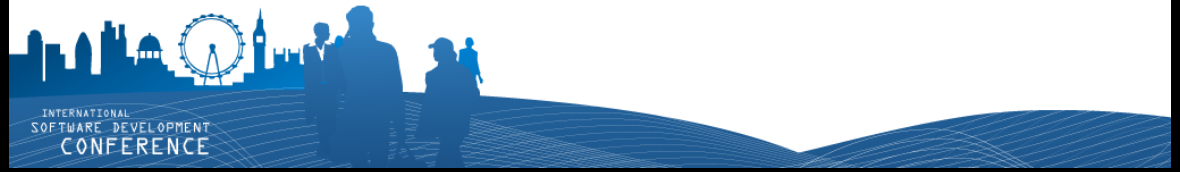
# How Netflix Leverages Multiple Regions to Increase Availability

Ruslan Meshenberg

10 März 2014

QCon

Please evaluate my talk via the mobile app!



# Netflix


NETFLIX
Visitor Parking


NETFLIX
HOUSE of CARDS
ORANGE is the BLACK

# Who am I?

- Director, Platform Engineering
- Led Multi-Regional Resiliency
- Leading NetflixOSS Program

- @rusmeshenberg

# Failure

fail·ure

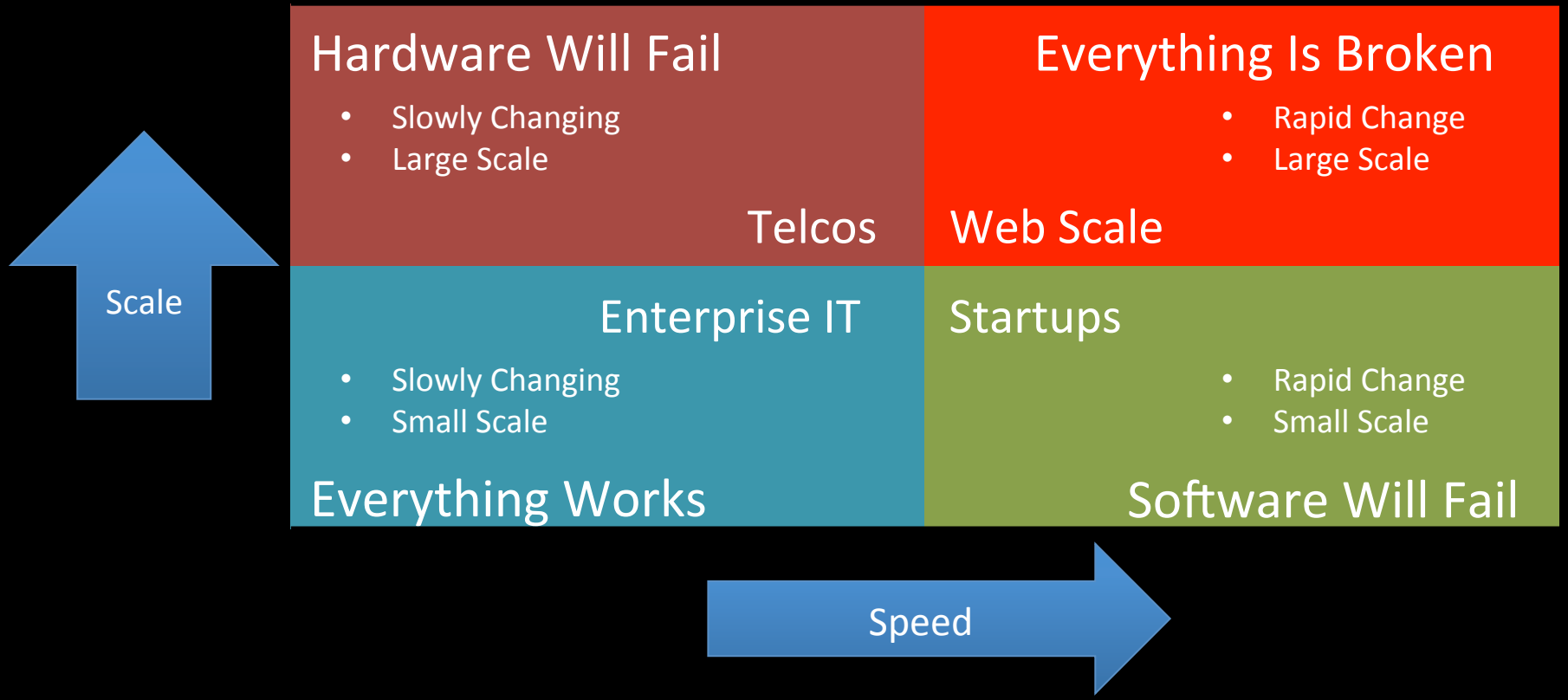
/ˈfāl(y)ər/

*noun*

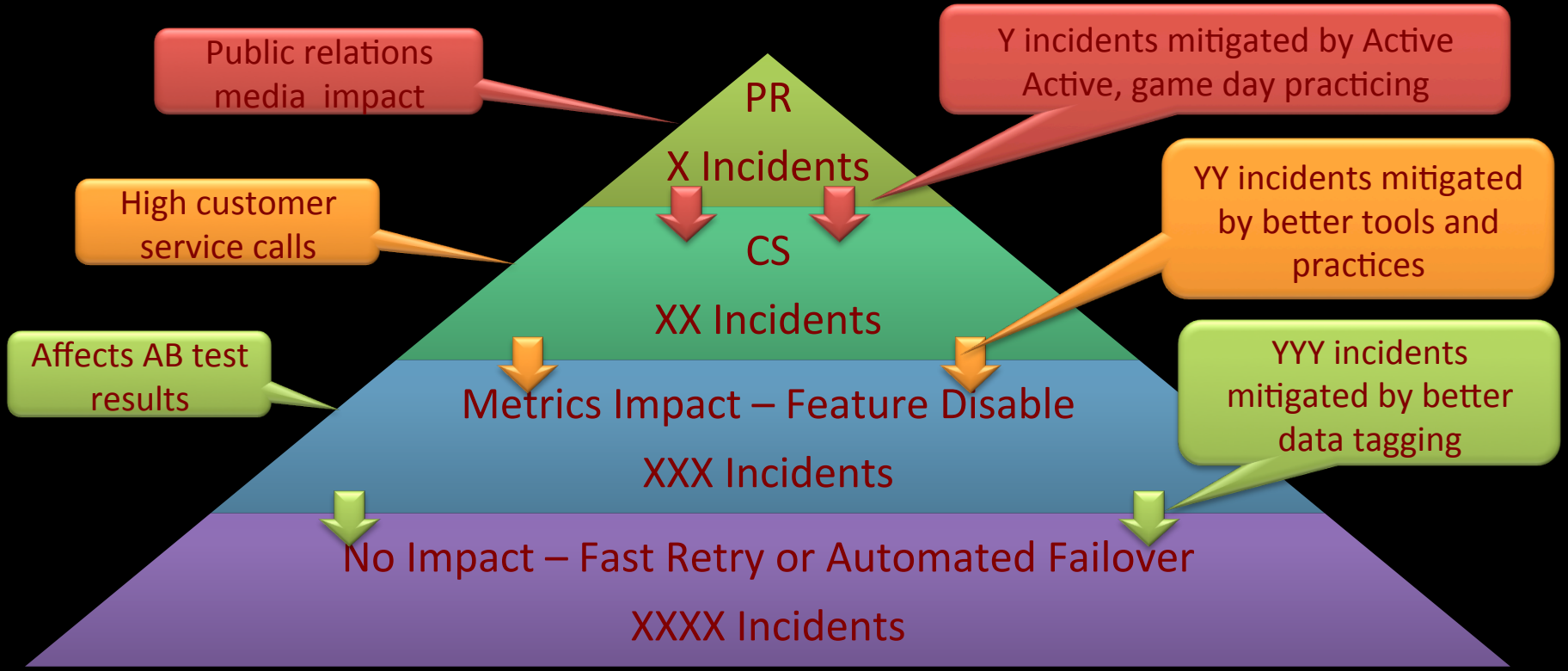
1. lack of succes
   "an economic p y that is doo failure"
   *synonyms:* lack o cess, nonf ment, defeat, collapse, foundering

2. the omission of expected or required action.
   "their failure to comply with the basic rules"

# Assumptions



# Incidents – Impact and Mitigation



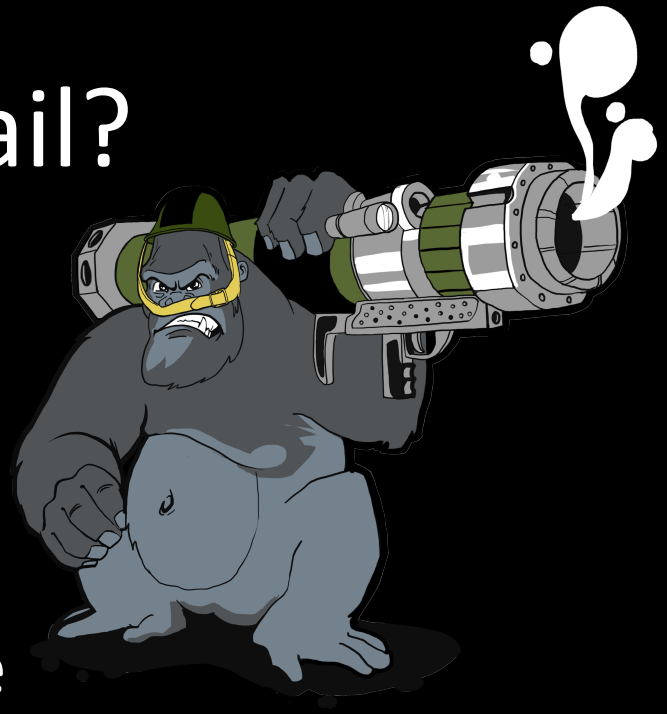
# Does an Instance Fail?

- It can, plan for it

- Bad code / configuration pushes
- Latent issues
- Hardware failure

- Test with Chaos Monkey

# Does a Zone Fail?



- Rarely, but happened before

- Routing issues
- DC-specific issues
- App-specific issues within a zone

- Test with Chaos Gorilla

# Does a Region Fail?

- Full region – unlikely, very rare
- Individual Services can fail region-wide
- Most likely, a region-wide configuration issue

- Test with Chaos Kong

# Everything Fails… Eventually

- Keep your services running by embracing isolation and redundancy
- Construct a highly agile and highly available service from ephemeral and assumed broken components

# Isolation

- Changes in one region should not affect others
- Regional outage should not affect others
- Network partitioning between regions should not affect functionality / operations

# Redundancy

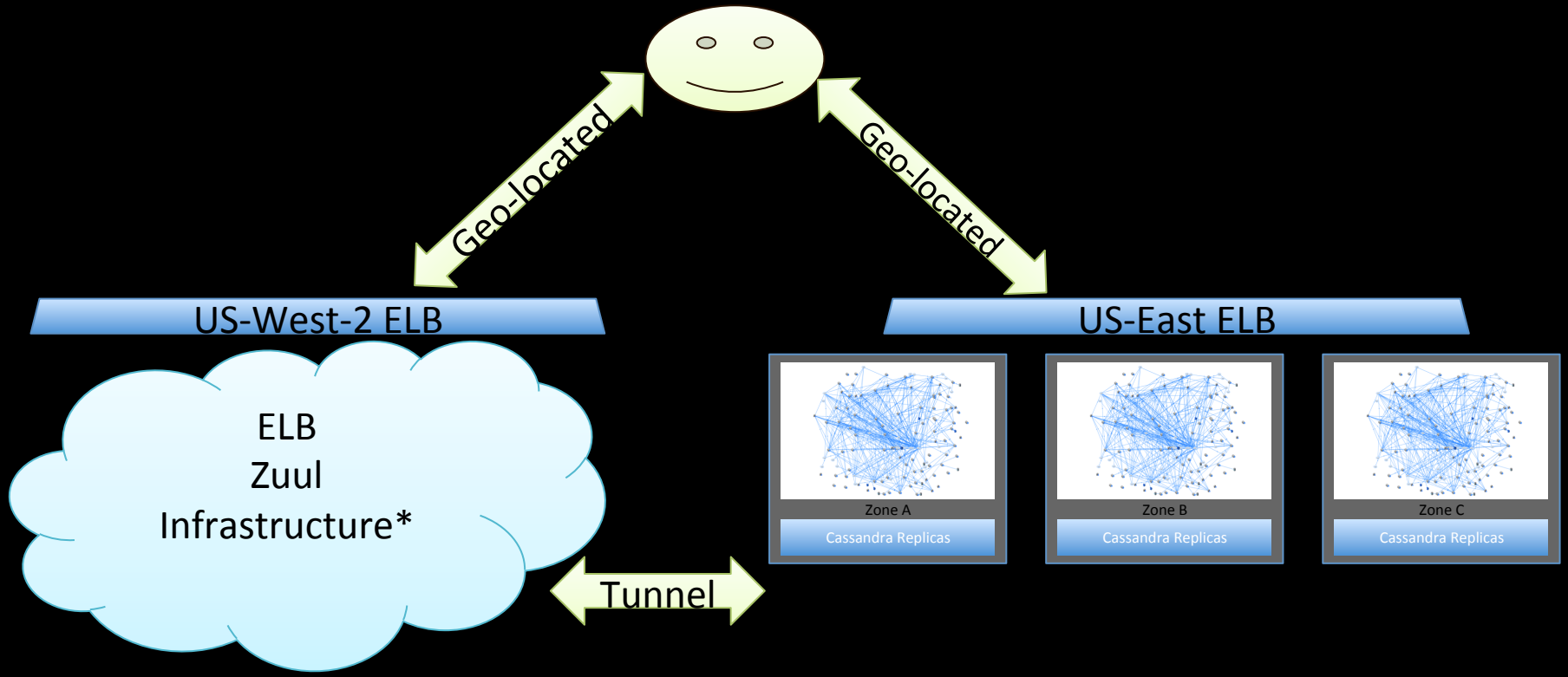
- Make more than one (of pretty much everything)
- Specifically, distribute services across Availability Zones and regions

# History: X-mas Eve 2012

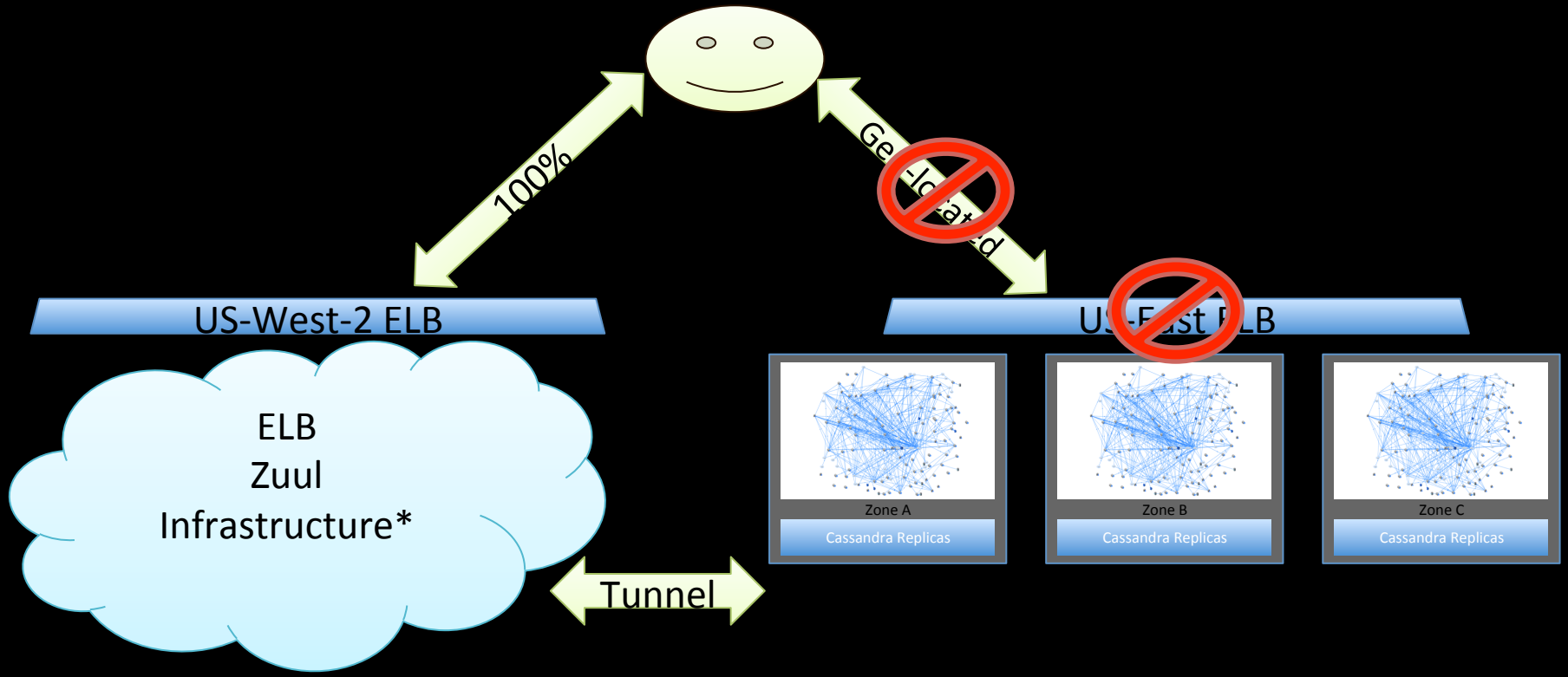
- Netflix multi-hour outage
- US-East1 regional Elastic Load Balancing issue

- *"...data was deleted by a maintenance process that was inadvertently run against the production ELB state data"*

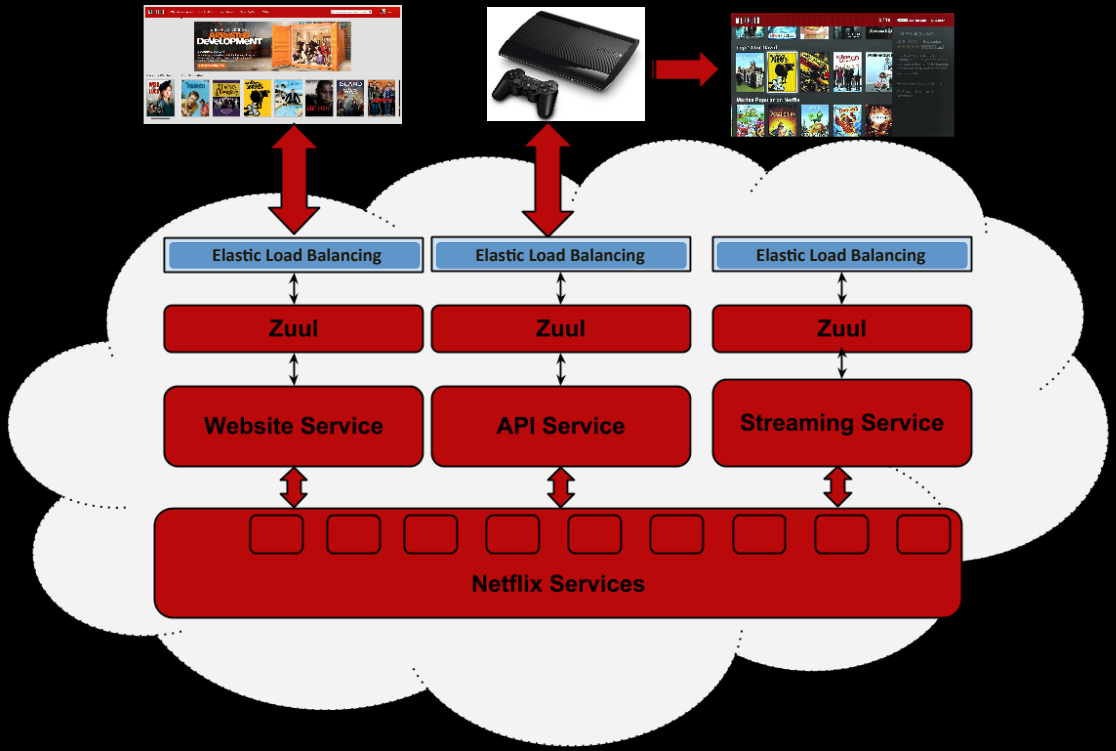
# Isthmus – Normal Operation



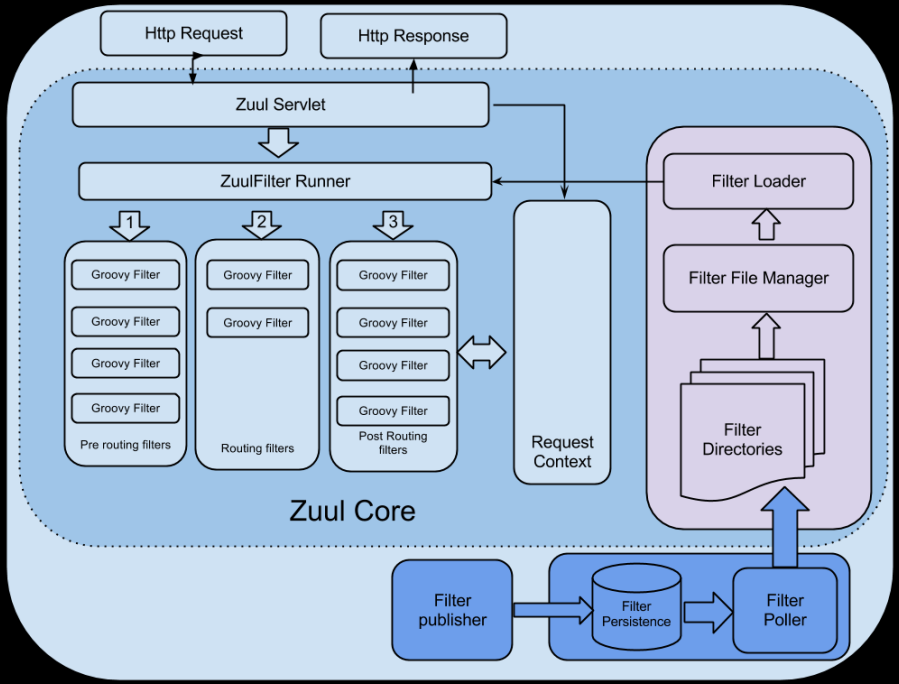
# Isthmus – Failover



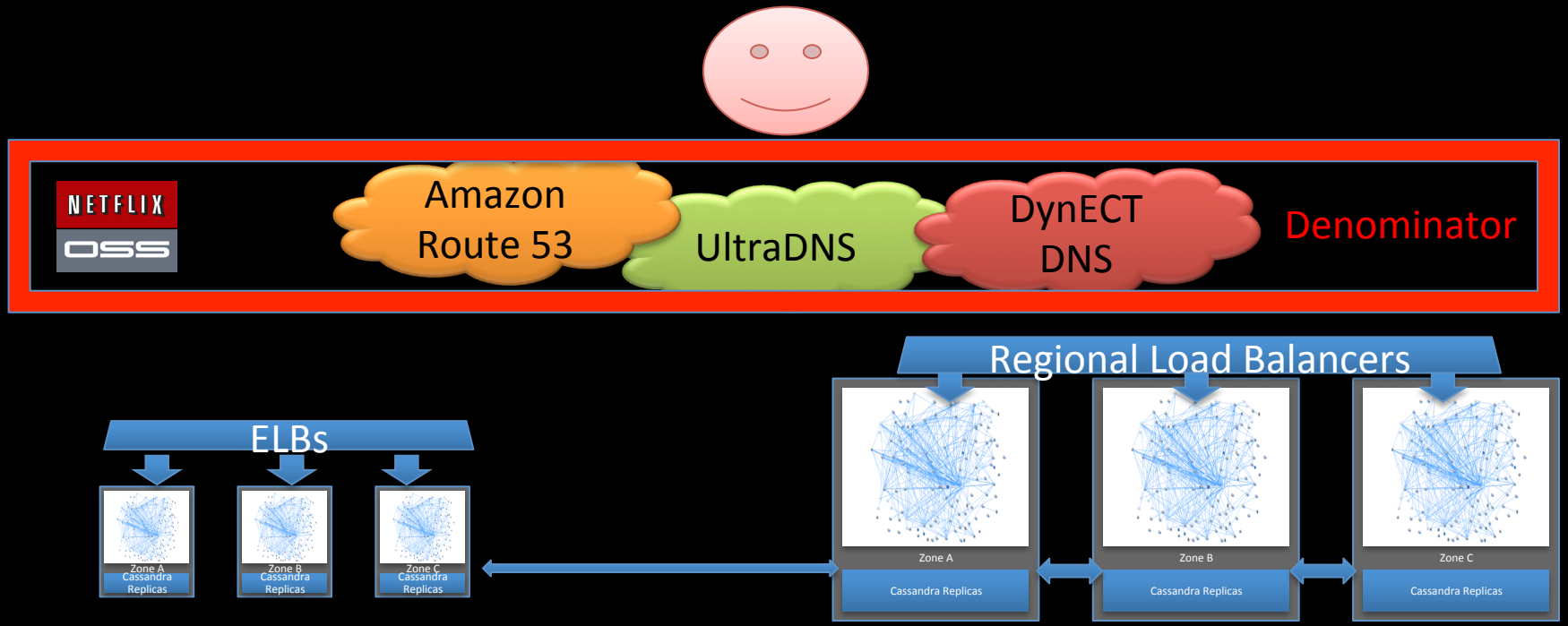
# Zuul Overview



# Zuul


Http Request
Http Response
Zuul Servlet
ZuulFilter Runner
1
2
3
Groovy Filter
Groovy Filter
Groovy Filter
Groovy Filter
Pre routing filters
Groovy Filter
Groovy Filter
Routing filters
Groovy Filter
Groovy Filter
Groovy Filter
Groovy Filter
Post Routing filters
Request Context
Filter Loader
Filter File Manager
Filter Directories
Zuul Core
Filter publisher
Filter Persistence
Filter Poller

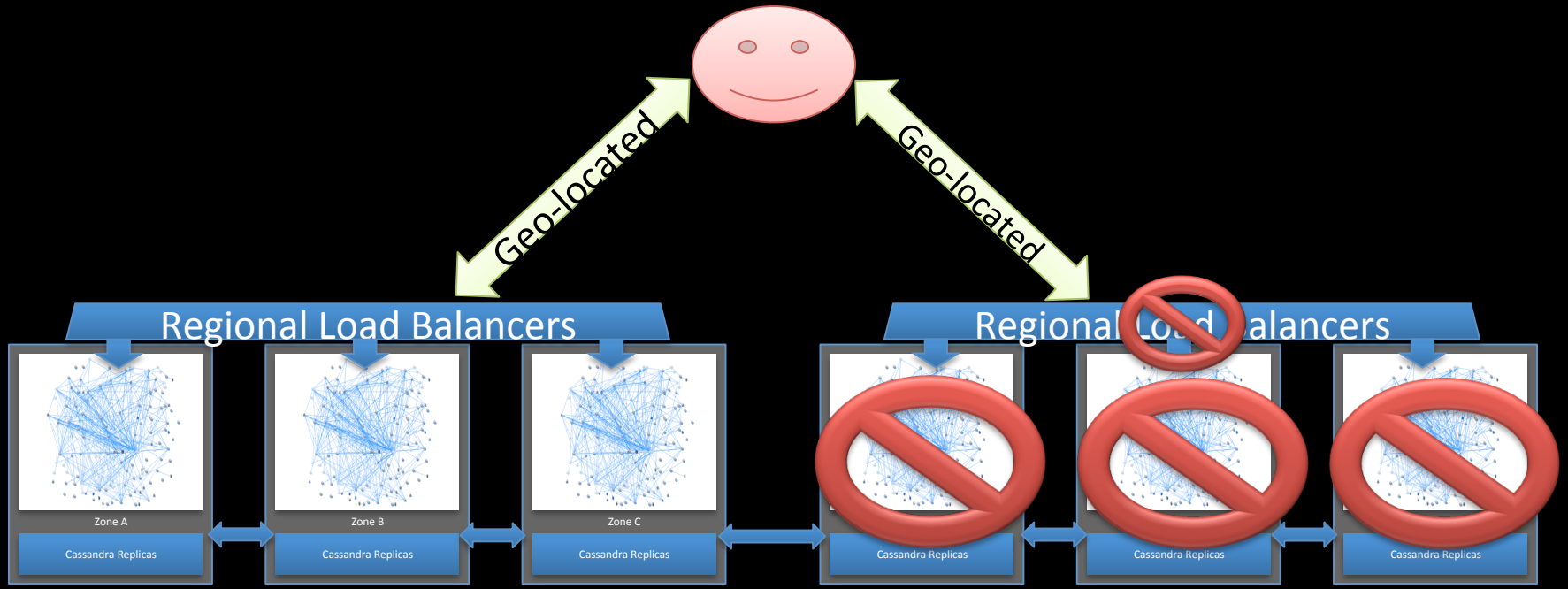
# Denominator – Abstracting the DNS Layer



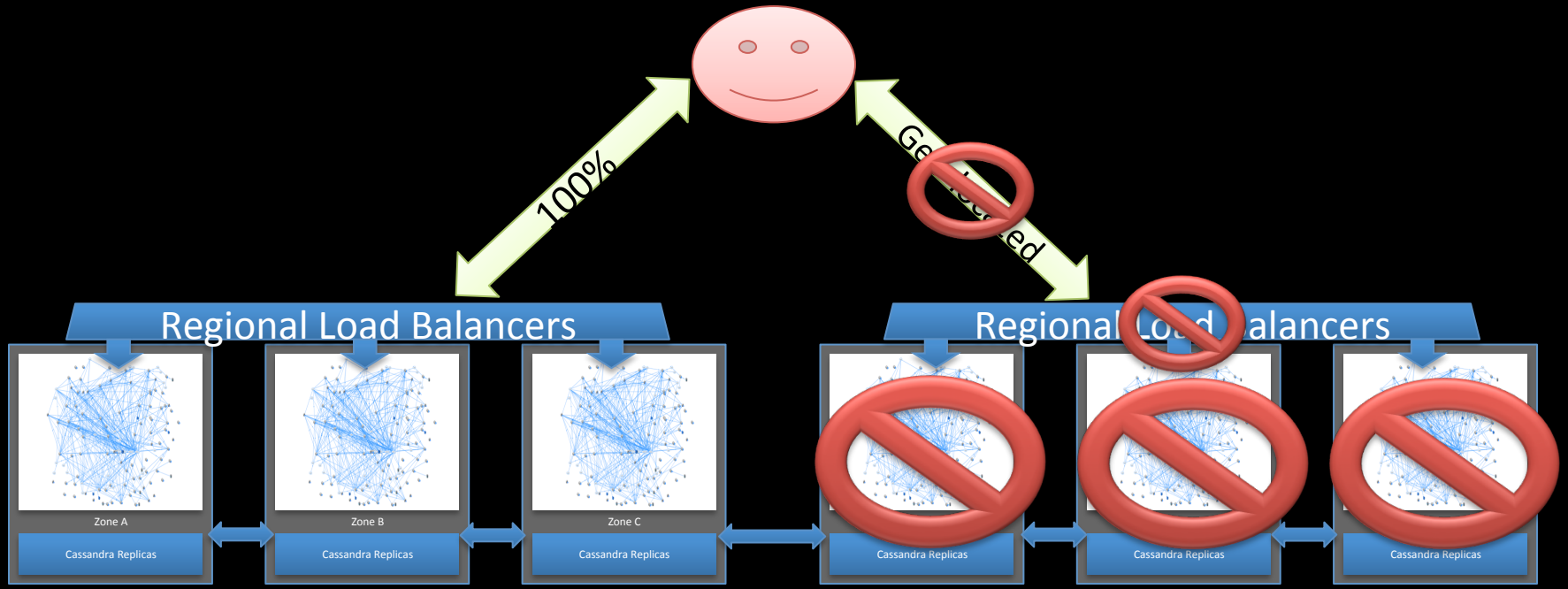
# Isthmus – Only for ELB Failures

- Other services may fail region-wide
- Not worthwhile to develop one-offs for each one

# Active-Active – Full Regional Resiliency

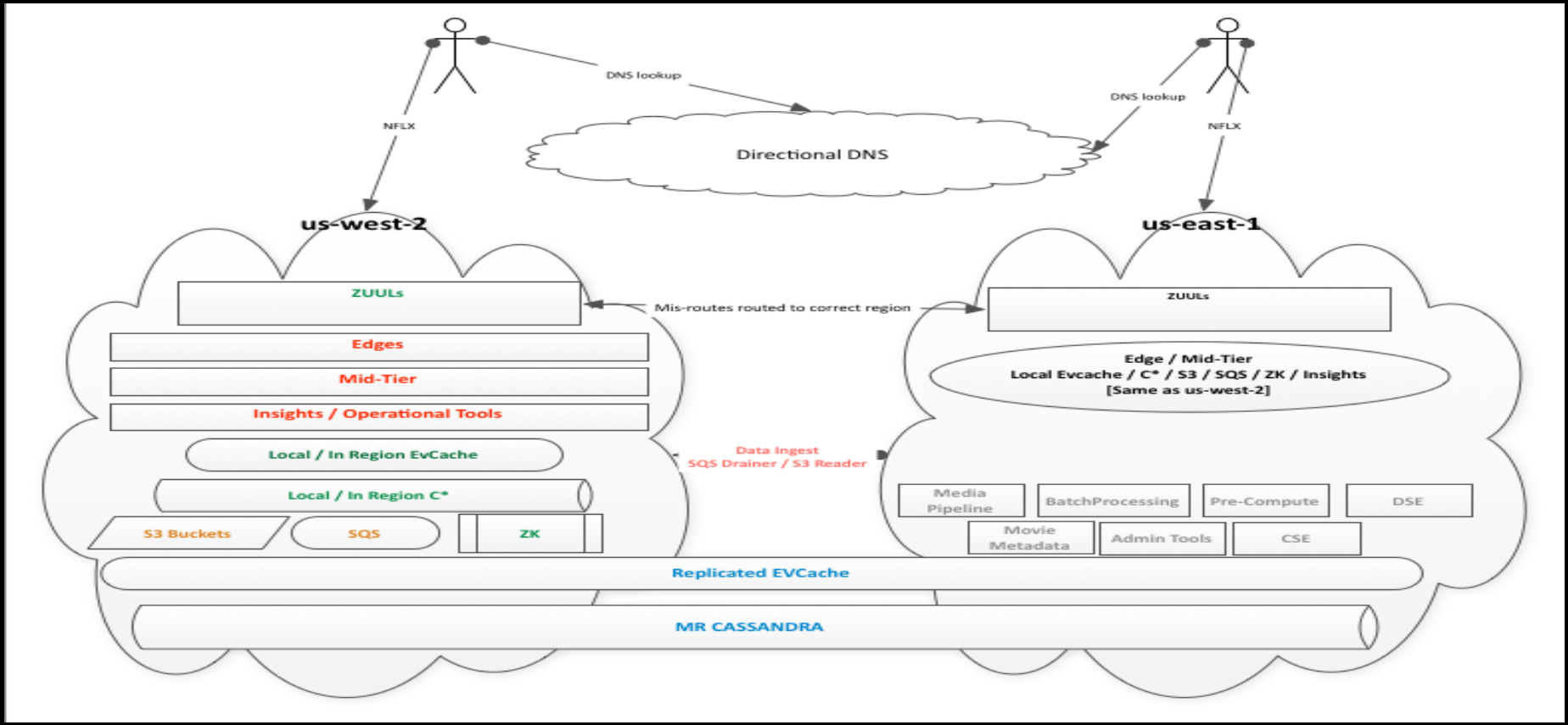


# Active-Active – Failover



# Can’t we just deploy in 2 regions?

# Active-Active Architecture



# 2 main challenges

- Routing the users to the services
- Replicating the data

# Separating the Data – Eventual Consistency

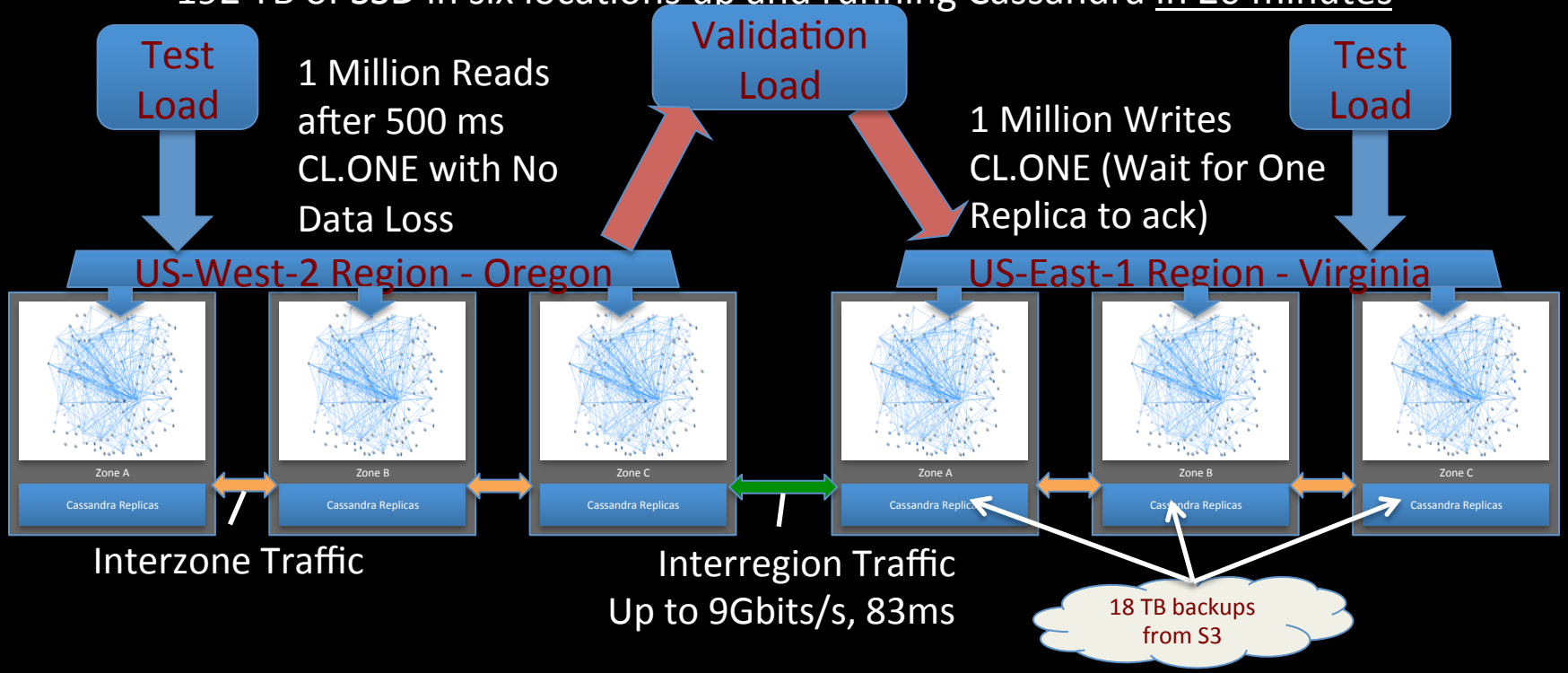
- 2–4 region Cassandra clusters
- Eventual consistency != hopeful consistency

# Benchmarking Global Cassandra

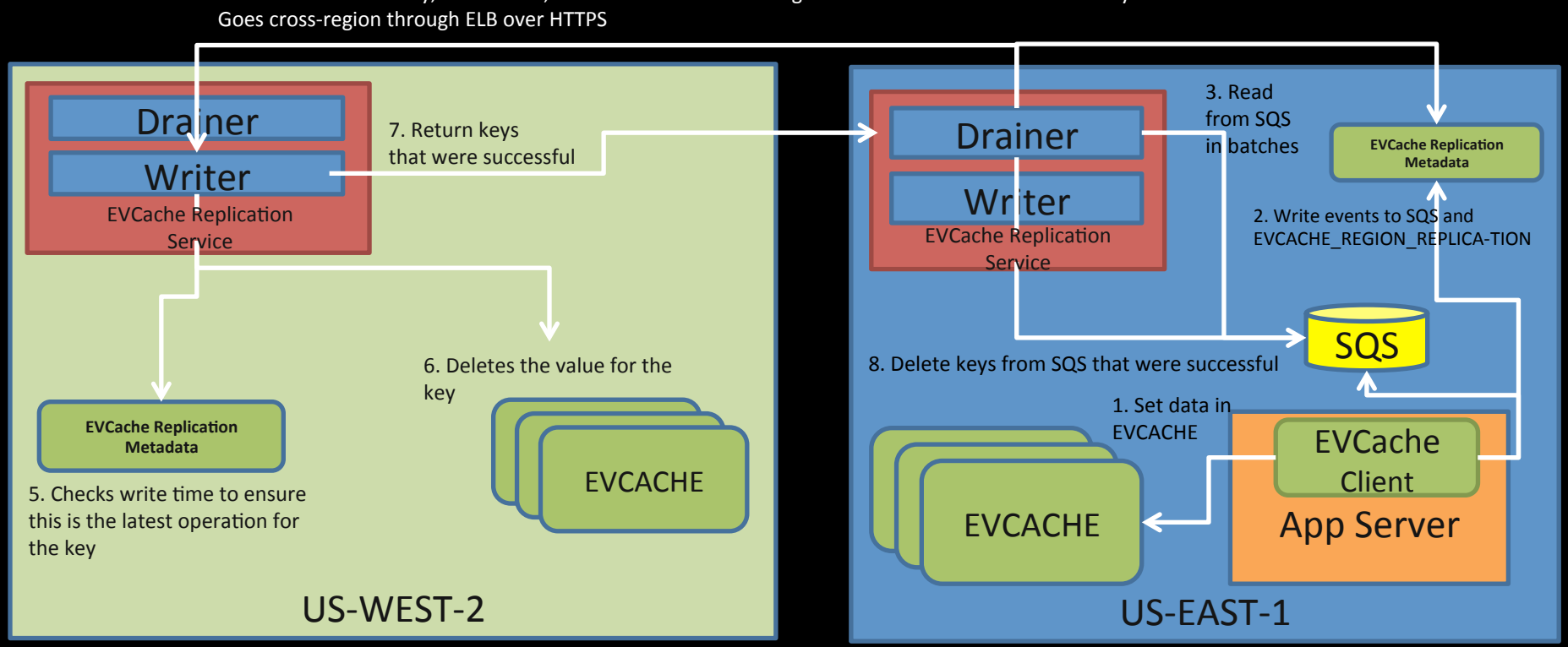
Write intensive test of cross-region replication capacity

16 x hi1.4xlarge SSD nodes per zone = 96 total

192 TB of SSD in six locations up and running Cassandra in 20 minutes

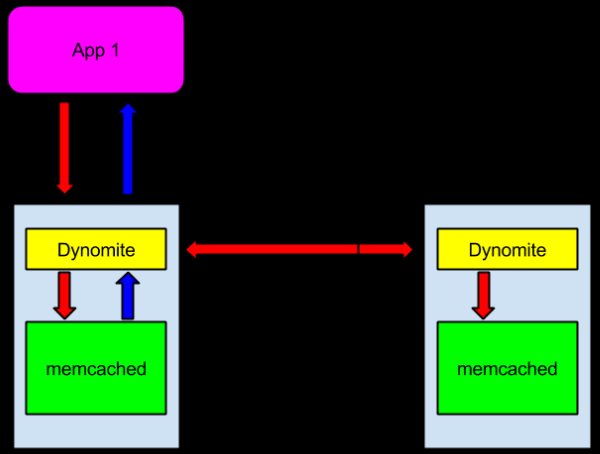


# Propagating EVCache Invalidations

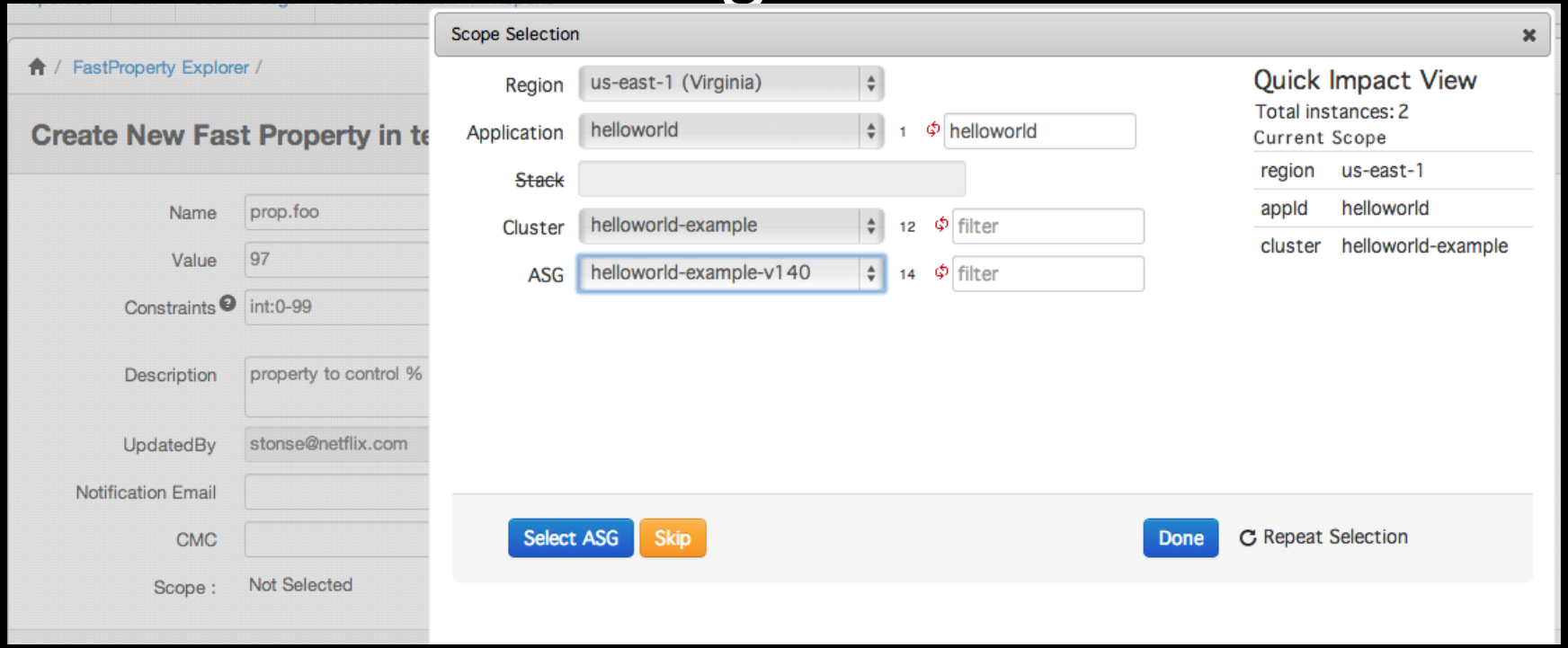


# Announcing Dinomyte

- Cross AZ & Region replication to existing Key Value stores
  - Memcached
  - Redis
- Thin Dynamo implementation for replication
- Keep native protocol
  - No code refactoring
- OSS soon!



# Archaius – Region-isolated Configuration



# Running Isthmus and Active-Active

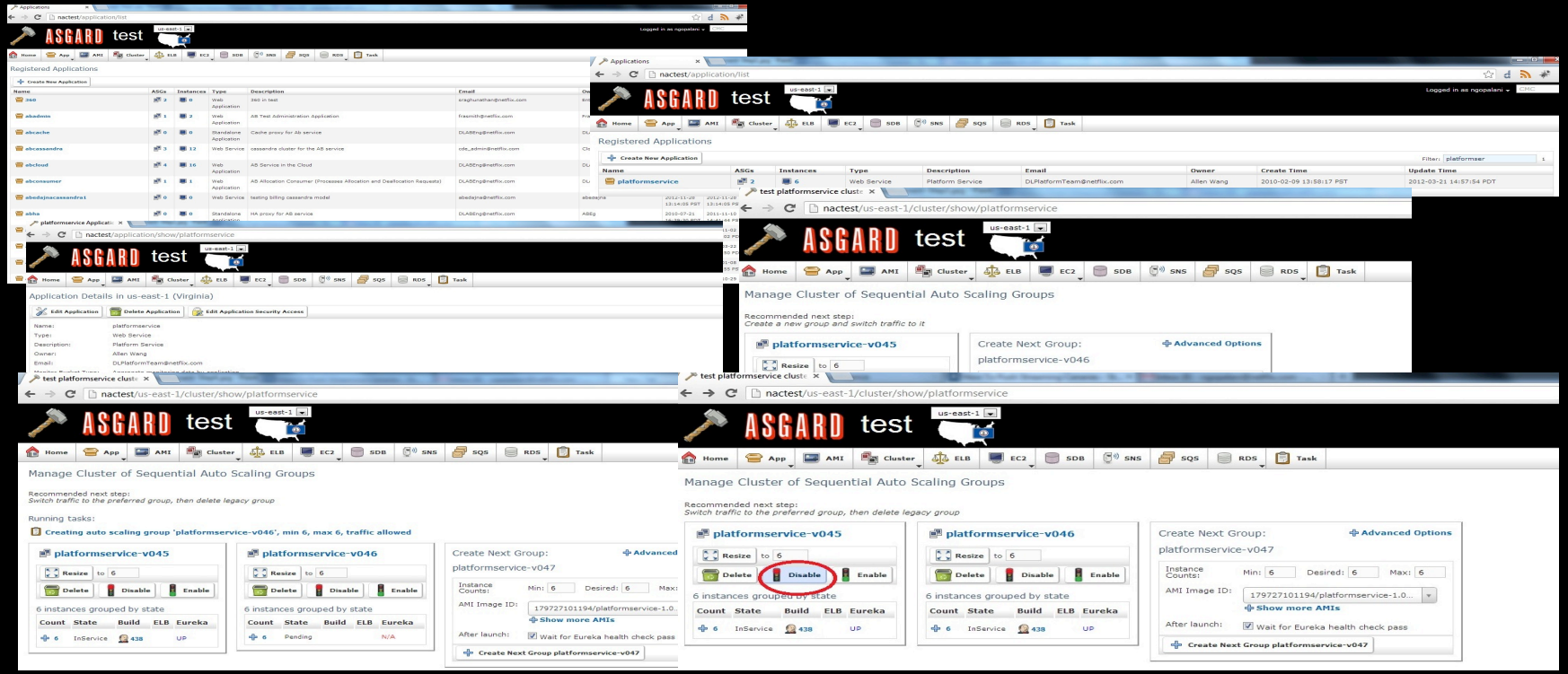
# Multiregional Monkeys

- Detect failure to deploy
- Differences in configuration
- Resource differences

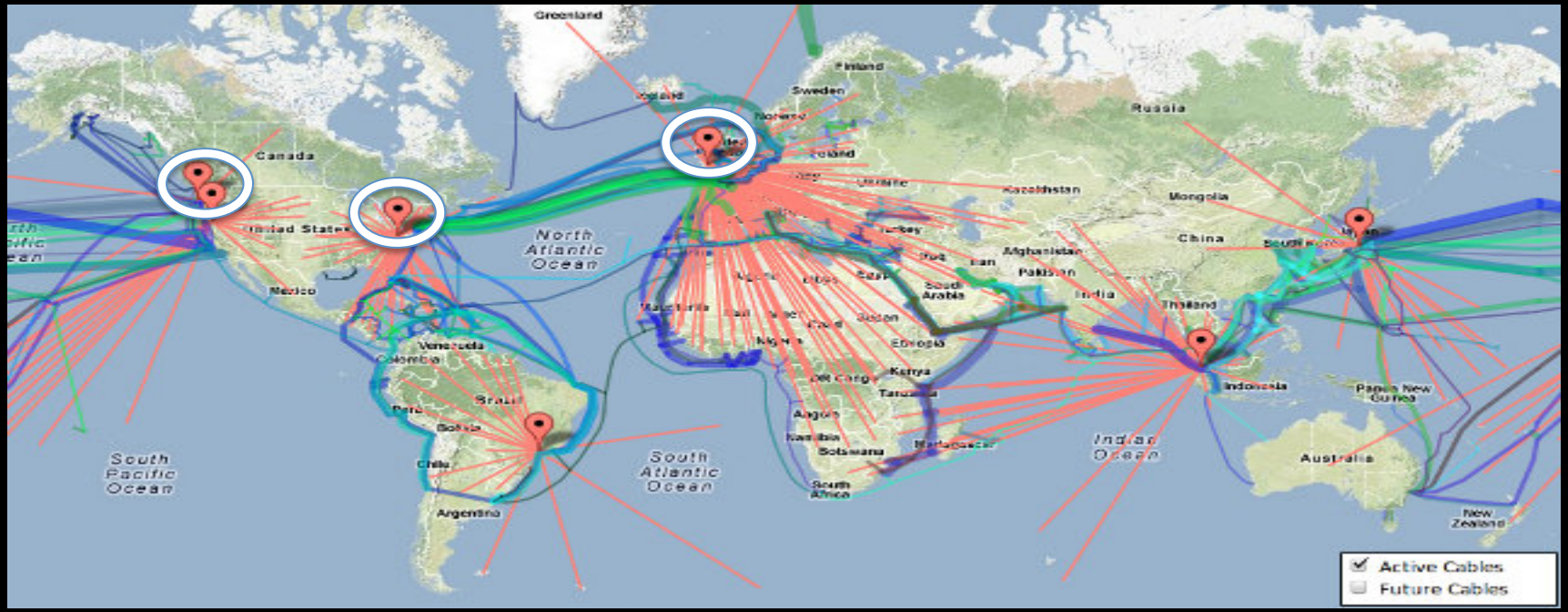
# Operating in N Regions

- Best practices: avoiding peak times for deployment
- Early problem detection / rollbacks
- Automated canaries / continuous delivery

# Application deployment



# 6 deployments




# At least 36 steps

# Automating deployment

Mimir Builds Deployments AMIs Clusters ASGs Workflows

## platformservice

Promotional Status | Builds | History | Tasks

**LATEST BUILD** — Promote to Test

Jenkins Job: WE-WAPP-platformservice
Status: Pass
Last Build: 659
Build Time: 09/27/2013

**TEST** — Promote to Production

TEST | US-WEST-2 | PLATFORMSERVICE-INT

| ASG Name | Build | Status | Instances | Created |
|---|---|---|---|---|
| platformservice-int-v009 | 658 | Down | 2 | 09/20/2013 |
| platformservice-int-v010 | 659 | Up | 2 | 09/27/2013 |

TEST | US-EAST-1 | PLATFORMSERVICE

| ASG Name | Build | Status | Instances | Created |
|---|---|---|---|---|
| platformservice-v105 | 659 | Up | 6 | 09/27/2013 |

TEST | US-WEST-2 | PLATFORMSERVICE

| ASG Name | Build | Status | Instances | Created |
|---|---|---|---|---|
| platformservice-v047 | 658 | Down | 2 | 09/20/2013 |
| platformservice-v048 | 659 | Up | 2 | 09/27/2013 |

TEST | EU-WEST-1 | PLATFORMSERVICE-INT

| ASG Name | Build | Status | Instances | Created |
|---|---|---|---|---|
| platformservice-int-v011 | 658 | Down | 2 | 09/20/2013 |
| platformservice-int-v012 | 659 | Up | 2 | 09/27/2013 |

TEST | US-EAST-1 | PLATFORMSERVICE-INT

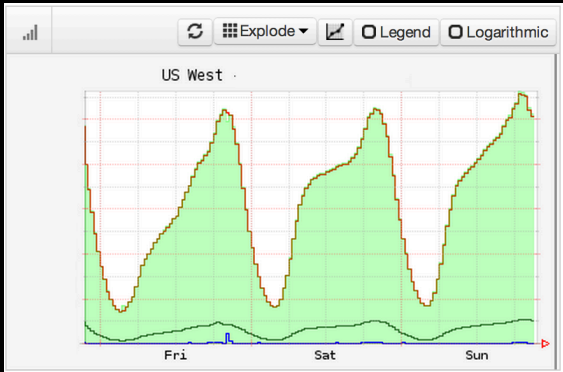
| ASG Name | Build | Status | Instances | Created |
|---|---|---|---|---|
| platformservice-int-v016 | 658 | Down | 2 | 09/20/2013 |
| platformservice-int-v017 | 659 | Up | 2 | 09/27/2013 |

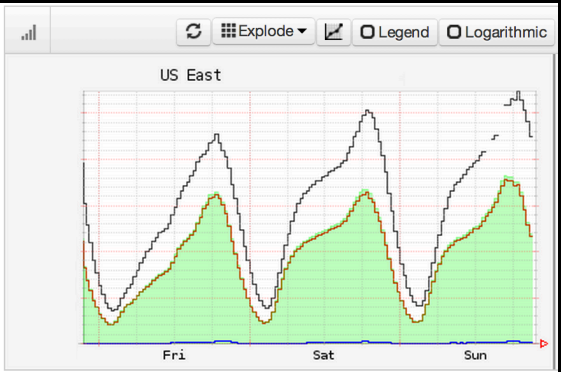
TEST | US-WEST-1 | PLATFORMSERVICE

| ASG Name | Build | Status | Instances | Created |
|---|---|---|---|---|

# Monitoring and Alerting

- Per region metrics
- Global aggregation
- Anomaly detection

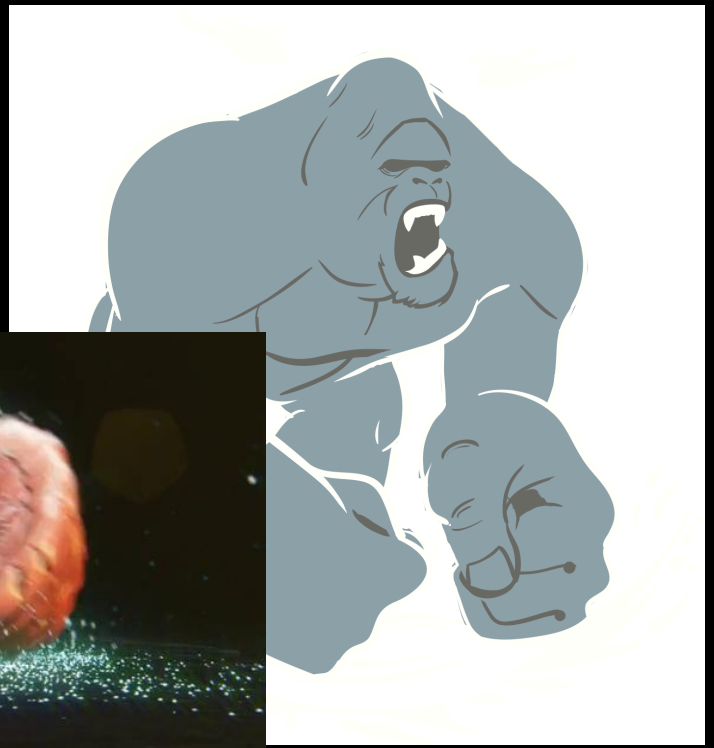


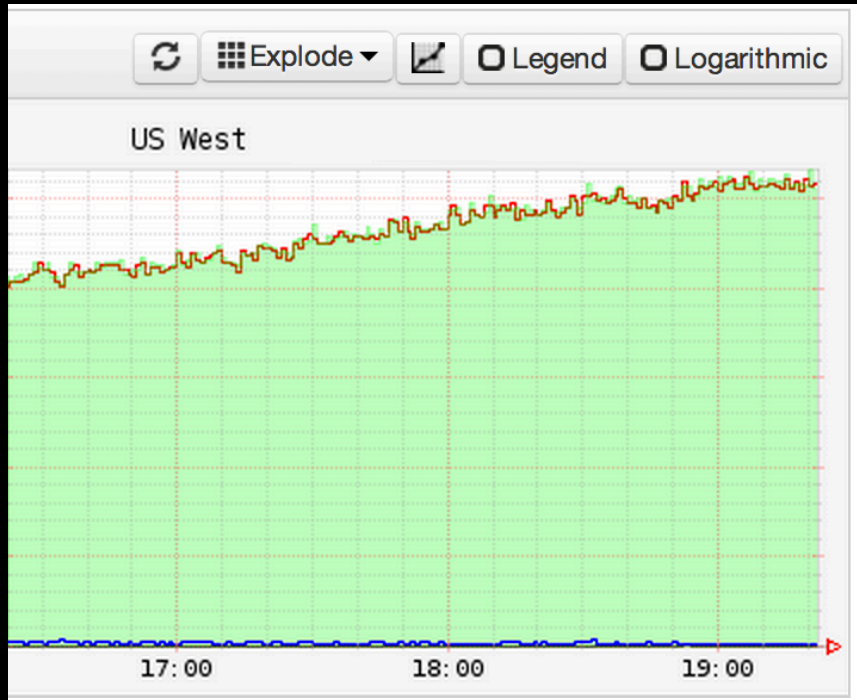
# Failover and Fallback

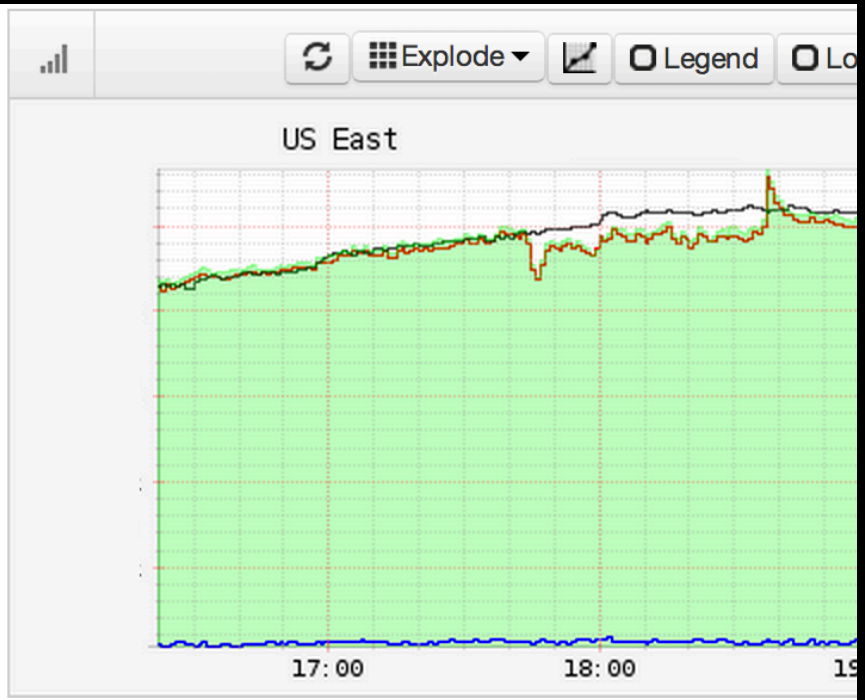
- Directional DNS changes (Or Route53)
- For fallback, ensure data consistency
- Some challenges
  - Cold cache
  - Autoscaling
- (Iterate, Automate)$^{+}$

# Validating the Whole Thing Works

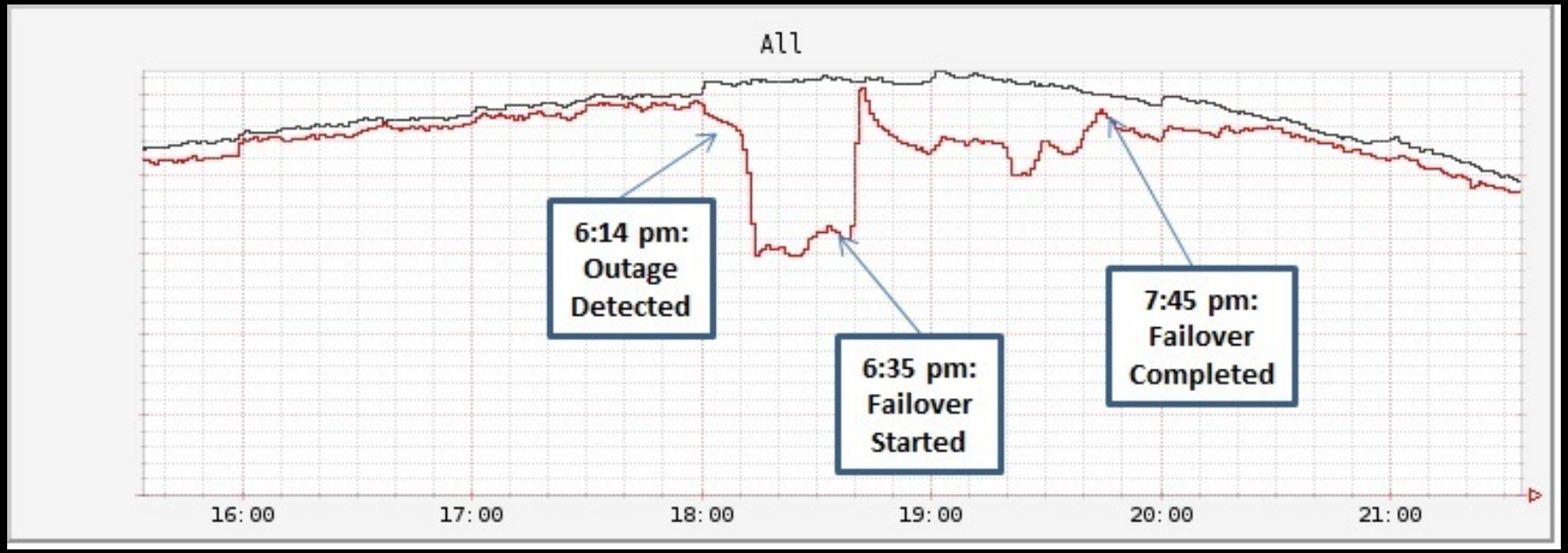



# Does Isolation Work?

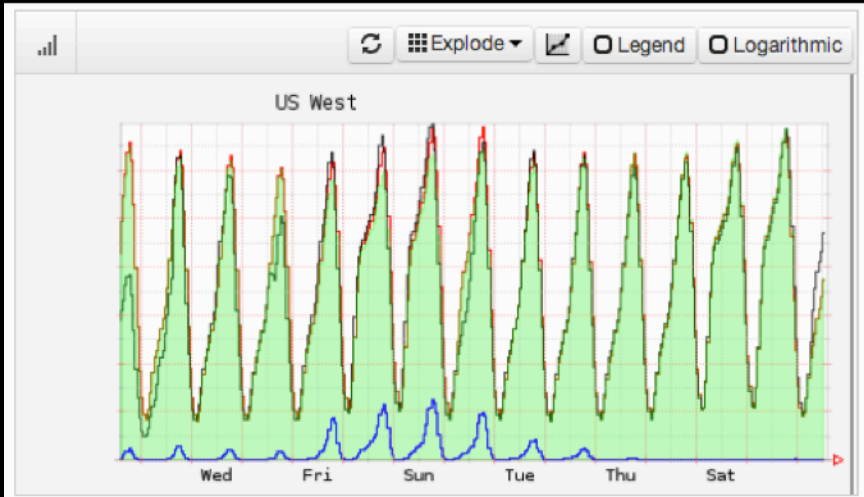


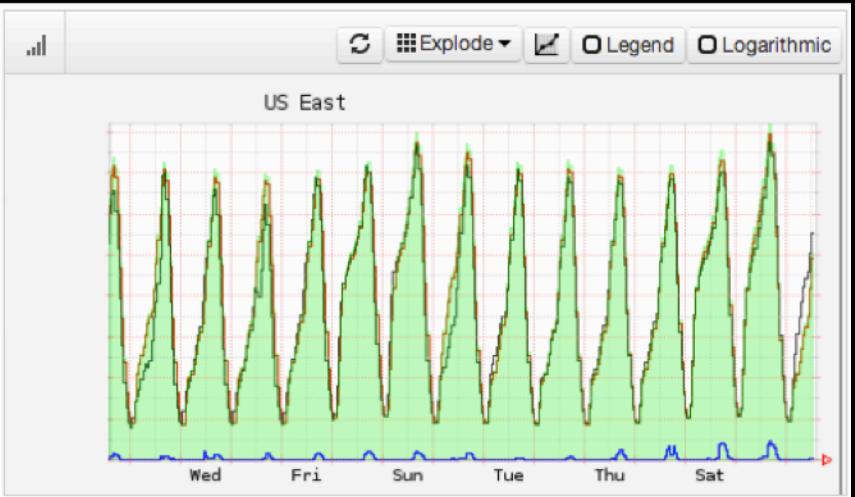



# Does Failover work?

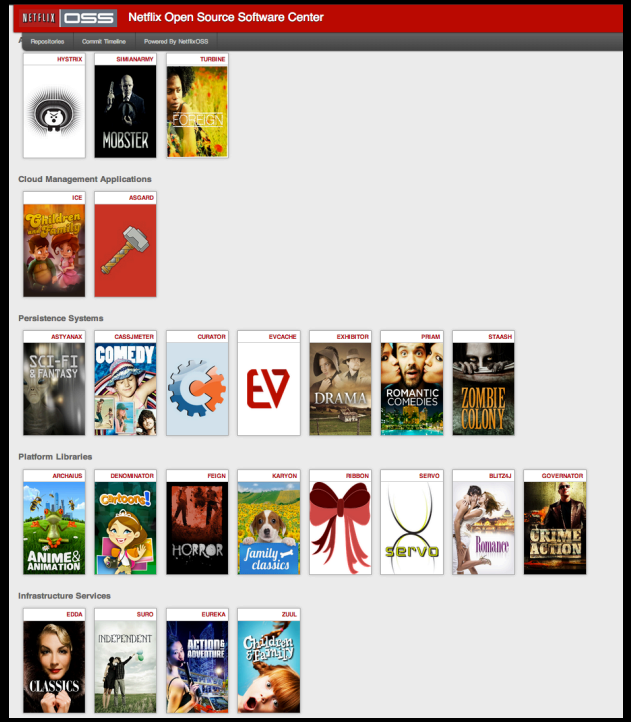


# Steady State



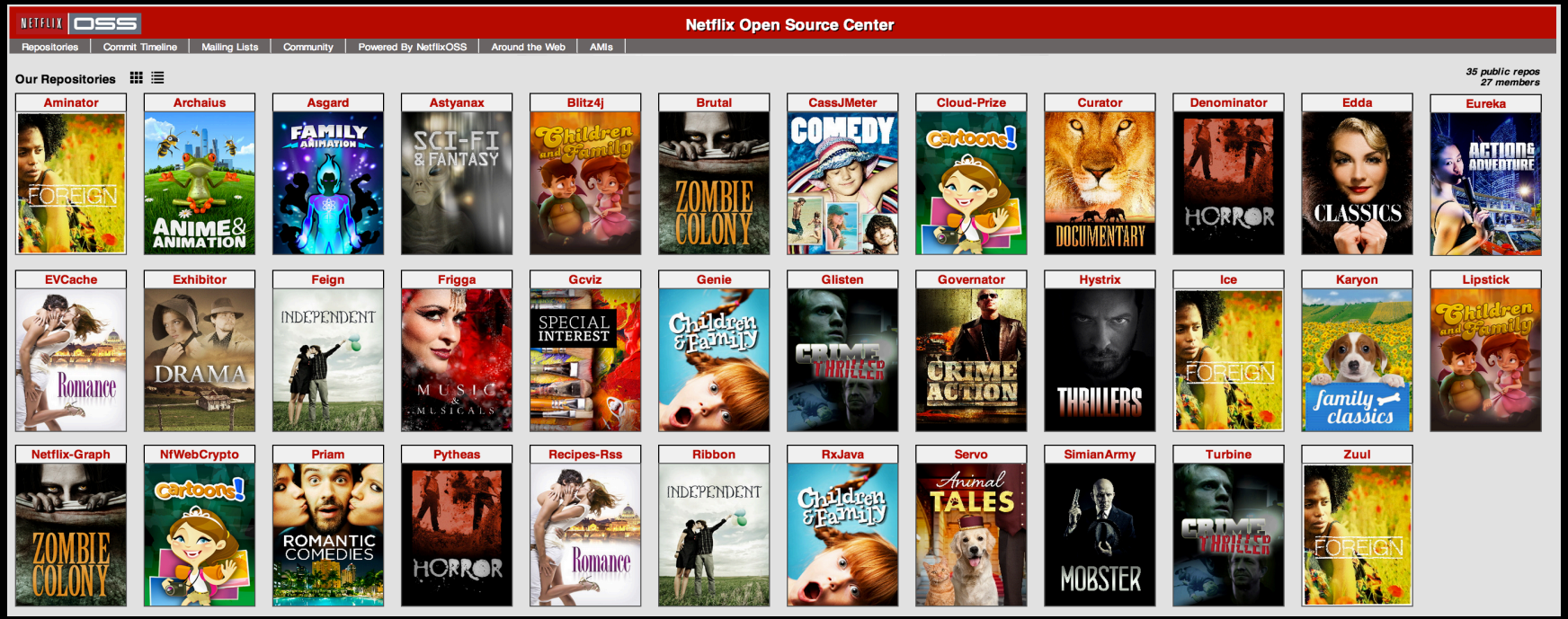



# Wii.FM?


NETFLIX OSS
Netflix Open Source Software Center
Repositories
Commit Timeline
Powered By NetflixOSS
HYSTRIX
SIMIANARMY
TURBINE
MOBSTER
FOREIGN
Cloud Management Applications
ICE
ASGARD
Persistence Systems
ASTYANAX
CASSJMETER
CURATOR
EVCACHE
EXHIBITOR
PRIAM
STAASH
SCI-FI & FANTASY
COMEDY
EV
DRAMA
ROMANTIC COMEDIES
ZOMBIE COLONY
Platform Libraries
ARCHAIUS
DENOMINATOR
FEIGN
KARYON
RIBBON
SERVO
BLITZ4J
GOVERNATOR
ANIME & ANIMATION
CARTOONS!
HORROR
family classics
servo
Romance
CRIME & ACTION
Infrastructure Services
EDDA
SURO
EUREKA
ZUUL
CLASSICS
INDEPENDENT
ACTIONS & ADVENTURE
Children & Family


We're here to help you get to global scale…
Apache Licensed Cloud Native OSS Platform
http://netflix.github.com

# Technical Indigestion – what do all these do?



# Asgard - Manage Red/Black Deployments

# Ice – Slice and dice detailed costs and usage

2013-04-01 12AM
End
2013-05-29 12AM
Cost Usage
Group by UsageType
Aggregate hourly
Plot Type area
Show Throughput
Factor Throughput
filter
ap-northeast-1
ap-southeast-1
eu-west-1
sa-east-1
us-east-1
ec2_instance
OndemandInstances
ReservedInstances
BorrowedInstances
LentInstances
UnusedInstances
c1.medium
c1.medium.windows
c1.xlarge
c1.xlarge.others
c1.xlarge.windows
Submit
Download

SHOW ALL HIDE ALL

| UsageType | Total | Max | Average |
|---|---|---|---|
| aggregated | $ | $ | $ |

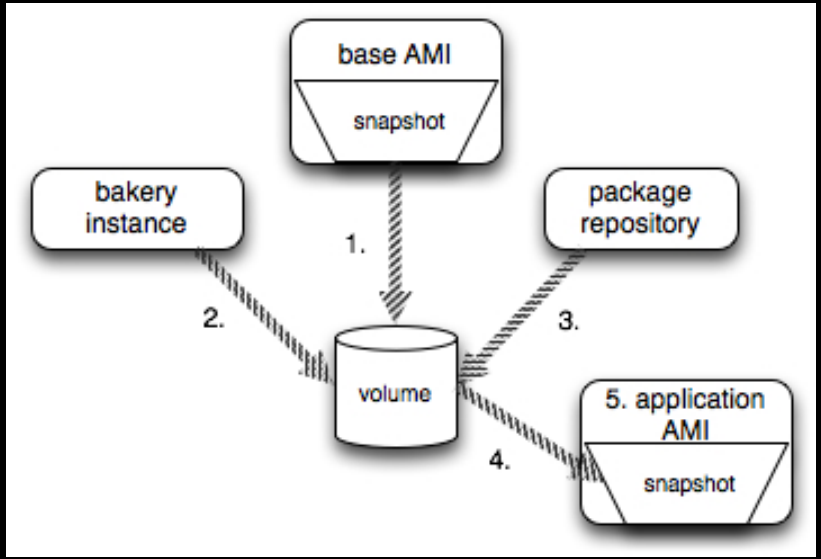
Monday, Apr 15, 12am, 2013
aggregated : $ / hour
: $ / hour (0.1%)
: $ / hour (0.0%)
: $ / hour (0.4%)
: $ / hour (0.0%)
: $ / hour (0.0%)
: $ / hour (3.3%)
: $ / hour (3.9%)
: $ / hour (0.0%)
: $ / hour (0.7%)
: $ / hour (0.1%)
: $ / hour (0.4%)
: $ / hour (0.2%)
: $ / hour (14.9%)
: $ / hour (52.5%)
: $ / hour (7.3%)
: $ / hour (15.6%)
: $ / hour (0.0%)
: $ / hour (0.1%)
: $ / hour (0.5%)

Cost per hour
8. Apr
15. Apr
22. Apr
29. Apr
6. May
13. May
20. May
27. May

NETFLIX OSS

# Automatically Baking AMIs with Aminator

- AutoScaleGroup instances should be identical
- Base plus code/config
- Immutable instances
- Works for 1 or 1000…
- Aminator Launch
  - Use Asgard to start AMI or
  - CloudFormation Recipe




NETFLIX OSS

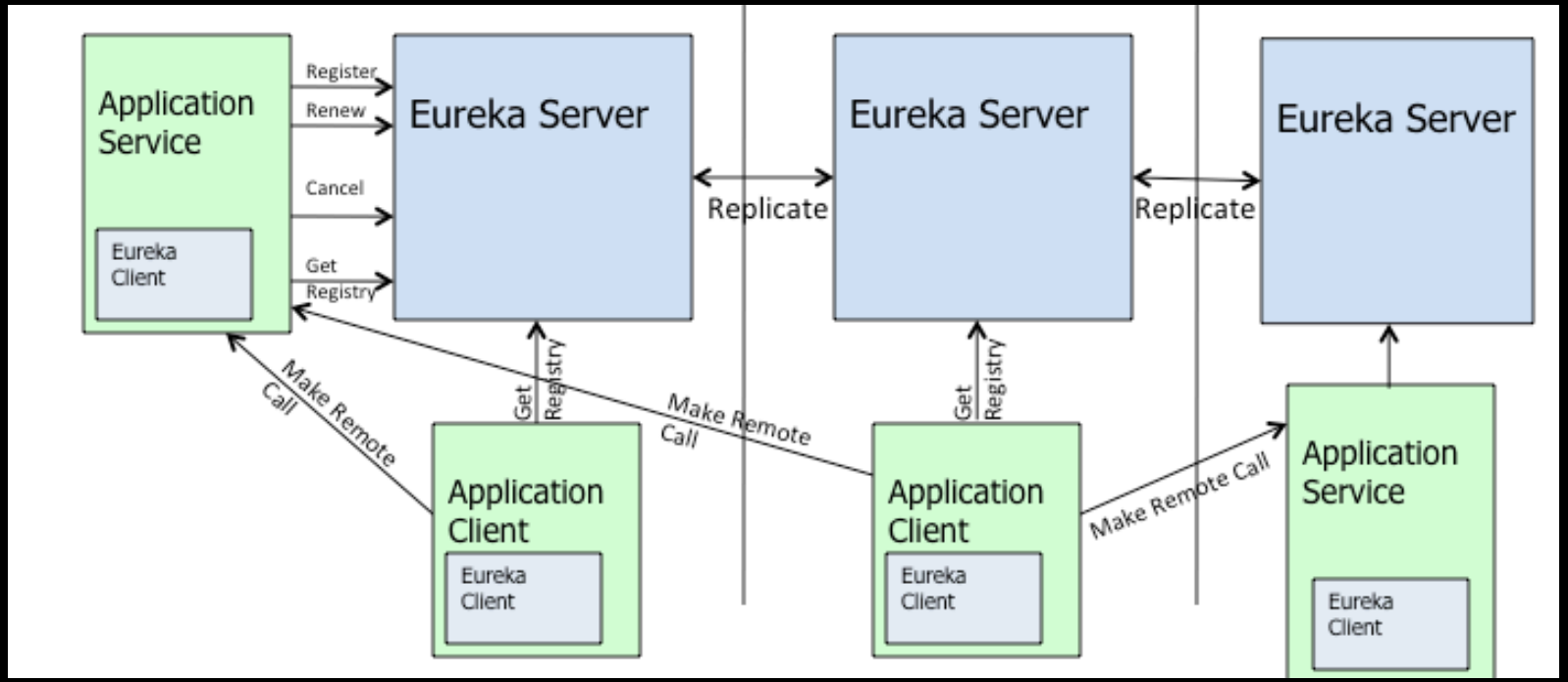
# Discovering your Services - Eureka

| | | | |
|---|---|---|---|
| **EVCACHE** | **ami-e794f78e** (3), **ami-c579ecac** (1), | **us-east-1c** (3), **us-east-1d** (1), | **UP** (4) - i-9fd, i-083, i-78e, i-45 |

- Map applications by name to
  - AMI, instances, Zones
  - IP addresses, URLs, ports
  - Keep track of healthy, unhealthy and initializing instances
- Eureka Launch
  - Use Asgard to launch AMI or use CloudFormation Template

NETFLIX OSS

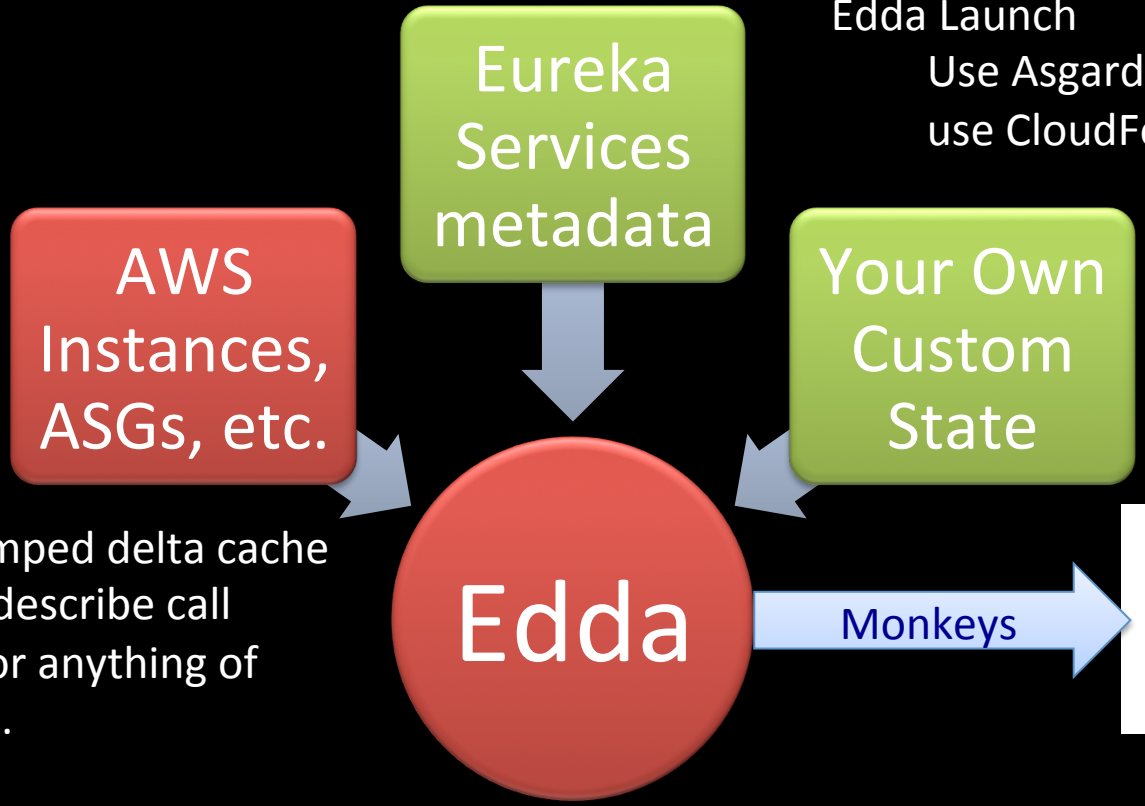
# Deploying Eureka Service – 1 per Zone



# Searchable state history for a Region / Account

Edda Launch

Use Asgard to launch AMI or use CloudFormation Template




Timestamped delta cache of JSON describe call results for anything of interest…


NETFLIX OSS

# Edda Query Examples

Find any instances that have ever had a specific public IP address

```
$ curl "http://edda/api/v2/view/instances;publicIpAddress=1.2.3.4;_since=0"
 ["i-0123456789","i-012345678a","i-012345678b”]
```

Show the most recent change to a security group

```
$ curl "http://edda/api/v2/aws/securityGroups/sg-0123456789;_diff;_all;_limit=2"
--- /api/v2/aws.securityGroups/sg-0123456789;_pp;_at=1351040779810
+++ /api/v2/aws.securityGroups/sg-0123456789;_pp;_at=1351044093504
@@ -1,33 +1,33 @@
 {
…
       "ipRanges" : [
         "10.10.1.1/32",
         "10.10.1.2/32",
+        "10.10.1.3/32",
-        "10.10.1.4/32"
…
 }
```

# Archaius – Property Console

Persisted Properties Console : **TEST** us-east-1

Properties | Diff | Journal Logs | Documentation | Reports

/ FastProperty Explorer /

**Create New Fast Property in te**

Name: prop.foo
Value: 97
Constraints: int:0-99
Description: property to control %
UpdatedBy: stonse@netflix.com
Notification Email:
CMC:
Scope : Not Selected

Select Scope

**Scope Selection**

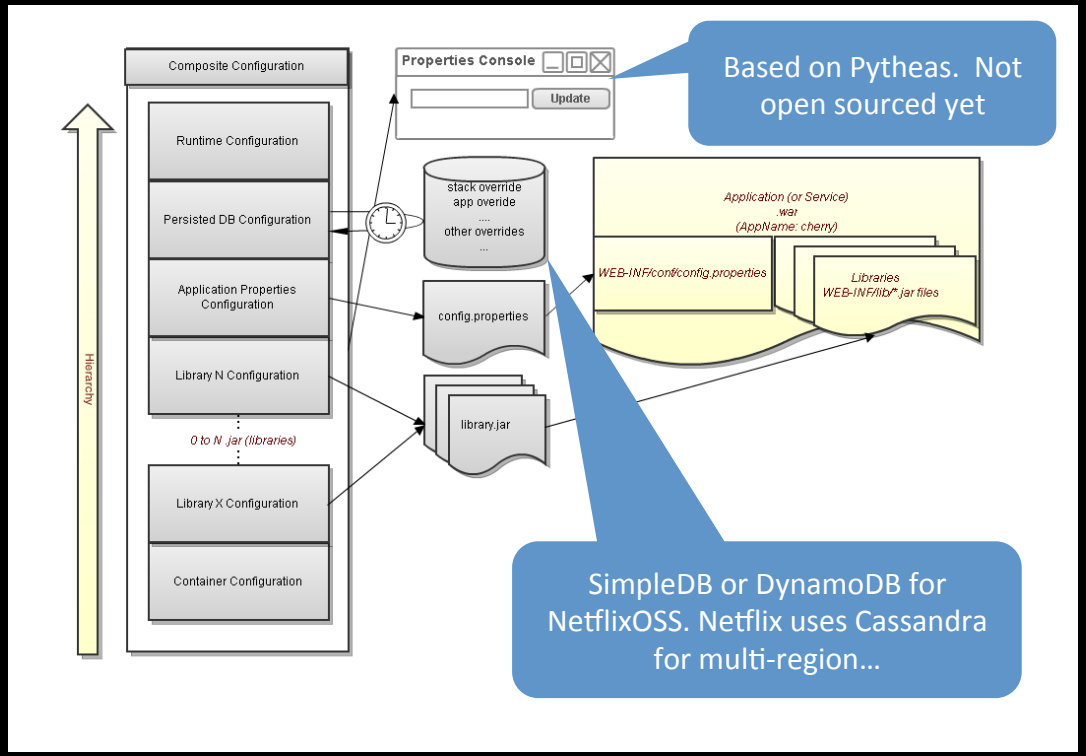
Region: us-east-1 (Virginia)
Application: helloworld 1 helloworld
~~Stack~~:
Cluster: helloworld-example 12 filter
ASG: helloworld-example-v140 14 filter

**Quick Impact View**
Total instances: 2
Current Scope

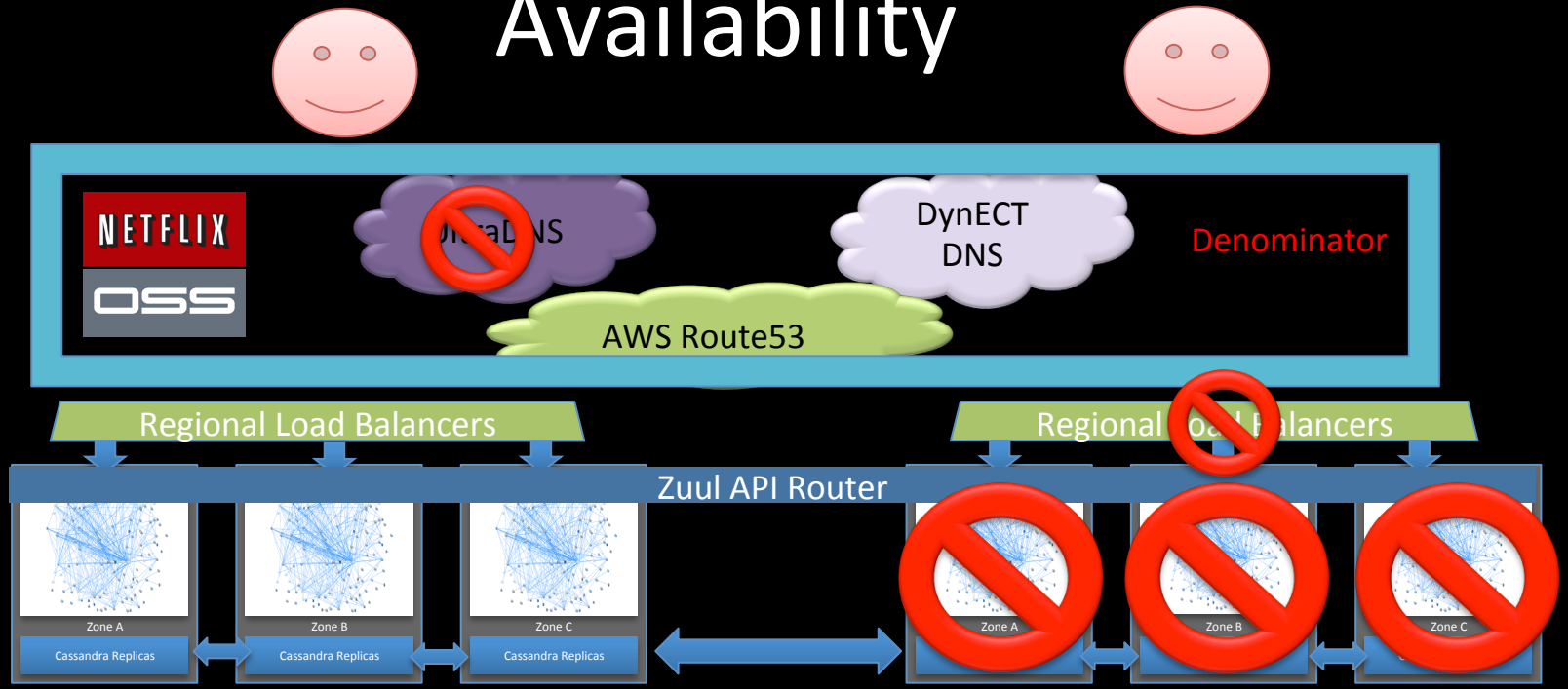
| | |
|---|---|
| region | us-east-1 |
| appId | helloworld |
| cluster | helloworld-example |

Select ASG | Skip | Done | Repeat Selection

# Archaius library – configuration management



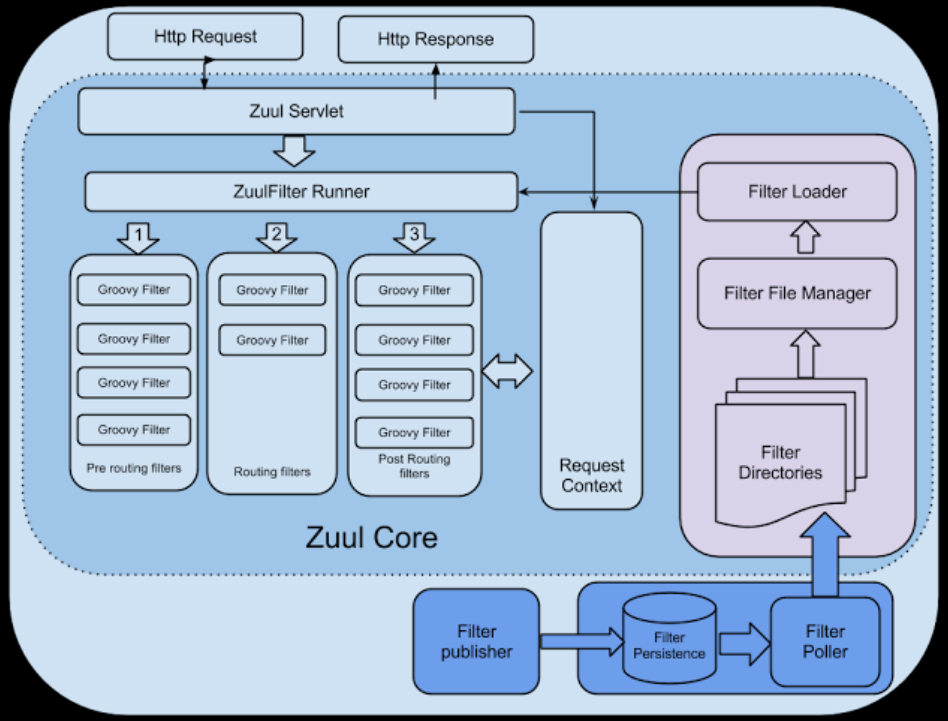
# Denominator: DNS for Multi-Region Availability



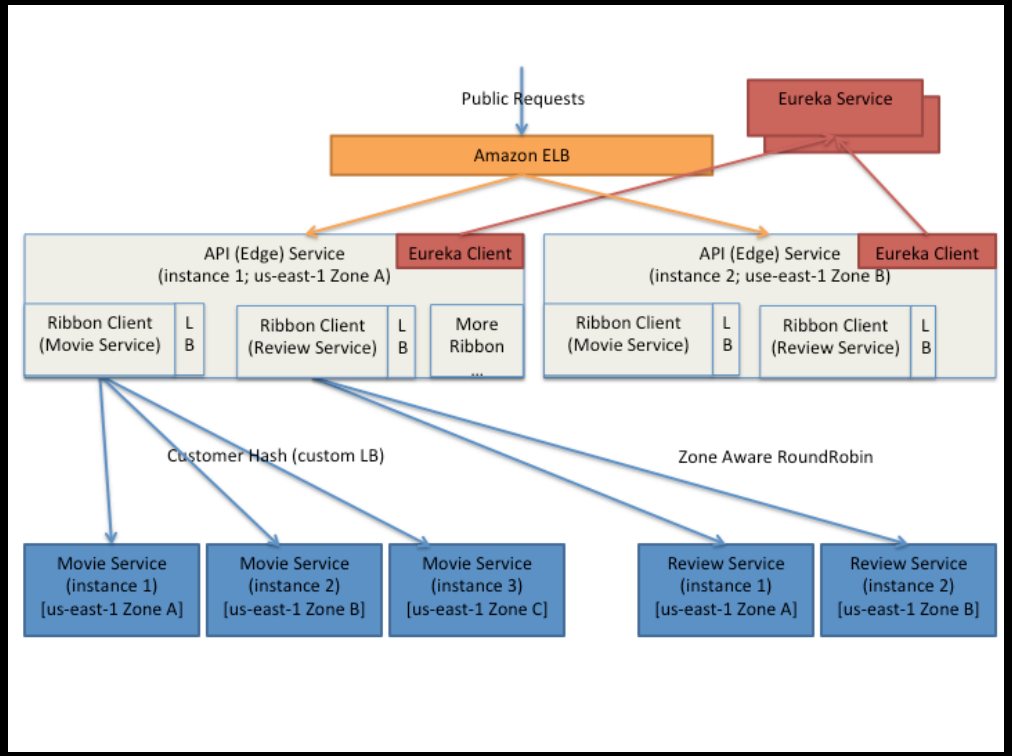

Denominator – manage traffic via multiple DNS providers with Java code

NETFLIX
OSS

# Zuul – Smart and Scalable Routing Layer



# Ribbon library for internal request routing



# Ribbon – Zone Aware LB

Average Active Requests = outstanding requests / (total instances - tripping instances)


# Karyon - Common server container

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

NETFLIX OSS

# Karyon

- Embedded Status Page Console
  - Environment
  - Eureka
  - JMX

PLATFORMSERVICE v1.0 us-east-1.prod

Astyanax | Counters | Debug Data | Diagnostics | Discovery | Environment | Info | Jars | JMX | NIWS | Properties | SLA | Tracers | Refresh | Machine Readable | filter

Filter:

- Catalina
- JMImplementation
- Library-atlas_plugin-com.netflix.library.management
- Library-epicplugin-com.netflix.library.management
- Library-nfastyanax-com.netflix.library.management
- Library-platform-com.netflix.library.management
- com.netflix.MonitoredResources
  - APPLICATION
  - ASTYANAX
    - cass_turtle.platformservice.astyanax
    - cass_turtle_platformservice
  - PLATFORM
- com.netflix.messaging
- com.netflix.servo
  - ActiveChukwaConsumers
  - ActiveChukwaSenders
  - ChunkQueueSize
  - DiscoveryClient_Cancel
  - DiscoveryClient_Failed
  - DiscoveryClient_FetchRegistry
  - DiscoveryClient_GetServiceUrlsFromDNS
  - DiscoveryClient_Refresh
  - DiscoveryClient_RefreshDelta
  - DiscoveryClient_Register
  - DiscoveryClient_Renew
  - DiscoveryClient_Reregister
  - DiscoveryClient_Retry
  - LBStats_AvailableZones
  - LBStats_CircuitBreakerTrippedCount
  - LoadBalancer_AllServerList
  - LoadBalancer_ChooseServer
  - LoadBalancer_UpServerList
  - PoolSize
  - SQS-changeMessageVisibilityTimeoutBatchCount

com.netflix.servo:name=LBStats_AvailableZones,*

Attributes

Refresh | Search:

| Key | Name | Value |
|---|---|---|
| {"class":"LoadBalancerStats","id":"akmsclient","level":"INFO","type":"INFORMATIONAL"} | value | [us-east-1e, us-east-1d, us-east-1c] |
| {"class":"LoadBalancerStats","id":"atlas_publish","level":"INFO","type":"INFORMATIONAL"} | value | [us-east-1e, us-east-1d, us-east-1c] |
| {"class":"LoadBalancerStats","id":"chukwatrackerclient","level":"INFO","type":"INFORMATIONAL"} | value | [us-east-1e, us-east-1d, us-east-1c] |
| {"class":"LoadBalancerStats","id":"defaultRestClient","level":"INFO","type":"INFORMATIONAL"} | value | [] |
| {"class":"LoadBalancerStats","id":"epic_publish","level":"INFO","type":"INFORMATIONAL"} | value | [us-east-1e, us-east-1d, us-east-1c] |
| {"class":"LoadBalancerStats","id":"epicplugin","level":"INFO","type":"INFORMATIONAL"} | value | [us-east-1e, us-east-1d, us-east-1c] |

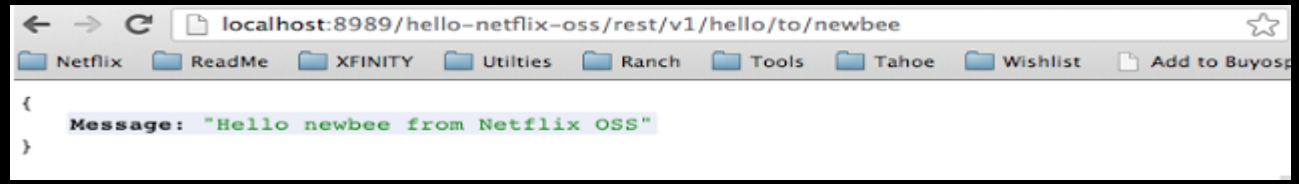
Showing 1 to 6 of 6 entries

Error: Visible: 898 Total: 898 Last updated: Mon Mar 04 2013 23:13:52

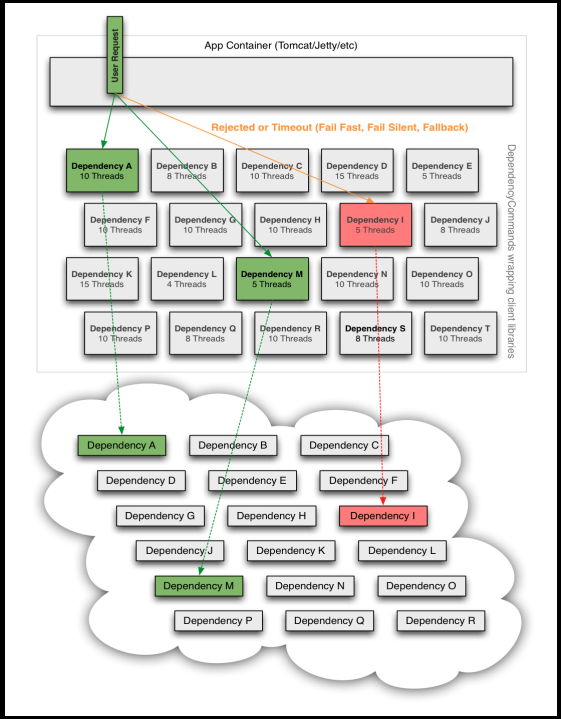
NETFLIX OSS

# Karyon

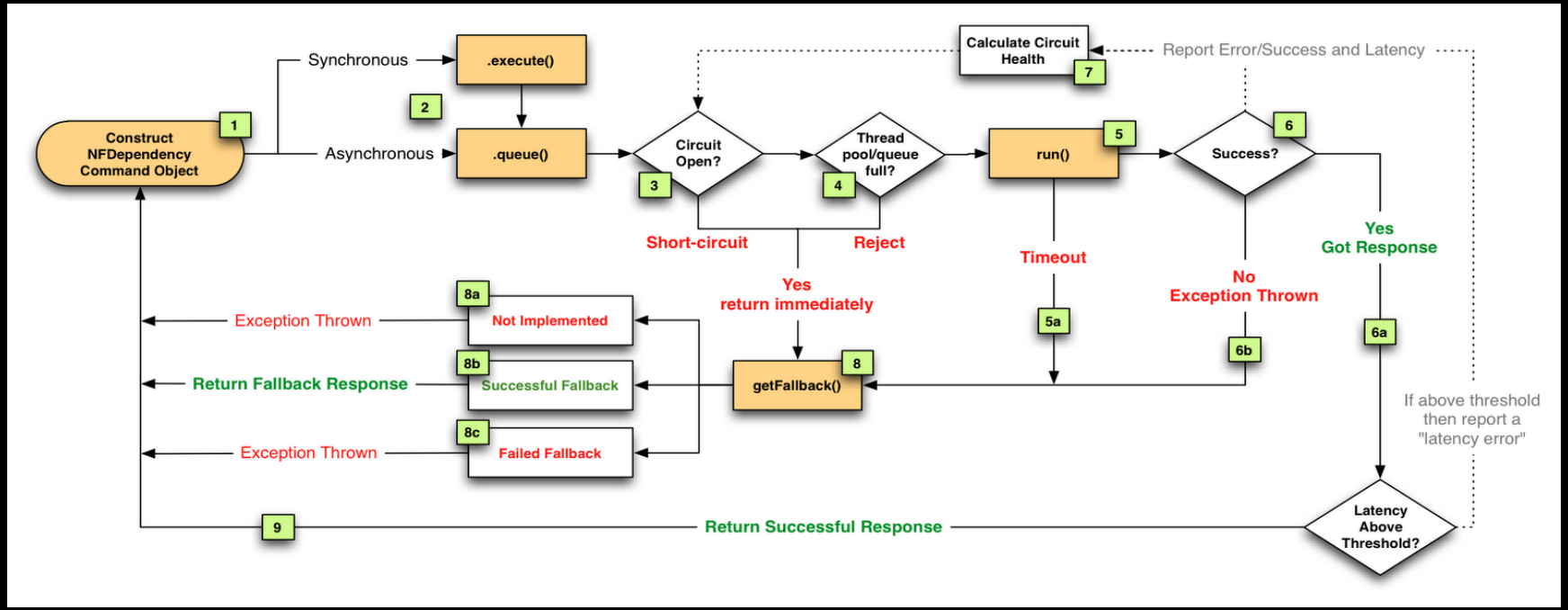
- Sample Service using Karyon available as "Hello-netflix-oss" on github



# Hystrix Circuit Breaker: Fail Fast -> recover fast

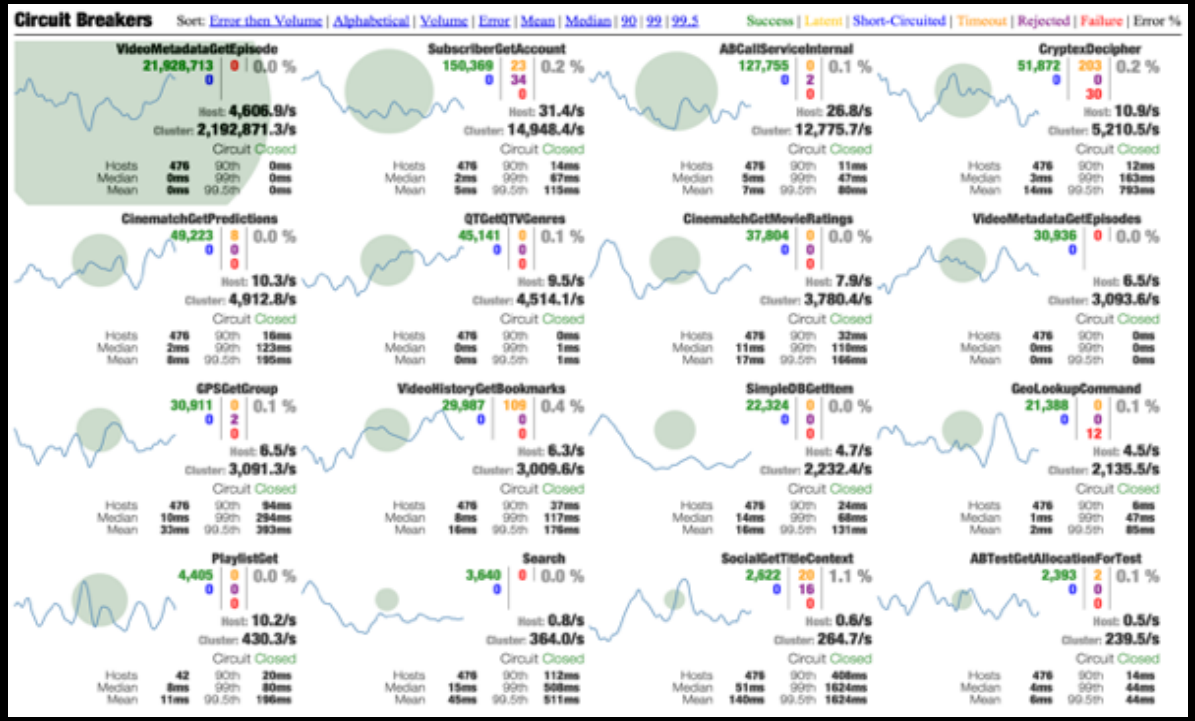


# Hystrix Circuit Breaker State Flow



# Turbine Dashboard

## Per Second Update Circuit Breakers in a Web Browser



# Either you break it, or users will


NETFLIX OSS

# Add some Chaos to your system


NETFLIX
OSS

# Clean up your room! – Janitor Monkey

Works with Edda history to clean up after Asgard

NETFLIX OSS

# Conformity Monkey

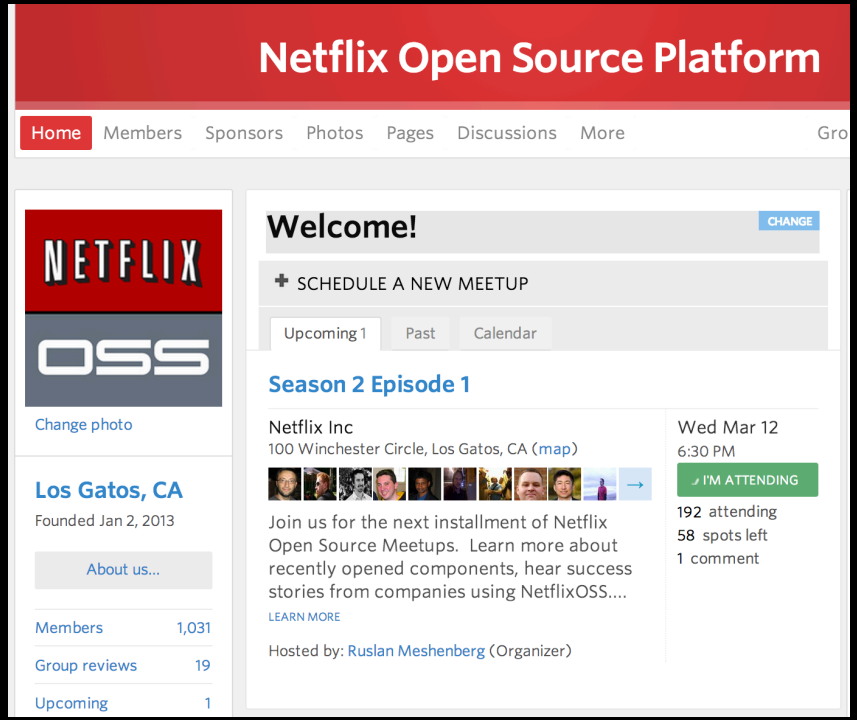
Track and alert for old code versions and known issues

Walks Karyon status pages found via Edda

# In use by many companies


netflixoss@netflix.com

# http://www.meetup.com/Netflix-Open-Source-Platform/



# Takeaways

*By embracing isolation and redundancy for availability*

*we transformed our Cloud Infrastructure to be resilient against Region-wide outages*

*Use NetflixOSS to scale your Enterprise*
*Contribute to existing github projects and add your own*

*http://netflix.github.com*

*@NetflixOSS*

*@rusmeshenberg*