redhat.

# INTRODUCTION TO SWIFT OBJECT STORAGE

## UNDERSTANDING THE ARCHITECTURE AND USE CASES

Thiago da Silva

redhat

$WHOAMI

**Thiago da Silva**

**Swift Core Reviewer**

thiago@redhat.com

thiagodasilvablog.wordpress.com

irc: tdasilva

thiagol11

thiagodasilva

# AGENDA

- Introduction
- Use cases
- Requirements
- Swift
- Questions

# DATA GROWTH

## BY 2020:

- **IDC**: 44 zettabytes of data created annually
- **Cisco**: The number of mobile devices connected to the internet will be about 1.5 per capita
- **Forbes**: 1.7 megabytes of new information will be created every second for each person on the planet

# Types of Data

# TYPES OF DATA

- **Structured:**
  - Organized
  - Relational
  - Contextualized
  - Easy to consume and analyse

SQL

# TYPES OF DATA

- **Unstructured:**
  - Data without any structure, order or schema
  - Documents, media files
  - 90% of generated data today

# Different storage systems for different types of data

# TYPES OF DATA STORAGE

- **Block**
- **File**
- **Object**
  - Logical architecture to manage data as objects
  - Objects contain data, metadata and unique id
  - Flat address structure
  - Always access objects as whole

# Use Cases for Object Storage

# USE CASES

## PUBLIC/PRIVATE CLOUDS

SOFTLAYER® an IBM Company

many 10+ PB clusters

NTT Communications

many PBs and 10+ millions of users

OVH.com Innovation is Freedom

75+ PB

# USE CASES

## WEB/MOBILE APPLICATIONS


# USE CASES

## DATA ARCHIVAL



# OBJECT STORAGE REQUIREMENTS

- Durability
- Availability
- Accessibility
- Scalability
- Low cost

# SWIFT

*an OpenStack Community Project*

# OPENSTACK SWIFT

Swift is a highly available, distributed, eventually consistent object/blob store. Organizations can use Swift to store lots of data efficiently, safely, and cheaply.

# OPENSTACK SWIFT

## COMMUNITY

- Founding project of OpenStack
- ~8 years in production
- Active community
  - 700+ contributors (total)
- ~30K LoC
- ~100K tests LoC

# OPENSTACK SWIFT

## OVERVIEW

- Access by REST API
  - URL is Object key: http://swift.com/v1/acc/cont/obj
- Objects grouped in Containers (buckets)
- Highly durable and distributed
  - Data replicated many times (or EC)
  - No SPoF
- Eventual Consistency (CAP)
  - Designed for HA and partition tolerance

# FUNCTIONALITY

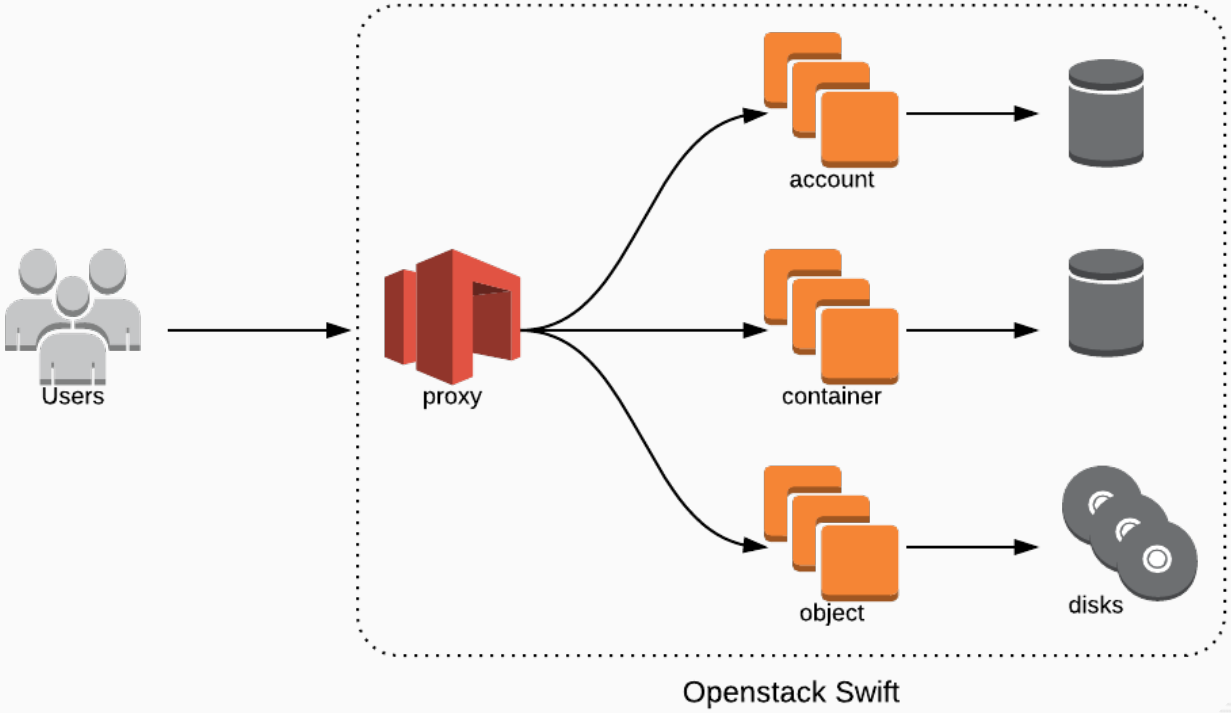
- Metadada
  - account, container, object
- **Object Versioning**
- **Object Expiration**
- Quota
  - Account/Container
- Encryption

# FUNCTIONALITY

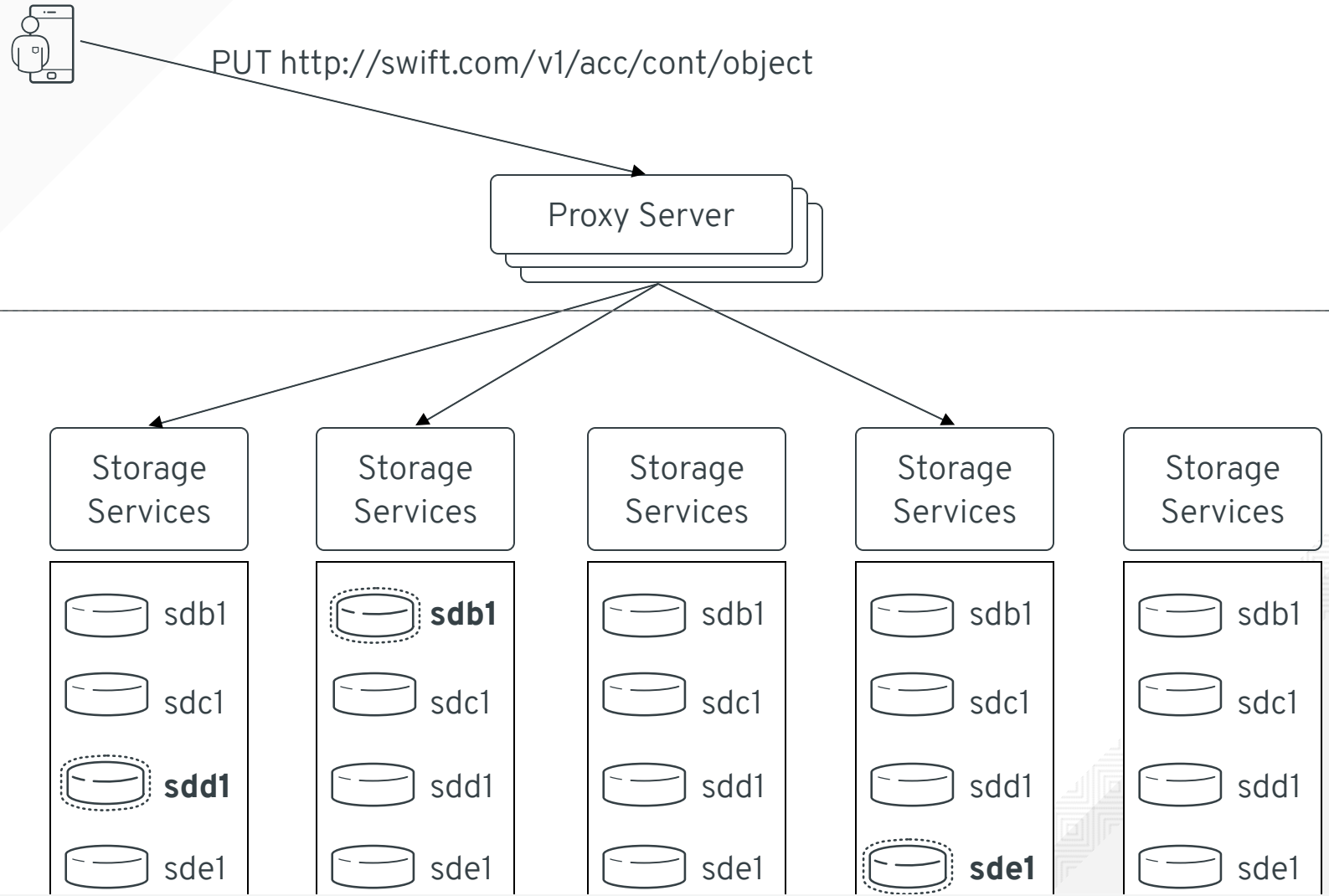
- Support for large objects
  - DLO, **SLO**
- Static Web
- **TempURL**
- **Authentication**
- ACLs
- Swift3 (third-party)

# Swift Architecture

# ARCHITECTURE

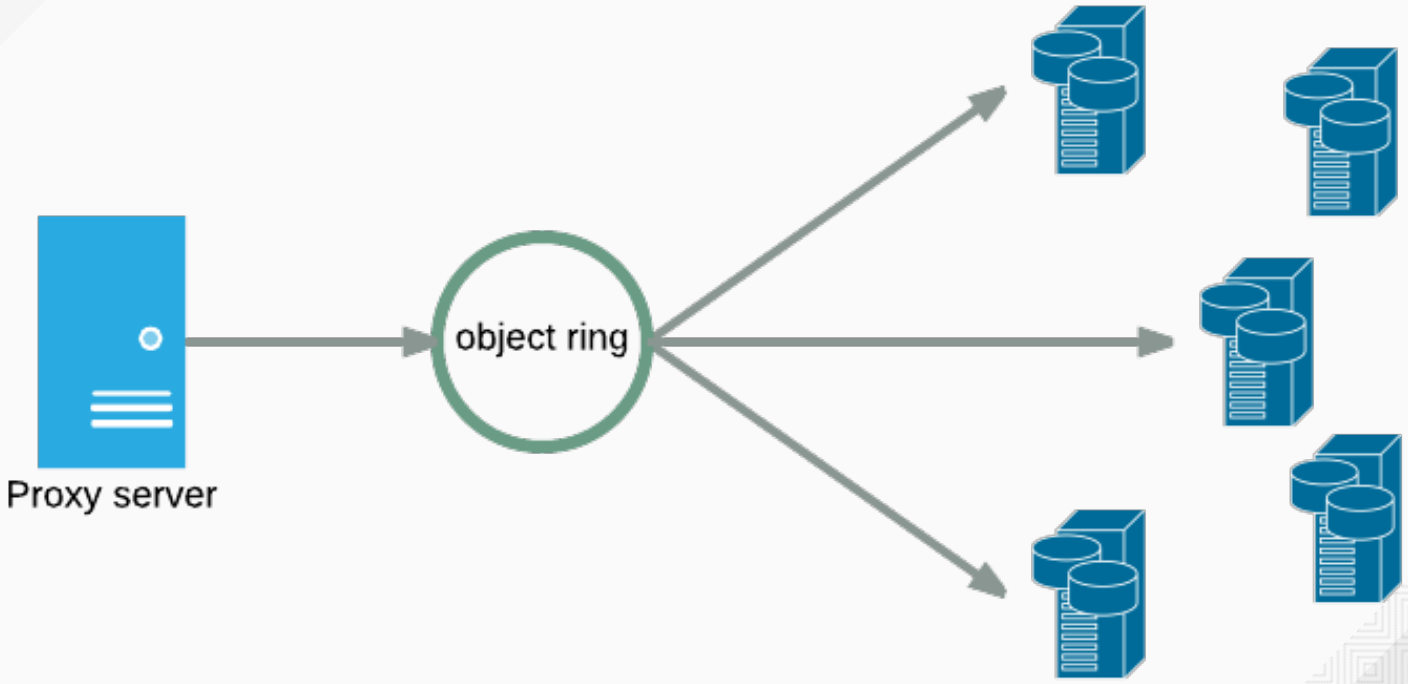


# ARCHITECTURE



# ARCHITECTURE

## THE RING



# ARCHITECTURE

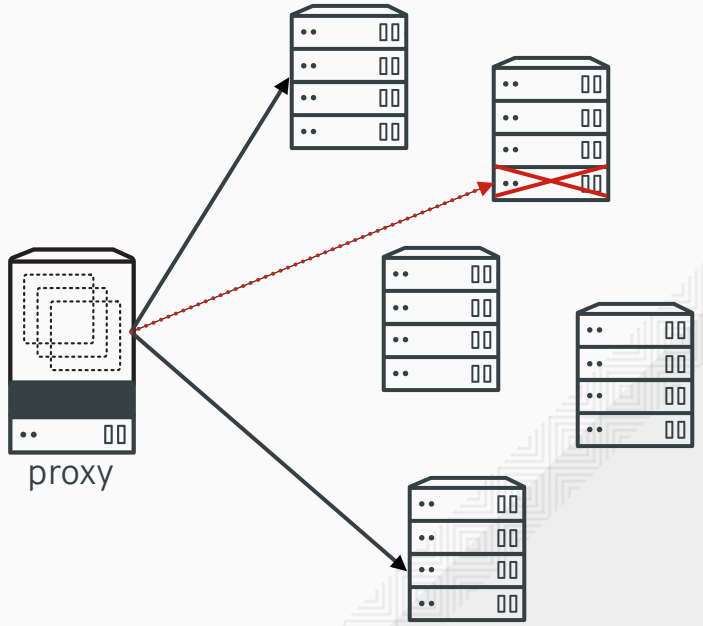
## STORAGE POLICIES

- Support for different data storage rules (i.e. policies)
  - 3 replicas
  - 2 replicas
  - Erasure Coding
  - Geographical location

# CONSISTENCY ENGINE

## BACKGROUND SERVICES

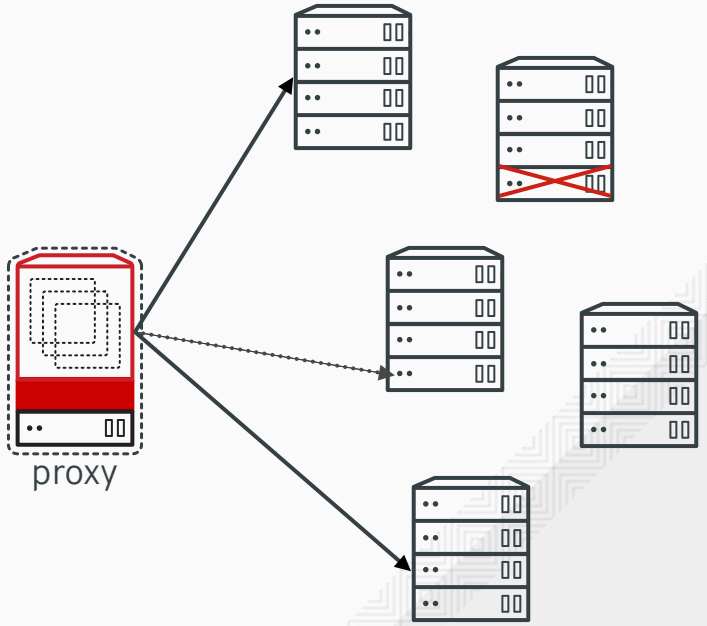
- Provide data healing (replicators, reconciler, reconstructor)
- bit-rot detection (auditors)
- update container, account information (updaters)
- expire objects (object-expirer)



# CONSISTENCY ENGINE

## BACKGROUND SERVICES

- Provide data healing
- bit-rot detection
- update container, account information
- expire objects

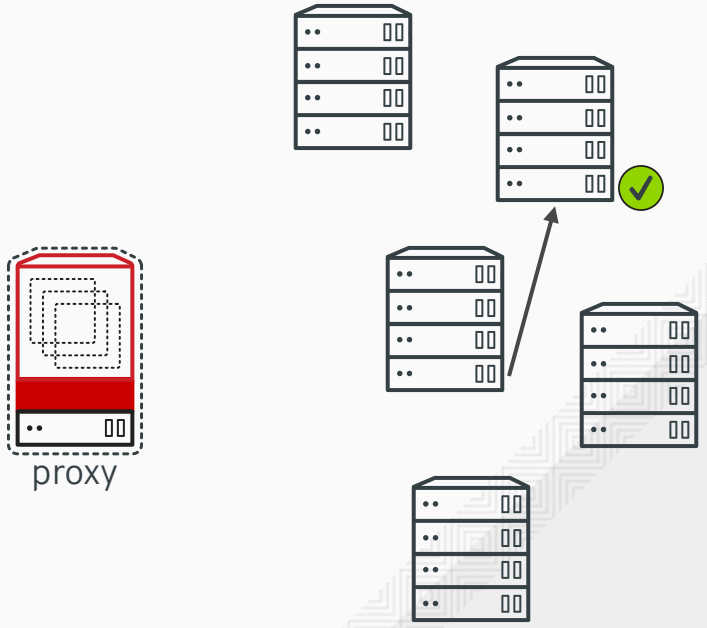



*"must guarantee durability, always write 3 replicas"*

# CONSISTENCY ENGINE

## BACKGROUND SERVICES

- Provide data healing
- bit-rot detection
- update container, account information
- expire objects



*"Swift sounds really cool, I'd like to contribute..."*

# WHAT WE ARE WORKING ON...

- Container sharding
- LSOF
- Data tiering
- New implementation of the S3 API middleware.
- Project hummingbird

# COME JOIN US...

- code: github.com/openstack/swift
- IRC: #openstack-swift on freenode

redhat®

# THANK YOU!