 0.06%
GCAP
Gain Capital Holdings Inc
4.7800
0.09 1.92%
KCG
Knight Capital Group Inc
3.9695
-2.97 -42.80%
42,178,624
VOLUME
3.27
OPEN
3.99
HIGH
3.21
LOW
3.96
5
3.97
41
Knight
EV
Eaton Vance
12.5900
0.47 3.88%
AB
Alliance Bernstein Holding L P
26.5950
0.01 0.02%
PRI
Primerica Inc
20.0700
0.09 0.45%
OLN
Olin
JCI
Johnson Controls
LCC
US Airways Group Inc.
MA
Mastercard
ALK
Alaska Air Group
DPZ
Domino's Pizza Inc
WGO
Winnebago Inds
WBC
Wabco Hold
CO SECURITIES

The problem is that:

Systems are inherently mutable!

But ideally:

Systems should behave as though they weren't!

Computer systems are in many ways open systems, providing the keys to the vault if one is so inclined to grab them. But in order to foster an air of immutability in our own systems, it's of utmost importance to create a façade of immutability. Immutability requires that we layer over and abstract the parts of our system that provide unrestrained mutability.

-- The Joy of Clojure


puppet
labs®

Describe how you'd like your systems to look, and Puppet does all the hard work for you!

```
file { “/etc/issue”:
  content => “Got an issue? Here’s a tissue!”,
}
```

```
file { “/etc/motd”:
  content => template(“Welcome to $hostname!”),
}
```

```
file { "/etc/sudoers":
  owner  => root,
  group  => root,
  mode   => 440,
  source => "puppet:///modules/sudo/sudoers"
}
```

```
package { 'ntp':
  ensure => installed,
}

service { 'ntpd':
  ensure     => running,
  enable     => true,
  subscribe => File['/etc/ntp.conf'],
}

file { '/etc/ntp.conf':
  ensure  => file,
  require => Package['ntp'],
  source  => "puppet:///modules/ntp/ntp.conf",
}
```

```
class ntp {

    package { 'ntp':
      ensure => installed,
    }

    service { 'ntpd':
      ensure     => running,
      enable     => true,
      subscribe => File['/etc/ntp.conf'],
    }

    file { '/etc/ntp.conf':
      ensure  => file,
      require => Package['ntp'],
      source  => "puppet:///modules/ntp/ntp.conf",
    }

}
```

```
node “webserver.mydomain.com” {
   include ntp
}

node “appserver.mydomain.com” {
   include ntp
}

node “database.mydomain.com” {
   include ntp
}
```

```
class ssh {

    @@sshkey { $hostname:
      type => dsa,
      key => $sshdsakey
    }

    Sshkey <<|  |>>

}
```

File “/tmp/foo/bar”

User “deepak”

Dir “/tmp/foo”

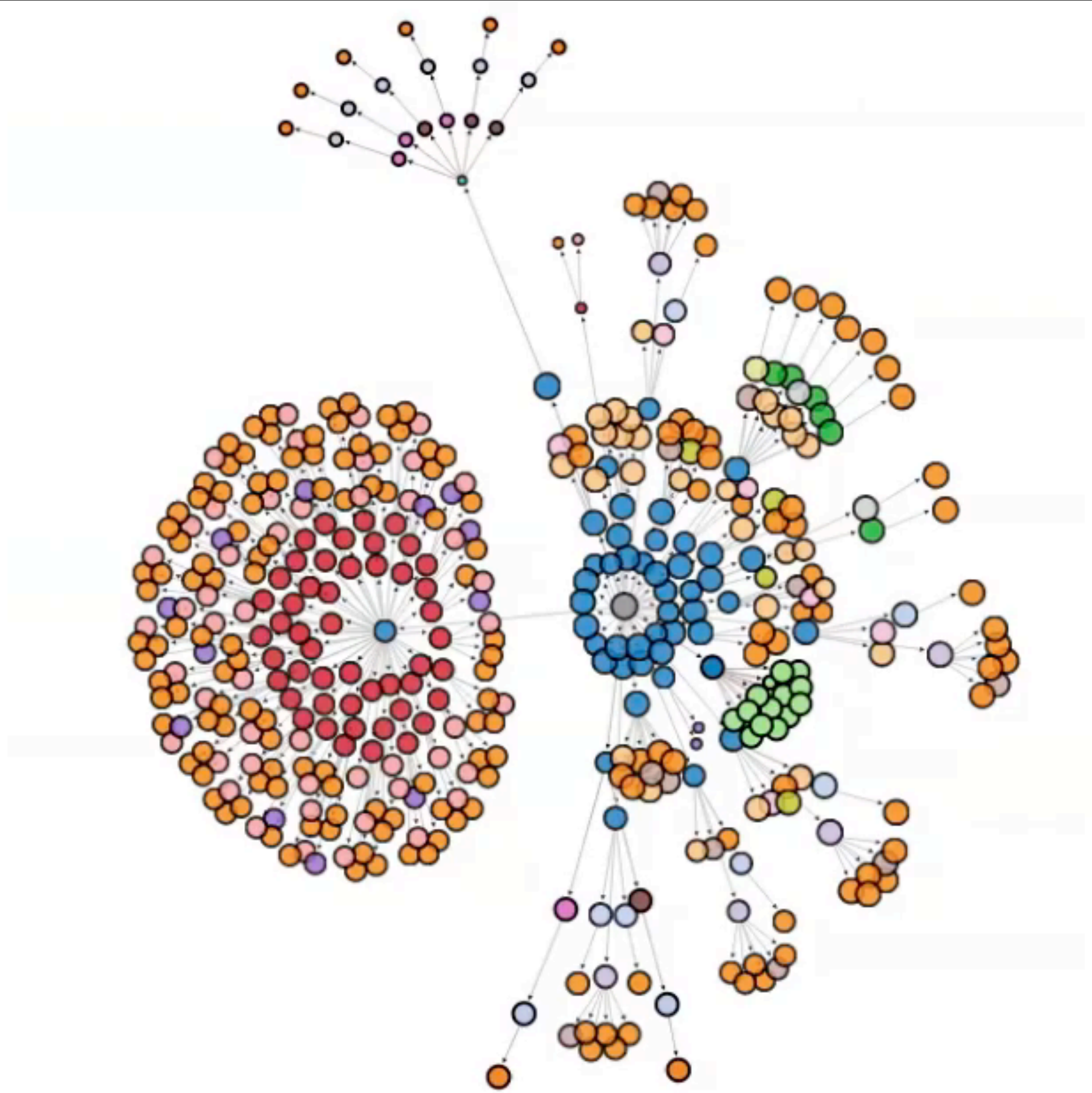
Dir “/tmp”

Dir “/tmp”

User “deepak”

Dir “/tmp/foo”

File “/tmp/foo/bar”



Idempotent, and only does what's necessary

Compensates for the inherent mutability of systems

Combats spooky action at a distance with automatic repair

Brings predictability to your systems

A foundation of predictability and reliability lets you perform higher-level operations on your infrastructure

VOLUME

every resource
every parameter
every relationship
every fact
for every node
updated all the time

Users leverage this data to do higher-order things with their infrastructure

key distribution
monitoring
clustered services
master/slave replication
load balancers
shared filesystems
firewall rules
...

Infrastructure as code

Infrastructure as data

User demand:

Store as much data as we can!
Much better queryability!

Oh yeah, but:

Don’t slow down the system!
Don’t compromise reliability!

We can rebuild it, we have the technology!

Speed is important

Parsing, validating, and manipulating incoming data is computationally expensive

Speed is important

The slower central storage is, the less agile sysadmins can be. That can cost money and uptime!

Reliability is important

If it's a critical part of managing infrastructure, it's got to be solid!

Deployment is important

Should be easy to package, install, configure, use, monitor, and upgrade.

Wait, isn't Puppet written in Ruby?

Lessons learned from writing the rest of our software in Ruby

It's...not speedy. Object creation, method calls, garbage collection, etc. "Magical" APIs amplify the problem.

Lessons learned from writing the rest of our software in Ruby

Can only use one core!
Workarounds compromise performance or simplicity

Lessons learned from writing the rest of our software in Ruby

Mutable state all over the damn place!

Lessons learned from writing the rest of our software in Ruby

Many struggles with the runtime. :(

-- Jeff Gagliardi

PuppetDB

Definitely Better!

Fast, safe storage of catalogs, facts, and events

*like, *way* faster!*

HTTP APIs

for resource, fact, node, report retrieval

plenty of data, just a "curl" away!

Storage &

CORS

Querying

Command
Query
Responsibility
Separation

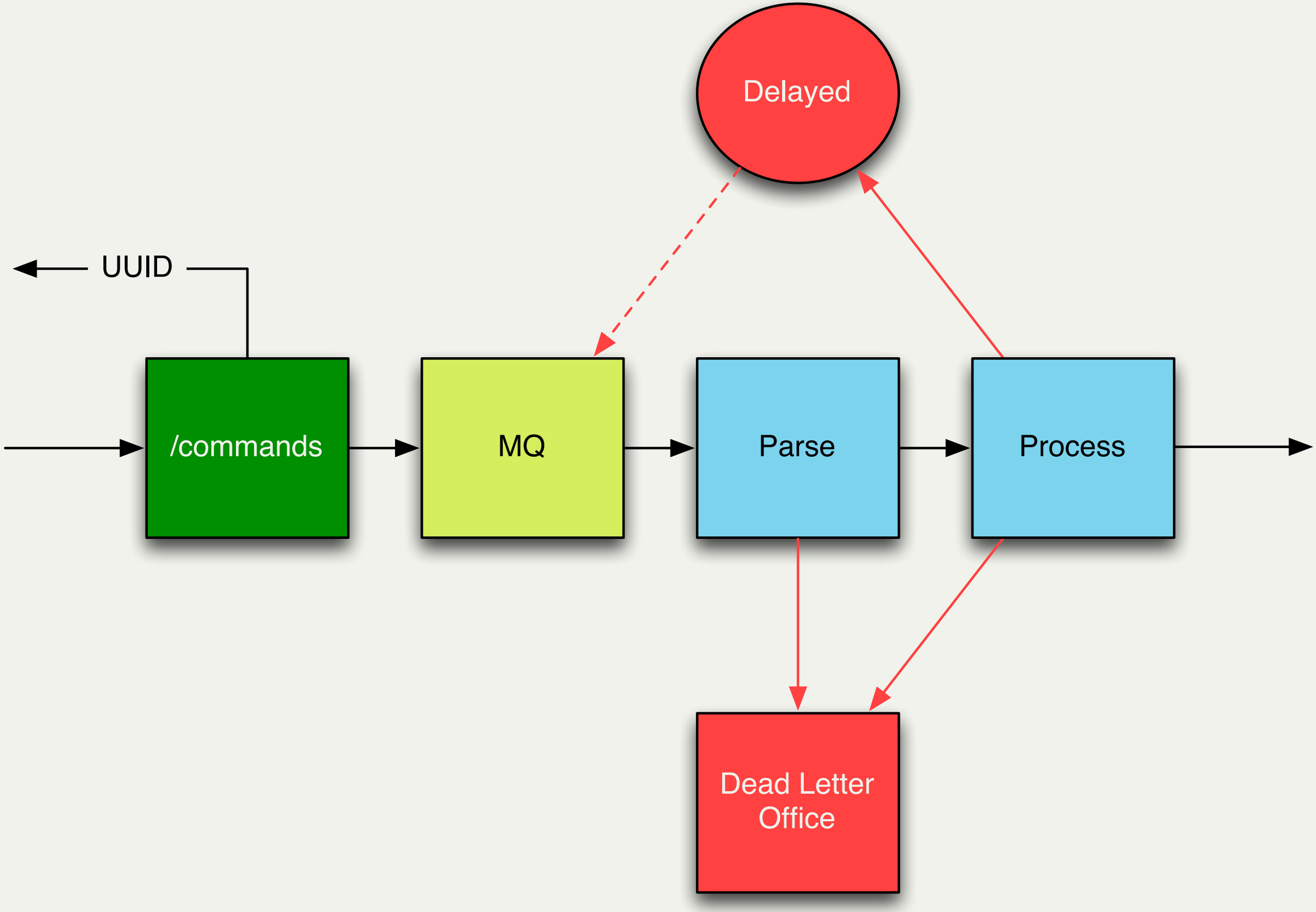
use a different model to update information than the model you use to read information

Writes

CQRS-esque write pipeline

async, parallel, MQ-based, with automatic retry

```
{
 :command "replace catalog"
 :version 2
 :payload {...}
}
```


Delayed
UUID
/commands
MQ
Parse
Process
Dead Letter
Office

```clojure
(defmulti process-command!
  (fn [{:keys [command version] :or {version 1}} _]
    [command version]))

(defmethod process-command! ["replace catalog" 1]
  [command options]
  (replace-catalog* command options))
```

```clojure
(defmulti process-command!
  (fn [{:keys [command version] :or {version 1}} _]
    [command version]))

(defmethod process-command! ["replace catalog" 2]
  [command options]
  (replace-catalog* command options))

(defmethod process-command! ["replace catalog" 1]
  [command options]
  (-> command
      (update-in [:payload] catalog-v1->v2)
      (replace-catalog* options)))
```

Command processors must be retry-aware

*expect failure, because it *will* happen.*

(demo)

Reads

Queries

are expressed in their own “language”

domain specific, AST-based query language

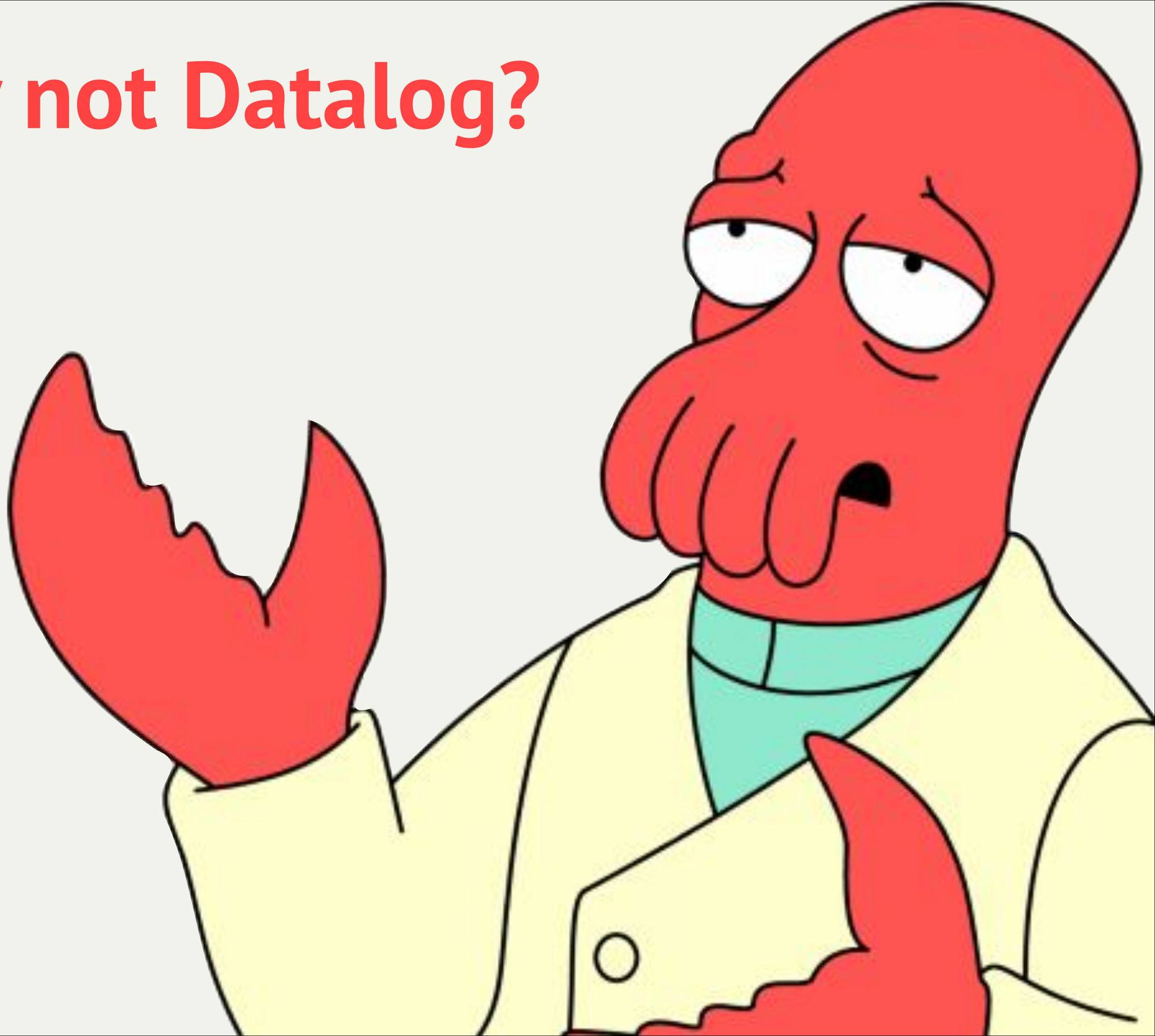
```
["and",
   ["=", "type", "User"],
   ["=", "title", "nick"]]
```

```
["and",
  ["=", ["fact", "operatingsystem"], "Debian"],
  ["<", ["fact", "uptime_seconds"], 10000]]
```

```
["and",
  ["=", "name", "ipaddress"],
  ["in", "certname",
    ["extract", "certname", ["select-resources",
                             ["and",
                               ["=", "type", "Class"],
                               ["=", "title", "Apache"]]]]
```

```
["or",
  ["=", "certname", "foo.com"],
  ["=", "certname", "bar.com"],
  ["=", "certname", "baz.com"]]
```

Why not Datalog?



We use core.match to walk the tree, compiling it to SQL

AST-based API lets users write their own languages and manipulate queries

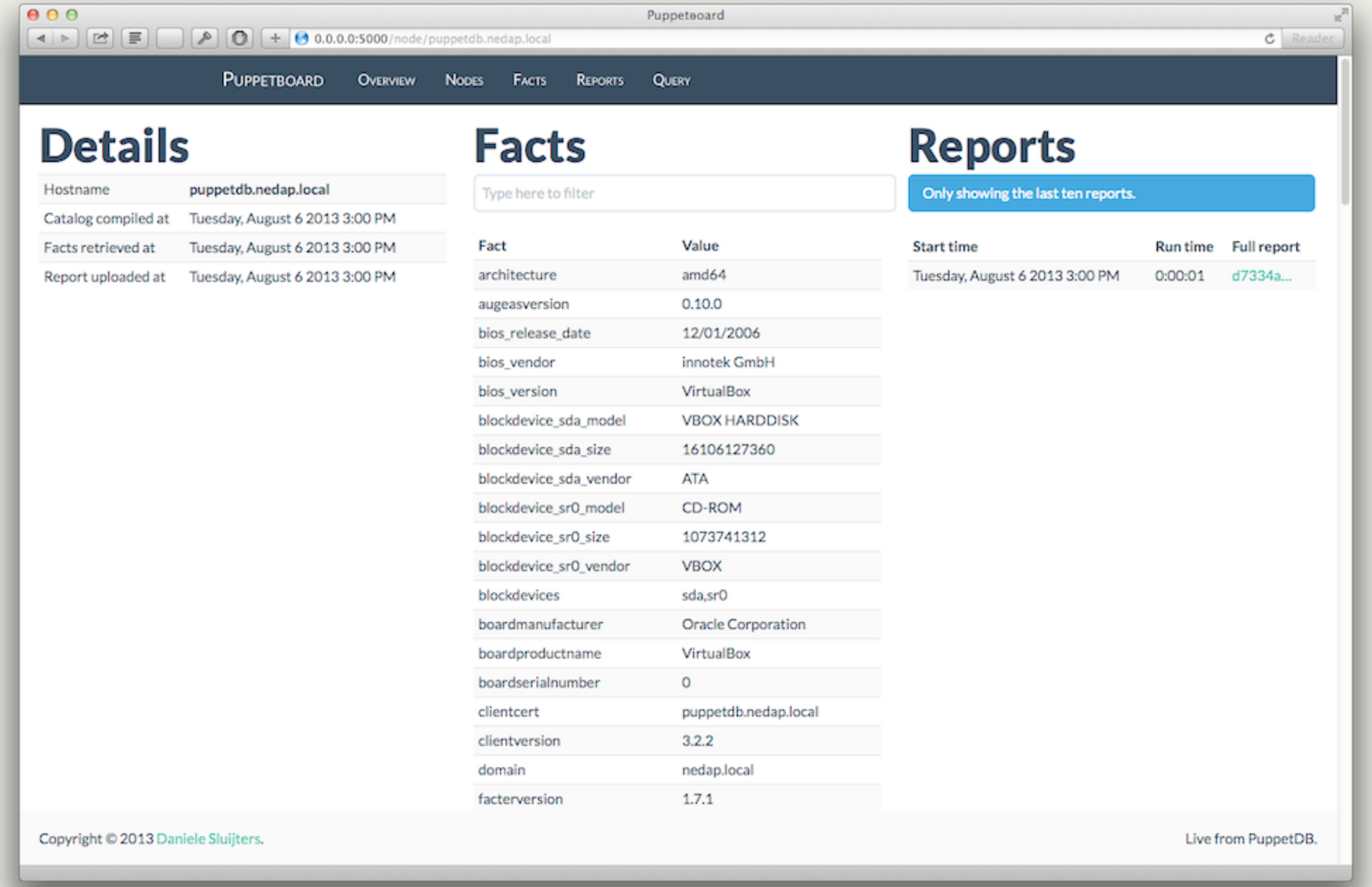
ah, you've got to love open source!

```
(Package[httpd] and country=fr)
or country=us


Package["mysql-server"]
and architecture=amd64
```

Erik Dalén, Spotify

https://github.com/dalen/puppet-puppetdbquery



daenny, Puppetboard

https://github.com/nedap/puppetboard

Chrome File Edit View History Bookmarks Window Help Tue 6:41 PM Wagner

Puppet Enterprise Event In

localhost:9292/#

puppet enterprise console

node requests (0) admin Help

Events Nodes Groups Classes Reports Inventory Search Live Management

Summary / Resources with failures / Exec[10gen-apt-update]

resources: with failures

2 resources that had events and the number of nodes affected:

Resource
Exec[10gen-apt-update]
Package[mongodb-10gen]

resource: Exec[10gen-apt-update]

49 events for this resource and the nodes affected:

3 Failed | 46 Changed

Event	Node	Class	Time
failure	dhcp69.redhat.backline.puppet...	Mongodb	2013-08-10 23:19:20
failure	dhcp71.redhat.backline.puppet...	Mongodb	2013-08-10 23:19:58
failure	dhcp80.redhat.backline.puppet...	Mongodb	2013-08-10 23:20:06
success	dhcp62.redhat.backline.puppet...	Mongodb	2013-08-10 23:55:03
success	dhcp55.redhat.backline.puppet...	Mongodb	2013-08-10 23:55:07
success	dhcp74.debian.backline.puppe...	Mongodb	2013-08-10 23:17:57
success	dhcp73.debian.backline.puppe...	Mongodb	2013-08-10 23:17:59
success	dhcp65.debian.backline.puppe...	Mongodb	2013-08-10 23:17:55
success	dhcp89.debian.backline.puppe...	Mongodb	2013-08-10 23:18:55
success	dhcp64.debian.backline.puppe...	Mongodb	2013-08-10 23:18:28
success	dhcp86.debian.backline.puppe...	Mongodb	2013-08-10 23:19:07
success	dhcp51.debian.backline.puppe...	Mongodb	2013-08-10 23:19:12
success	dhcp57.debian.backline.puppe...	Mongodb	2013-08-10 23:18:37
success	dhcp98.debian.backline.puppe...	Mongodb	2013-08-10 23:18:45
success	dhcp52.debian.backline.puppe...	Mongodb	2013-08-10 23:19:43
success	dhcp67.debian.backline.puppe...	Mongodb	2013-08-10 23:19:54

Puppet Enterprise, Event Inspector

https://puppetlabs.com

Foreman Integration (CERN)

https://github.com/cernops/puppetdb_foreman

Web UI

https://github.com/dima-exe/puppetdb-db

Web UI

https://github.com/gbougeard/puppetdb-frontend

Ruby

https://github.com/dalen/puppet-puppetdbquery

Ruby (DataMapper)

https://github.com/dalen/dm-puppetdb-adapter

Ruby

https://github.com/ripienaar/ruby-puppetdb

Python

https://github.com/nedap/pypuppetdb

Python

https://github.com/arcus-io/puppetdb-python

Python

https://github.com/JHaals/puppetdb-grep

Java

https://github.com/thallgren/puppetdb-javaclient

Go

https://github.com/nightlyone/puppetquery

Scala

https://github.com/gbougeard/puppetdb-frontend

CoffeeScript

https://gist.github.com/pmuellr/5591686

Node.js

https://github.com/nightfly19/minidb

MCollective

https://github.com/ploubser/mcollective-puppetdb-discovery

Rundeck

https://github.com/sirhopcount/puppetdb-rundeck

Rundeck

https://github.com/martin2110/puppetdb-rundeck

OpenStack

https://github.com/bodepd/puppet-openstack_puppetdb

Vagrant

https://github.com/grahamgilbert/vagrant-puppetmaster

PowerDNS

https://github.com/evenup/evenup-pdns

```clojure
(def query-app
  (app
    [&]
    {:get (fn [{:keys [params] :as request}]
            (perform-query (params "query")))}))

(def resources-app
  (app
   []
   query-app

   [type title &]
   (comp query-app
         (partial http-q/restrict-resource-query-to-type type)
         (partial http-q/restrict-resource-query-to-title title))

   [type &]
   (comp query-app
         (partial http-q/restrict-resource-query-to-type type))))
```

```clojure
(defn restrict-resource-query-to-type
  [type query]
  (conj ["and"
         ["=" "type" type]]
        query))
```

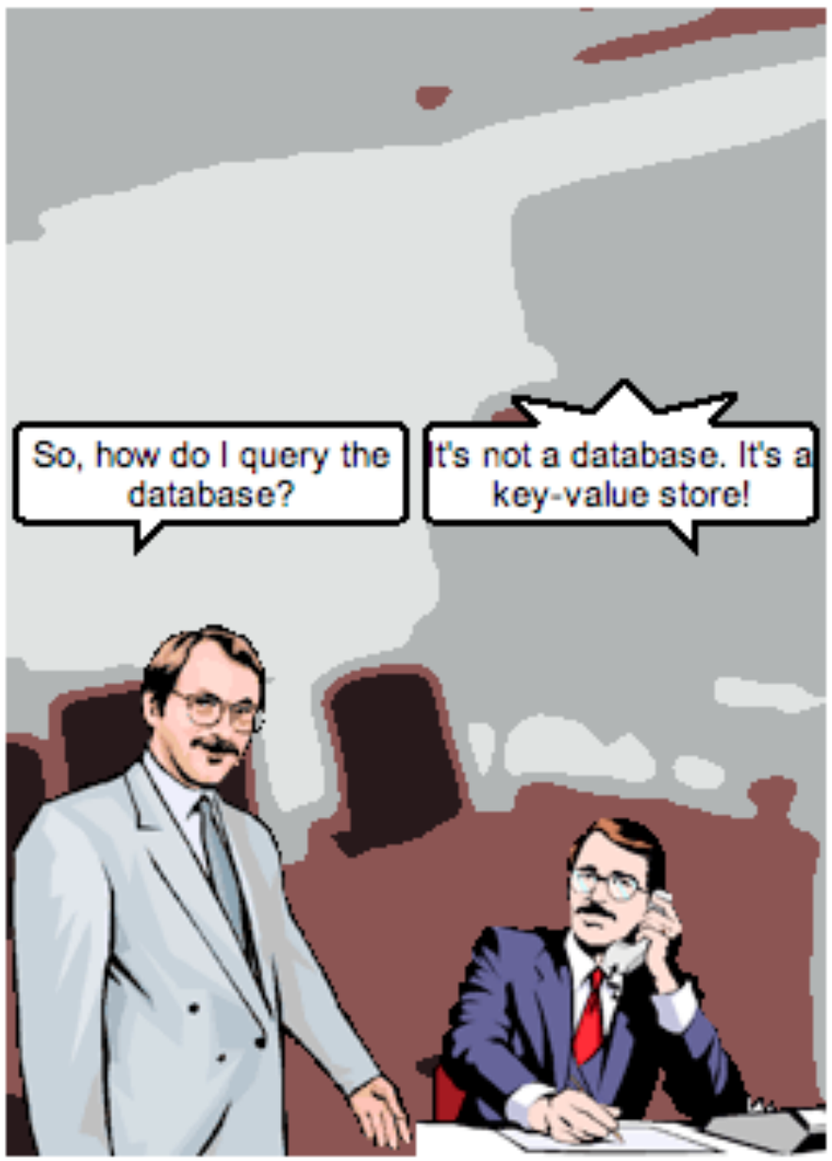
Storage

Relational Database, embedded or PostgreSQL

because they're actually pretty fantastic at ad-hoc queries, aggregation, windowing, etc. while maintaining safety

Fault-tolerance

by @jrecursive



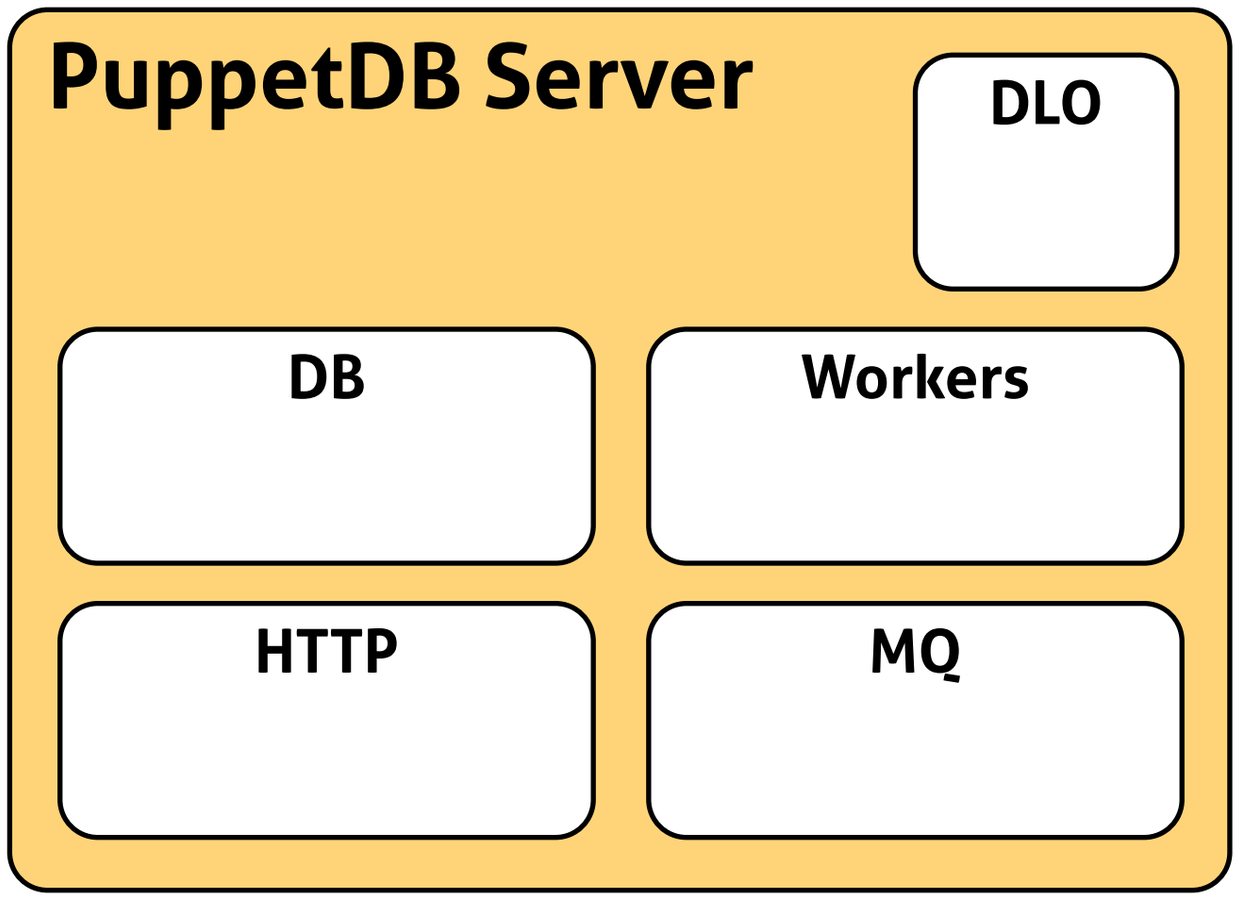
Relational Database, embedded or PostgreSQL

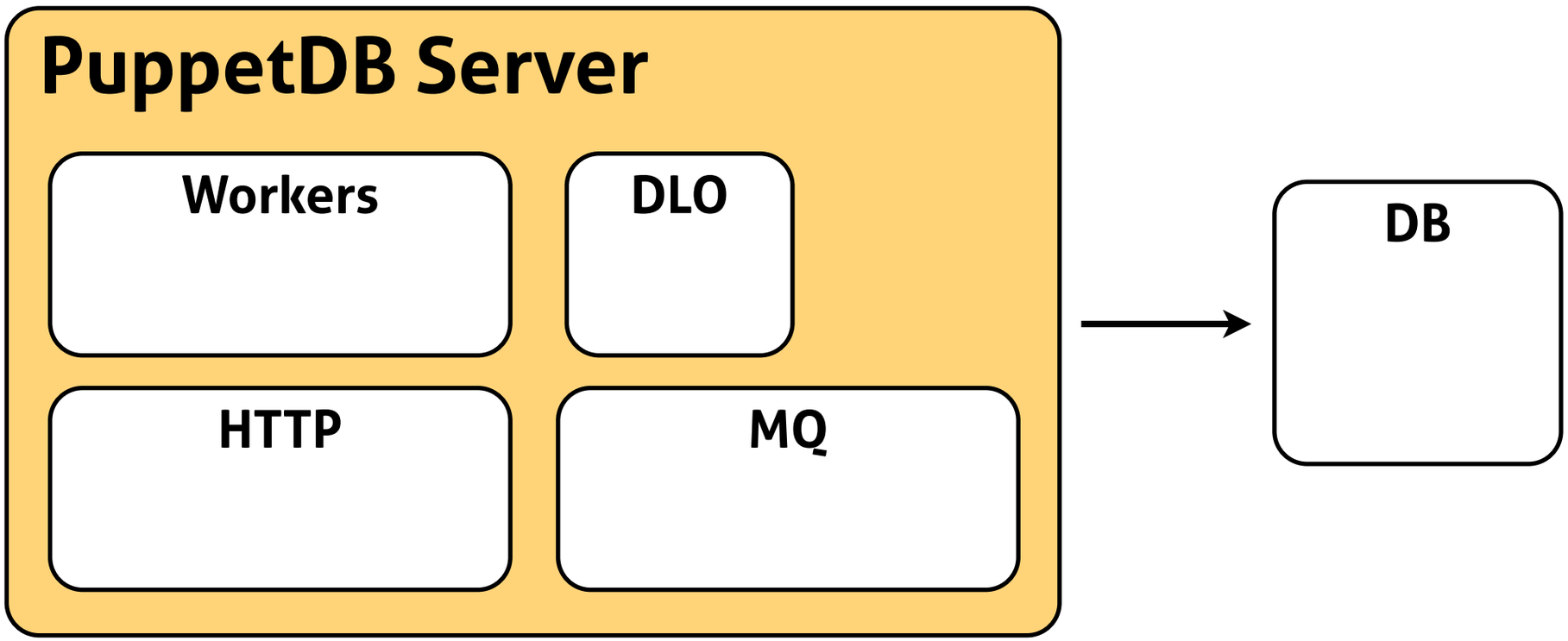
we use arrays, recursive queries, indexing inside complex structures

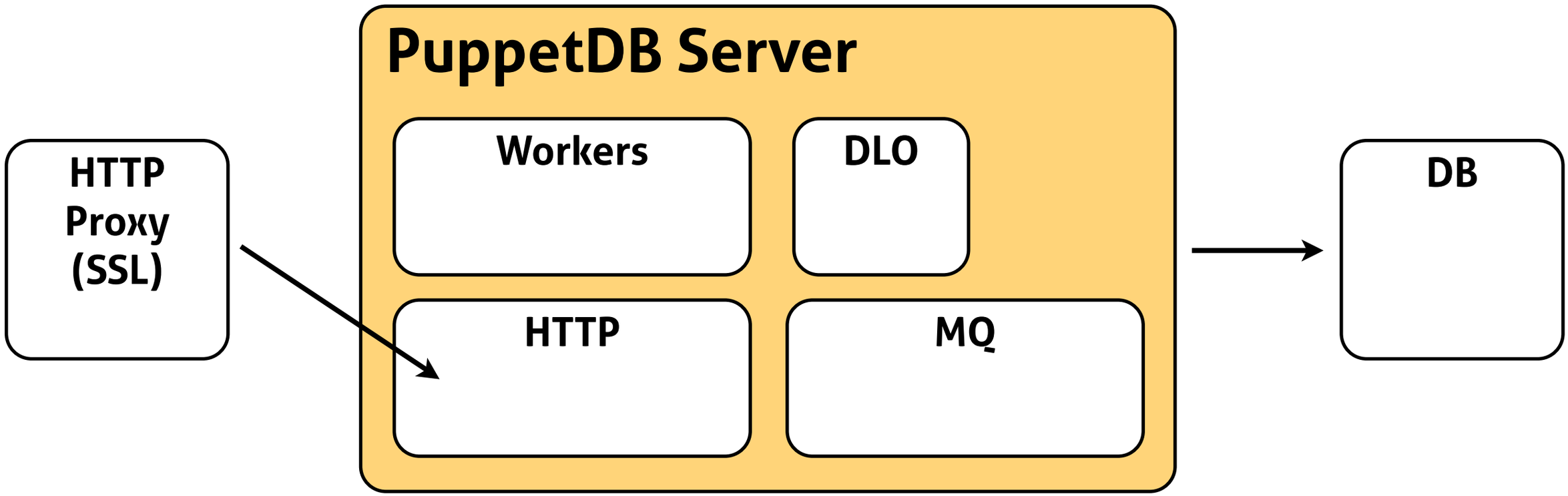
Relational Database, embedded or PostgreSQL

schema is oriented towards the types of queries we encounter

Deployment


PuppetDB Server
DLO
DB
Workers
HTTP
MQ


PuppetDB Server
Workers
DLO
HTTP
MQ
DB


PuppetDB Server
HTTP Proxy (SSL)
Workers
DLO
HTTP
MQ
DB

Results

Thousands of production deployments

Small shops with a dozen hosts, large shops with thousands of hosts, intercontinental deployments...

There is a new deployment of PuppetDB every 15 minutes.

Dean Wilson @unixdaemon 7h

Did some basic performance testing of #**puppetdb** and reduced our compile times from 180ish second to mid 20s. #massivewin

Expand

Greg Mason @greg_mason 25 Sep

according to #**puppetdb**, I'm currently managing 71,777 resources with Puppet.

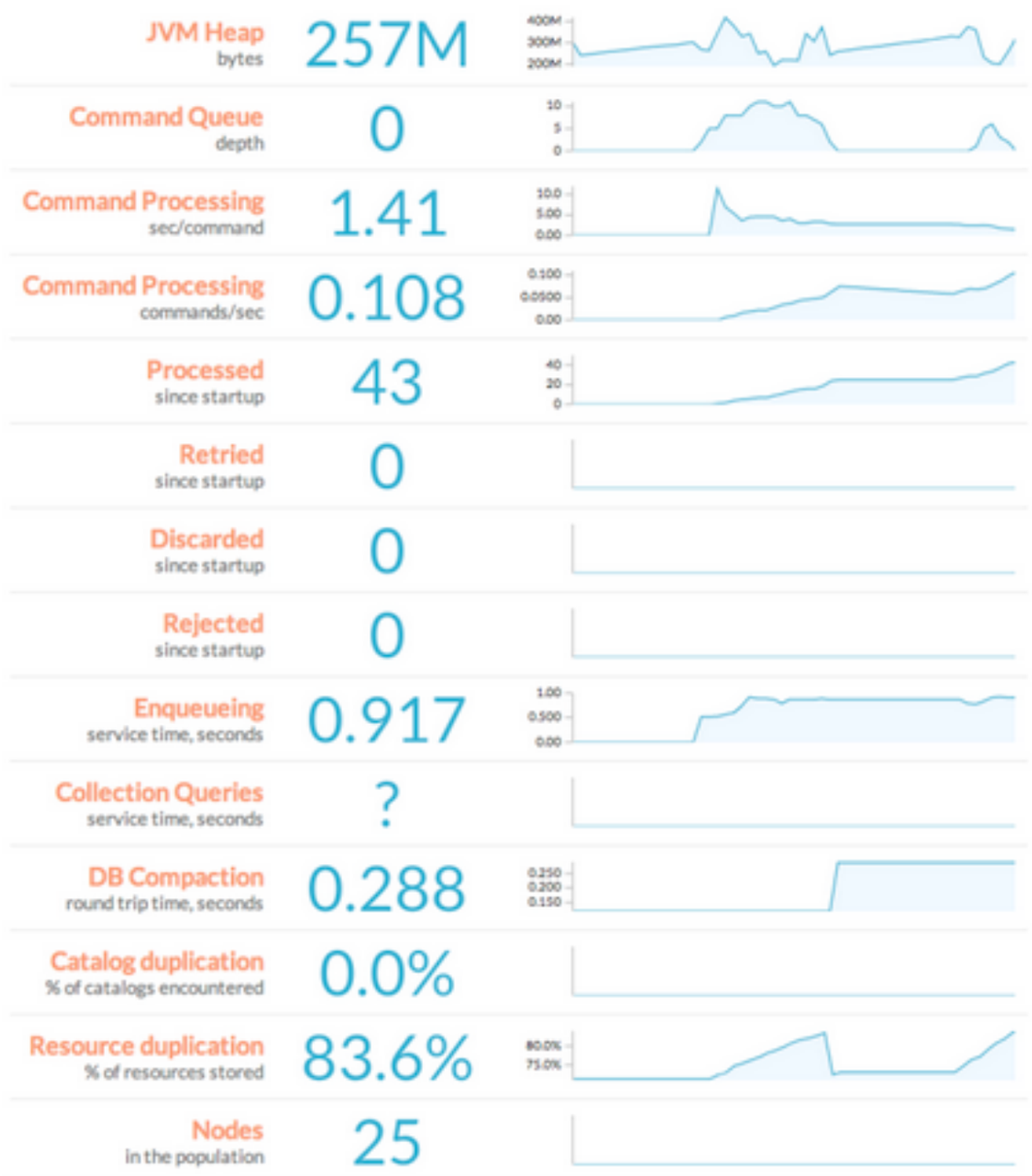
Expand

ripienaar @ripienaar 13 Jun

puppetdb + 25 masterless nodes with embedded database twitpic.com/9vz7y2

Retweeted by D Giridharagopal

Hide photo Reply Retweeted Favorite



Matt Nicholson @sjoeboo 20 Jul

@glarizza @grim_radical two pic.twitter.com/mvJuOiJF

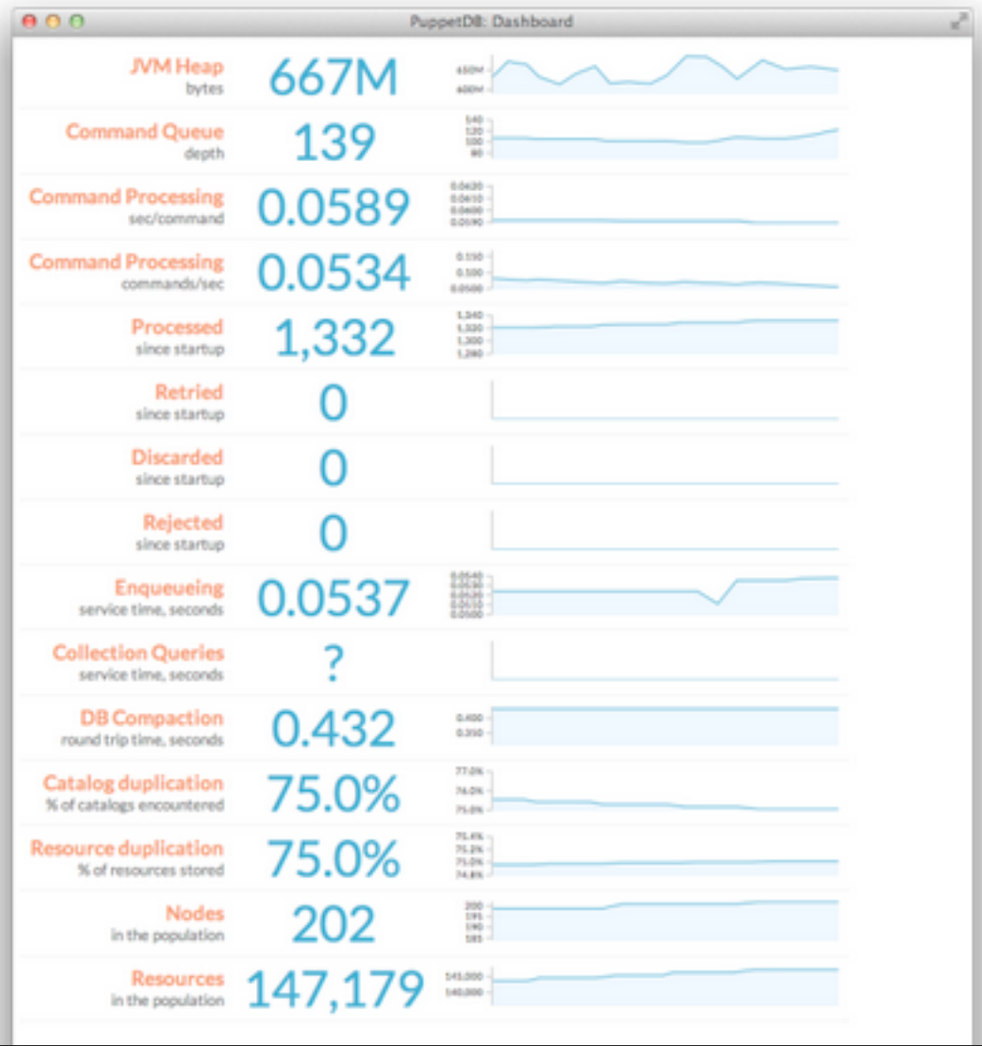
Hide photo

Lars Fronius @LarsFronius 23 May

This #puppetdb dashboard thingy is actually so awesome. Oh, and puppetdb is such a big step in performance! #puppetize twitpic.com/9oi290

Retweeted by D Giridharagopal

Hide photo Reply Retweeted Favorite



Thousands of deployments,
Lots of threads per install,
Zero deadlocks,
Zero bugs involving clojure state

companion Ruby code has
~10x the defect rate

Users dig the uberjar, makes deployment straightforward and less error-prone

Nobody cares that it's Clojure. Users just want stuff to work.

Monitoring and instrumentation is a big deal. Users want easy ways to consume metrics.

Nagios

https://github.com/jasonhancock/nagios-puppetdb

Nagios

https://github.com/favoretti/puppetdb-external-naginator

Munin

https://github.com/vpetersson/munin_puppetdb

Munin

https://github.com/dalen/puppetdb-muninplugins

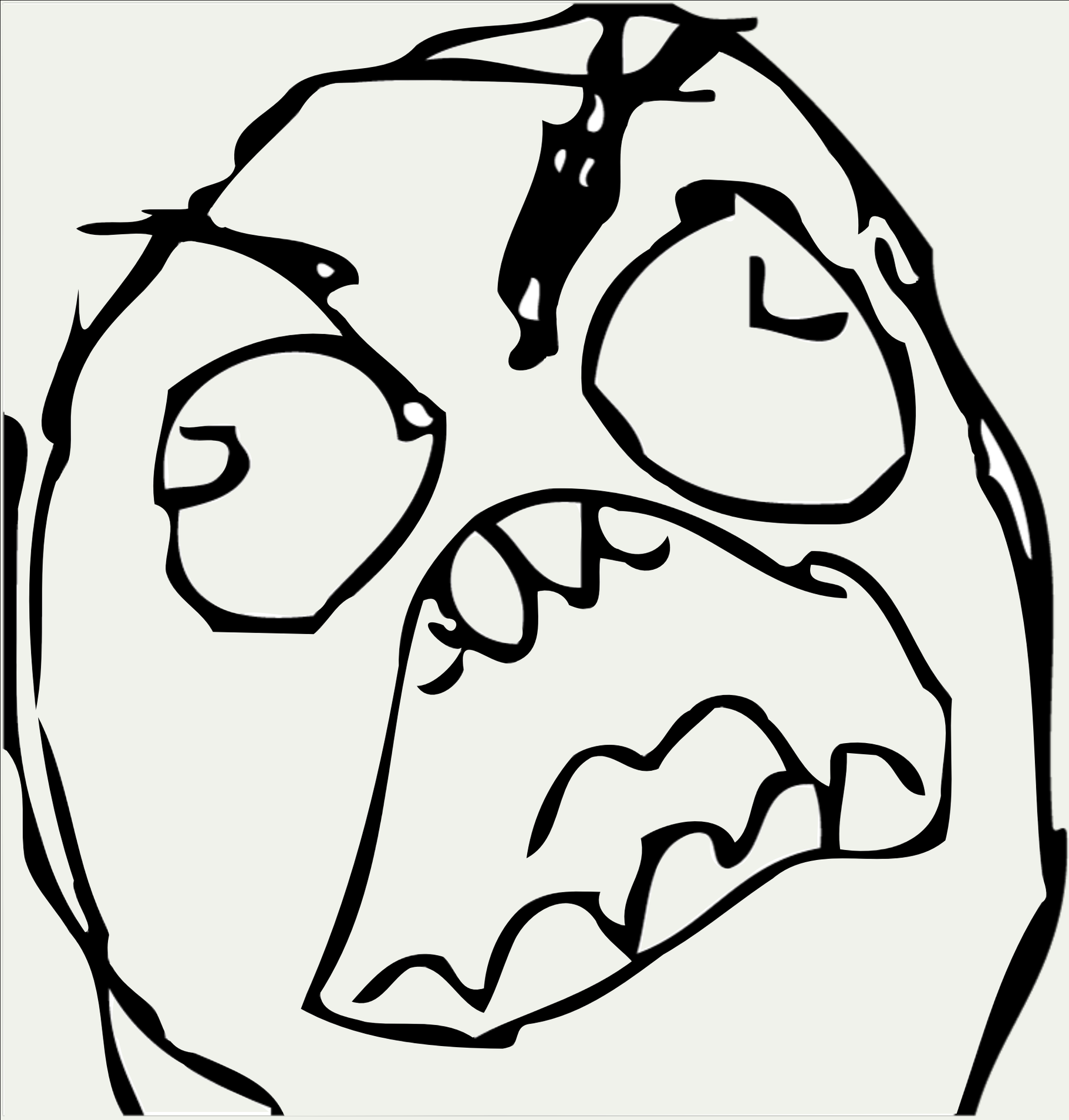
Collectd

https://gist.github.com/mfournier/5615125

Open source

http://github.com/puppetlabs/puppetdb

If you build it, they will come.



FFFFFFF

FFFFFFF

FFFFFF

FFFUU

UUUU

UUUU

UUUU

UUUU-

Think about the entire stack

I don’t care what tools you use, but *please use good ones*

deepak giridharagopal

deepak@puppetlabs.com
@grim_radical [github twitter freenode]

We're hiring!