Cloud Native Architecture at Netflix

Yow December 2013 (Brisbane)

Adrian Cockcroft

@adrianco @NetflixOSS

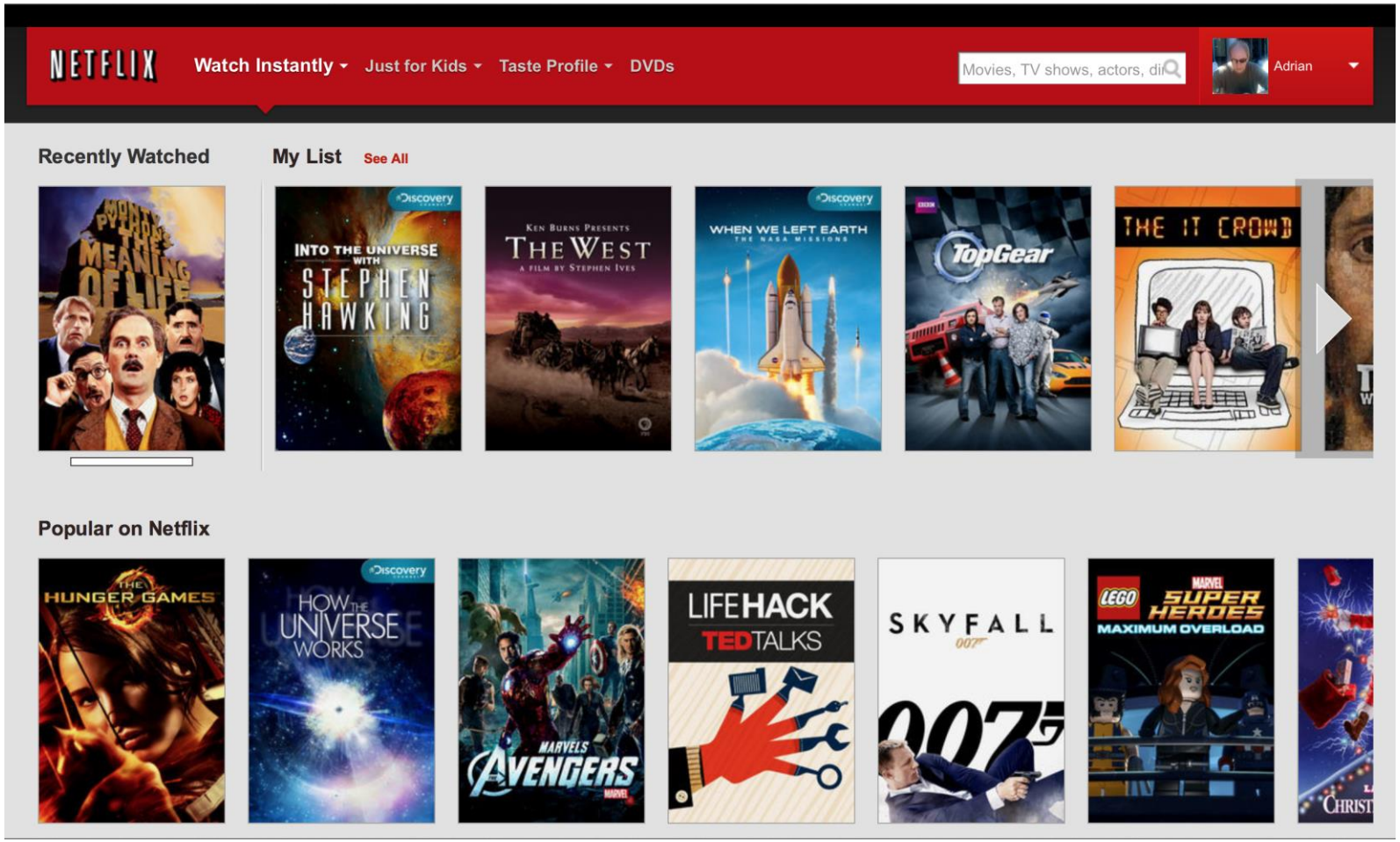
http://www.linkedin.com/in/adriancockcroft

Netflix History (current size)

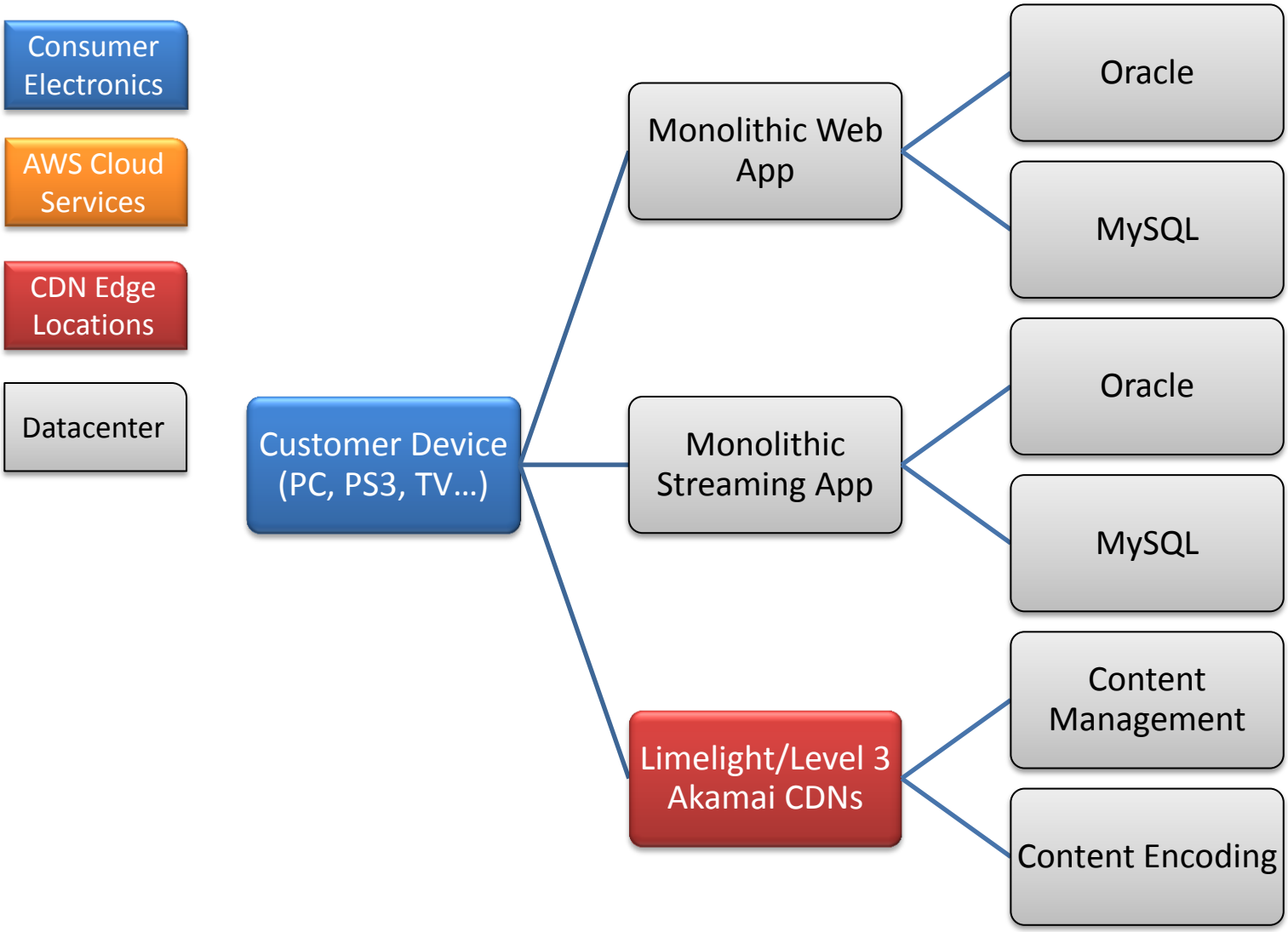
- 1998 DVD Shipping Service in USA (~7M users)
- 2007 Streaming video in USA (~31M users)
- International streaming video (~9M users)
  - 2010 Canada
  - 2011 Latin America
  - 2012 UK and Ireland
  - 2012 Nordics
  - 2013 Netherlands

Netflix Member Web Site Home Page

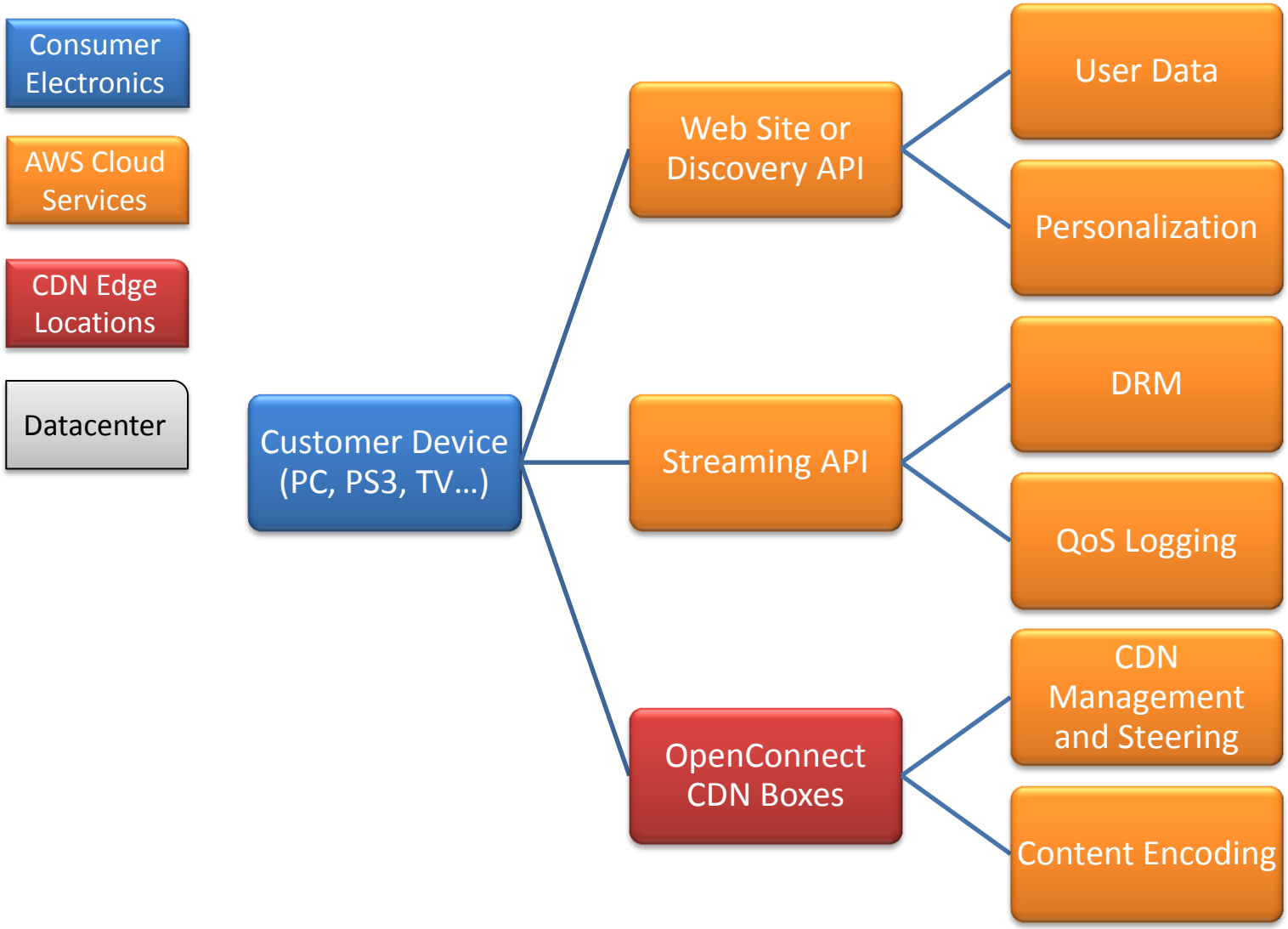
Personalization Driven – How Does It Work?



How Netflix Used to Work



How Netflix Streaming Works Today



2H 2013

	Upstream		Downstream		Aggregate	
Rank	Application	Share	Application	Share	Application	Share
1	BitTorrent	36.35%	Netflix	31.62%	Netflix	28.18%
2	HTTP	6.03%	YouTube	18.69%	YouTube	16.78%
3	SSL	5.87%	HTTP	9.74%	HTTP	9.26%
4	Netflix	4.44%	BitTorrent	4.05%	BitTorrent	7.39%
5	YouTube	3.63%	iTunes	3.27%	iTunes	2.91%
6	Skype	2.76%	MPEG - Other	2.60%	SSL	2.54%
7	QVoD	2.55%	SSL	2.05%	MPEG - Other	2.32%
8	Facebook	1.54%	Amazon Video	1.61%	Amazon Video	1.48%
9	FaceTime	1.44%	Facebook	1.31%	Facebook	1.34%
10	Dropbox	1.39%	Hulu	1.29%	Hulu	1.15%
		66.00%		76.23%		73.35%

sandvine

Table 2 - Top 10 Peak Period Applications - North America, Fixed Access

1H 2013

	Upstream		Downstream		Aggregate	
Rank	Application	Share	Application	Share	Application	Share
1	BitTorrent	34.81%	Netflix	32.25%	Netflix	28.88%
2	HTTP	7.53%	YouTube	17.11%	YouTube	15.43%
3	SSL	5.81%	HTTP	11.11%	HTTP	10.66%
4	Netflix	5.38%	BitTorrent	5.57%	BitTorrent	9.23%
5	Skype	4.88%	MPEG	2.58%	SSL	2.39%
6	YouTube	3.71%	Hulu	2.41%	MPEG	2.30%
7	Facebook	1.71%	iTunes	1.90%	Hulu	2.16%
8	Apple Photostream	1.34%	SSL	1.89%	iTunes	1.71%
9	Dropbox	1.21%	Flash Video	1.72%	Flash Video	1.53%
10	Carbonite	0.99%	Facebook	1.48%	Facebook	1.52%
Top 10		**67.38%**		**78.03%**		**75.82%**

sandvine

Table 2 - Top 10 Peak Period Applications - North America, Fixed Access

2H 2012

	Upstream		Downstream		Aggregate	
Rank	Application	Share	Application	Share	Application	Share
1	BitTorrent	36.8%	Netflix	33.0%	Netflix	28.8%
2	HTTP	9.83%	YouTube	14.8%	YouTube	13.1%
3	Skype	4.76%	HTTP	12.0%	HTTP	11.7%
4	Netflix	4.51%	BitTorrent	5.89%	BitTorrent	10.3%
5	SSL	3.73%	iTunes	3.92%	iTunes	3.43%
6	YouTube	2.70%	MPEG	2.22%	SSL	2.23%
7	PPStream	1.65%	Flash Video	2.21%	MPEG	2.05%
8	Facebook	1.62%	SSL	1.97%	Flash Video	2.01%
9	Apple PhotoStream	1.46%	Amazon Video	1.75%	Facebook	1.50%
10	Dropbox	1.17%	Facebook	1.48%	RTMP	1.41%
	Top 10	68.24%	Top 10	79.01%	Top 10	76.54%

sandvine

Table 3 - Top 10 Peak Period Applications (North America, Fixed Access)

Netflix Scale

- Tens of thousands of instances on AWS
  - Typically 4 core, 30GByte, Java business logic
  - Thousands created/removed every day
- Thousands of Cassandra NoSQL nodes on AWS
  - Many hi1.4xl - 8 core, 60Gbyte, 2TByte of SSD
  - 65 different clusters, over 300TB data, triple zone
  - Over 40 are multi-region clusters (6, 9 or 12 zone)
  - Biggest 288 m2.4xl – over 300K rps, 1.3M wps

Reactions over time

2009 “You guys are crazy! Can’t believe it”

2010 “What Netflix is doing won’t work”

2011 “It only works for ‘Unicorns’ like Netflix”

2012 “We’d like to do that but can’t”

2013 “We’re on our way using Netflix OSS code”

YOW! Workshop

175 slides of Netflix Architecture

See bit.ly/netflix-workshop

A whole day…

This Talk

Abstract the principles from the architecture

Objectives:

Scalability

Availability

Agility

Efficiency

Principles:

Immutability

Separation of Concerns

Anti-fragility

High trust organization

Sharing

Outcomes:

- Public cloud – scalability, agility, sharing
- Micro-services – separation of concerns
- De-normalized data – separation of concerns
- Chaos Engines – anti-fragile operations
- Open source by default – agility, sharing
- Continuous deployment – agility, immutability
- DevOps – high trust organization, sharing
- Run-what-you-wrote – anti-fragile development

When to use public cloud?

NETFLIX OSS

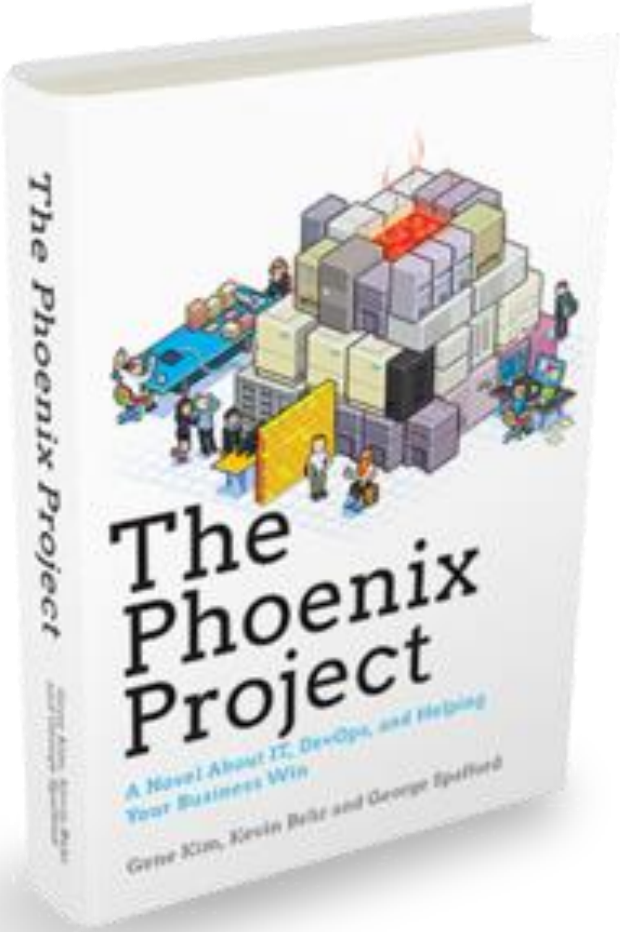
adrian cockcroft @adrianco 10 Apr

Baffling-late-adopters as a Service

Retweeted by Andrew Clay Shafer

Expand

"This is the IT swamp draining manual for anyone who is neck deep in alligators." **-**
Adrian Cockcroft, Cloud Architect at Netflix



Goal of Traditional IT: Reliable hardware running stable software

SCALE

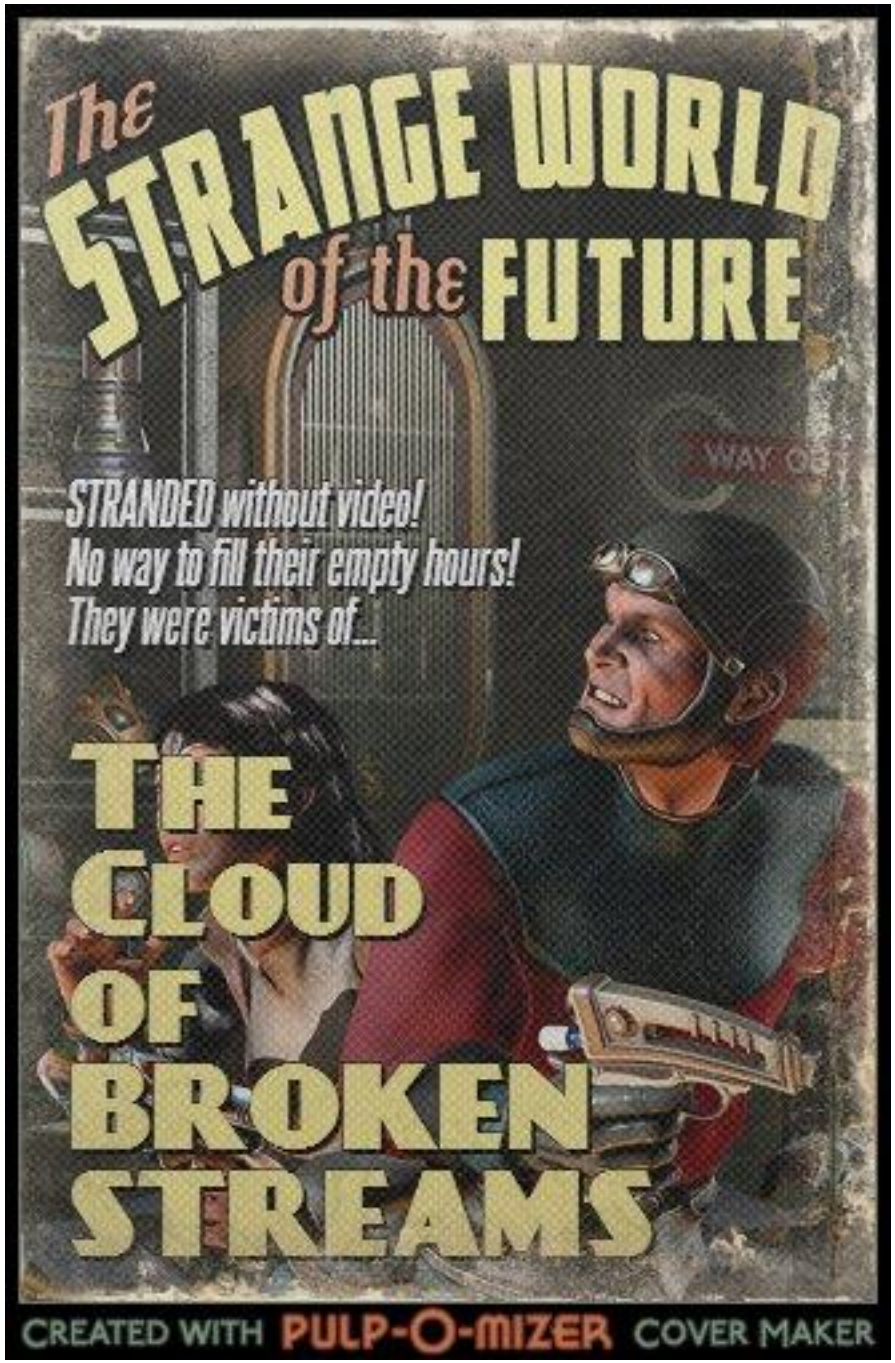
Breaks hardware

....*SPEED*

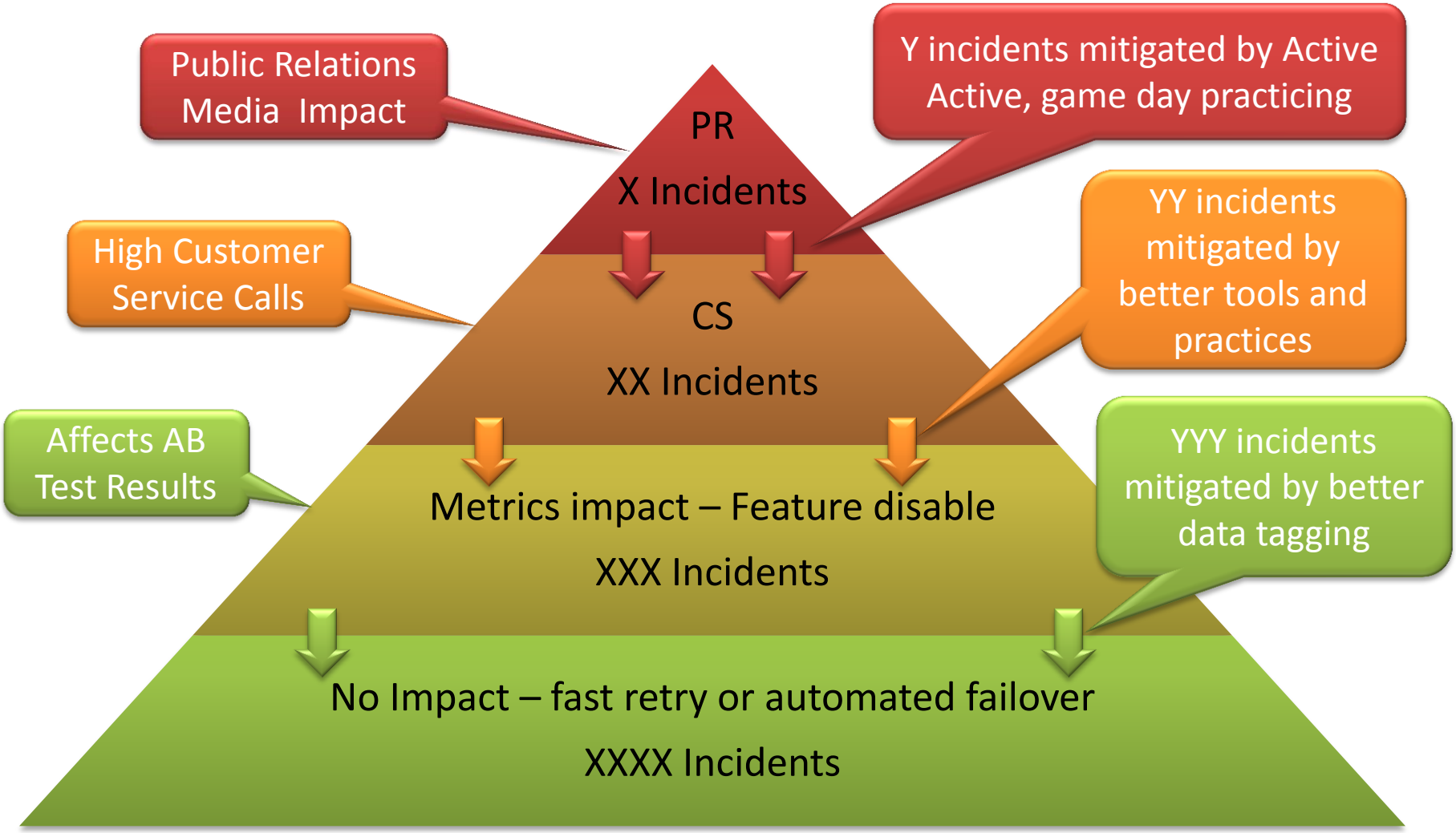
Breaks software

SPEED at SCALE

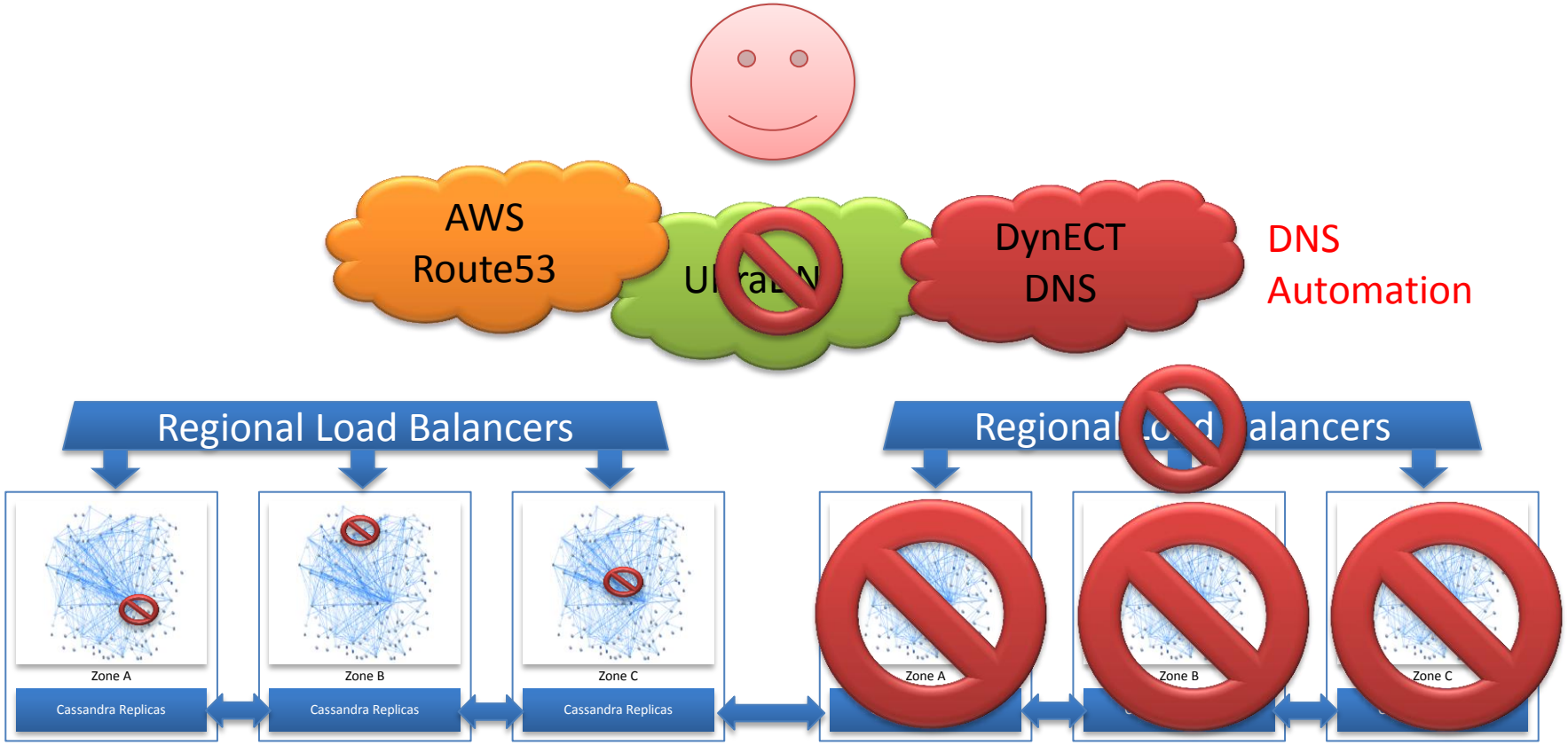
Breaks everything


The STRANGE WORLD of the FUTURE
STRANDED without video!
No way to fill their empty hours!
They were victims of...
THE CLOUD OF BROKEN STREAMS
CREATED WITH PULP-O-MIZER COVER MAKER

Incidents – Impact and Mitigation



Web Scale Architecture

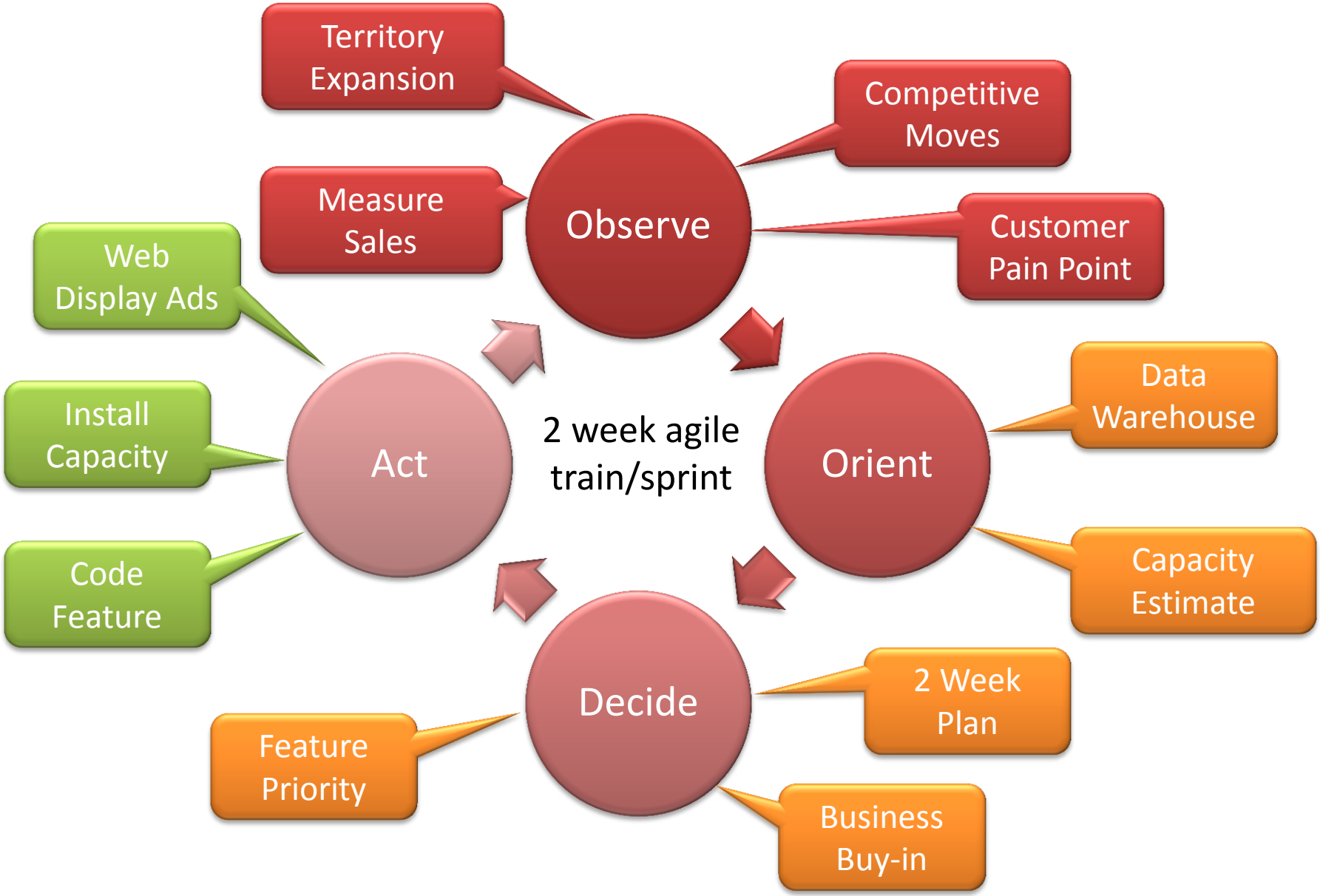


“Get inside your adversaries' OODA loop to disorient them”

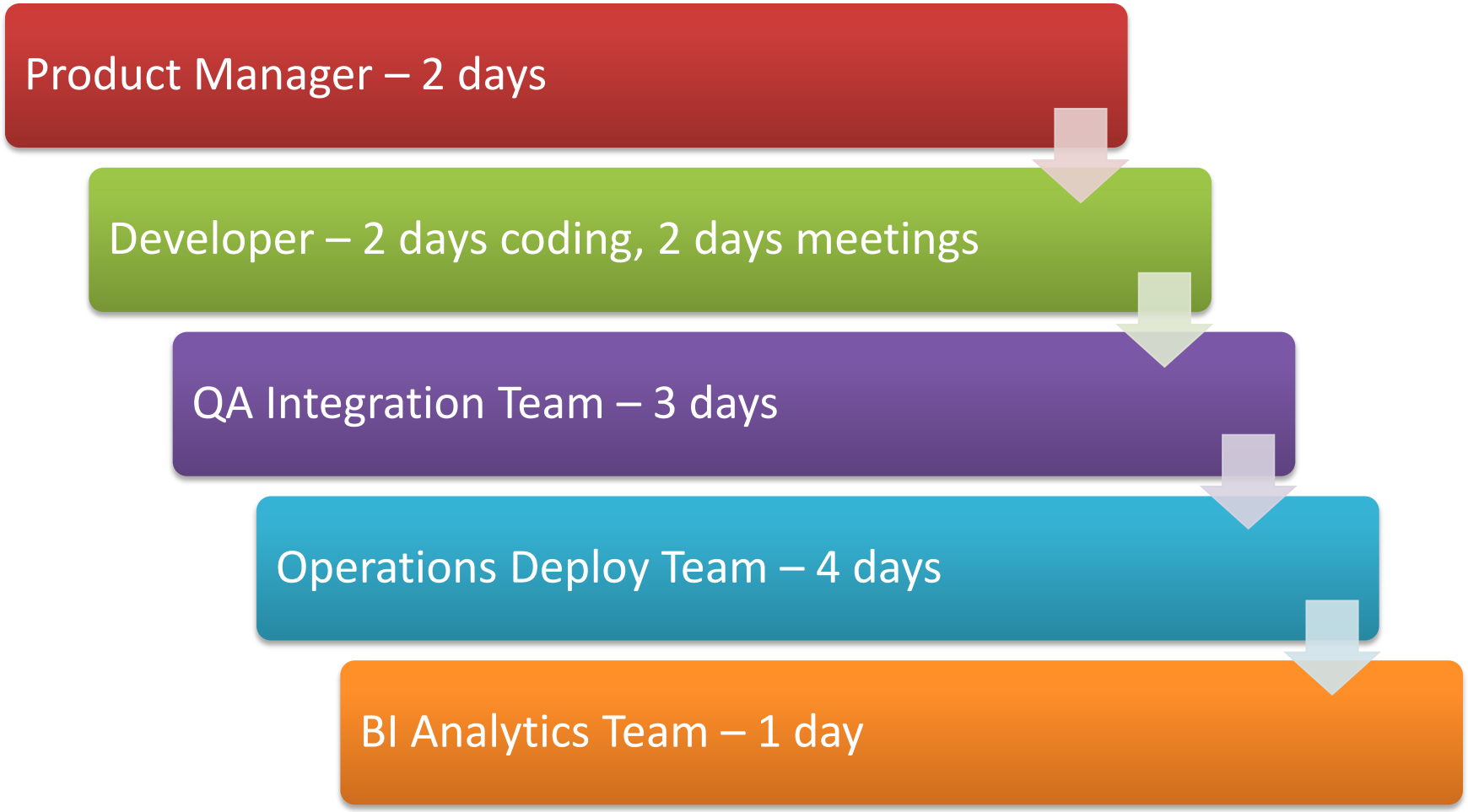
Colonel Boyd, USAF

“Agile” vs. “Continuous Delivery”

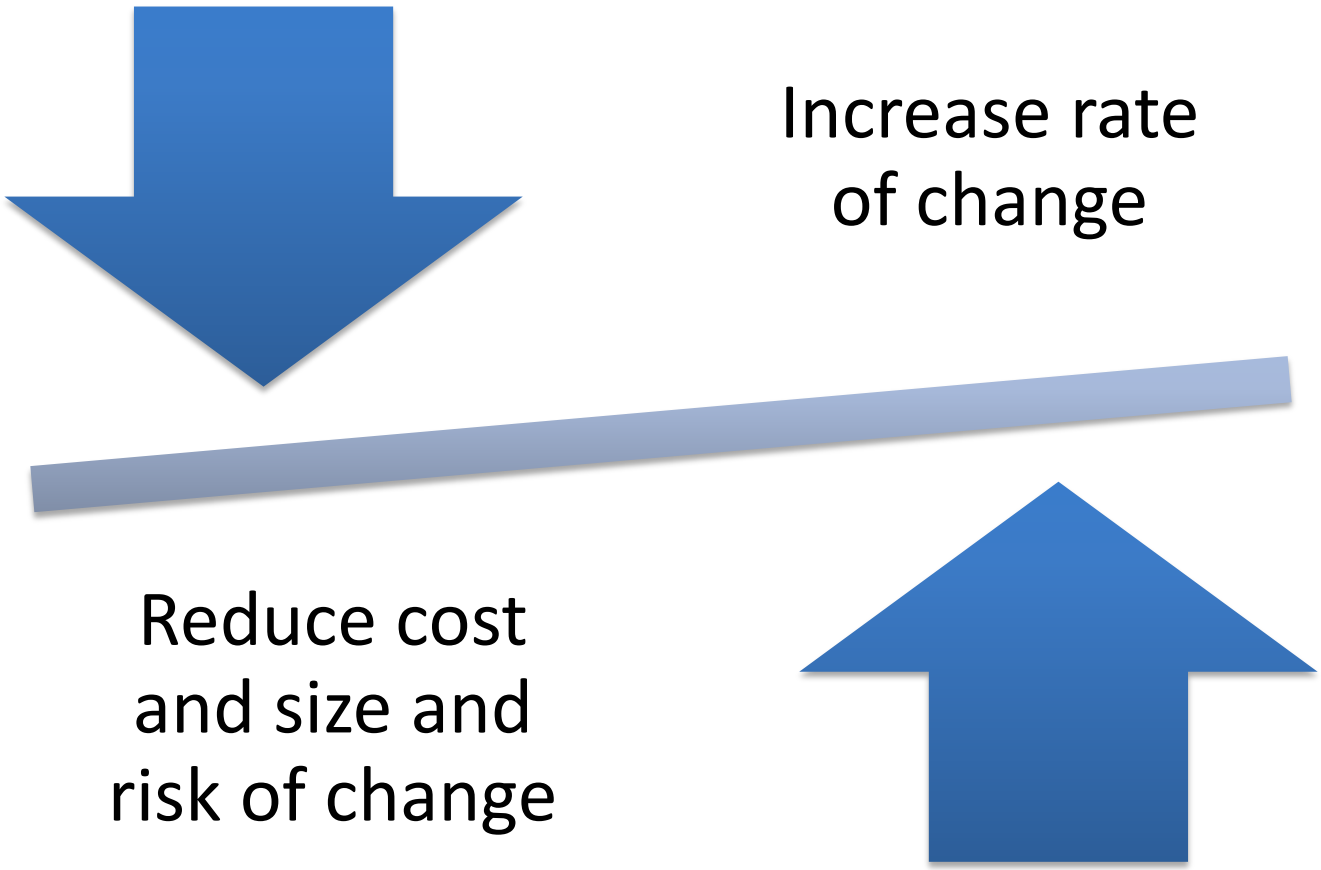
Speed Wins


Territory Expansion
Competitive Moves
Measure Sales
Observe
Customer Pain Point
Web Display Ads
Data Warehouse
Install Capacity
Act
2 week agile train/sprint
Orient
Capacity Estimate
Code Feature
Decide
2 Week Plan
Feature Priority
Business Buy-in

2 Week Train Model Hand-Off Steps



What's Next?


Increase rate
of change
Reduce cost
and size and
risk of change

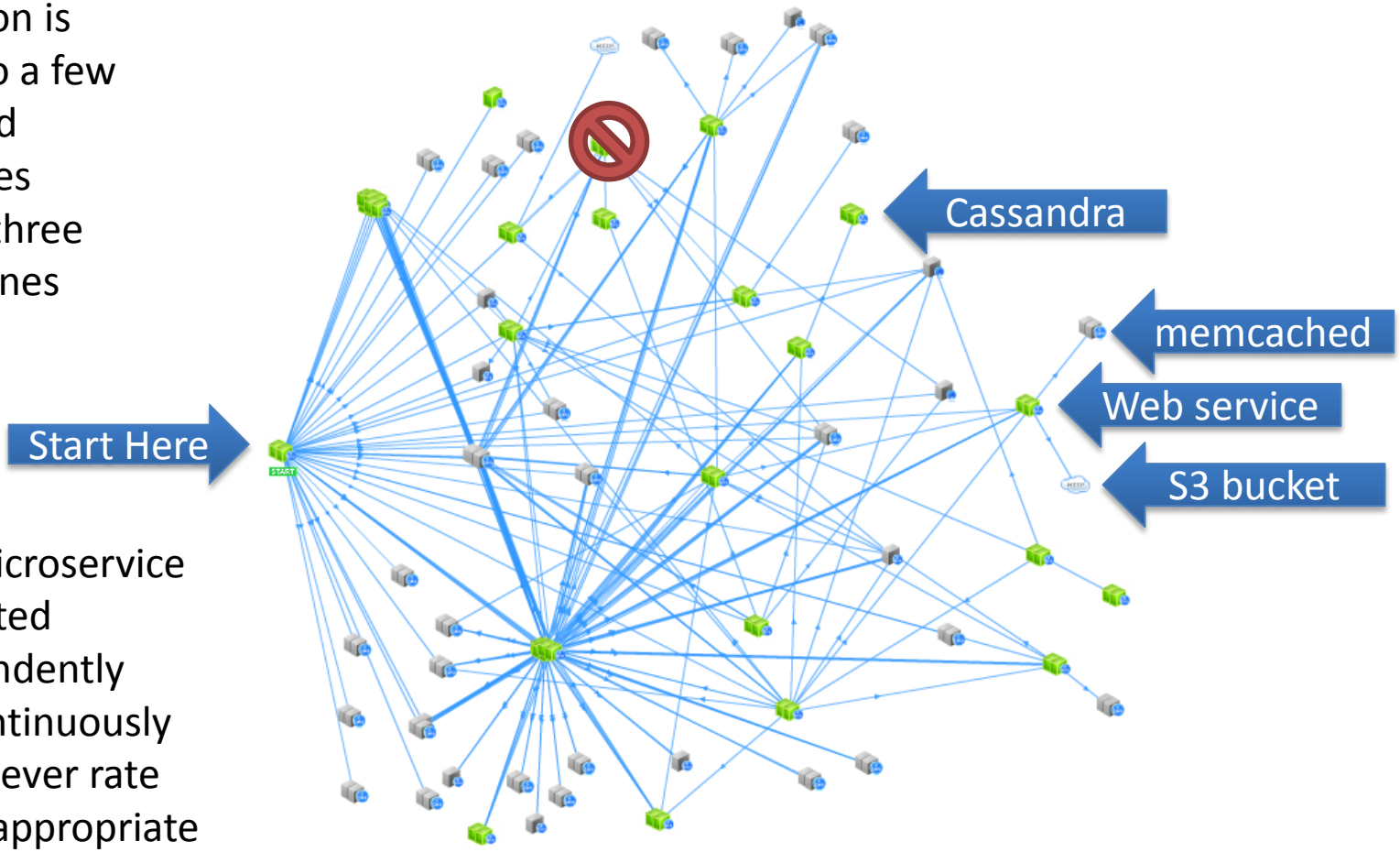
NETFLIX OSS

Cloud Native

Construct a highly agile and highly available service from ephemeral and assumed broken components

Cloud Native Microservices

Each icon is three to a few hundred instances across three AWS zones



Each microservice is updated independently and continuously at whatever rate seems appropriate

Continuous Deployment

No time for handoff to IT

NETFLIX OSS

Developer Self Service

Freedom and Responsibility

NETFLIX OSS

Developers run what they wrote

Root access *and* pagerduty

IT is a Cloud API

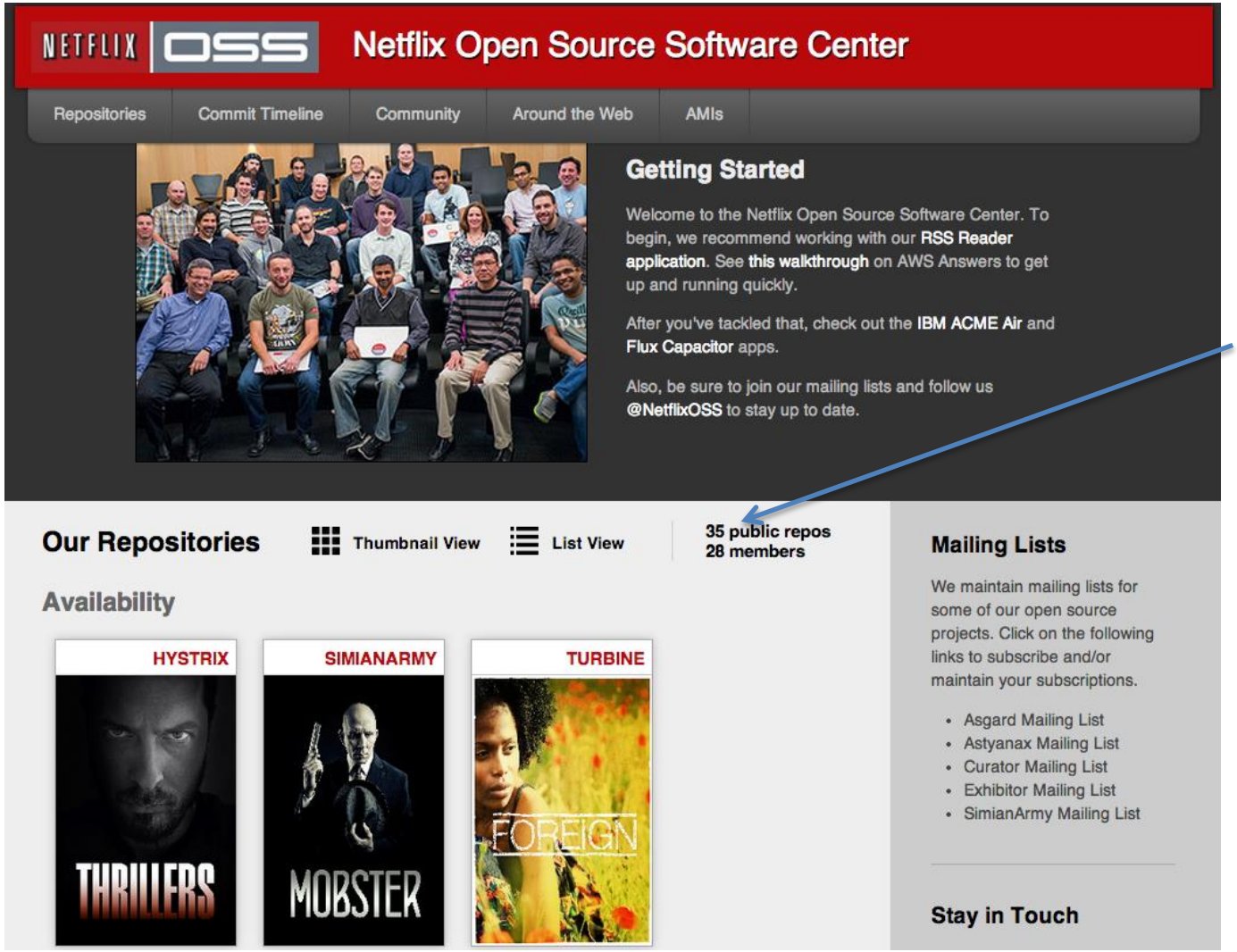
DEVops automation

NETFLIX OSS

Github all the things!

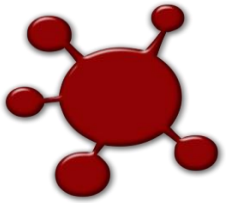
Leverage social coding

Netflix.github.com



~~35 repos~~
~~36 repos~~
37 today

Karyon Microservice Template



Bootstrapping

- Dependency & Lifecycle management via Governator.
- Service registry via Eureka.
- Property management via Archaius
- Hooks for Latency Monkey injection testing
- Preconfigured status page and heathcheck servlets

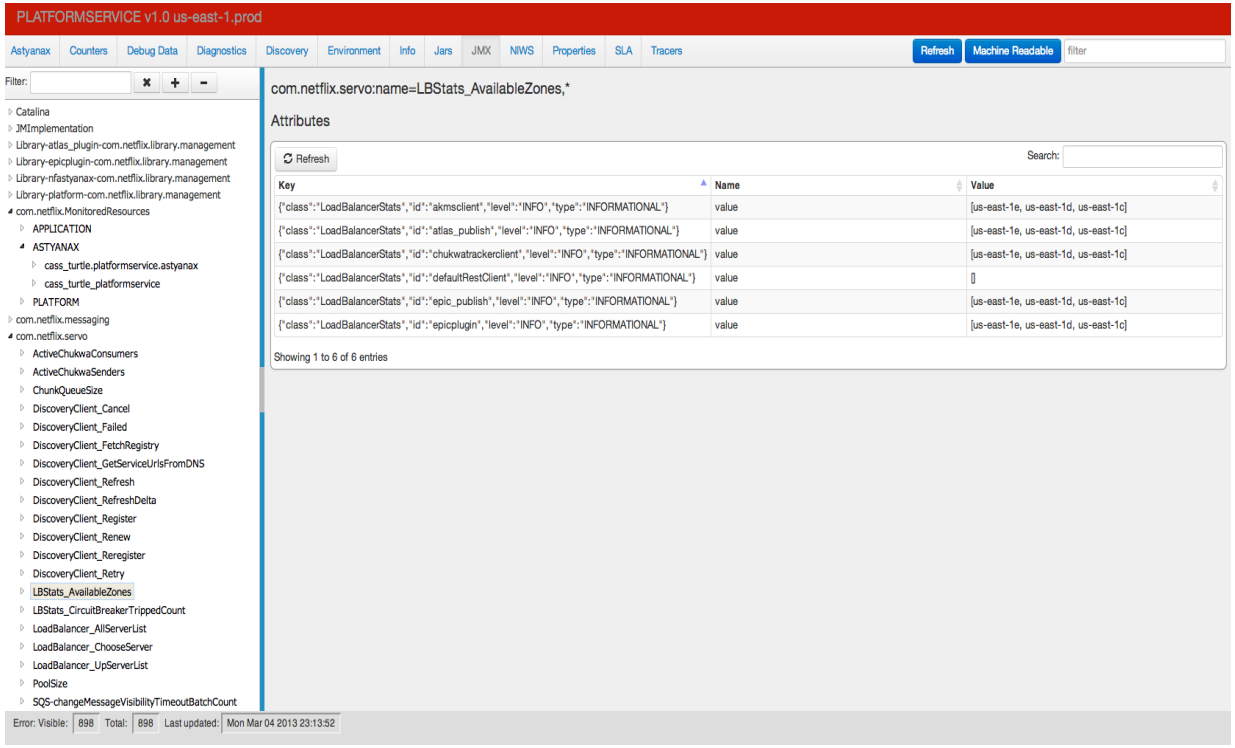
Karyon Microservice Status Page

Eureka discovery service metadata

Environment variables

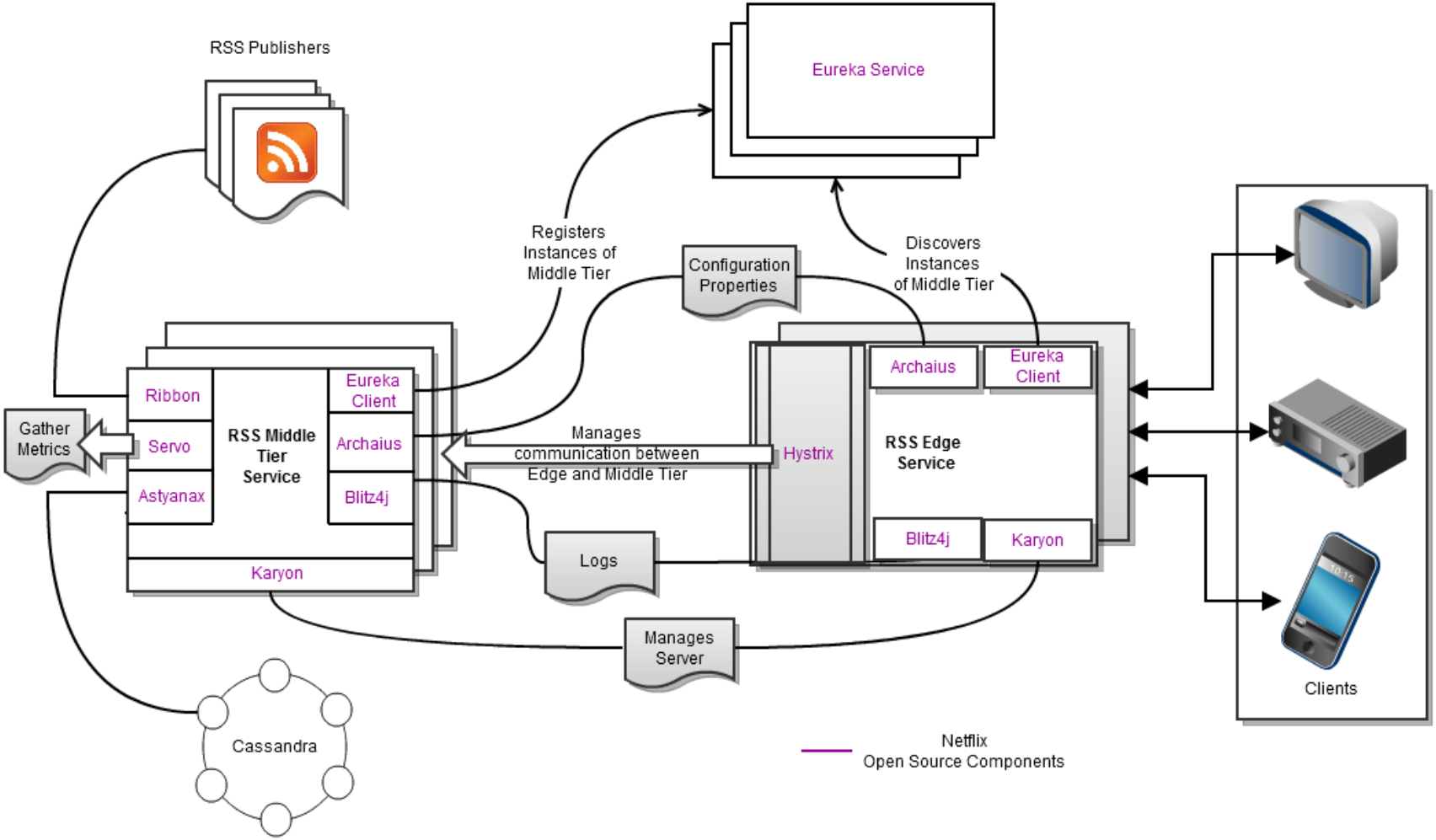
JMX

Versions

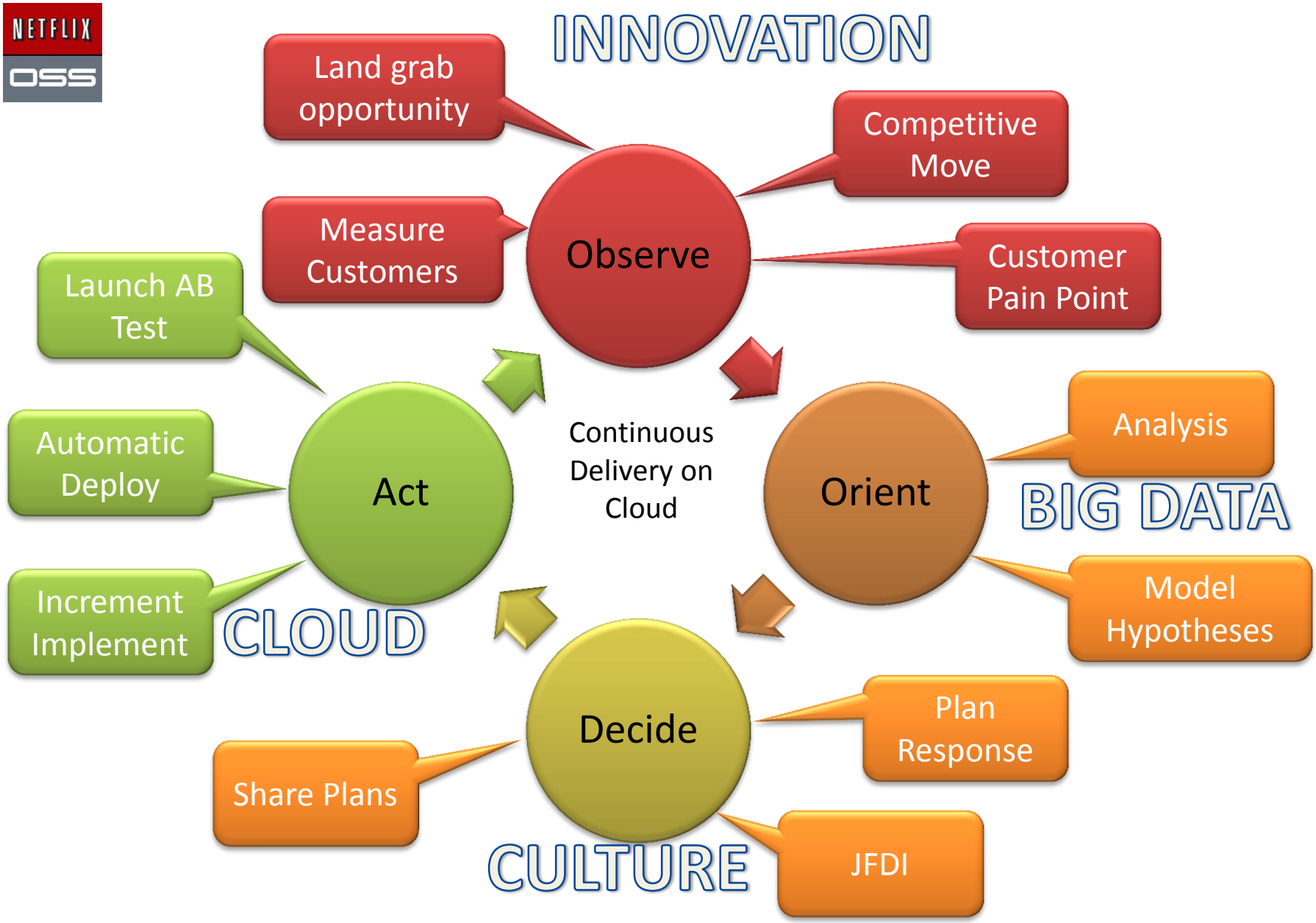


Conformity Monkey Support

Sample Application – RSS Reader



Putting it all together…


NETFLIX
OSS
INNOVATION
Land grab opportunity
Competitive Move
Measure Customers
Observe
Customer Pain Point
Launch AB Test
Automatic Deploy
Act
Continuous Delivery on Cloud
Orient
Analysis
BIG DATA
Model Hypotheses
Increment Implement
CLOUD
Decide
Plan Response
Share Plans
CULTURE
JFDI

Continuous Deploy Hand-Off

Product Manager - 2 days

A/B test setup and enable

Self service hypothesis test results

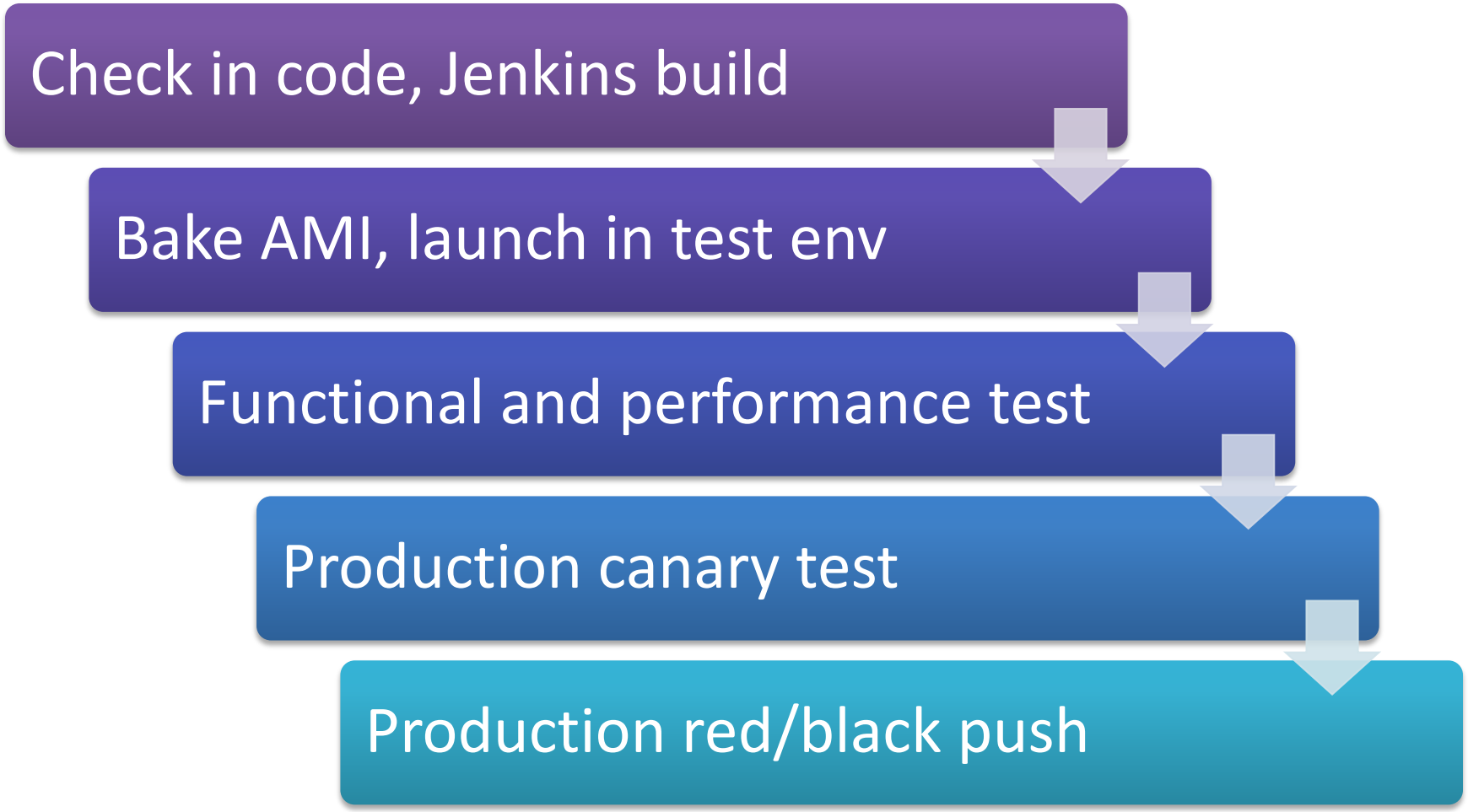
Developer – 2 days

Automated test

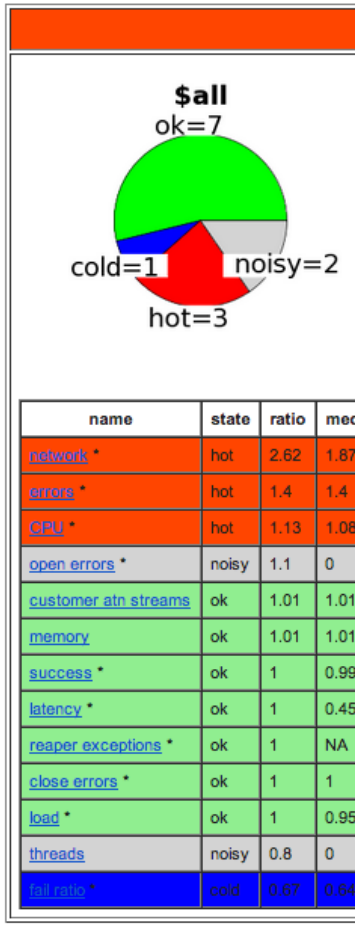
Automated deploy, on call

Self service analytics

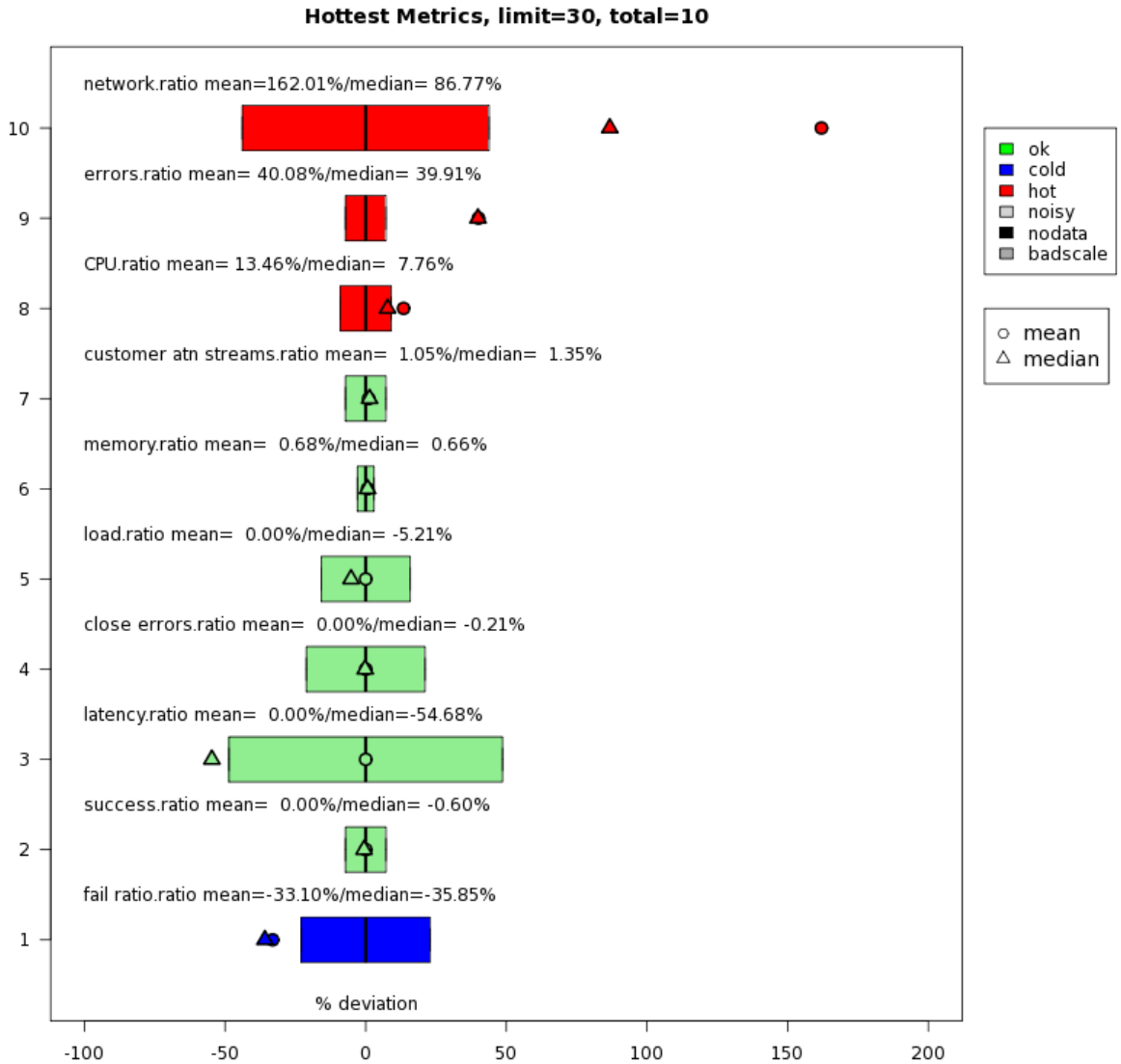
Continuous Deploy Automation



Bad Canary Signature

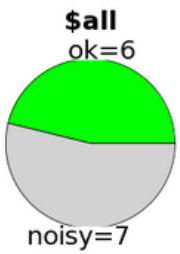


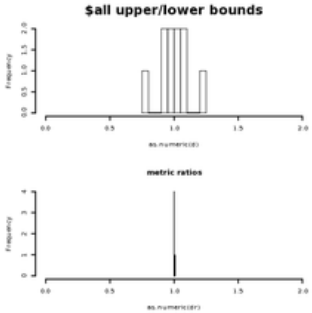
name	state	ratio	med
network *	hot	2.62	1.87
errors *	hot	1.4	1.4
CPU *	hot	1.13	1.08
open errors *	noisy	1.1	0
customer atn streams	ok	1.01	1.01
memory	ok	1.01	1.01
success *	ok	1	0.99
latency *	ok	1	0.45
reaper exceptions *	ok	1	NA
close errors *	ok	1	1
load *	ok	1	0.95
threads	noisy	0.8	0
fail ratio *	cold	0.67	0.64

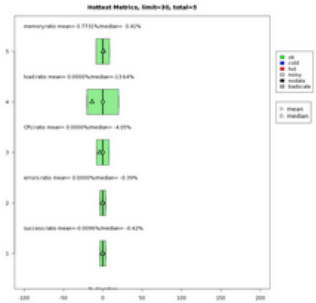


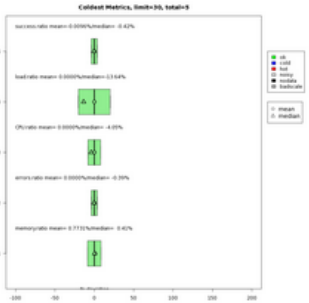
Happy Canary Signature

$all: state=noisy (98%)



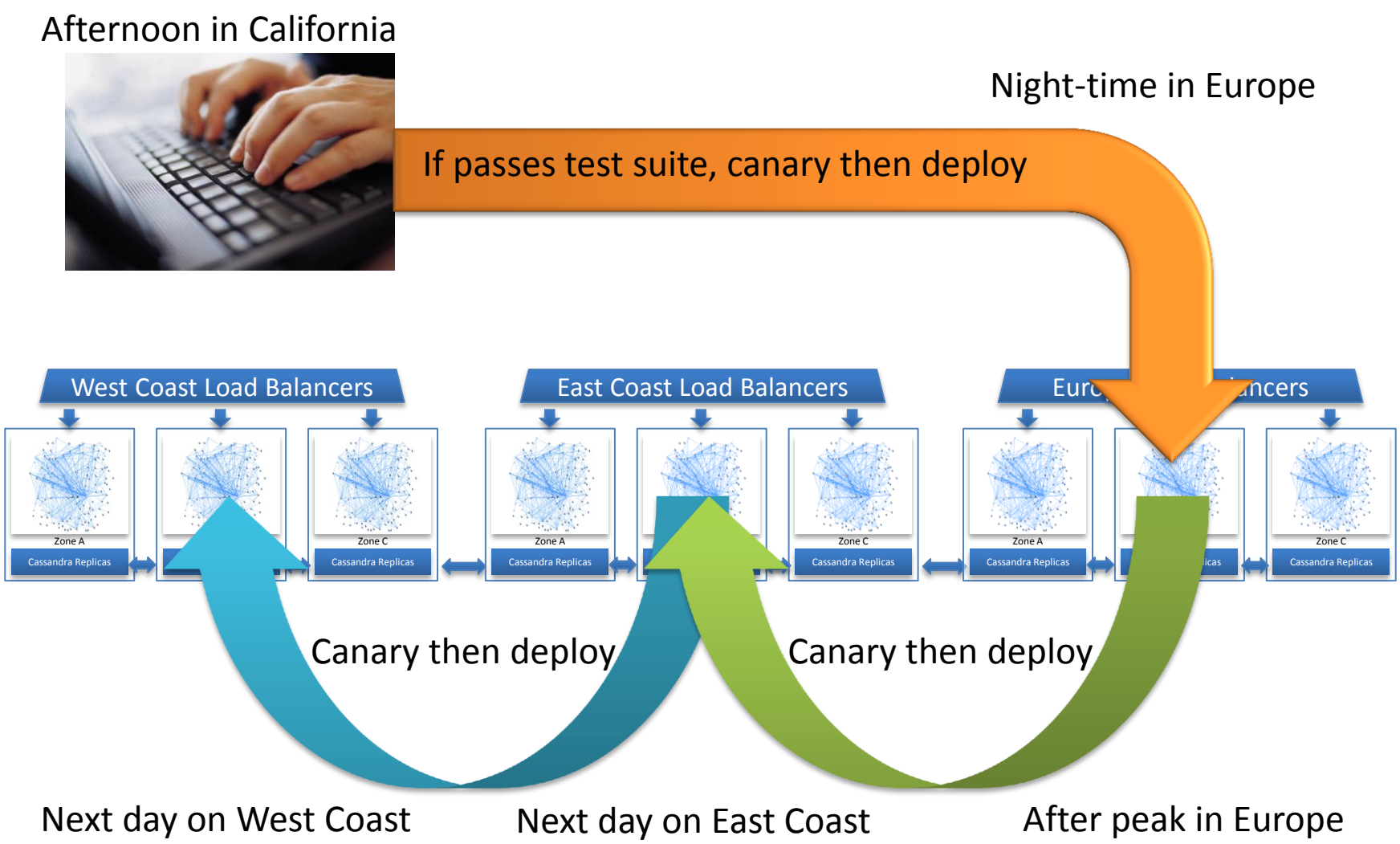






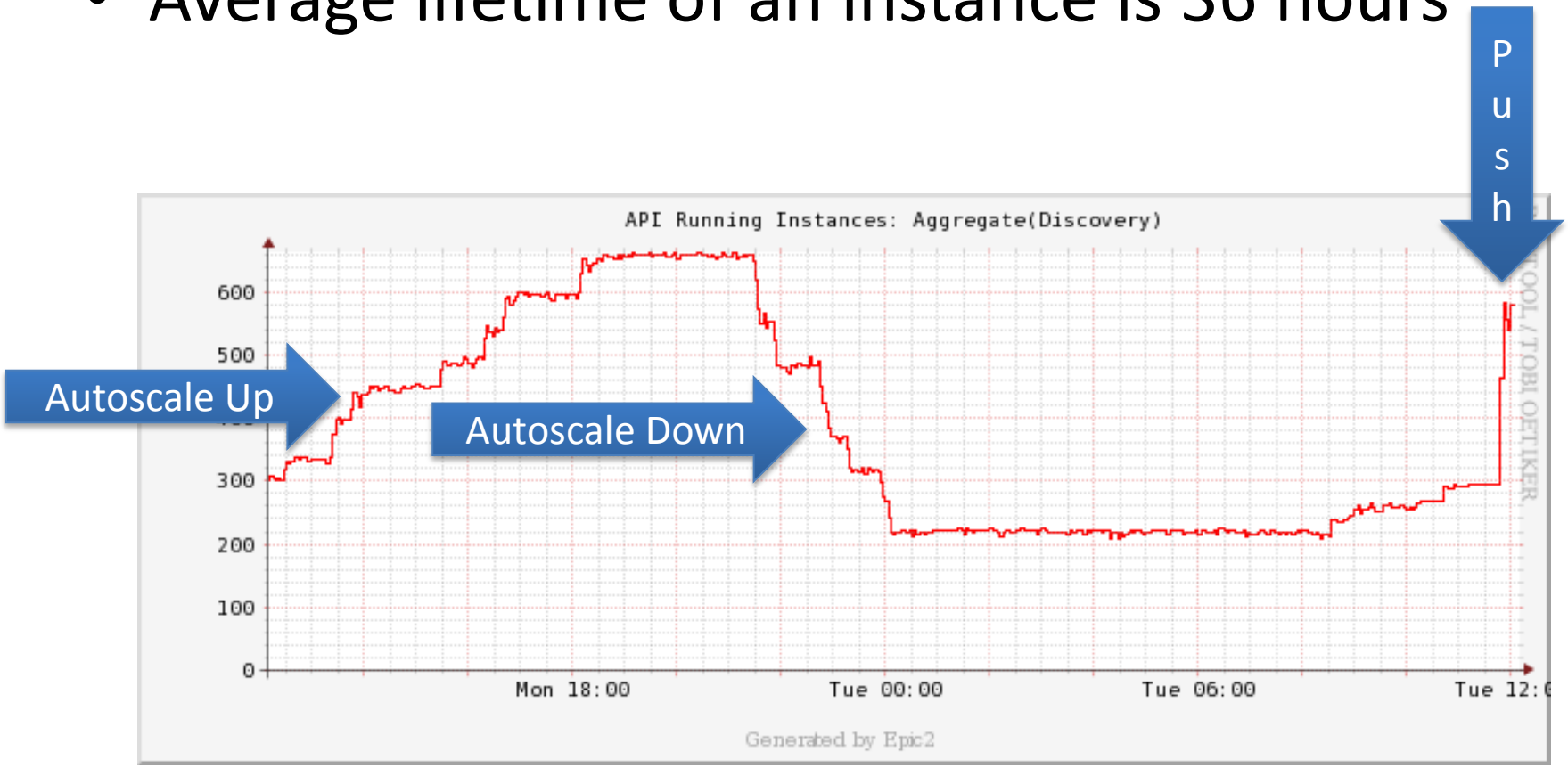
name	state	ratio	median	lower	upper
close errors *	noisy	1.19	0	0	6.13
failure *	noisy	1.18	0	0	6.14
fail ratio *	noisy	1.15	0	0	7.05
memory	ok	1.01	1	0.91	1.09
errors *	ok	1	1	0.96	1.04
latency *	noisy	1	0.59	0.31	1.69
customer atn streams *	ok	1	NA	1	1
CPU *	ok	1	0.96	0.92	1.08
load *	ok	1	0.86	0.8	1.2
network *	noisy	1	0.21	0	2.34
success	ok	1	1	0.96	1.04
threads	noisy	0.83	0.84	0.29	1.71
recovered	noisy	0.75	0	0	2.54

Global Deploy Automation



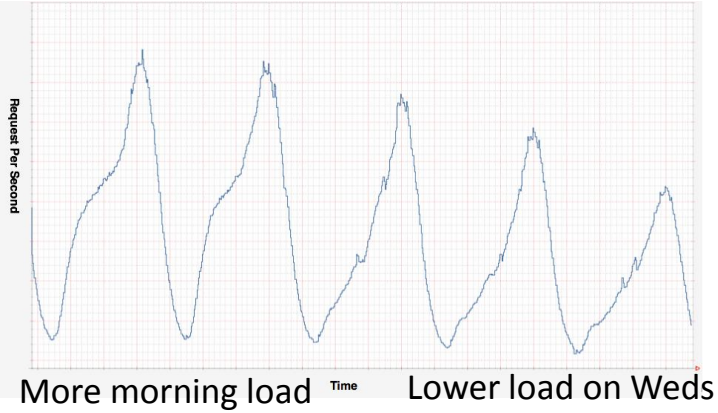
Ephemeral Instances

- Largest services are autoscaled
- Average lifetime of an instance is 36 hours



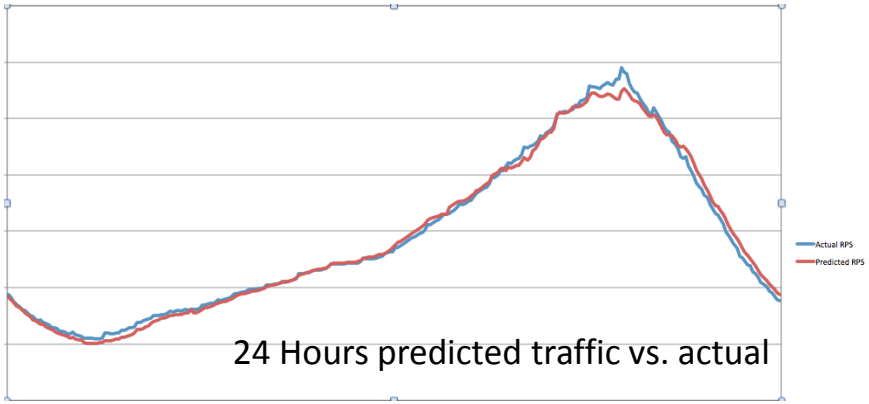
Scryer - Predictive Auto-scaling

See techblog.netflix.com

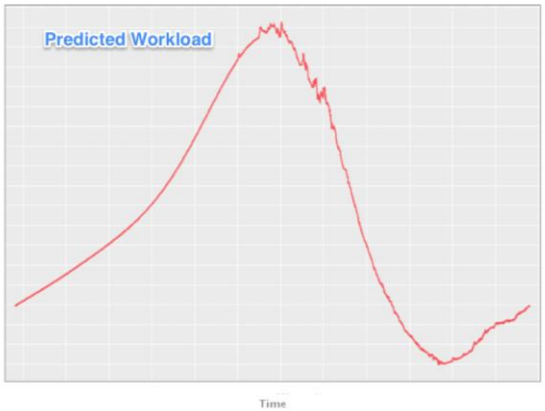


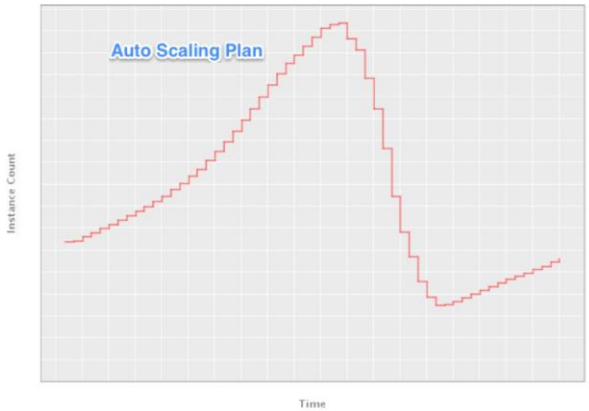
More morning load
Sat/Sun high traffic

Lower load on Weds



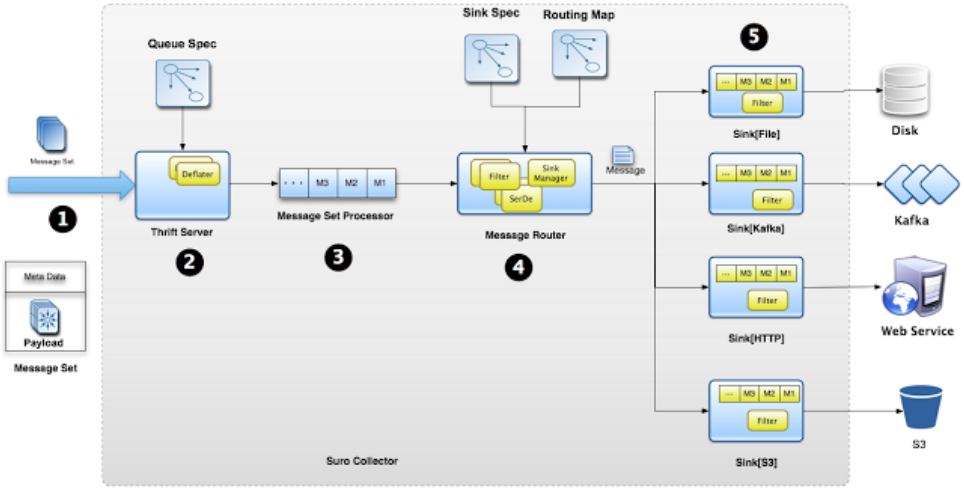
24 Hours predicted traffic vs. actual





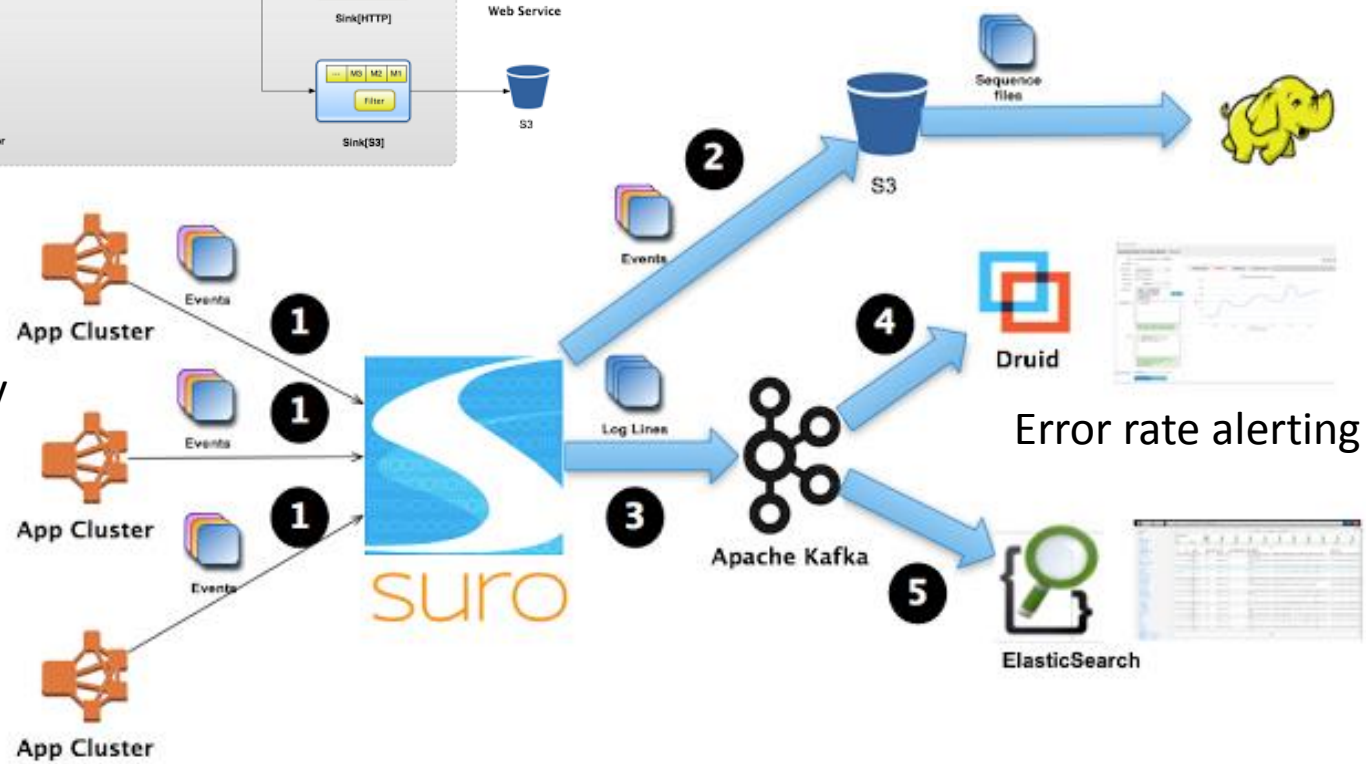
FFT based prediction driving AWS Autoscaler to plan minimum capacity

Suro Event Pipeline



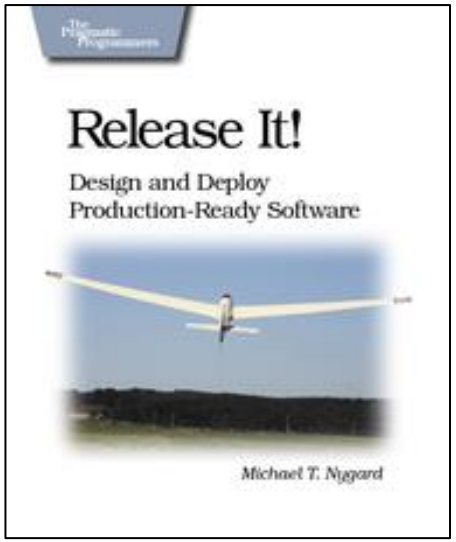
Cloud native, dynamic, configurable offline and realtime data sinks

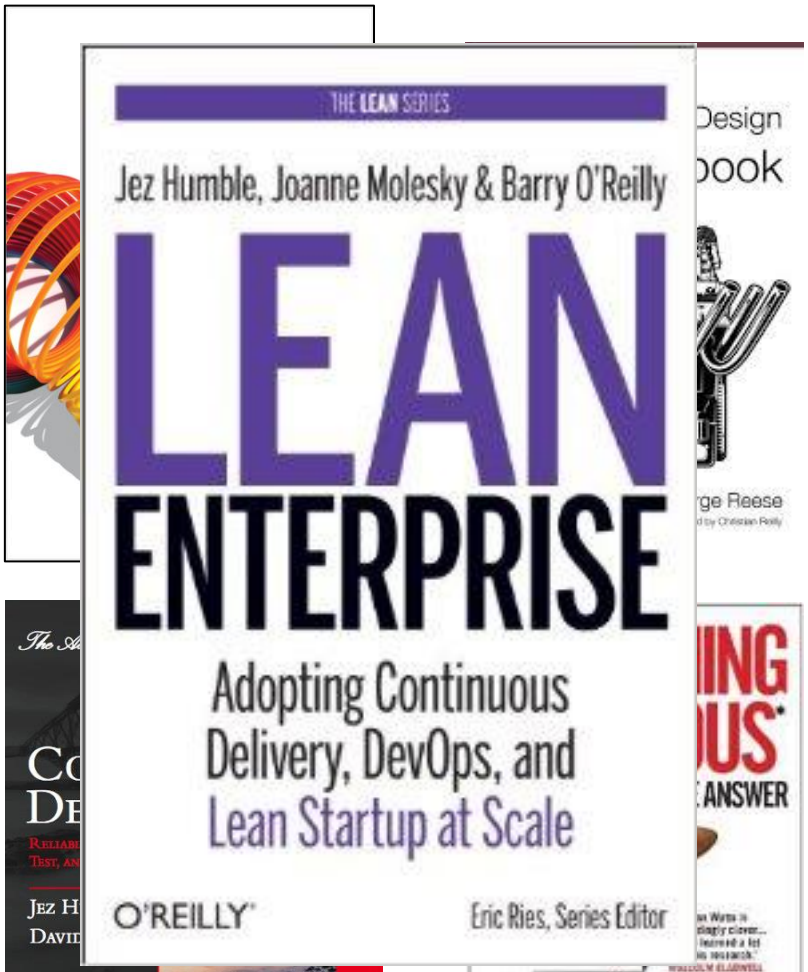
1.5 Million events/s
80 Billion events/day

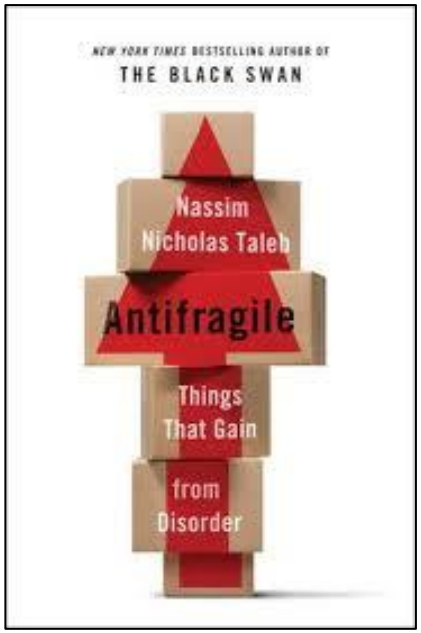


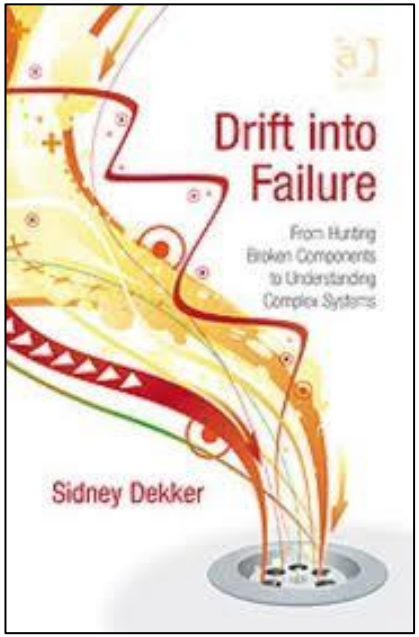
Error rate alerting

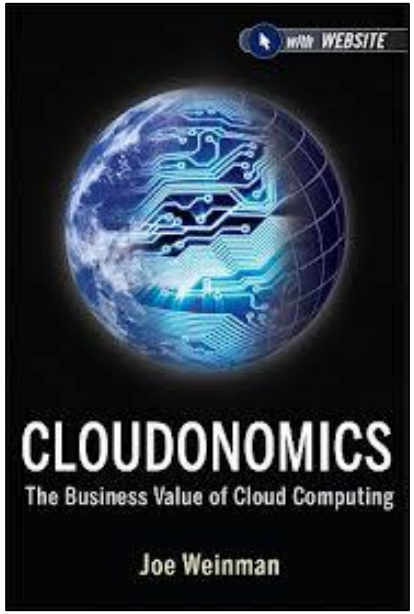
Inspiration











Principles Revisited:

Immutability

Separation of Concerns

Anti-fragility

High trust organization

Sharing

Takeaway

Speed Wins

Assume Broken

Cloud Native Automation

Github is the "app store" and resumé

@adrianco @NetflixOSS
http://netflix.github.com