OpenStack Swift Workshop

Joe Arnold

joe@swiftstack.com
swiftstack.com
@joearnold

Darrell Bishop

darrell@swiftstack.com
@swifterdarrell

San Francisco, CA

April 20, 2012

Getting Ready...

Creating Virtualbox Environment

Virtualbox->File->Import Appliance...

Start 'Swift VB 32bit Demo'

```
ssh -p 3333 -l demo localhost
```

Username: "demo"

password: "password"

Uses of Swift
Overview of Architecture

9 minutes

Deployment Workshop

47 minutes

swiftstack

Uses of Swift

Overview of Architecture

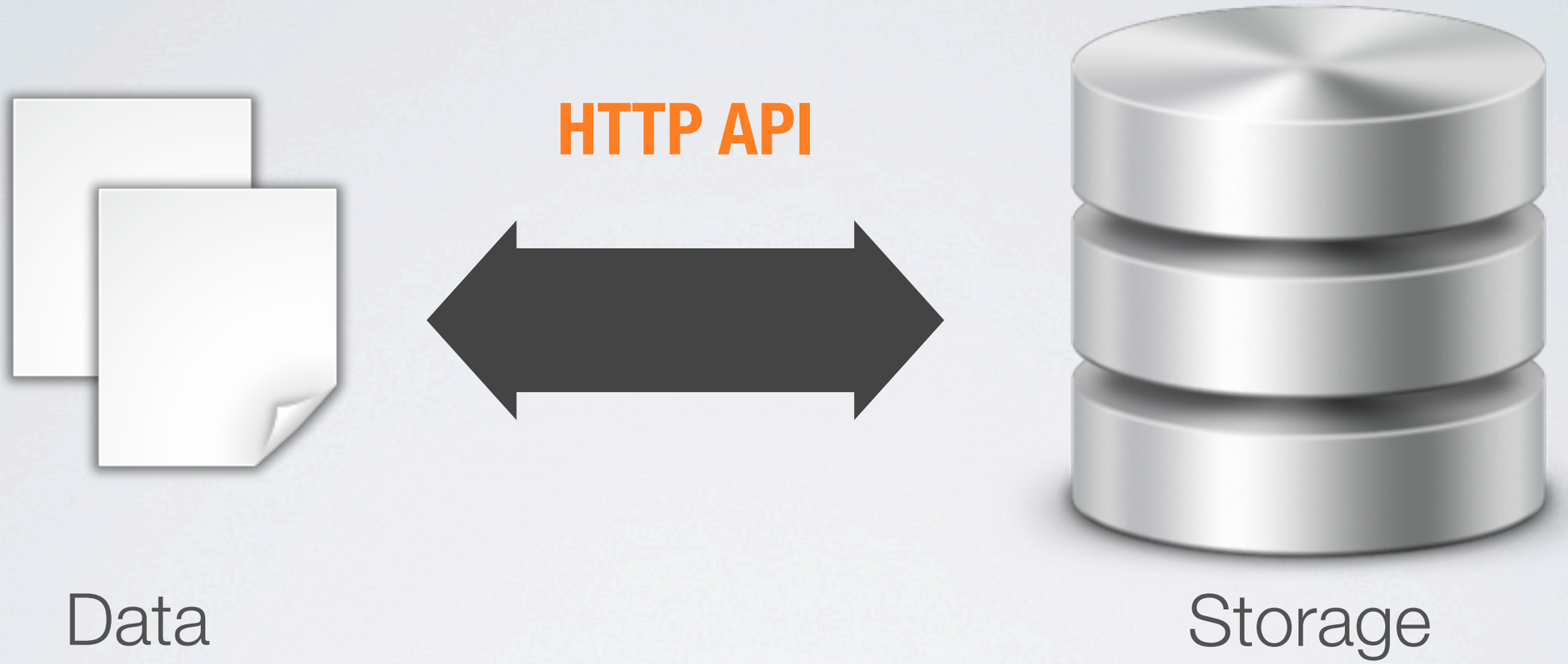
9 minutes

Deployment Workshop

47 minutes

swiftstack

Swift is Similar to Amazon S3



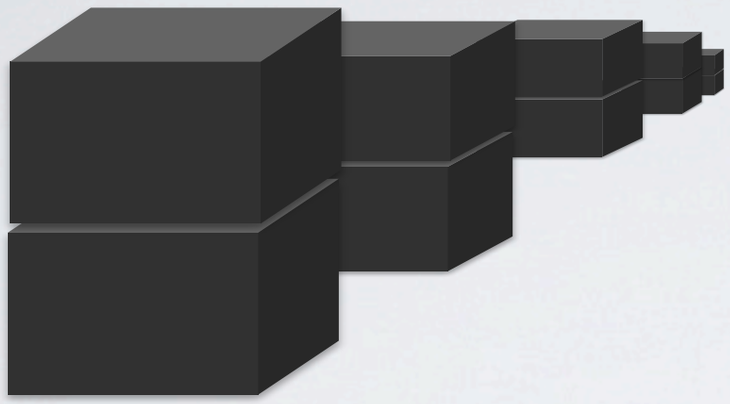
NOT blocks

NOT filesystem

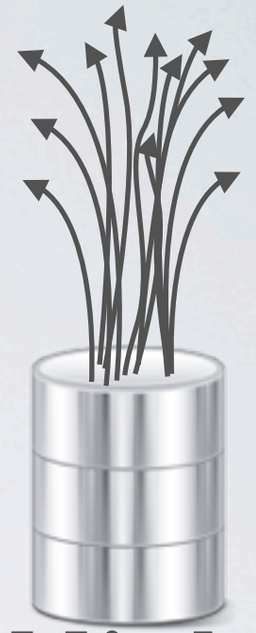
NOT SAN/NAS/DAS

swiftstack

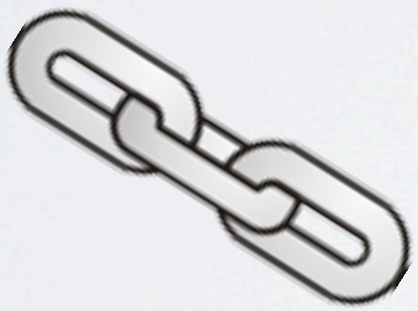
Swift Attributes



Scalable

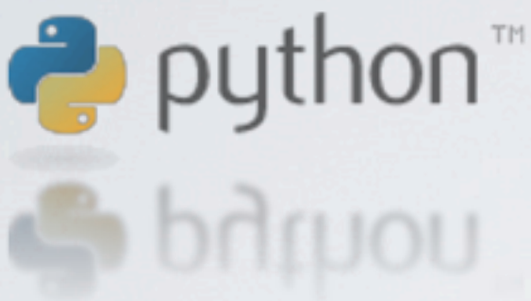


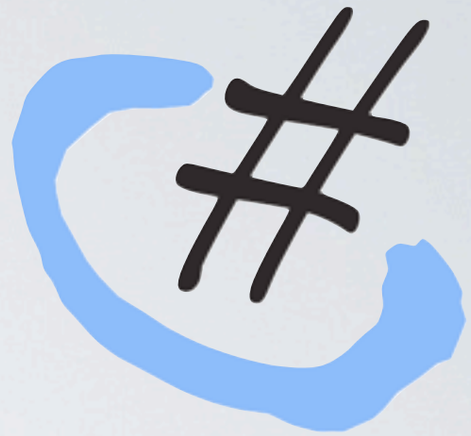
High concurrency



Durable

easy to use





jclouds

swiftstack

SwiftStack Distribution

Load Balancing

SSL Termination

Authentication

OpenStack Object Storage

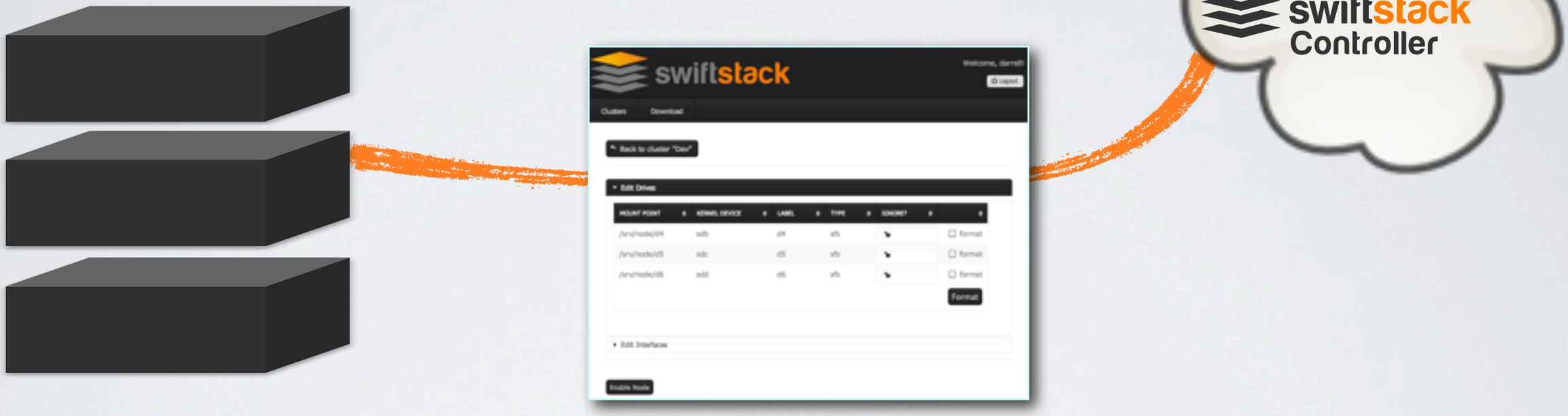
OS, Node & Drive Monitoring

Swift Runtime Monitoring

Operating System + Kernel

swiftstack

SwiftStack Controller



Uses of Swift

▶ Overview of Architecture

9 minutes

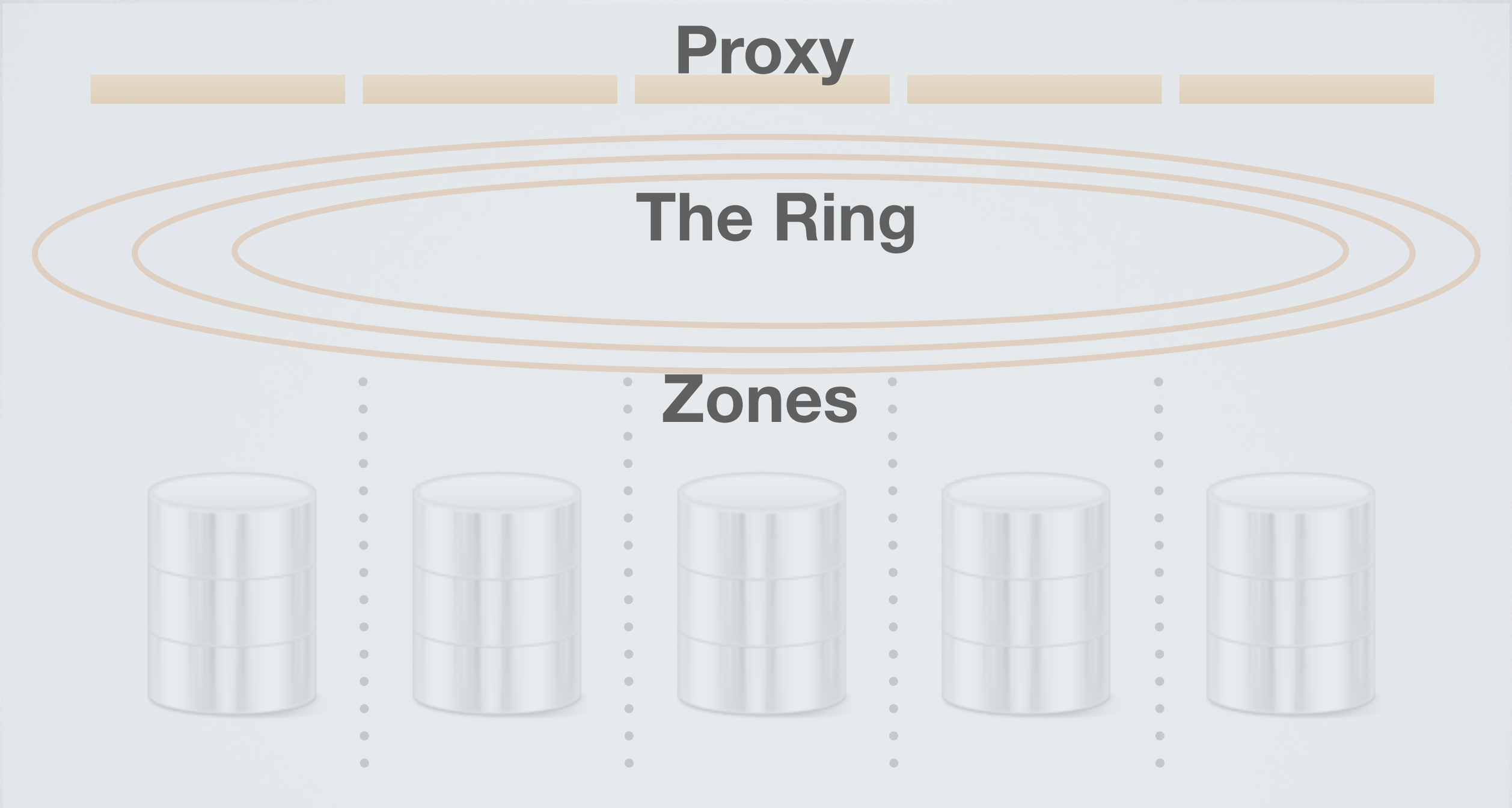
Deployment Workshop

47 minutes

swiftstack

Swift Architecture

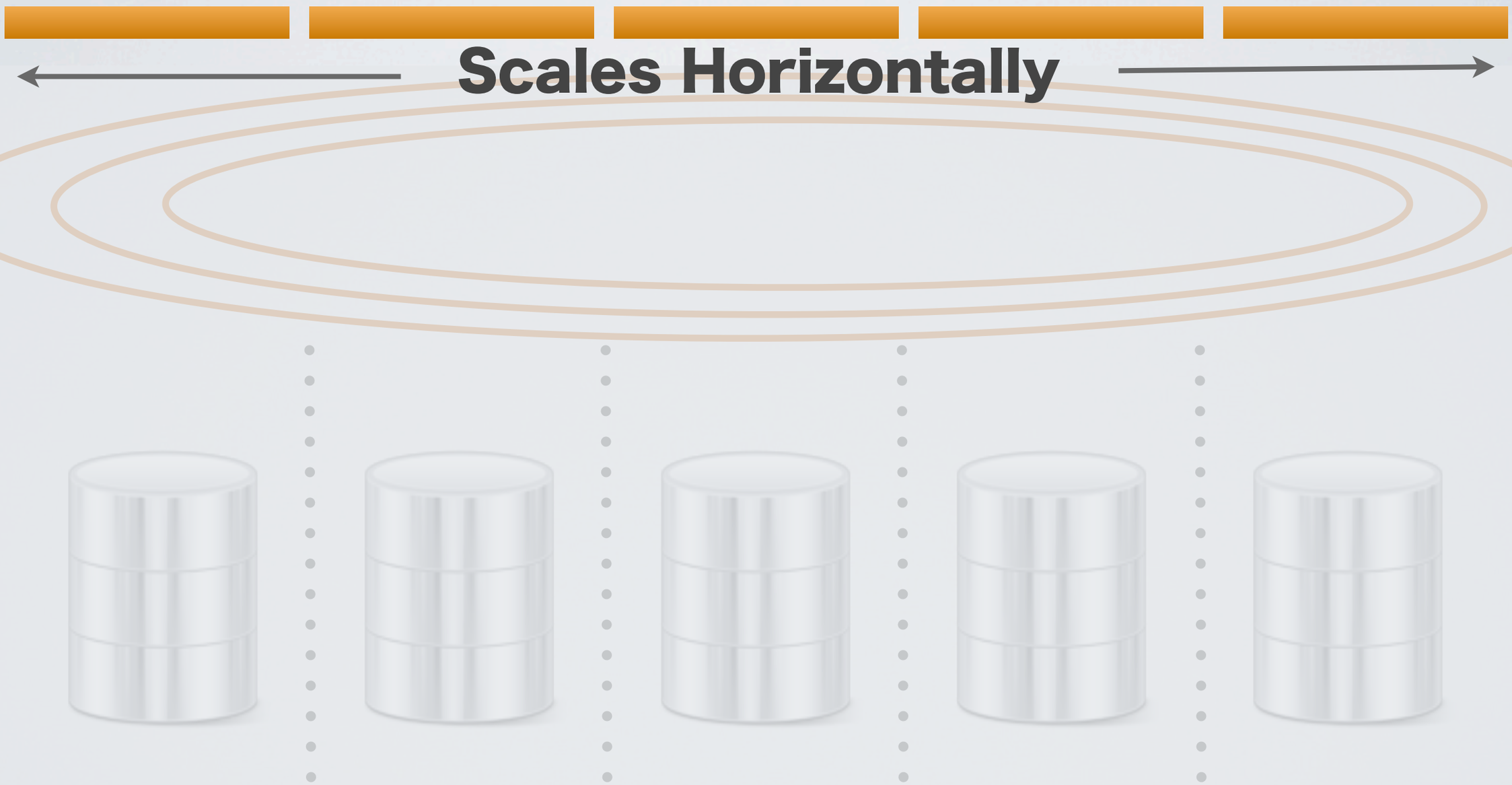
Overview



swiftstack

Proxy

handles incoming requests



swiftstack

The Ring

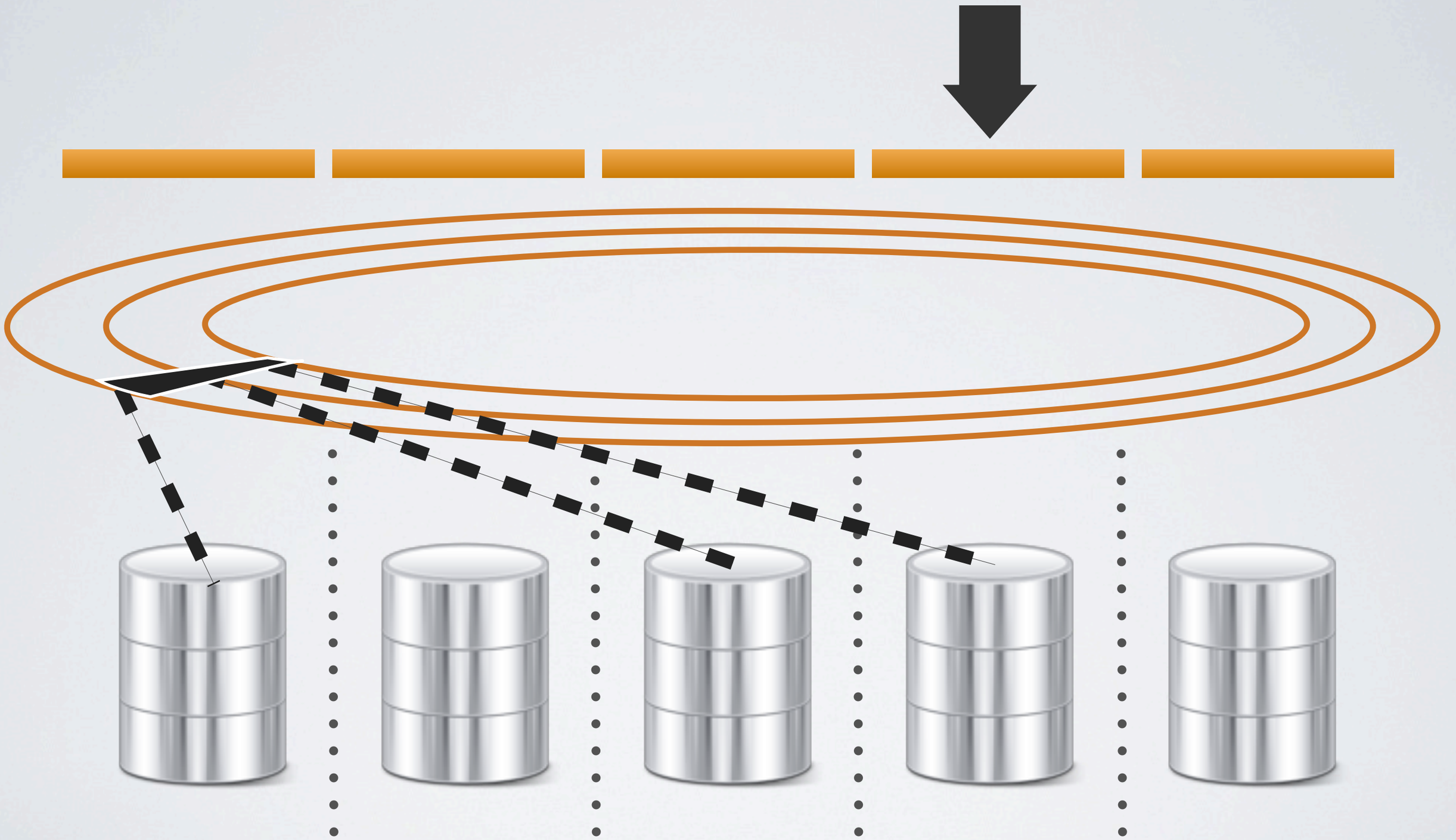
maps data to disk

Storage Zones

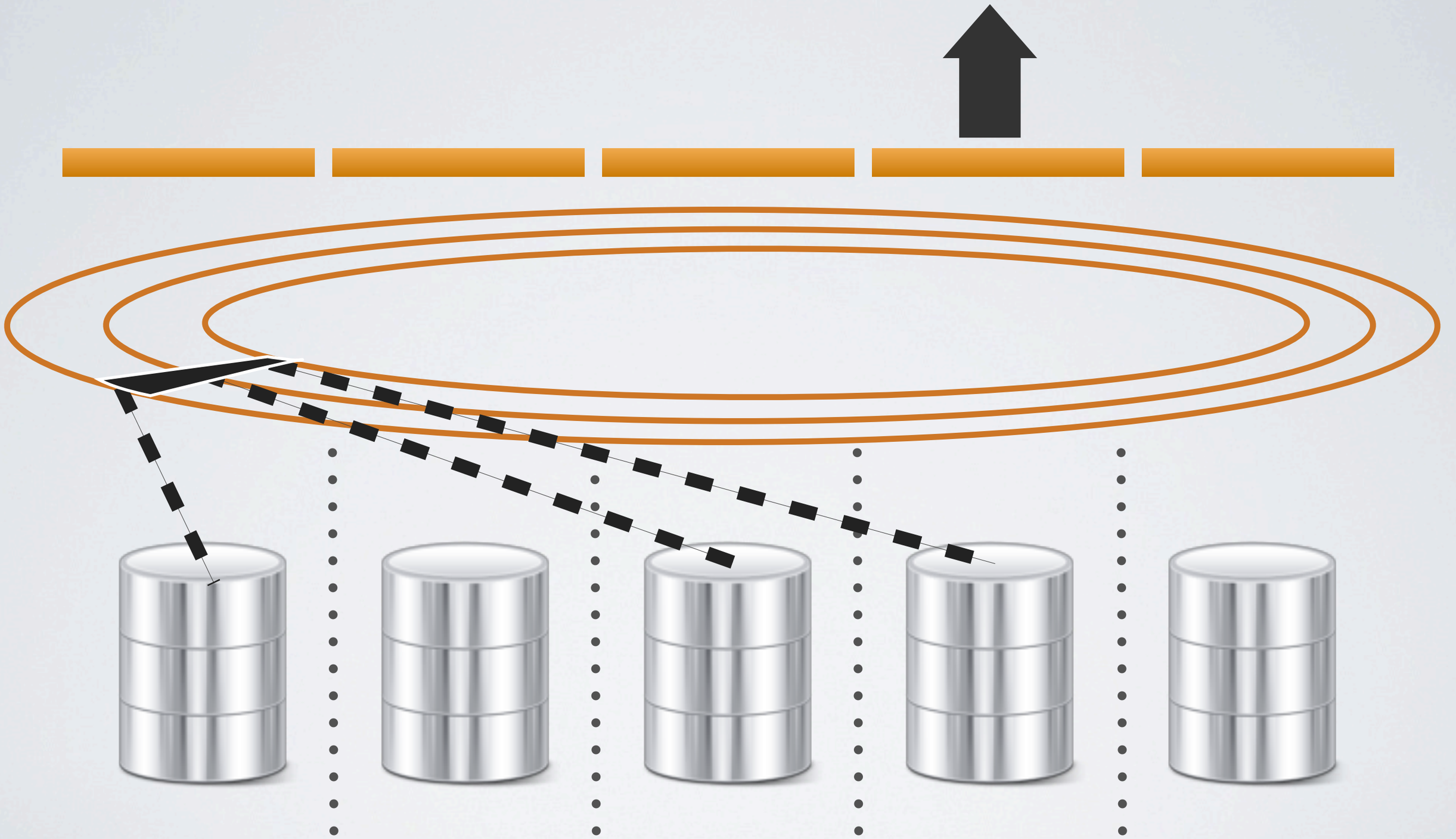
isolate physical failures

swiftstack

Quorum Writes



Single-Disk Reads



swiftstack

Replication

process runs continuously

swiftstack

Uses of Swift
Overview of Architecture

9 minutes

Deployment Workshop

47 minutes

swiftstack

Workshop Philosophy

Learn the Guts step-by-step by typing.

(NOT run our magic deployment tool.)

Follow along

-or-

Sit back and watch some typing

Getting Ready...

Creating Virtualbox Environment

Virtualbox->File->Import Appliance...

`Swift VB 32bit Demo.ova`

Start 'Swift VB 32bit Demo' VM

On host OS, SSH to localhost, port 3333
(port forwards to guest OS)

SSH to (host) localhost, port 3333

```
$ ssh -p 3333 -l demo localhost
```

Username: "`demo`"

Password: "`password`"

Formatting Devices

```
$ sudo su -
# mkfs.xfs -f -i size=512 -L d1 /dev/sdb
# mkfs.xfs -f -i size=512 -L d2 /dev/sdc
# mkfs.xfs -f -i size=512 -L d3 /dev/sdd
# mkfs.xfs -f -i size=512 -L d4 /dev/sde
```

Handy Commands

Mounted Filesystems

```
# df
```

Formatted Filesystems

```
# blkid
```

Mounting Drives

```
# mkdir -p /srv/node/d1
# mkdir -p /srv/node/d2
# mkdir -p /srv/node/d3
# mkdir -p /srv/node/d4

# mount -t xfs -L d1 /srv/node/d1
# mount -t xfs -L d2 /srv/node/d2
# mount -t xfs -L d3 /srv/node/d3
# mount -t xfs -L d4 /srv/node/d4
```

Handy Commands

```
# for i in 1 2 3 4; do
> mkdir -p /srv/node/d$i
> mount -t xfs -L d$i /srv/node/d$i
> done
# chown -R swift:swift /srv/node
```

Installing Swift

```
# apt-get install curl gcc bzr python-configobj python-coverage python-dev python-nose
python-setuptools python-simplejson python-xattr python-webob python-eventlet python-
greenlet debhelper python-sphinx python-all python-openssl python-pastedeploy python-
netifaces bzr-builddeb xfsprogs
(get and install Swift)
# apt-get update
```

...those should be already installed

Creating the Builder Files

swift-ring-builder (account|container|object).builder create <part_power> <replicas> <min_part_hours>

```
# cd /etc/swift
# swift-ring-builder account.builder create 18 3 1
# swift-ring-builder container.builder create 18 3 1
# swift-ring-builder object.builder create 18 3 1
```

Creating the Builder Files

Accounts

Containers

Objects

Builder files contain a record of all devices in the cluster

Creating the Builder Files

Swift Partitions

Creating the Builder Files

How many Swift Partitions?

How big will you be when you grow up?

Rule of thumb: 100 partitions * the number of drives that you think you will ever have, carried up to the nearest power of 2.

96 Drives * 100 = 9600

2^13 = 8192

2^14 = 16384

Creating the Builder Files

Replicas

Go with 3.

Creating the Builder Files

Mini Part Hours

Ensures that only one replica is in flight

A good default configuration setting is 24 hours.

swiftstack

Creating the Builder Files

```
$ ls /etc/swift
account.builder  backups  container.builder
object.builder (+ *.conf)
```

Extra Credit

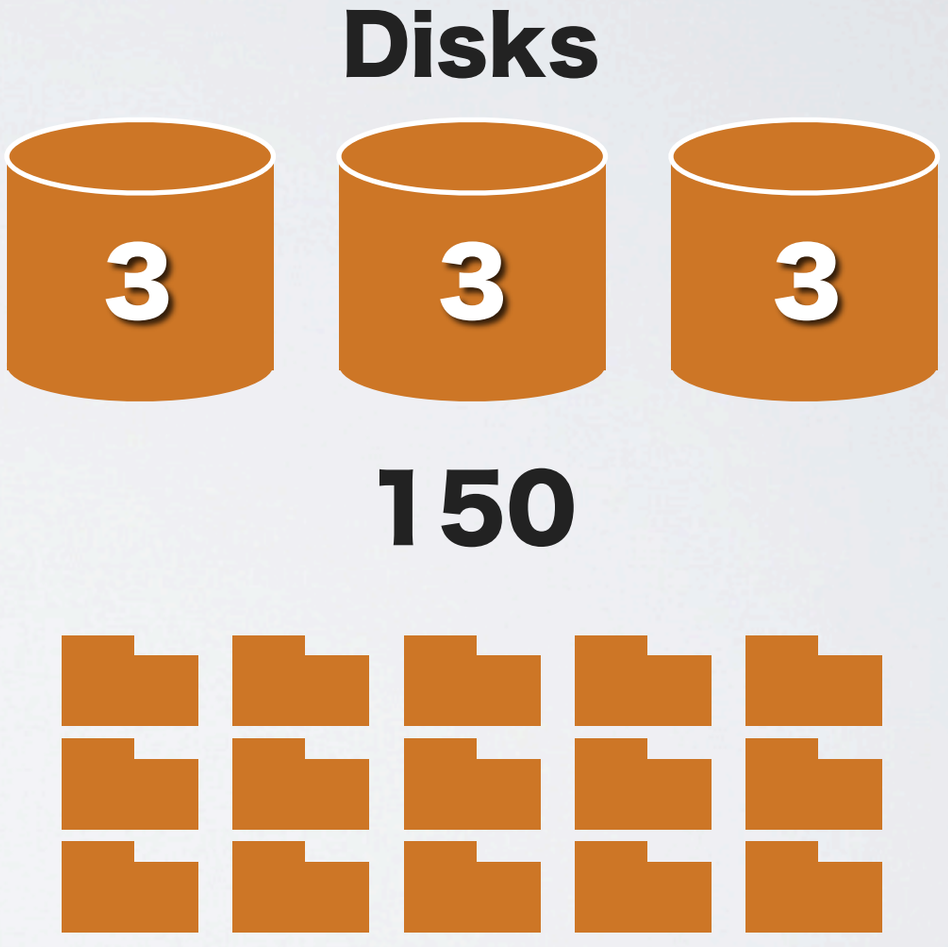
```
# python
Python 2.7.1+ (r271:86832, Apr 11 2011, 18:05:24)
[GCC 4.5.2] on linux2
Type "help", "copyright", "credits" or "license" for more
information.
>>> import pickle
>>> print pickle.load(open('object.builder'))
{'_replica2part2dev': None, '_last_part_gather_start': 0,
'min_part_hours': 1, 'replicas': 3, 'parts': 1024, 'part_power':
10, 'devs': [], 'devs_changed': False, 'version': 0,
'_last_part_moves_epoch': None, '_last_part_moves': None,
'_remove_devs': []}
```

Creating the Rings: Zones

```
swift-ring-builder  (account|container|object).builder  add
 z<zone>-<ip>:<port>/<device_name>  <weight>
```

Creating the Rings: Weights

```
swift-ring-builder  (account|container|object).builder  add
 z<zone>-<ip>:<port>/<device_name>  <weight>
```



swiftstack

Creating the Rings: Adding Devices

Validate your rings

```
# swift-ring-builder account.builder
# swift-ring-builder container.builder
# swift-ring-builder object.builder
```

Note that there are no partitions assigned to any drives yet. We must use “rebalance” to assign partitions and create the actual ring files.

swiftstack

Creating the Rings: Do It (Rebalance)!

```
# cd /etc/swift
# swift-ring-builder account.builder rebalance
# swift-ring-builder container.builder rebalance
# swift-ring-builder object.builder rebalance

# ls /etc/swift/*.ring.gz
account.ring.gz container.ring.gz object.ring.gz
```

Validate your rings

```
# swift-ring-builder account.builder
# swift-ring-builder container.builder
# swift-ring-builder object.builder
```

Let's Start Swift!

```
# swift-init main restart
```

Handy Commands

Tail the log file

```
$ tail -f /var/log/swift/all.log
```

Get a Token!

```
$ curl -v
    -H 'X-Auth-User: admin:admin'
    -H 'X-Auth-Key: admin'
     http://127.0.0.1:8080/auth/v1.0/

< HTTP/1.1 200 OK
< X-Storage-Url: http://127.0.0.1:8080/v1/AUTH_admin
< X-Storage-Token: AUTH_tk265318ae5e7e46f1890a441c08b5247f
< X-Auth-Token: AUTH_tk265318ae5e7e46f1890a441c08b5247f
< X-Trans-Id: txc75adf112791425e82826d6e3989be4d
< Content-Length: 0
```

Extra Credit: Token in memcache?

```
# python
>>> import swift.common.memcached as memcached
>>> memcache = memcached.MemcacheRing(['127.0.0.1:11211'])
>>> print memcache.get('AUTH_/user/admin:admin')
AUTH_tk265318ae5e7e46f1890a441c08b5247f
>>> print memcache.get('AUTH_/token/AUTH_tk265318ae5e7e46f1890a441c08b5247f')
(1308804765.9103661, 'admin,admin:admin')
>>>    (Hit Ctrl-D to exit)
```

swiftstack

Use a Token!

```
$ curl -v -H 'X-Auth-Token: <your x-auth-token>' <your x-storage-url>

< HTTP/1.1 204 No Content
< X-Account-Object-Count: 0
< X-Account-Bytes-Used: 0
< X-Account-Container-Count: 0
< Accept-Ranges: bytes
< X-Trans-Id: txafe3c83ed76e46d2a9536dd61d9fcf09
< Content-Length: 0
```

swiftstack

Uploading

Handy Command

Let's use the command 'swift' from here on out.

```
$ swift -U admin:admin -K admin
    -A http://127.0.0.1:8080/auth/v1.0/  upload
    test_container /vmlinuz

$ swift -U admin:admin -K admin
  -A http://127.0.0.1:8080/auth/v1.0/  list
  test_container
```

Where is the object?

```
# cd /srv/node
# find . -name '*.data'
# swift-object-info /full/path/to/file.data
# swift-get-nodes /etc/swift/object.ring.gz \
AUTH_admin test_container vmlinuz
```

Replication

Start Swift replicator services

```
# swift-init account-replicator start
# swift-init container-replicator start
# swift-init object-replicator start
```

Is rsync running?

```
$ rsync localhost::
$ rsync localhost::account
$ rsync localhost::container
$ rsync localhost::object
```

DESTROY!!!!

Delete all data in a zone!

```
# cd /srv/node/d1
# find .
# rm -rf *; find .
# sleep 20
# find .
```

Handy Commands

Tail the log file

```
# tail -f /var/log/swift/all.log
# tail -f /var/log/rsyncd.log
```

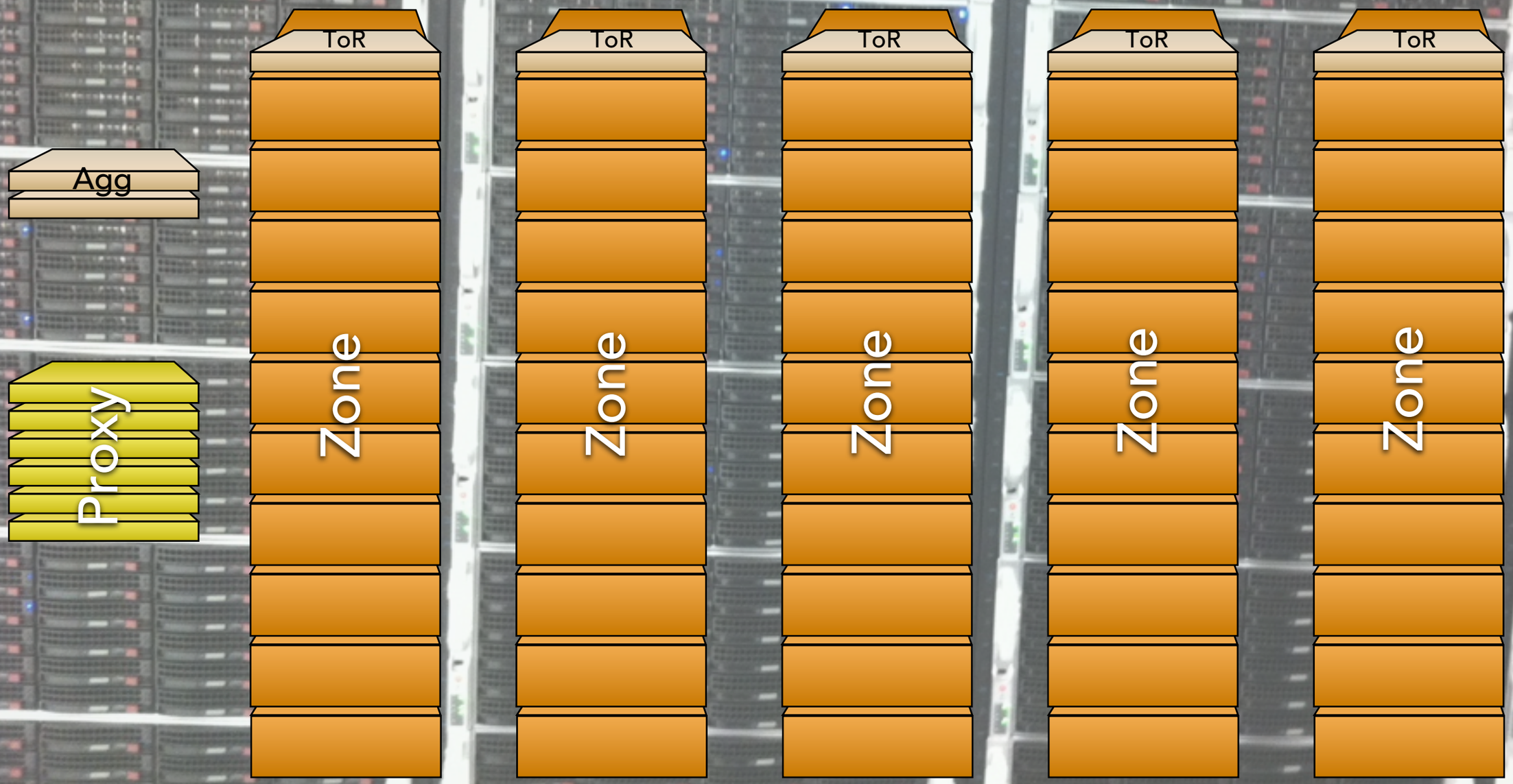
24-36 disk systems
SATA desktop drives
24-48 GB RAM

100 Terabyte

2 Proxy/LB Servers
4 Object Stores (36-drive, 3TB)

1.5 Petabyte

2 Agg Switches
6 Proxy/LB Servers
5 ToR Switches
50 Object Stores (36-drive / 3TB



Thank You!

This presentation + VBox Image

Swift technical white paper

Contact: joe@swiftstack.com

joe@swiftstack.com
darrell@swiftstack.com

swiftstack