

# Using Monitoring and Metrics to learn in Development

Patrick Debois  
*Atlassian*

INTERNATIONAL  
SOFTWARE DEVELOPMENT  
CONFERENCE

<http://jedi.be/blog>  
@patrickdebois

gotocn.com

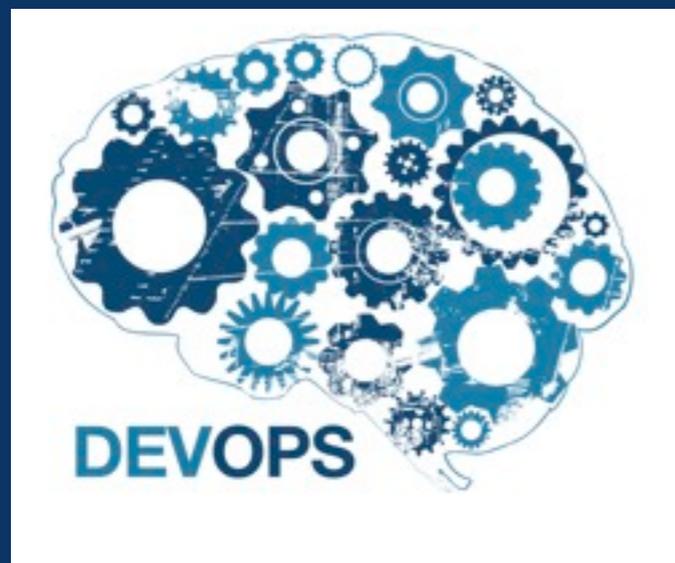
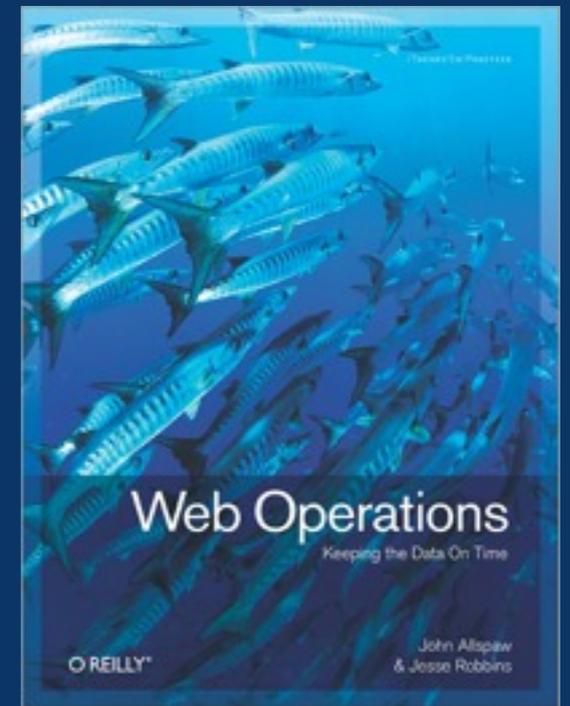


HOSTEDOPS

<https://my.atlassian.com/ondemand/>



<http://www.cutter.com>



<http://devopsdays.org>



Vagrant &  
Veewee



<http://itrevolution.com>

# Your Code



ISBN 0 8203 4255 1

C-4055 CAREER EXAMINATION SERIES

**PASSBOOKS**

Preferred By  
More Test Takers

- Up-to-Date
- Easy to Use
- All Tests
- No Fillers

*This is your  
PASSBOOK® for...*

# Certified Professional Coder (CPC)

*Test Preparation Study Guide  
Questions & Answers*

**NLC**

NATIONAL LEARNING CORPORATION

PLASTIC BOUND -  
LIES FLAT FOR  
STUDY EASIER!

“Test Driven Development”

- CI Coward

# Testing in Controlled Environment

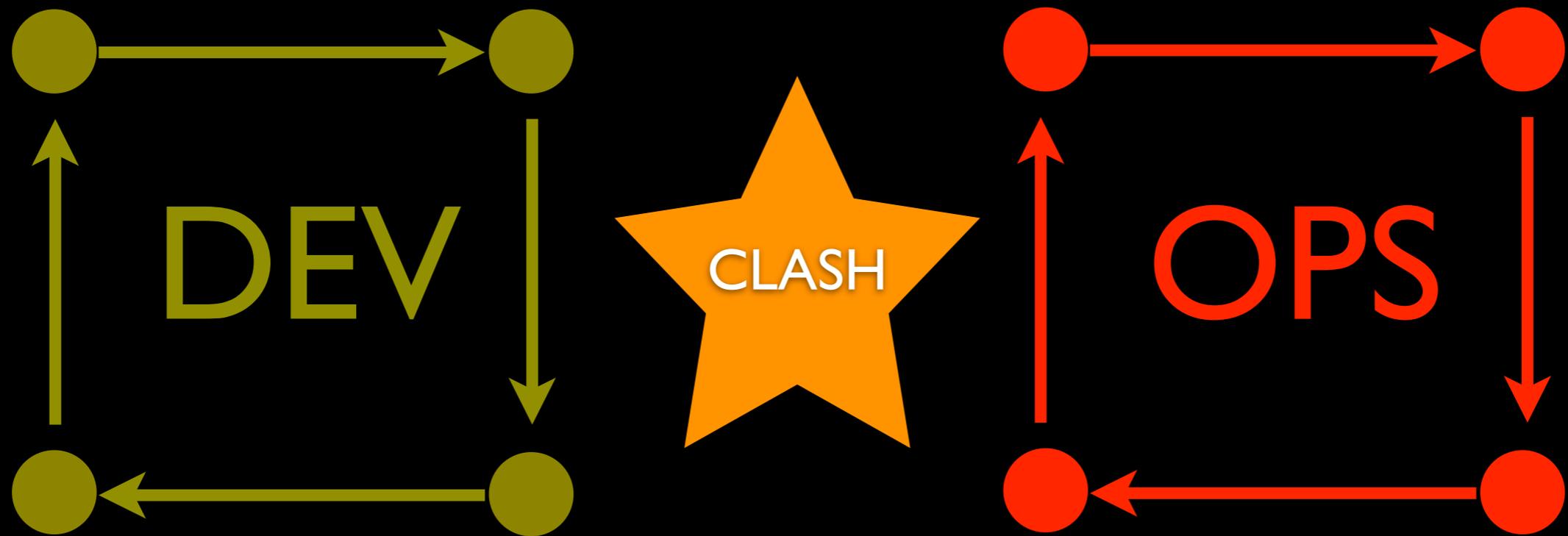


# Your Code in Test



“Until code runs in production ,  
your code is just inventory”  
- Agile Anonymous

# Continuous Delivery





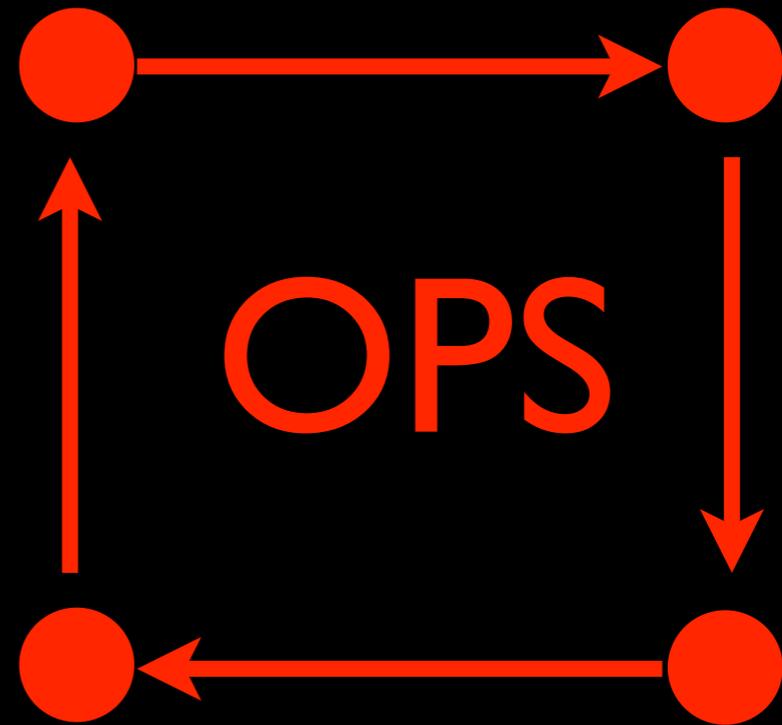
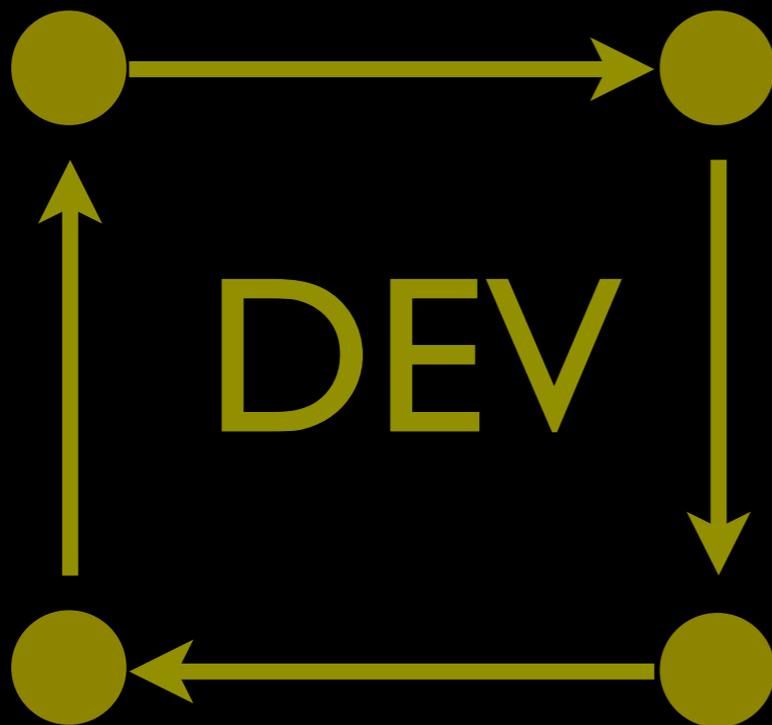
Shit Happens  
in production

“You go to production  
with the code you have,  
not the code you wish you had”  
- Devops Rumsfeld

Dreamer

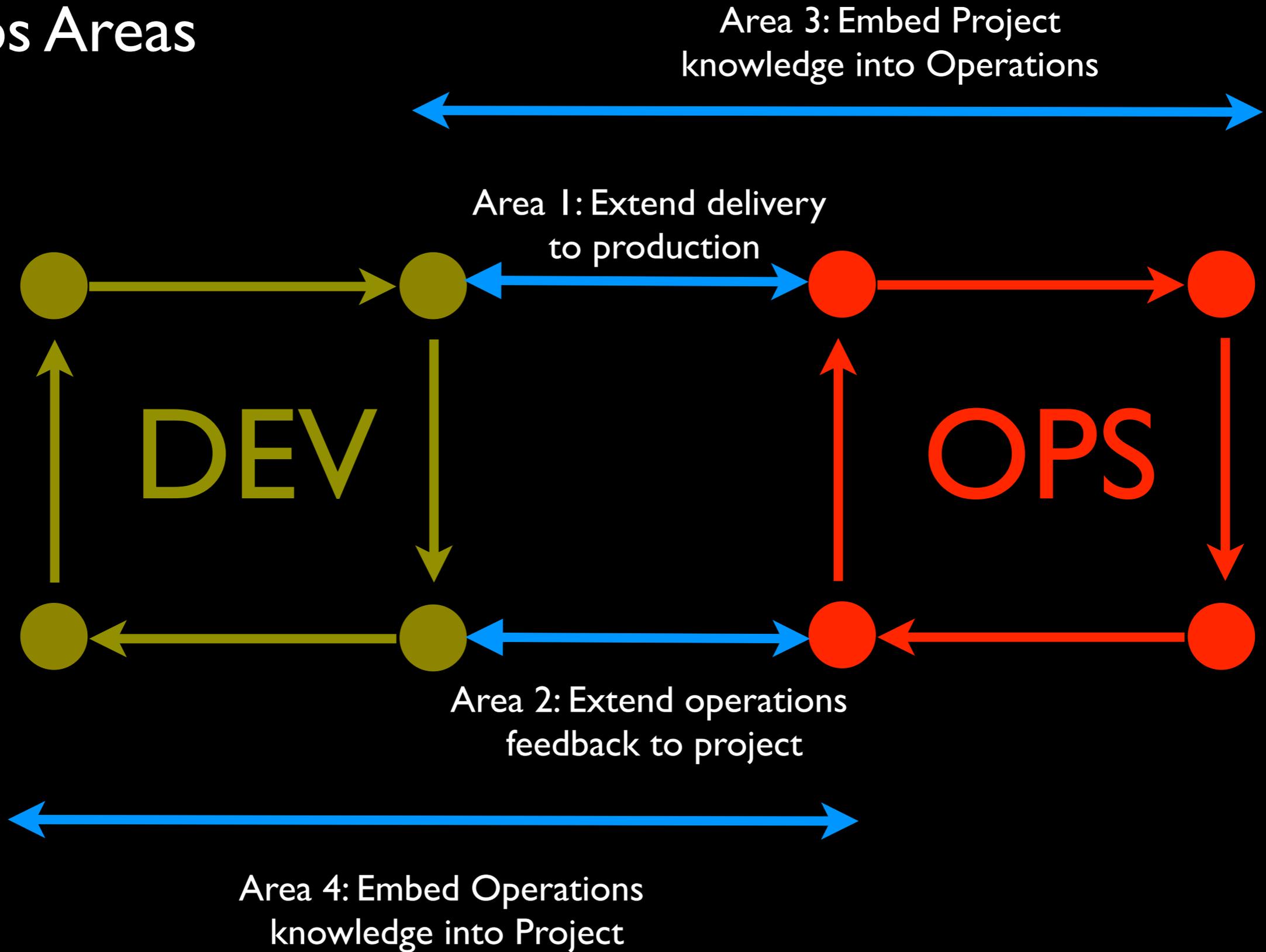
Realist

Critic

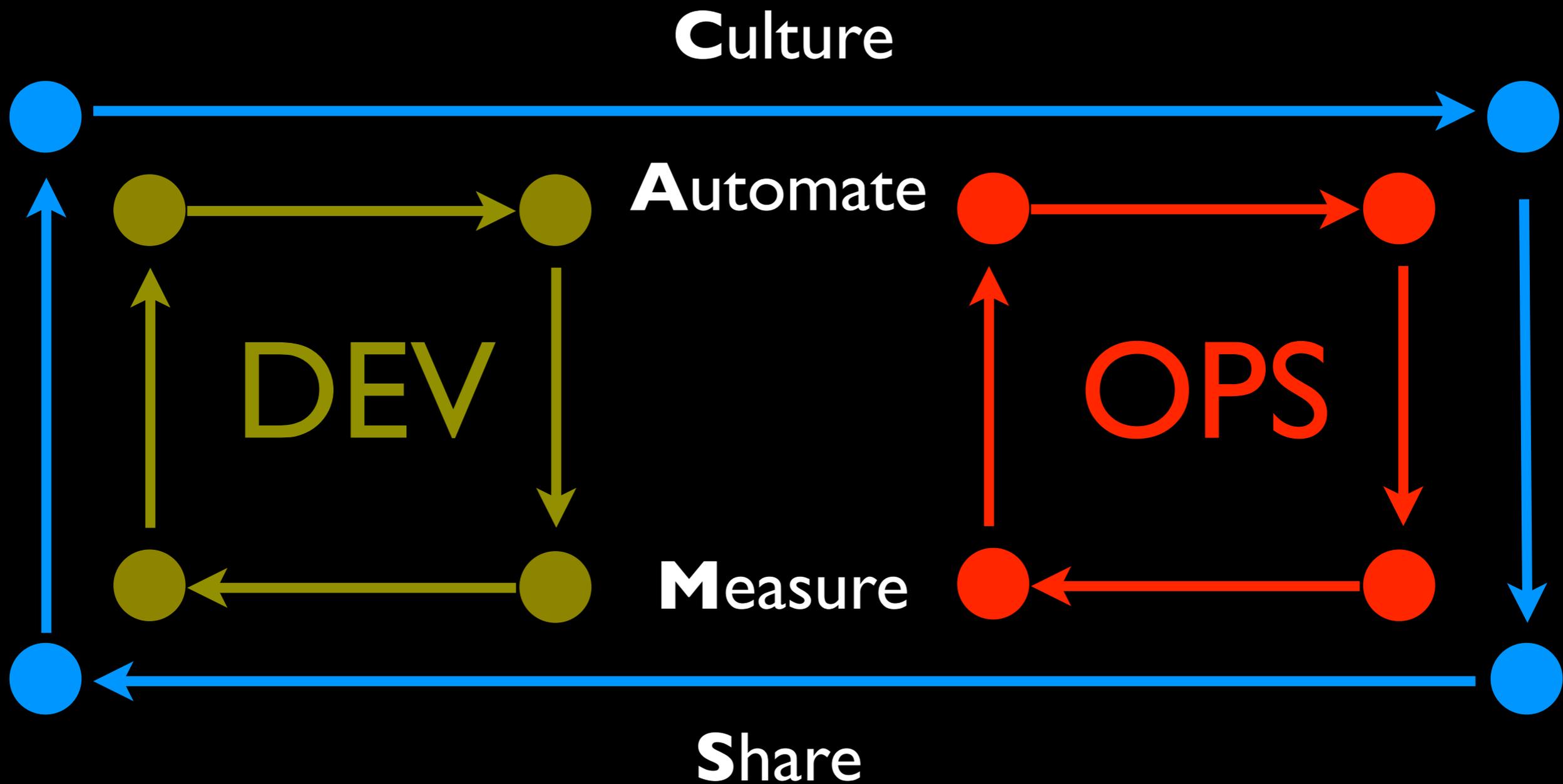


# Three Phases of Creativity (Disney)

# Devops Areas

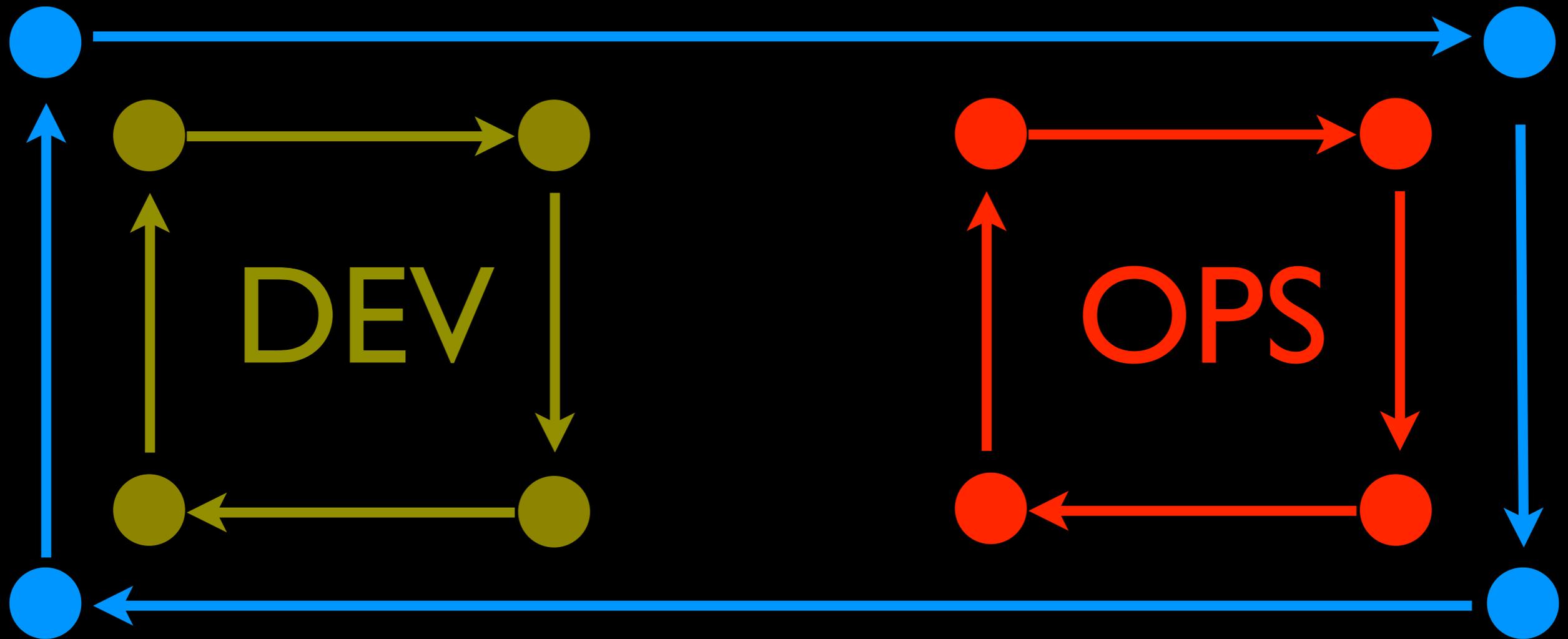


<http://jedi.be/blog/2012/05/12/codifying-devops-area-practices/>



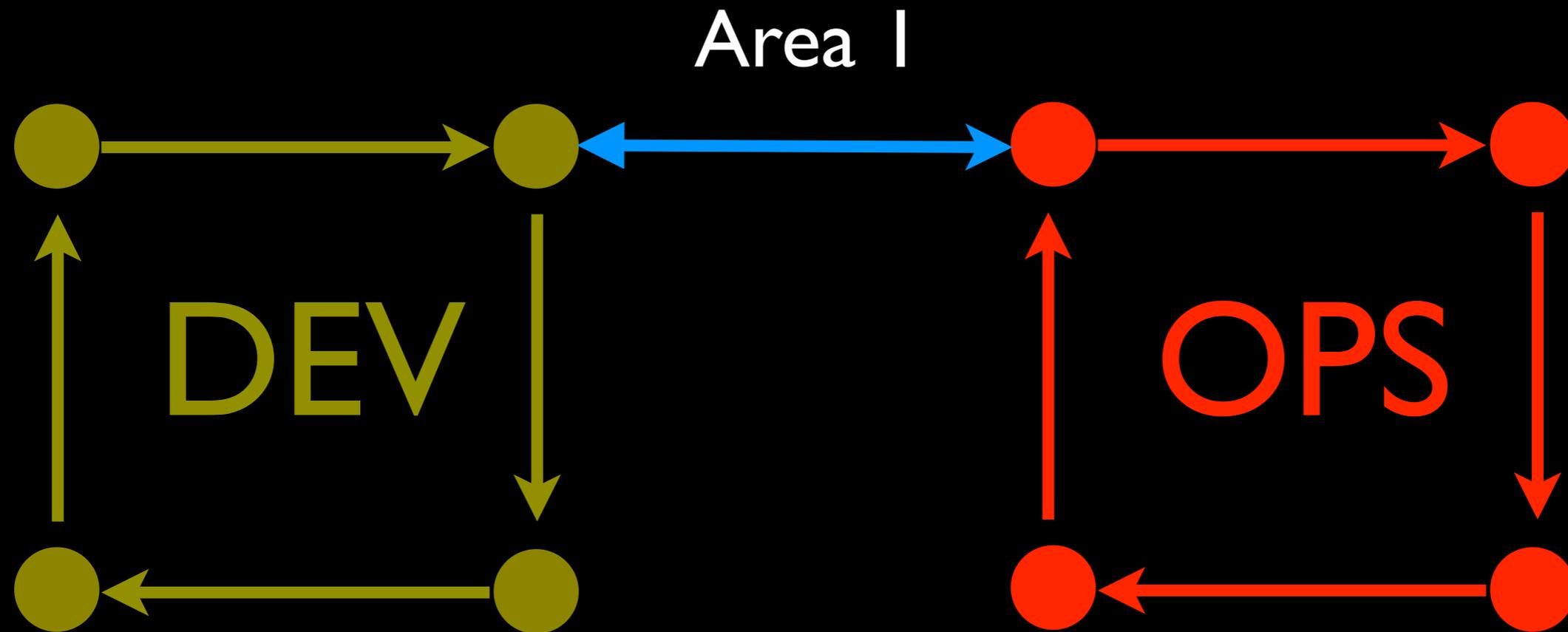
<http://www.opscode.com/blog/2010/07/16/what-devops-means-to-me/>

Smaller and more frequent changes



Faster and better feedback

# Area I: Extend delivery to production “think Jez Humble”



# Continuous Integration Maturity Model

Intro	Novice	Intermediate	Advanced	Insane
<ul style="list-style-type: none"><li>• Using Source Control</li><li>• Nightly Builds</li><li>• Current Issues Tracked &amp; Knowledge base</li></ul>	<ul style="list-style-type: none"><li>• Builds triggered on Commit</li><li>• Automated Deploy to Dev</li><li>• Unit Tests Run on every build</li><li>• Generate Change Log</li><li>• Collect Code Coverage Data</li><li>• Code Integrity Checks</li><li>• Static Analysis Used</li><li>• Run-time Analysis</li><li>• Automated API Documentation generation</li></ul>	<ul style="list-style-type: none"><li>• Automated Deployment to testing environments</li><li>• On demand deployments to controlled environments</li><li>• Per Env. Smoke Test</li><li>• Manual Test Results in CI Server</li><li>• Business visibility, reports</li><li>• Flag a CI build as Release Candidate (promotion)</li><li>• Product Activity Metrics</li><li>• Auto-update Defect Tracking</li><li>• Pre-commit Builds</li><li>• Data roll-up</li><li>• Automatic Cleanup of old Data</li></ul>	<ul style="list-style-type: none"><li>• Automated Func. Testing</li><li>• Multi-Threaded / Scalable build systems</li><li>• Change Reporting / SQA Impacts</li><li>• Defect Trending</li><li>• Identify Problem Code from Metrics</li><li>• Auto-Deploy to Prod</li><li>• Auto-Rollback in Prod</li><li>• Environment Monitoring</li><li>• Alerts based on build metric thresholds</li><li>• Security Scans</li></ul>	<ul style="list-style-type: none"><li>• Continuous Deployment to Prod</li></ul>

# Infrastructure Code Repository

DEV

TEST

PROD

WEB

WEB

WEB

APP

APP

APP

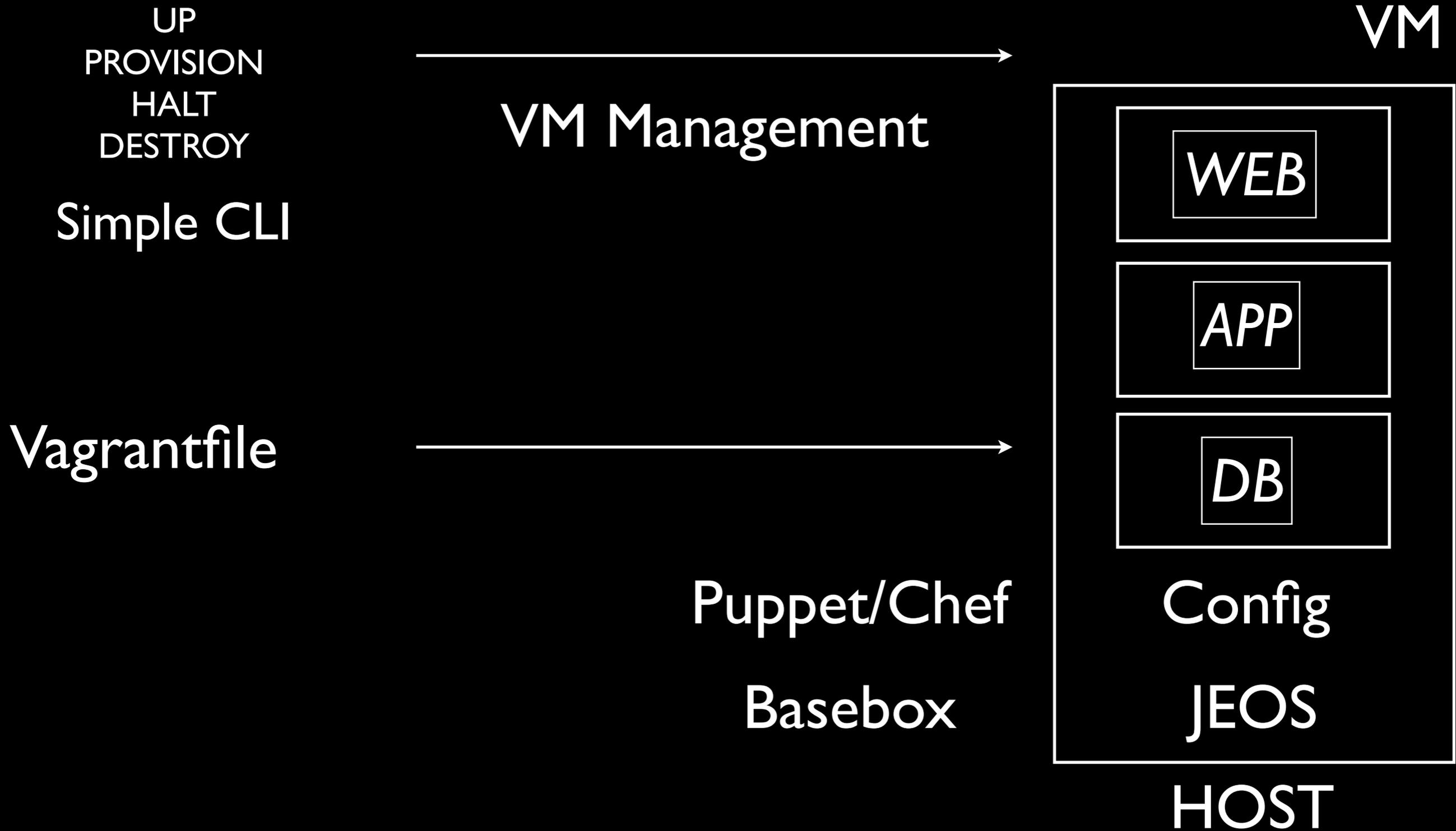
DB

DB

DB

**Reuse** “Infra code” across Environments

# Reuse “workflow” across Environments



<http://vagrantup.com/>

# Integrate with **Continuous Integration**

DEV

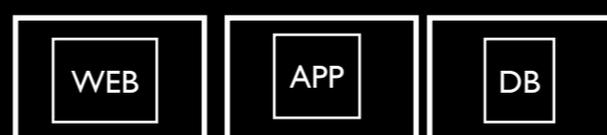
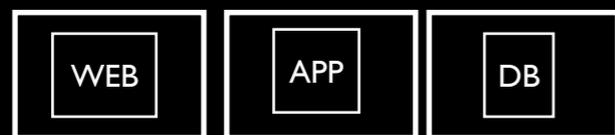
TEST

PROD

APP



INFRA



Infrastructure  
Code Repository

Application  
Code Repository

# Continuous Integration to Continuous Delivery

**Faster/Delivery**



DEV

TEST

PROD

APP



INFRA



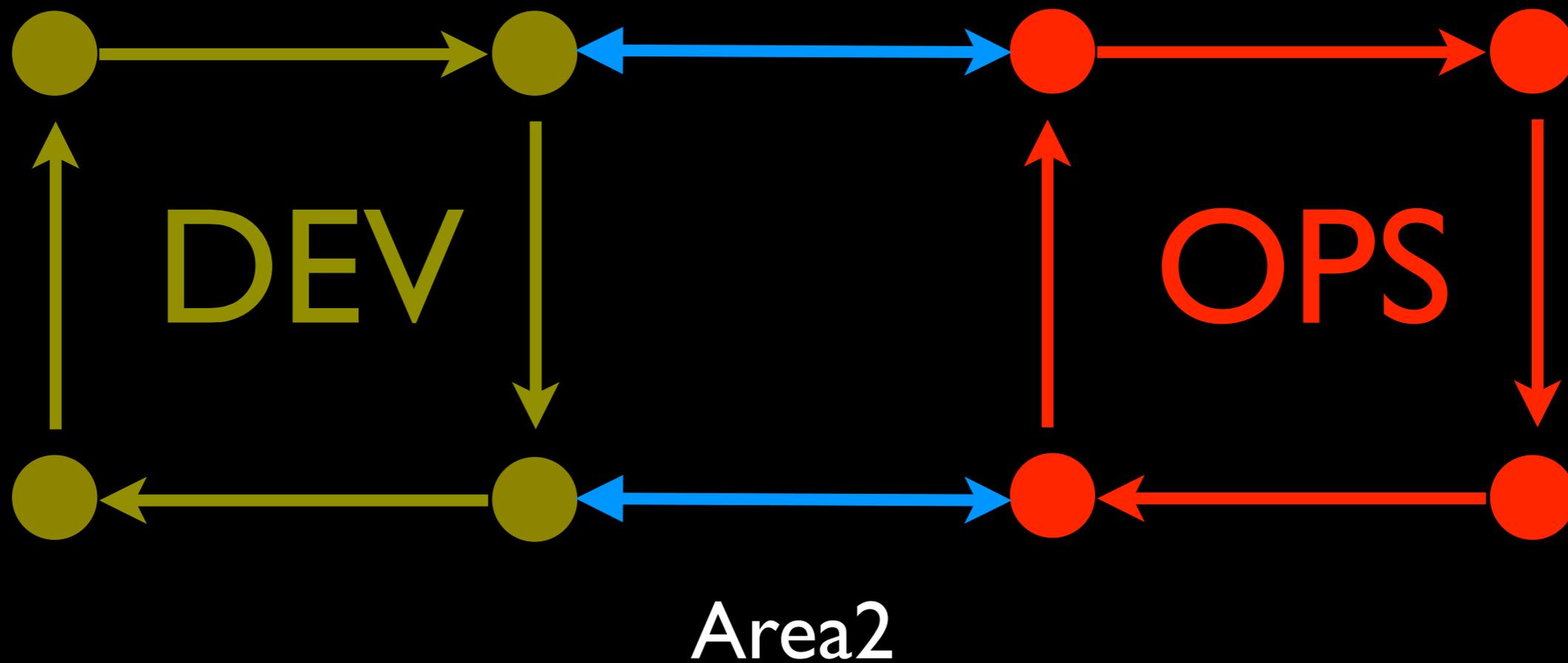
**Faster/Feedback**



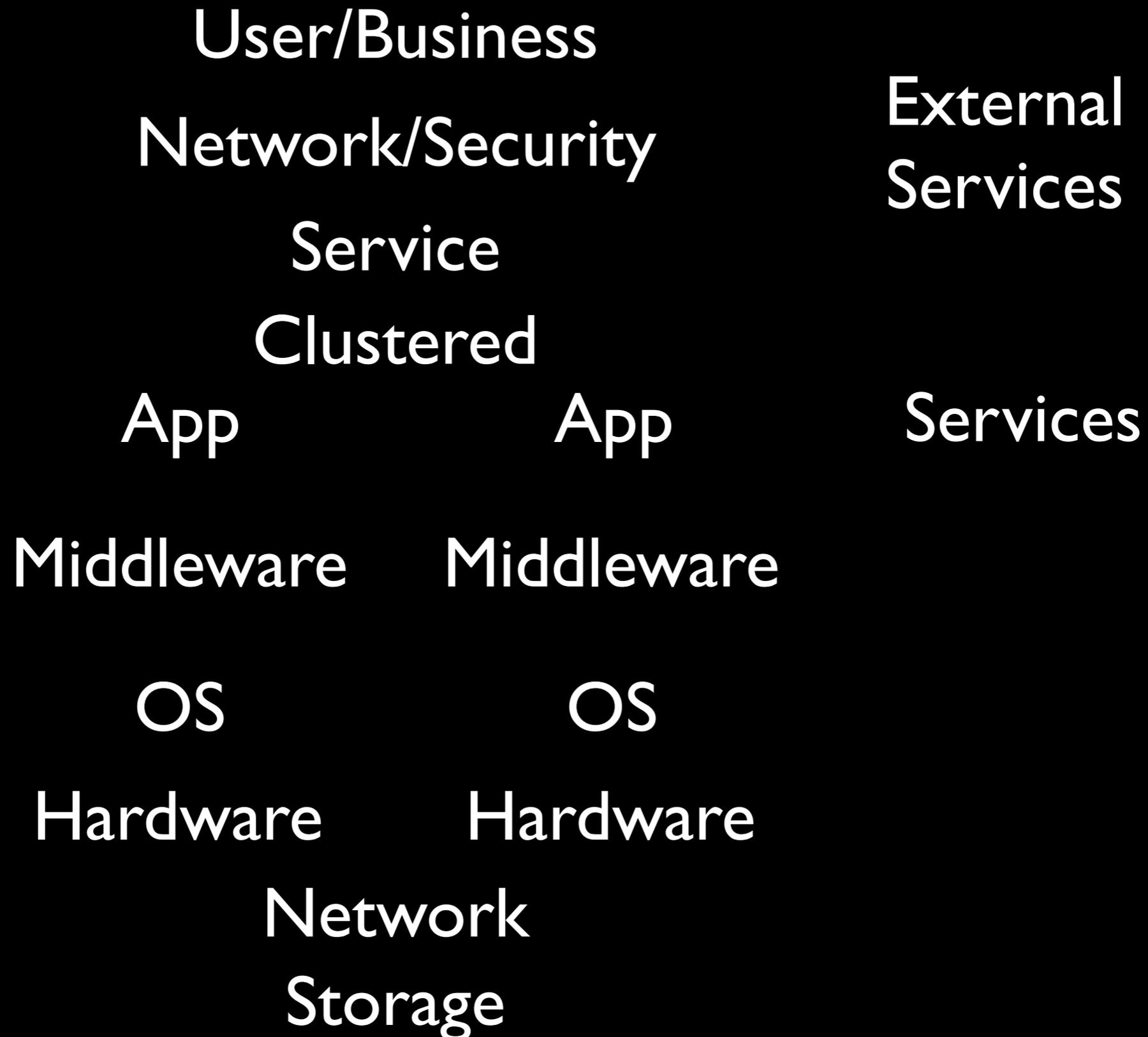
Infrastructure  
Code Repository

Application  
Code Repository

# Area 2: Extend operations feedback to project think “John Allspaw”



# Monitoring Layers



# Monitoring Layers

Behavioral (larger coverage)



Diagnostics

“What part made it fail”  
~unit test

CPU    Memory    Network

“Is it working?”  
~behavioral test

# Monitoring Driven Development

Create a monitor check before implementing a feature

**UC-1 Availability:  
100%**

**Information Radiator**



Monitoring  
Wonderland

<http://jedi.be/blog/2012/01/03/monitoring-wonderland-survey-introduction/>

<http://jedi.be/blog/2012/01/03/monitoring-wonderland-nagios-the-mighty-beast/>

<http://jedi.be/blog/2012/01/04/monitoring-wonderland-moving-up-the-stack-application-user-metrics/>

<http://jedi.be/blog/2012/01/03/monitoring-wonderland-metrics-api-gateways/>

<http://jedi.be/blog/2012/01/04/monitoring-wonderland-visualization/>



“traditional  
In house monitoring”



# Nagios

## General

- Home
- Documentation

## Monitoring

- Tactical Overview
- Service Detail
- Host Detail
- Status Overview
- Status Summary
- Status Grid
- Status Map
- 3-D Status Map
- Service Problems
- Host Problems
- Network Outages

- Comments
- Downtime

- Process Info
- Performance Info
- Scheduling Queue

## Reporting

- Trends
- Availability
- Alert Histogram
- Alert History
- Alert Summary
- Notifications
- Event Log

## Configuration

- View Config

**Current Network Status**  
 Last Updated: Tue Jul 6 11:22:11 CEST 2004  
 Updated every 90 seconds  
 Nagios® - [www.nagios.org](http://www.nagios.org)  
 Logged in as *bingel*

- [View History For all hosts](#)
- [View Notifications For All Hosts](#)
- [View Host Status Detail For All Hosts](#)

**Host Status Totals**

Up	Down	Unreachable	Pending
15	2	0	0
<b>All Problems</b>		<b>All Types</b>	
2		17	

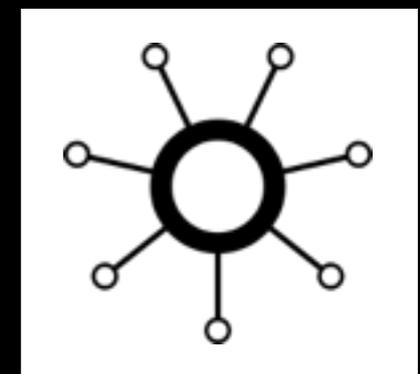
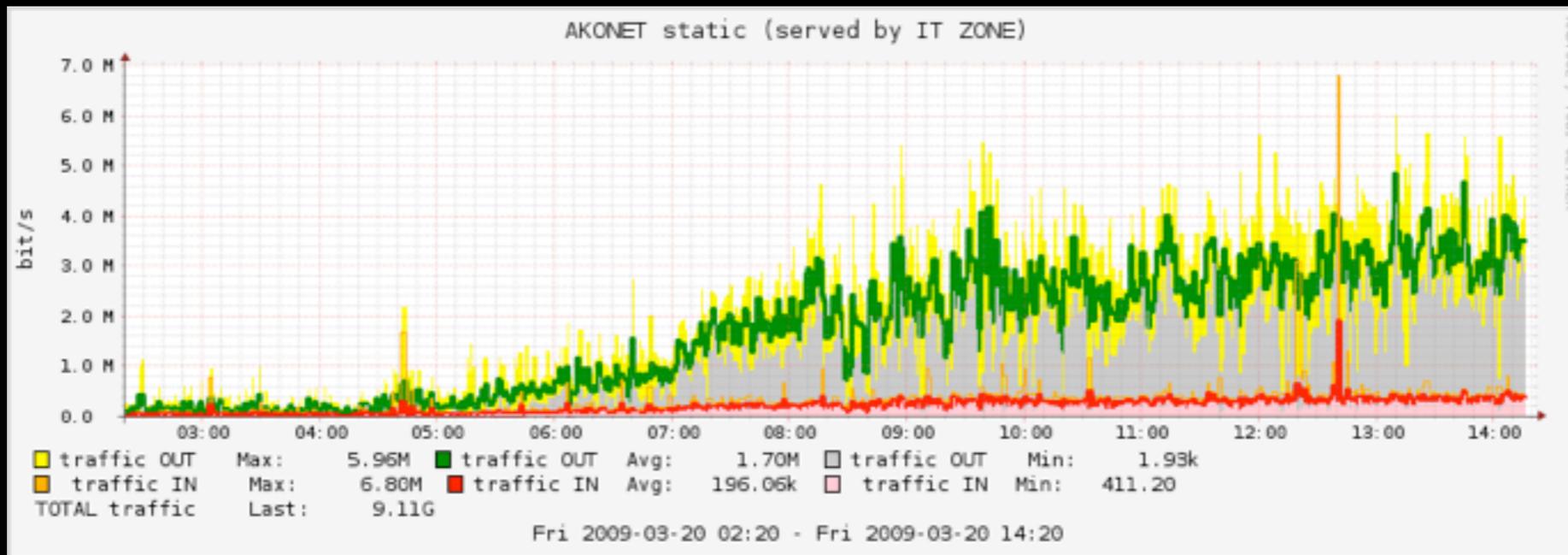
**Service Status Totals**

Ok	Warning	Unknown	Critical	Pending
29	1	1	4	0
<b>All Problems</b>		<b>All Types</b>		
6		35		

### Service Status Details For All Hosts

Host ↑↓	Service ↑↓	Status ↑	Last Check ↑↓	Duration ↑↓	Attempt ↑	Status Information
<a href="#">bart</a>	<a href="#">Diskusage C:</a>	OK	2004-07-06 11:10:19	20d 0h 7m 26s	1/3	C:\ - total: 3.91 Gb - used: 2.62 Gb (87%) - free 1.28 Gb (33%)
	<a href="#">Diskusage D:</a>	WARNING	2004-07-06 11:14:13	20d 0h 7m 20s	3/3	D:\ - total: 29.99 Gb - used: 26.78 Gb (89%) - free 3.20 Gb (11%)
	<a href="#">HTTP</a>	OK	2004-07-06 11:21:41	14d 1h 37m 47s	1/3	HTTP ok: HTTP/1.1 200 OK - 0.027 second response time
	<a href="#">MS-Exchange</a>	OK	2004-07-06 11:20:58	20d 0h 8m 6s	1/3	All services are running
	<a href="#">SMTP</a>	OK	2004-07-06 11:21:33	22d 1h 53m 13s	1/3	SMTP OK - 0 second response time
<a href="#">cixten</a>	<a href="#">Diskusage C:</a>	OK	2004-07-06 11:19:45	0d 22h 2m 24s	1/3	C:\ - total: 39.06 Gb - used: 8.01 Gb (21%) - free 31.05 Gb (79%)
	<a href="#">Terminalserver Sessions</a>	OK	2004-07-06 11:18:01	0d 22h 29m 13s	1/3	11
<a href="#">ftp.sunet.se</a>	<a href="#">PING</a>	OK	2004-07-06 11:21:30	0d 0h 31m 24s	1/10	PING OK - Packet loss = 0%, RTA = 36.37 ms
<a href="#">haubits</a>	<a href="#">PING</a>	OK	2004-07-06 11:21:21	193d 8h 33m 3s	1/10	PING OK - Packet loss = 0%, RTA = 1.36 ms
<a href="#">tknsgw1</a>	<a href="#">PING</a>	OK	2004-07-06 11:21:21	56d 22h 30m 57s	1/10	PING OK - Packet loss = 0%, RTA = 6.10 ms
	<a href="#">if-traffic</a>	OK	2004-07-06 11:21:20	63d 18h 45m 23s	1/10	OK: rate[IN]=250 kbit/s OK: rate[OUT]=286 kbit/s
<a href="#">tknsgw2</a>	<a href="#">PING</a>	CRITICAL	2004-04-20 12:39:00	77d 20h 45m 1s	10/10	PING CRITICAL - Packet loss = 100%
	<a href="#">if-traffic</a>	UNKNOWN	2004-04-20 12:39:00	77d 20h 45m 1s	10/10	check_snmp_counter: ERROR during get-request: No response from remote host '62.119.68.186'
<a href="#">jasper</a>	<a href="#">Diskusage C:</a>	OK	2004-07-06 11:15:46	77d 1h 13m 47s	1/3	C:\ - total: 4.00 Gb - used: 2.99 Gb (75%) - free 1.01 Gb (25%)
	<a href="#">Diskusage E:</a>	OK	2004-07-06 11:20:41	140d 1h 3m 7s	1/3	E:\ - total: 4.00 Gb - used: 1.46 Gb (36%) - free 2.54 Gb (64%)
	<a href="#">MS-Exchange</a>	OK	2004-07-06 11:21:41	61d 22h 57m 53s	1/3	All services are running
	<a href="#">MS-Exchange NotesConnector</a>	OK	2004-07-06 11:21:41	61d 22h 57m 57s	1/3	All services are running
	<a href="#">SMTP</a>	OK	2004-07-06 11:21:43	61d 22h 58m 4s	1/3	SMTP OK - 0 second response time
<a href="#">lenin</a>	<a href="#">SMTP</a>	OK	2004-07-06 11:21:51	20d 2h 18m 14s	1/3	SMTP OK - 0 second response time

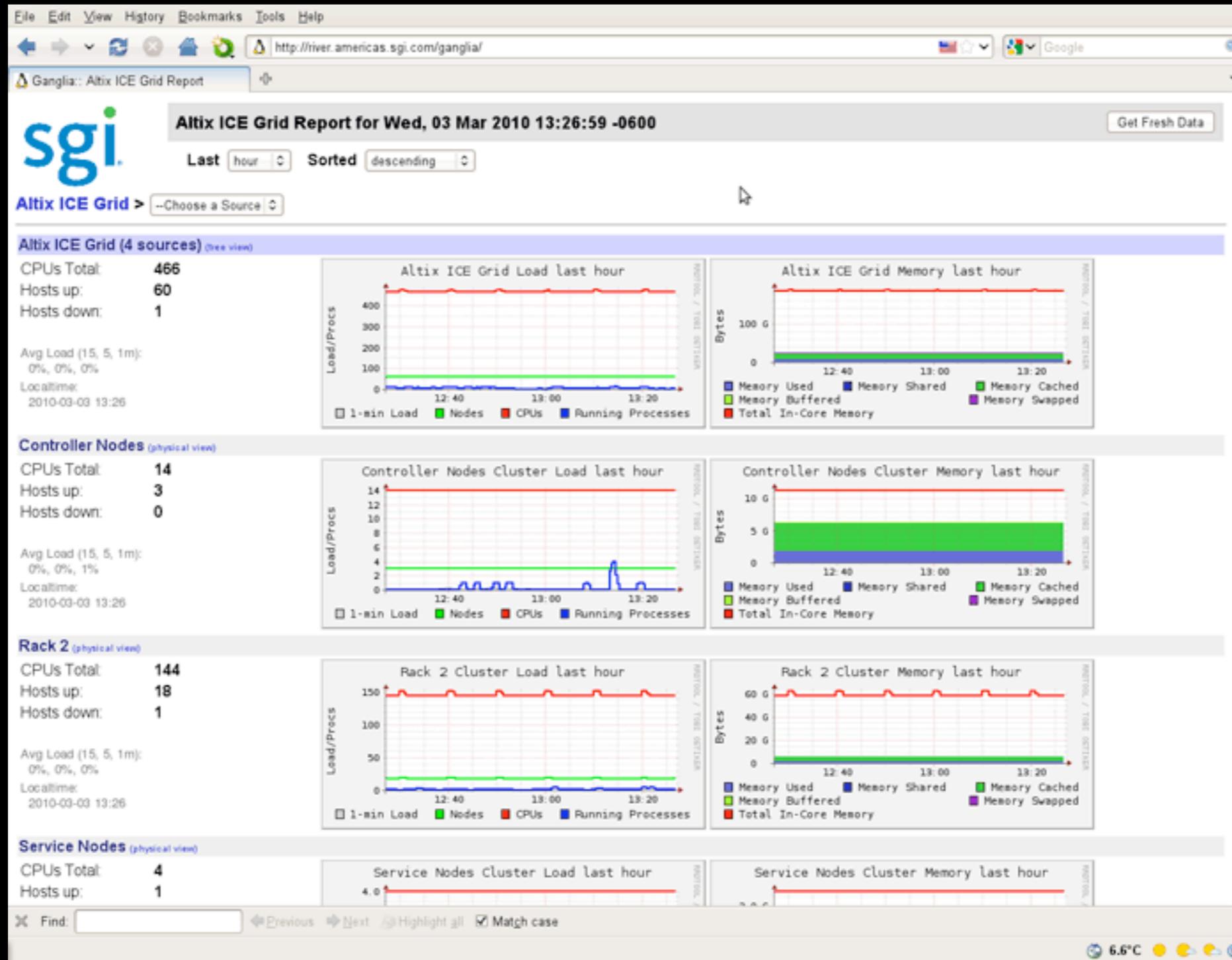
# “traditional In house metrics”



<http://munin-monitoring.org/>

<http://collectd.org/>

# Ganglia



<http://ganglia.info/>

[https://github.com/  
monitoringsucks](https://github.com/monitoringsucks)

“1980” style UI

No API

Hard to scale

# Static configurations

Hard to expose “data”  
UI vs API

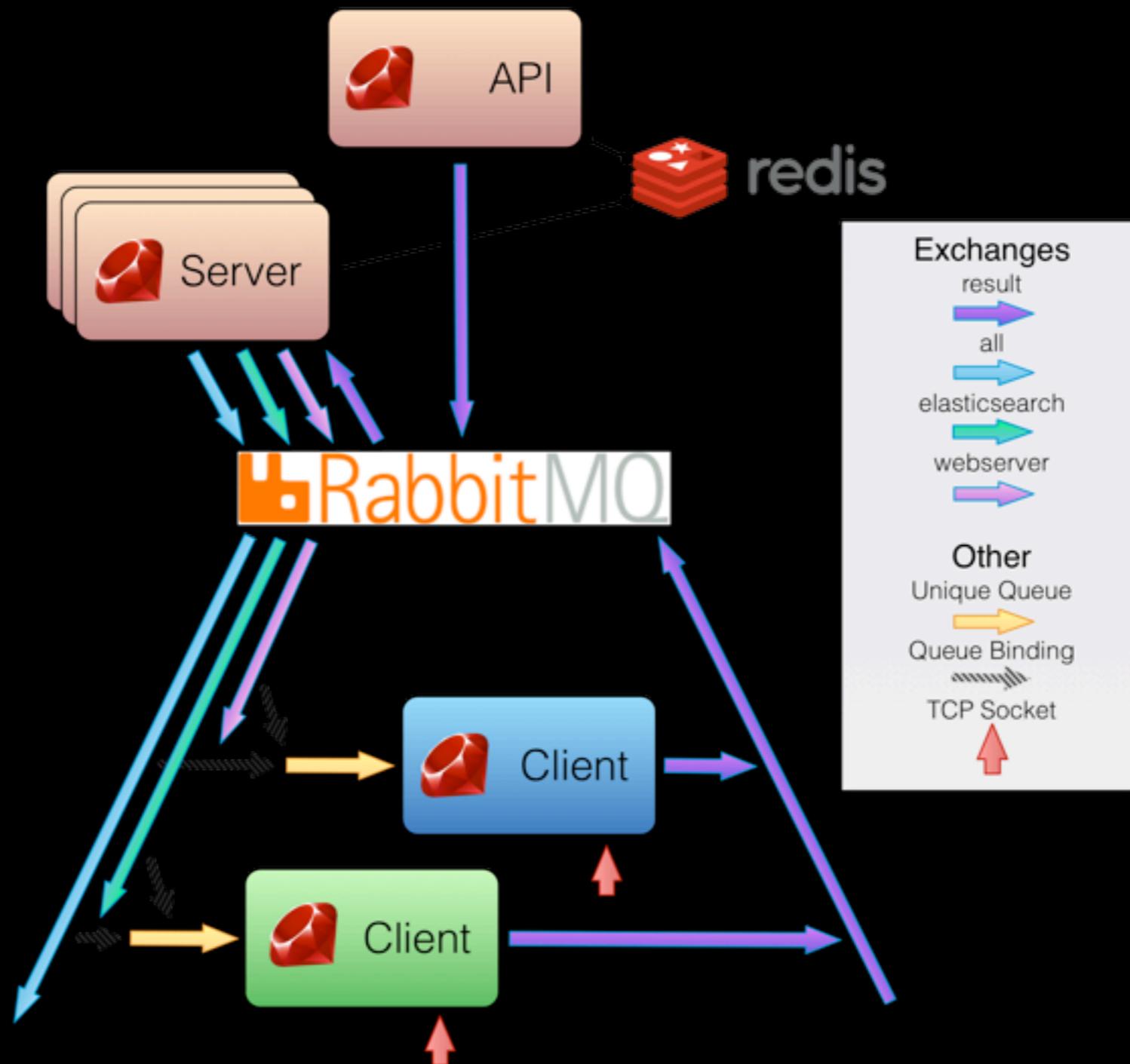
Not self-servicing



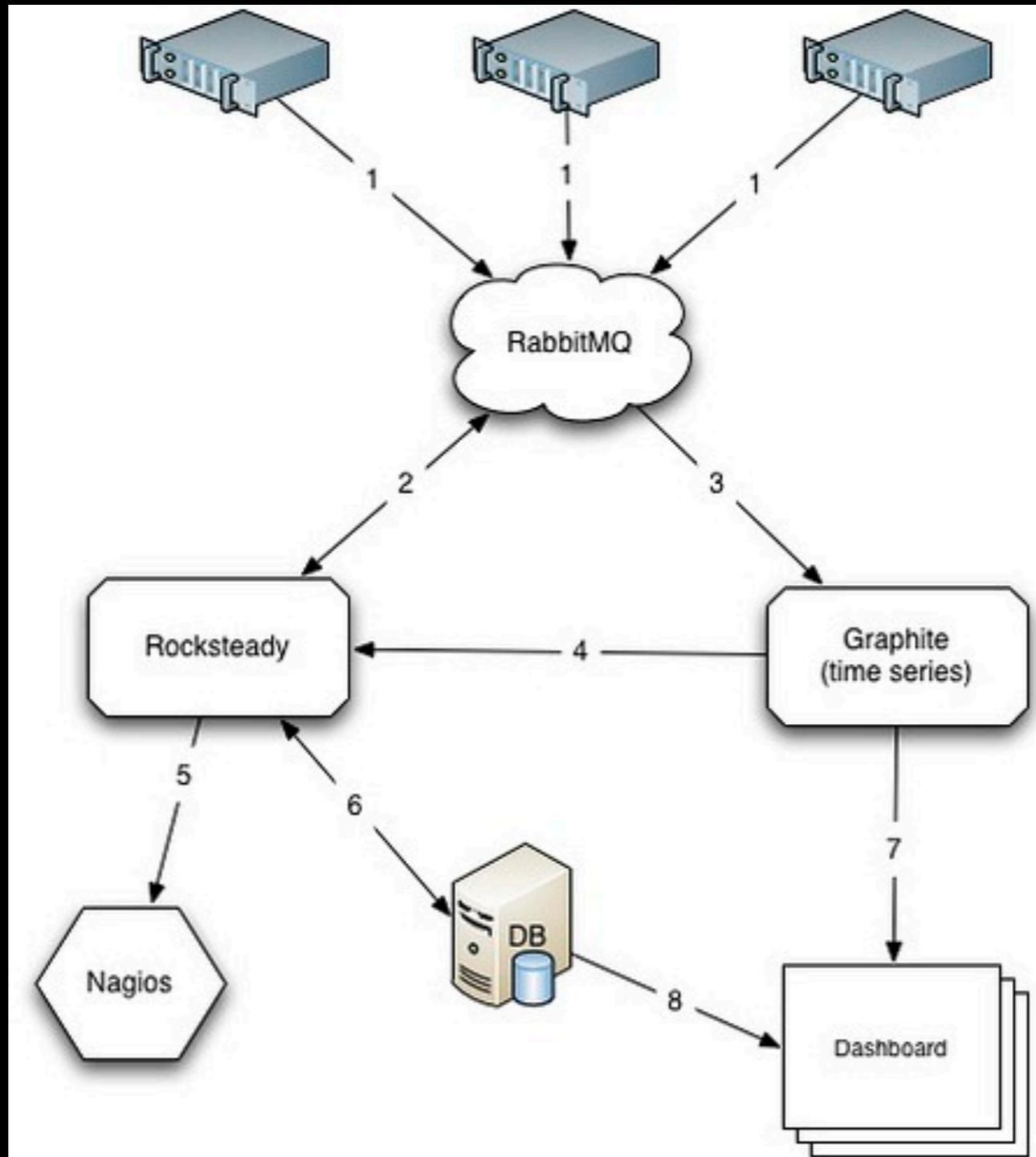
# “Monitoring as a service”



# Sonian Cloud Monitoring "Sensu"



<https://github.com/sensu/sensu>



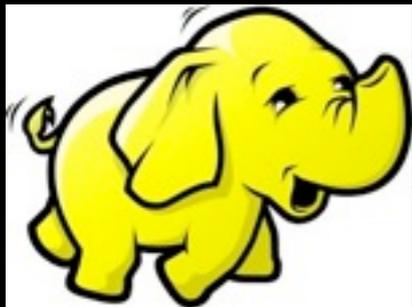
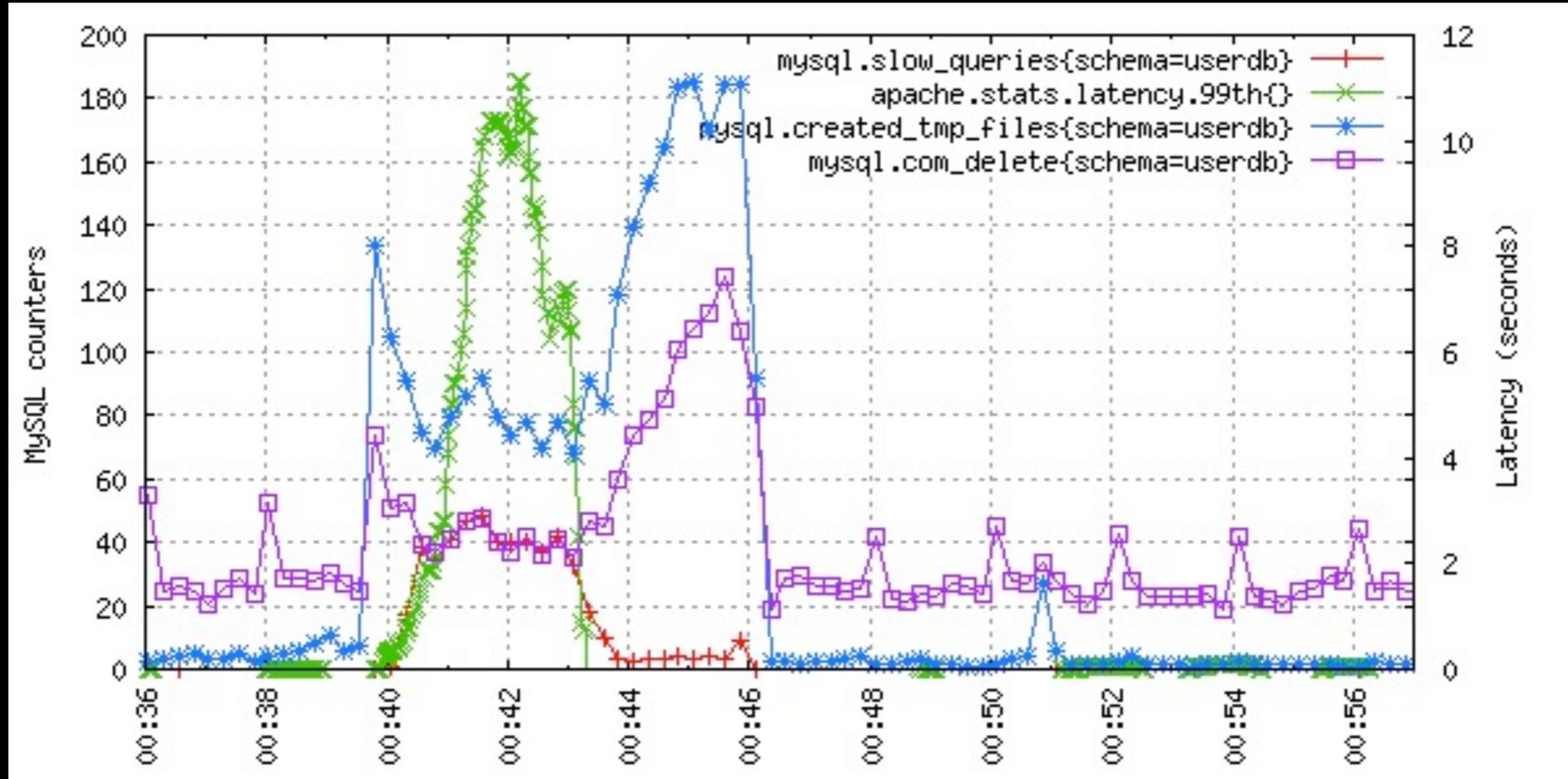
<http://code.google.com/p/rocksteady/>



Reconnoiter's goal is to better the world of monitoring by marrying fault detection and trending together. Through ease of configuration and ongoing maintenance, Reconnoiter encourages monitoring important technical metrics along side critical business metrics to improve awareness and ultimately accountability.

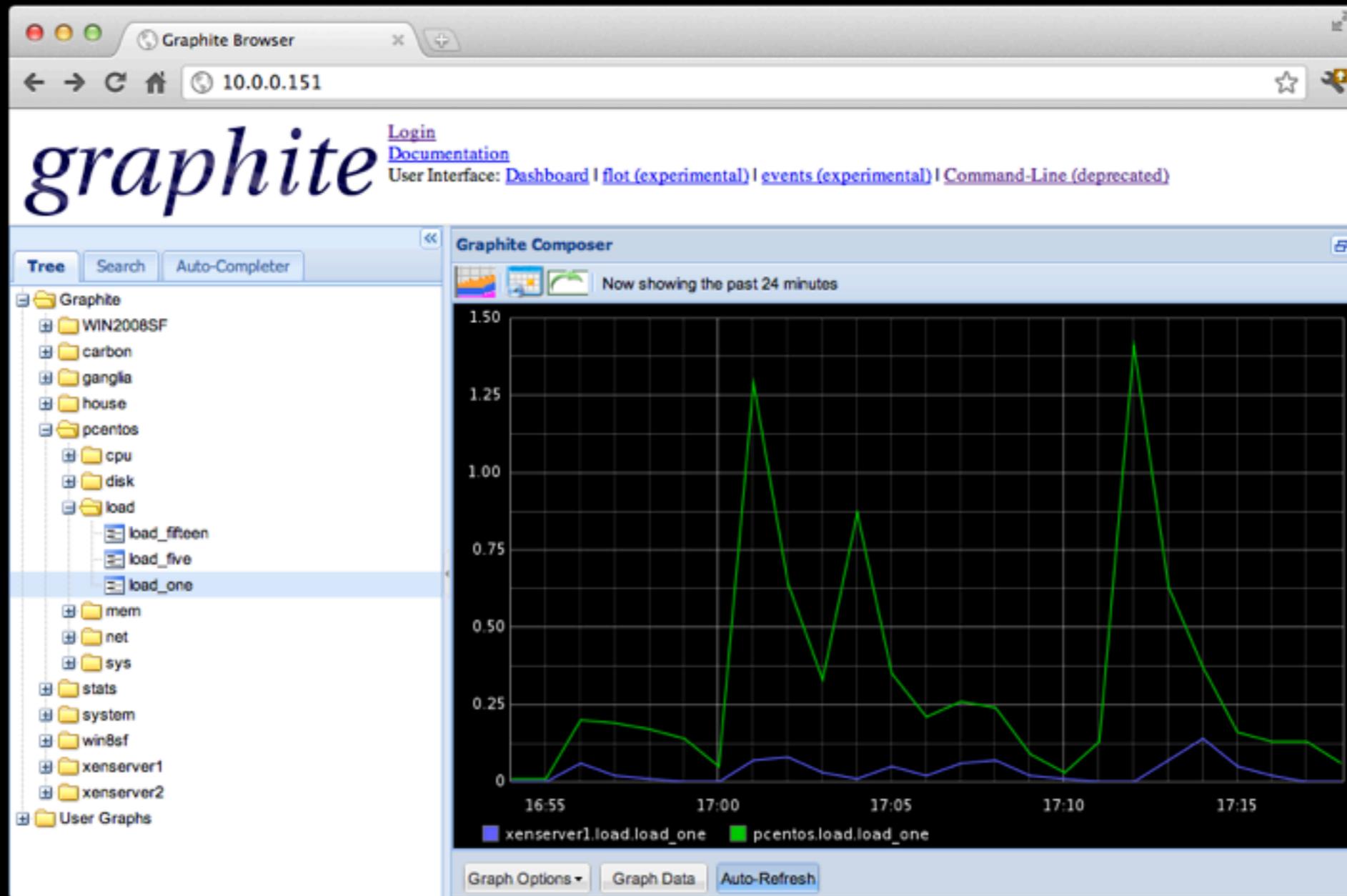
<http://labs.omniti.com/labs/reconnoiter>

# Scalable writes



<http://opentsdb.net/>

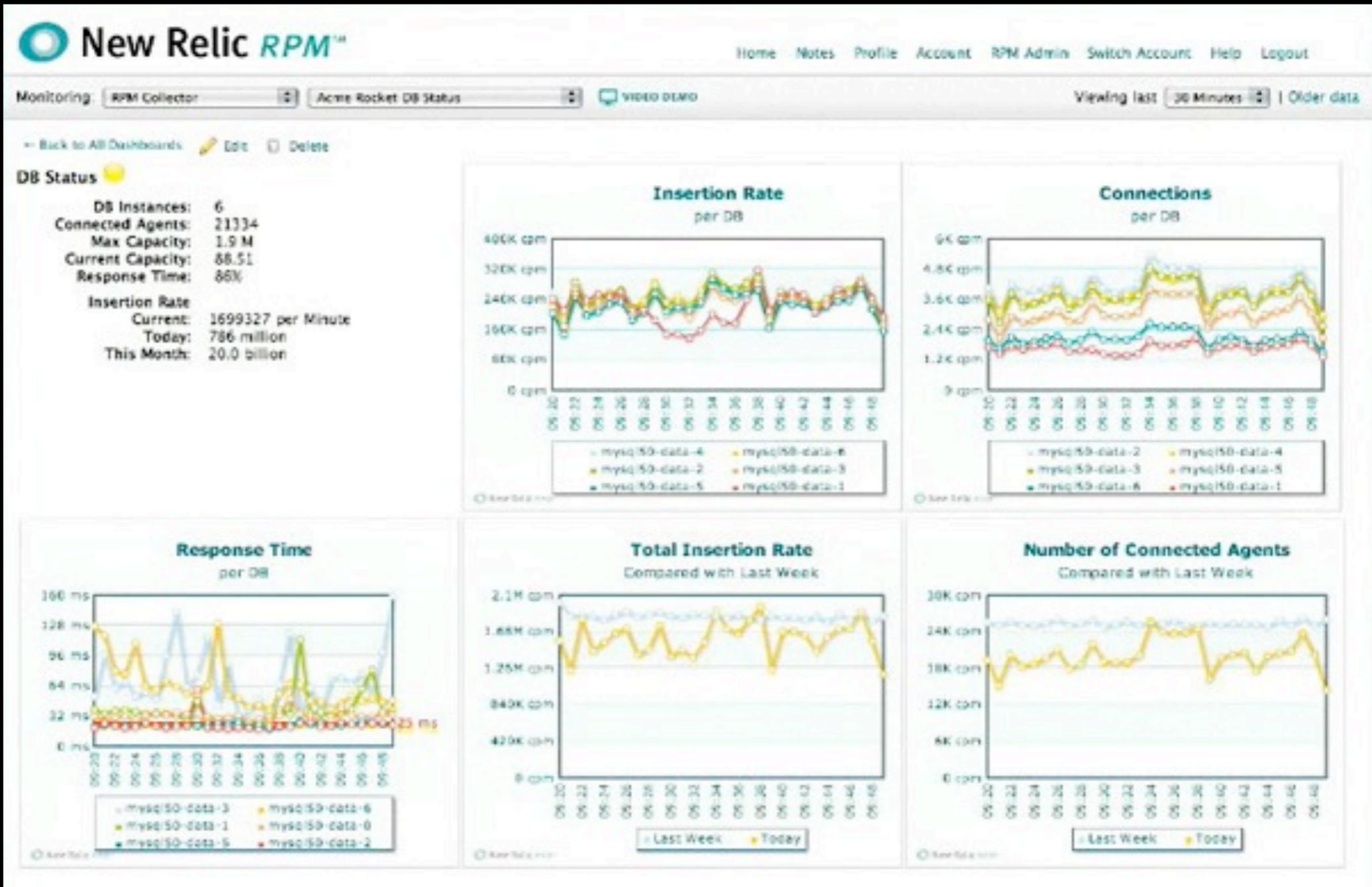
# Selfservicing Graphs



<http://graphite.wikidot.com/>

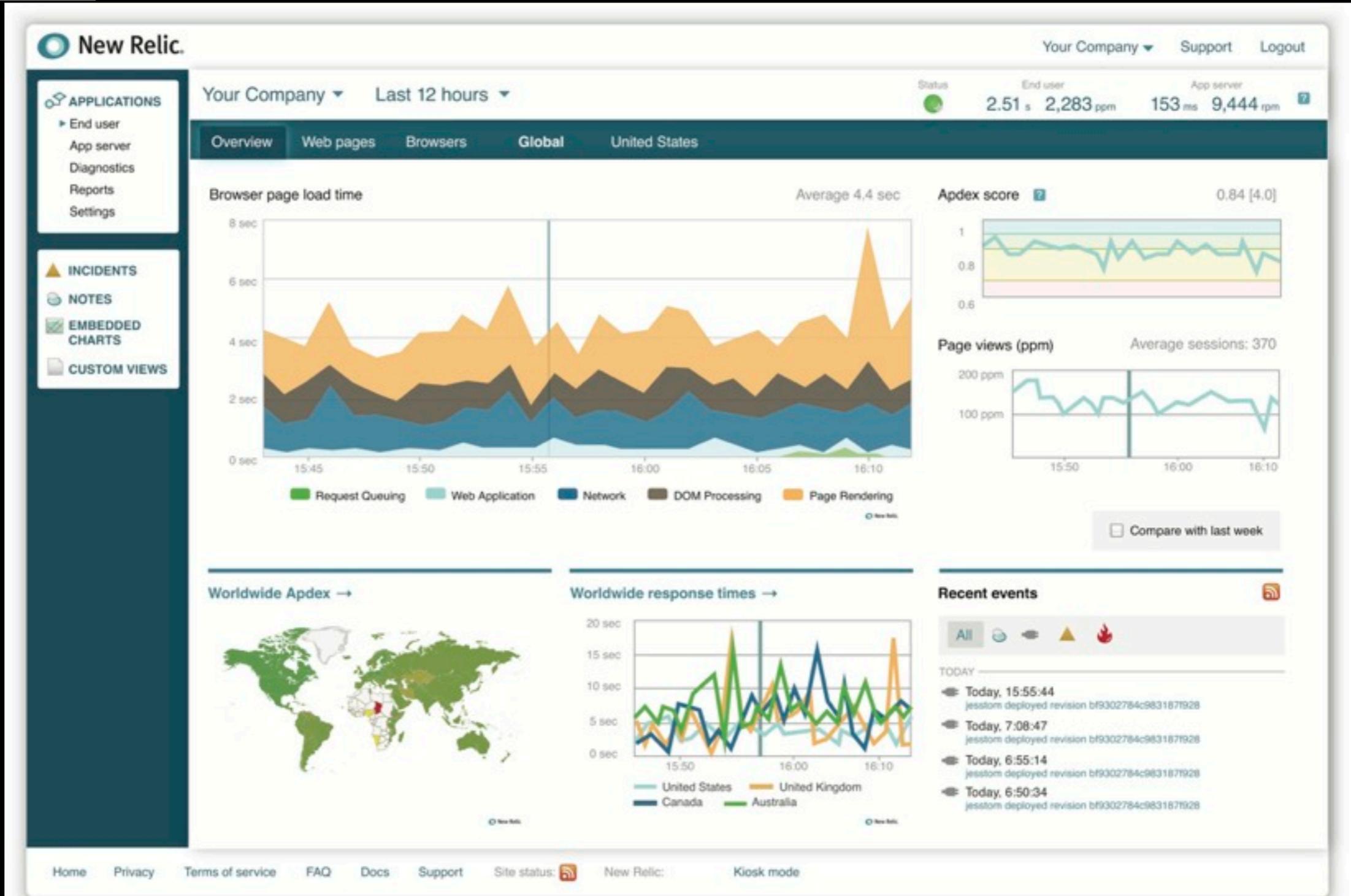


# “App Metrics”



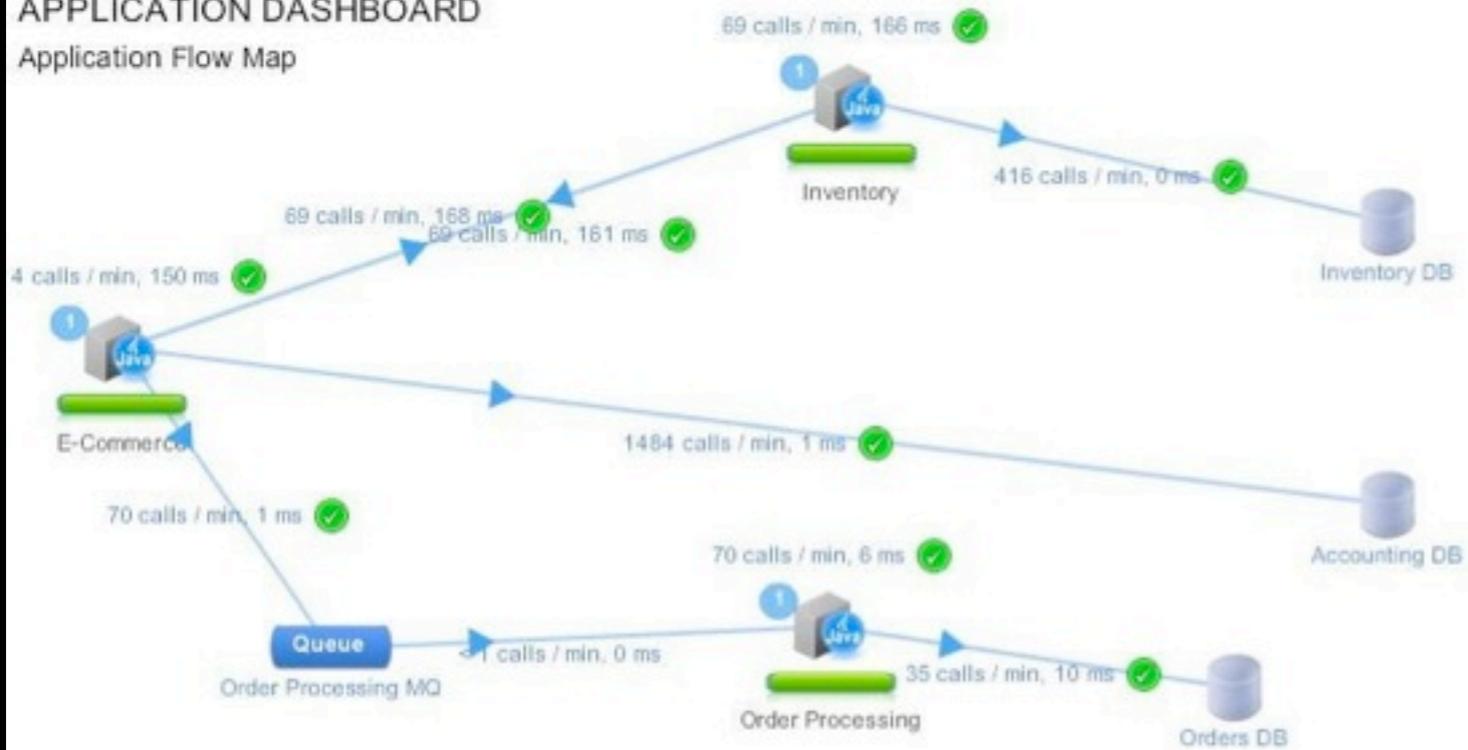


# “End-User Metric



# APPLICATION DASHBOARD

## Application Flow Map



No Incidents in the time range [View All](#)

## Request Summary

Category	Percentage	Count
Normal	99.2%	6371
Slow	0.0%	0
Very Slow	0.7%	45
Errors	0.1%	5
Stalls	0.1%	4

Total Requests: 6420

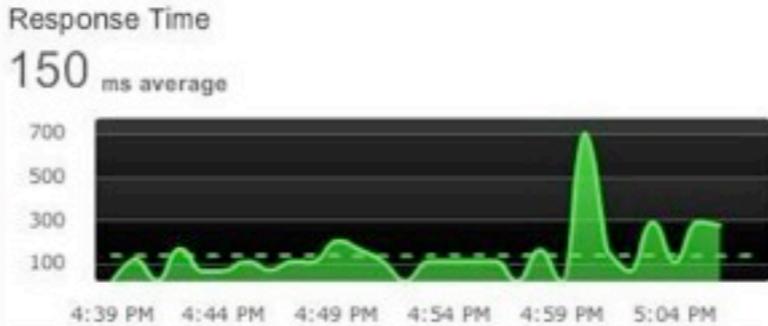
## Top Transactions by Load

Name	calls/min
GetAllItems	36
Checkout	36
Logout	36
AddToCart	36
Login	36

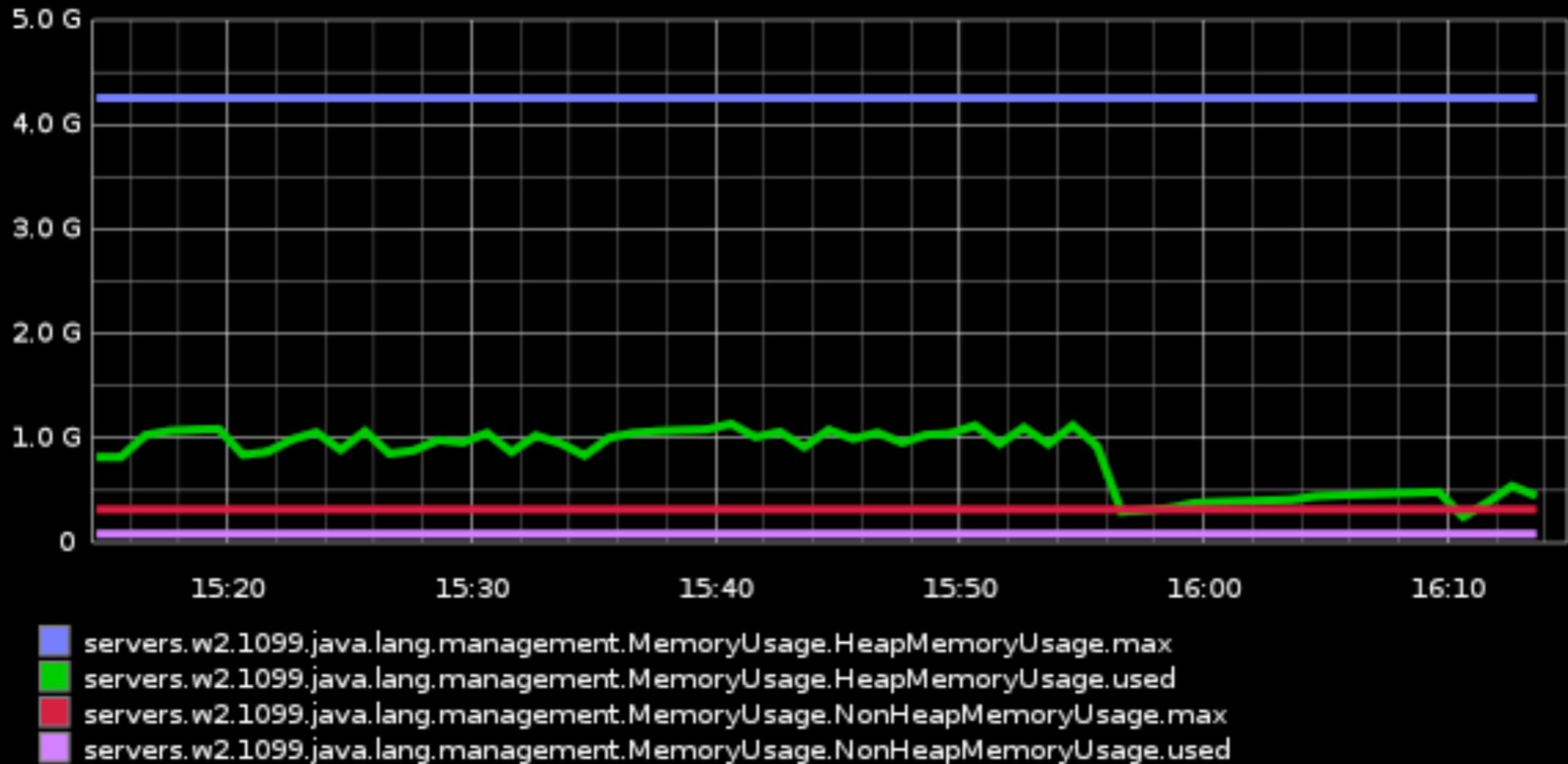
## Top Transactions by Resp. Time

Name	resp.time
Checkout	300
Logout	266
AddToCart	234
GetAllItems	19
Login	14

[Explain this View](#)



<http://www.appdynamics.com/>



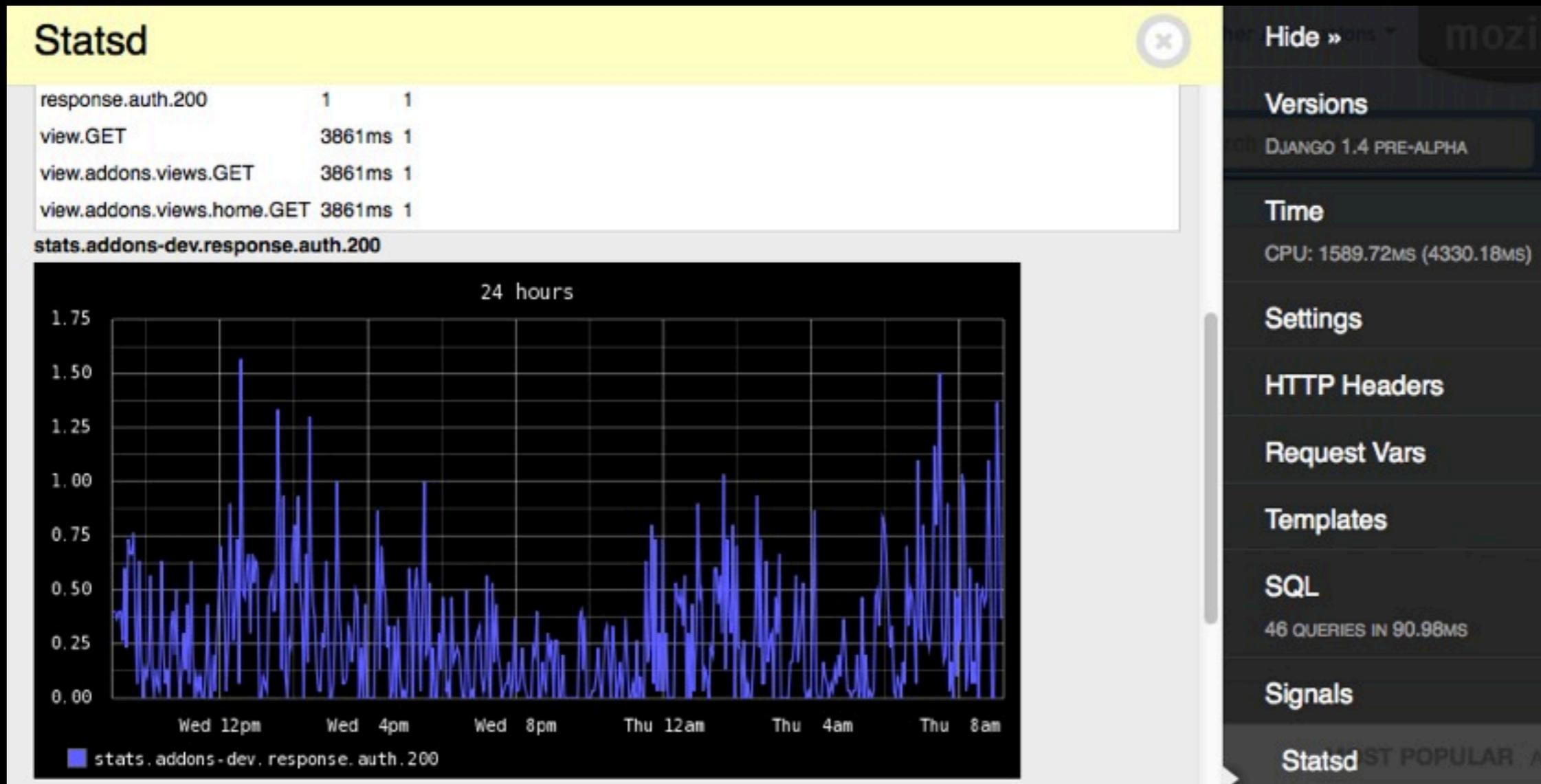
<https://github.com/lookfirst/jmxtrans>

# APP

# STATSD

# GRAPHITE

counting  
timing  
sampling



<https://github.com/etsy/statsd>

# Metrics Java Library

<http://metrics.codahale.com/>

<https://github.com/tritonrc/metricsd>

# Business Level Metrics

Usage patterns - Google Analytics

# A/B Testing in Prod

Metrics / Information

# Meta-Ops Metrics

Know context



# Logs as feedback

<http://logstash.net/>

<http://graylog2.org/>

<https://github.com/rashidkpc/Kibana>

# Unix Pipe Analogy

## inputs

- amqp
- exec
- file
- gelf
- redis
- stdin
- stomp
- syslog
- tcp
- twitter
- xmpp
- zeromq

## filters

- date
- dns
- gelfify
- grep
- grok
- grokdiscovery
- json
- multiline
- mutate
- split

## outputs

- amqp
- elasticsearch
- elasticsearch\_river
- file
- ganglia
- gelf
- graphite
- internal
- loggly
- mongodb
- nagios
- null
- redis
- statsd
- stdout
- stomp
- tcp
- websocket
- xmpp
- zabbix
- zeromq

# Integrate with **Continuous Integration**

DEV

TEST

PROD

APP



INFRA



Infrastructure  
Code Repository

Application  
Code Repository

# Reuse Tools across dev & ops

DEV

TEST

PROD

APP



INFRA



Monitoring/  
Metrics

Infrastructure  
Code Repository

Application  
Code Repository

# Integrate with **Continuous Monitoring**

DEV

TEST

PROD

APP



INFRA



Monitoring/  
Metrics

Monitoring/  
Metrics

Monitoring/  
Metrics

Infrastructure  
Code Repository

Application  
Code Repository

# Reuse Workflow

Debugging

Post-Mortem

DEV

TEST

PROD

APP



INFRA



Monitoring/  
Metrics

Monitoring/  
Metrics

Monitoring/  
Metrics

# Metrics Driven Engineering

[http://www.slideshare.net/mikebrittain/metricsdriven-  
engineering](http://www.slideshare.net/mikebrittain/metricsdriven-engineering)

# Measure Anything, Measure Everything

Posted by Ian Malpass | Filed under [data](#), [engineering](#), [infrastructure](#)

If Engineering at Etsy has a religion, it's the Church of Graphs. If it moves, we track it. Sometimes we'll draw a graph of something that isn't moving yet, just in case it decides to make a run for it. In general, we tend to measure at three levels: network, machine, and application. (You can read more about our graphs in Mike's [Tracking Every Release](#) post.)

“The ~~future is~~ metrics are here  
they are just not evenly  
distributed yet”

- Morpheus Monitoring

<http://jedi.be/blog>  
<https://github.com/monitoringsucks>

