LeoFS

Reliable, Scaling and High Performance Storage System

Yosuke Hara - @yosukehara

A Researcher of R.I.T. and **Tech Lead LeoFS**

with **Masahiro Sanjo**, Coordinator of R.I.T.

楽天 Rakuten

R.I.T.
Rakuten
Institute of Technology

LeoFS is an Unstructured Object Storage for the Web and a highly available, distributed, eventually consistent storage system.

LeoFS was published as OSS on July of 2012

leo-project.net/leofs

楽天 Rakuten

Overview

Brief Benchmark Report

Multi Data Center Replication

NFS Support

LeoFS Administration at Rakuten

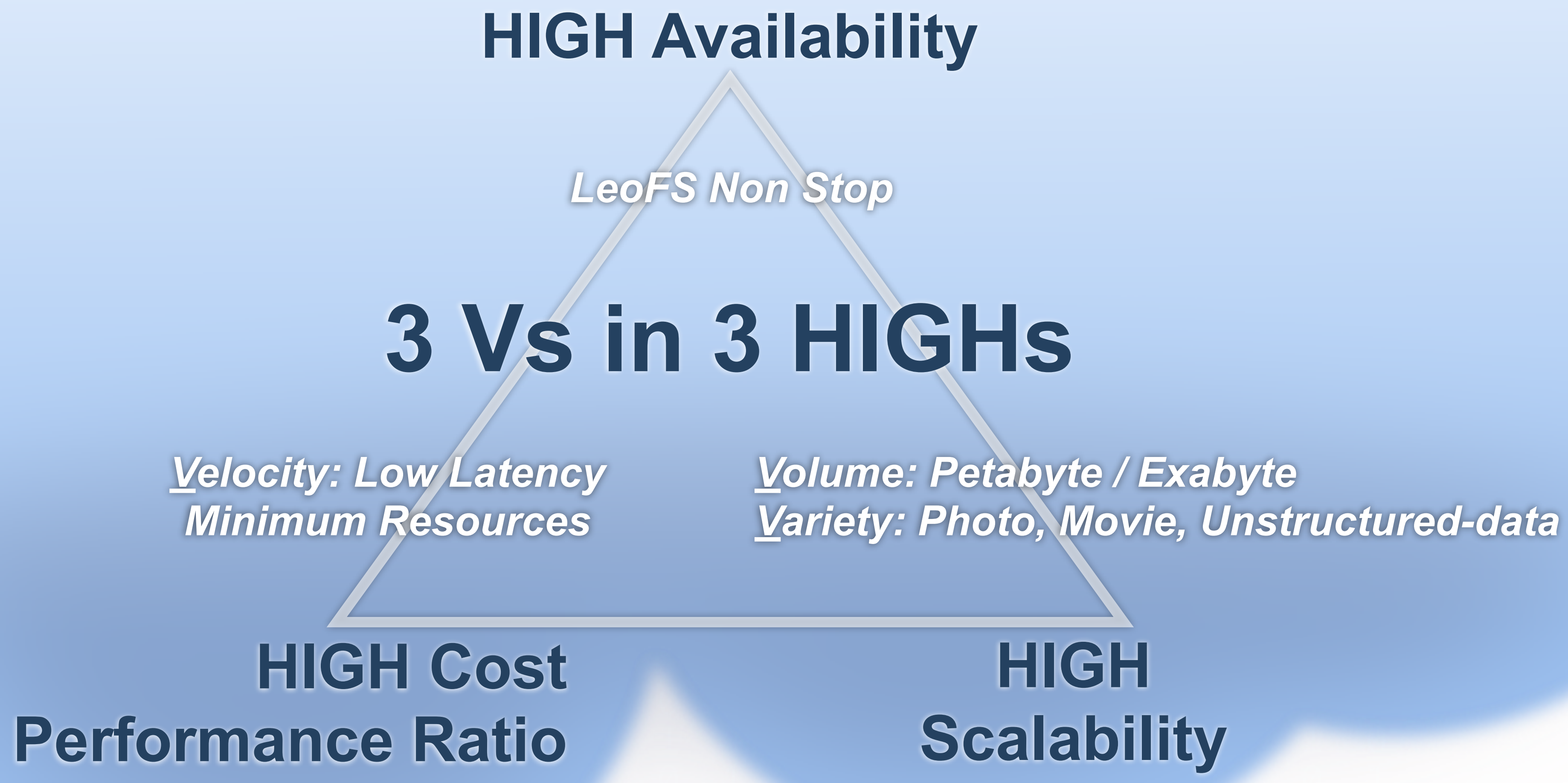
Future Plans

- LeoFS QoS

Overview

LeoFS

The Lion of Storage Systems



HIGH Availability

LeoFS Non Stop

3 Vs in 3 HIGHs

Velocity: Low Latency
Minimum Resources

Volume: Petabyte / Exabyte
Variety: Photo, Movie, Unstructured-data

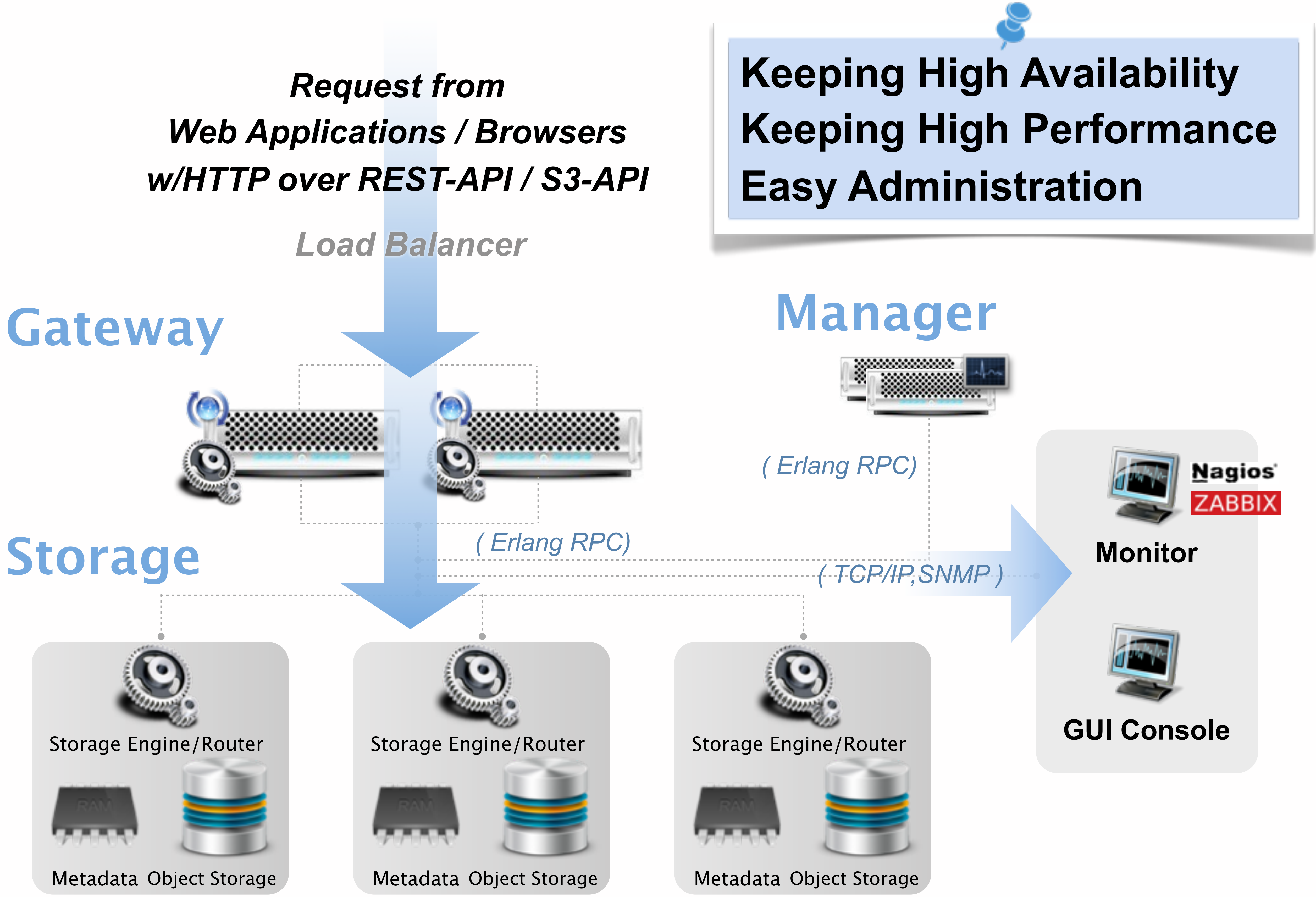
HIGH Cost
Performance Ratio

HIGH
Scalability

楽天 Rakuten

LeoFS Overview

Keeping High Availability
Keeping High Performance
Easy Administration



楽天 Rakuten

LeoFS Gateway

LeoFS Overview - Gateway

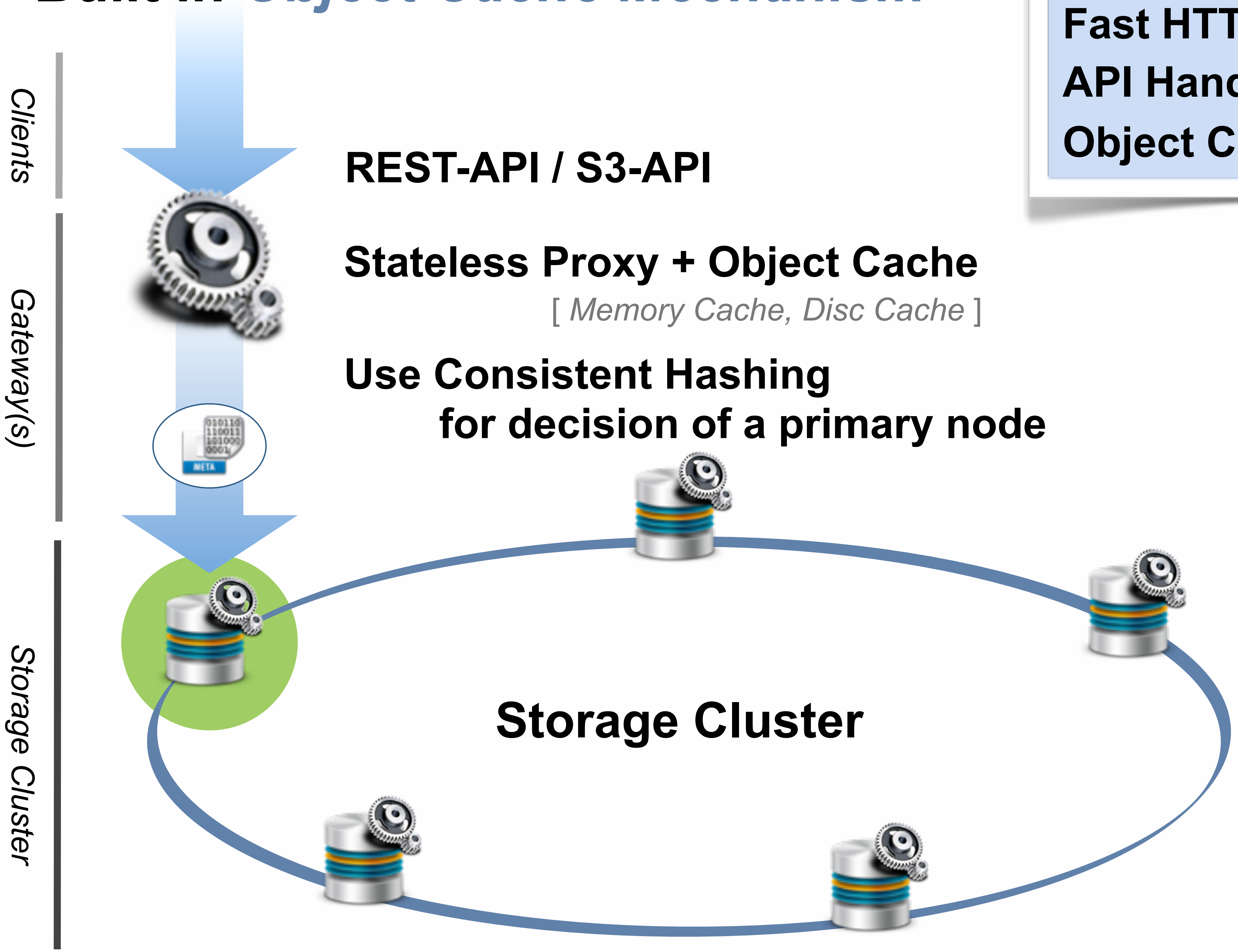
HTTP Request and Response

Built in *Object Cache Mechanism*

Fast HTTP Server - Cowboy

API Handler

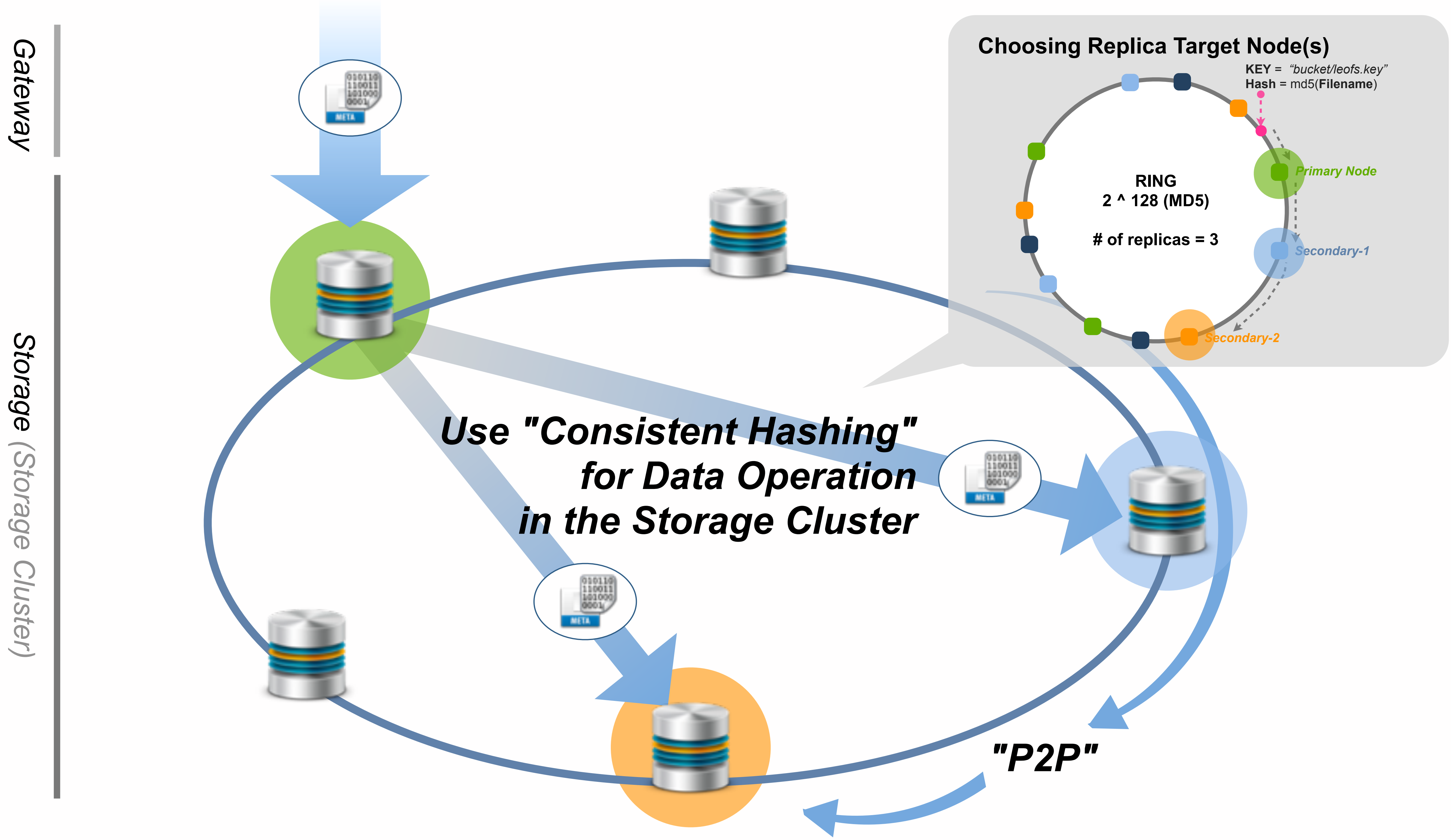
Object Cache Mechanism


Clients
Gateway(s)
Storage Cluster
REST-API / S3-API
Stateless Proxy + Object Cache
[*Memory Cache, Disc Cache*]
Use Consistent Hashing
for decision of a primary node
META
Storage Cluster

LeoFS Storage

LeoFS Overview - Storage

WRITE: Auto Replication
READ : Auto Repair of an Inconsistent Object with Async



Gateway

Storage (Storage Cluster)

Choosing Replica Target Node(s)

KEY = *"bucket/leofs.key"*
Hash = md5(**Filename**)

RING
2 ^ 128 (MD5)

# of replicas = 3

Primary Node

Secondary-1

Secondary-2

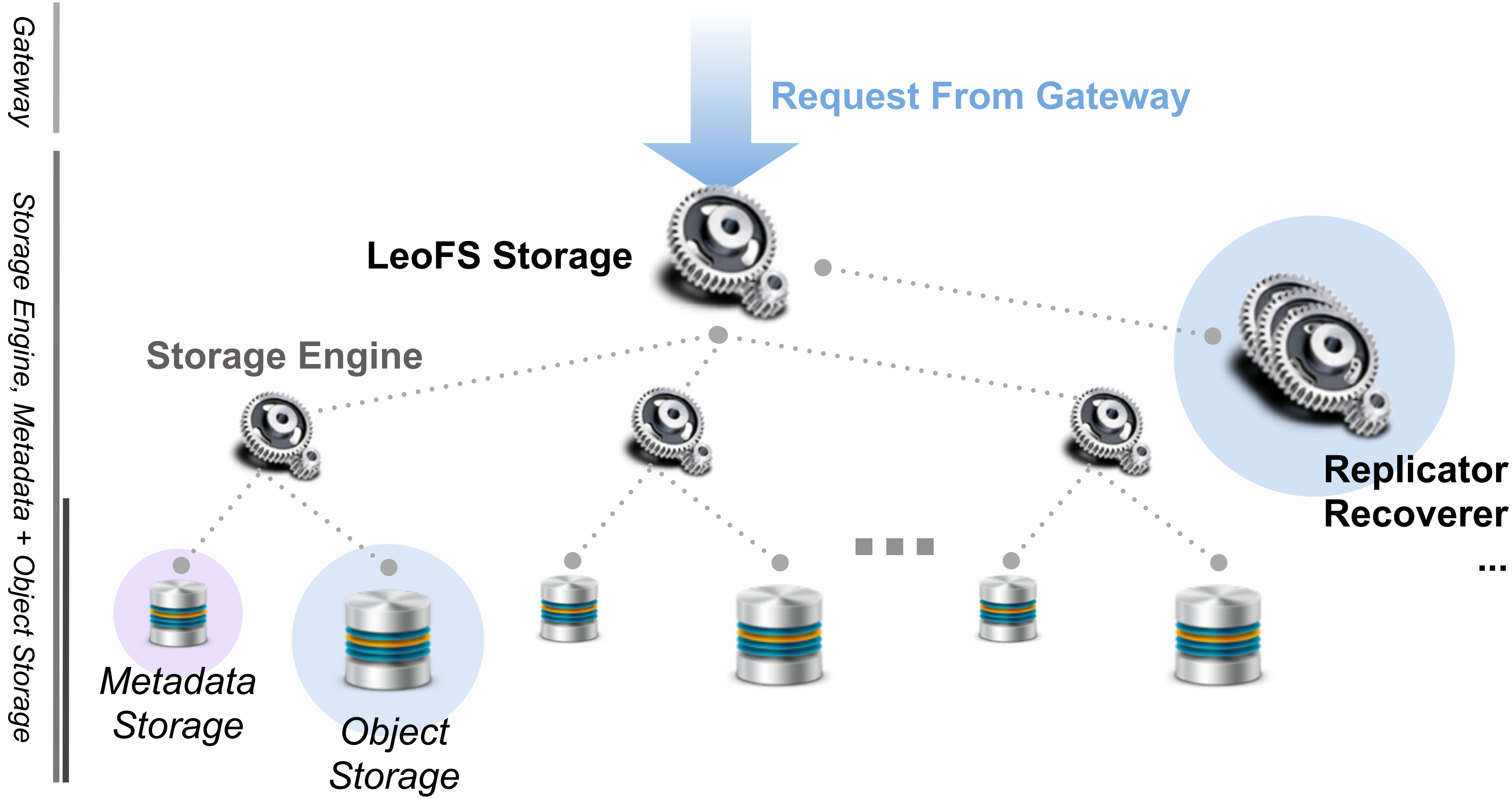
Use "Consistent Hashing" for Data Operation in the Storage Cluster

"P2P"

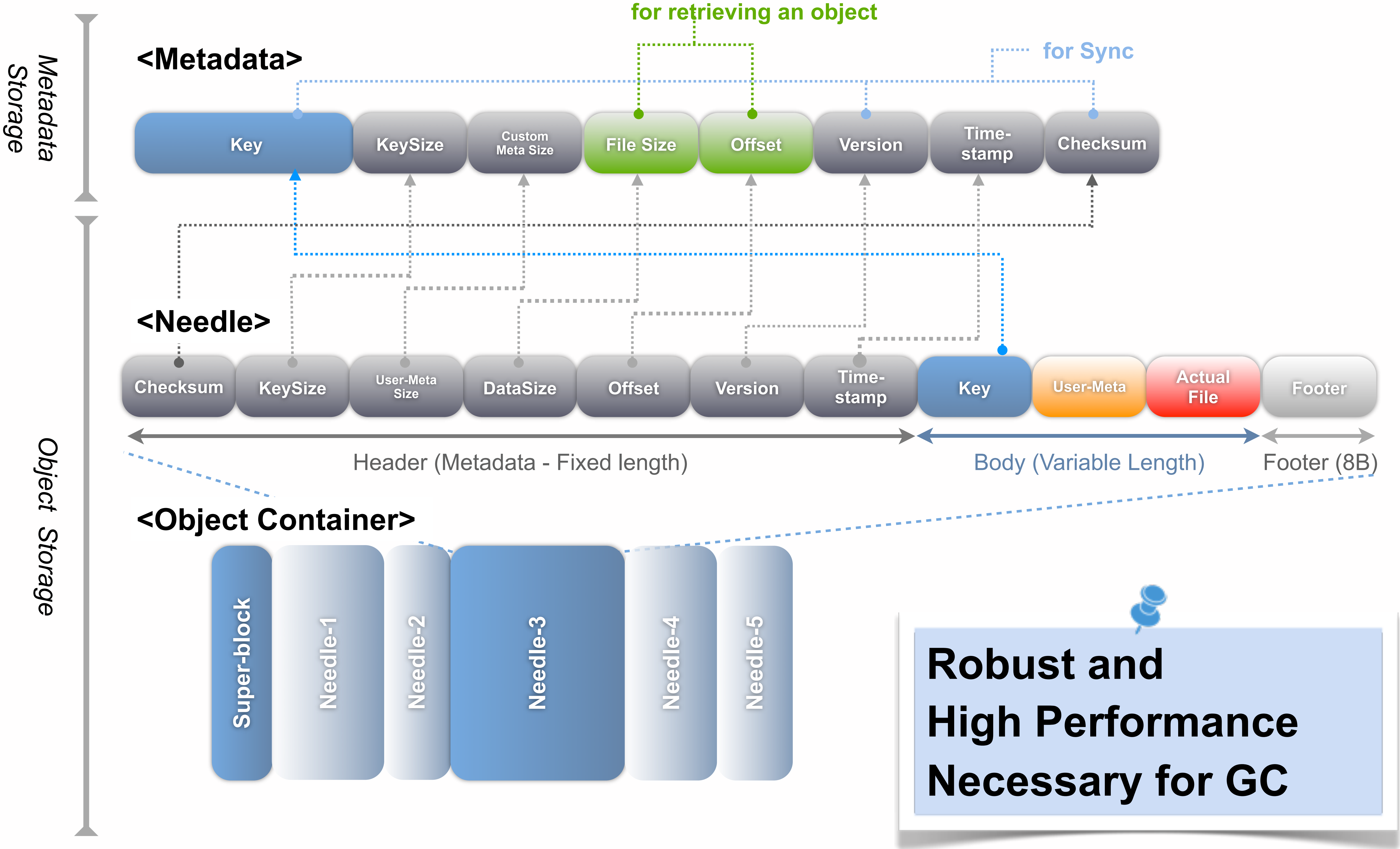
LeoFS Overview - Storage

Storage consists of *Object Storage* *and* *Metadata Storage*

Includes *Replicator* and *Recoverer* for the eventual consistency



LeoFS Overview - Storage - Data Structure



Robust and
High Performance
Necessary for GC

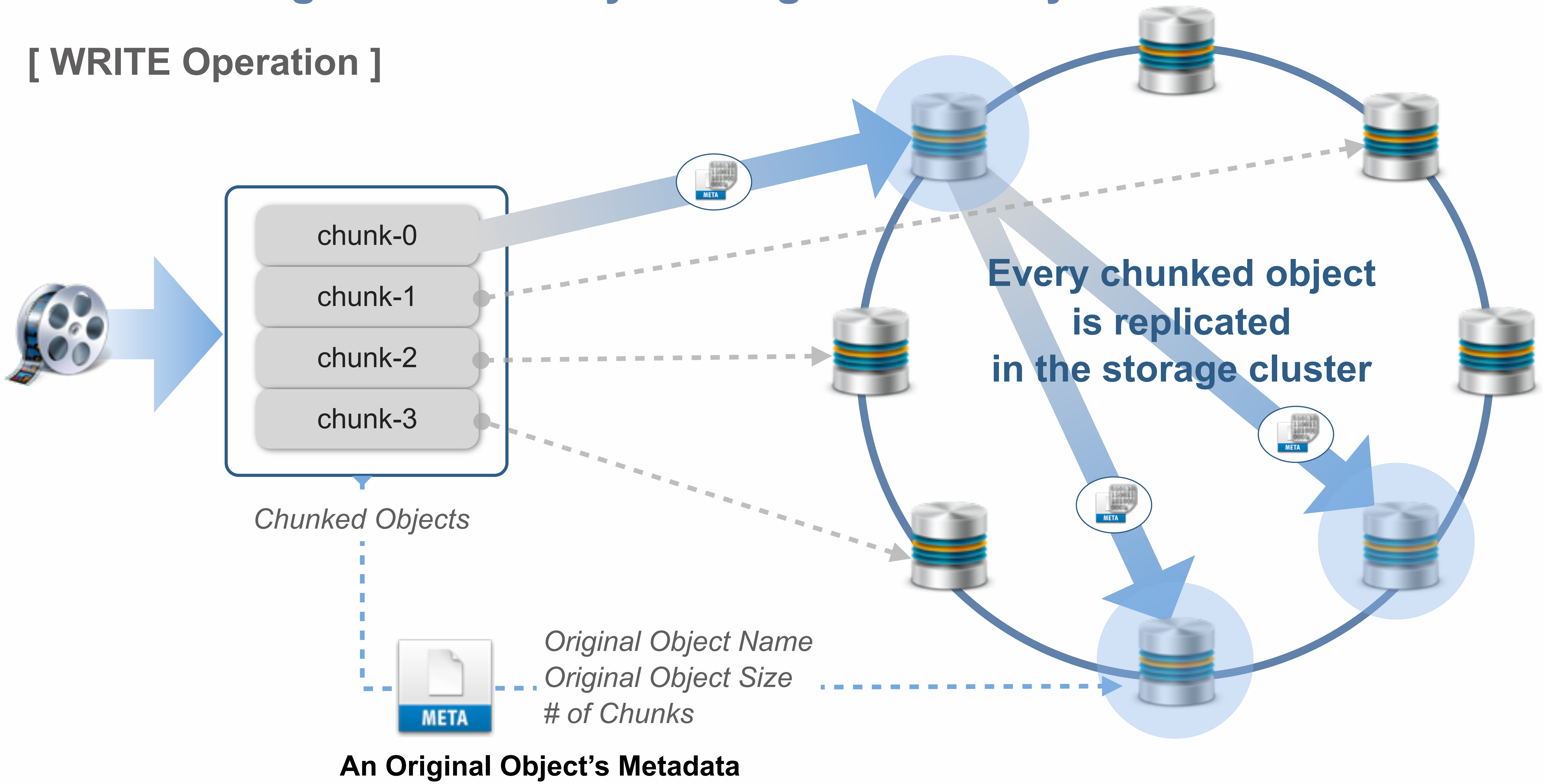
楽天 Rakuten

LeoFS Overview - Storage - Large Object Support

To Equalize Disk Usage in Every Storage Node

To Realise High I/O efficiency and High Availability

[WRITE Operation]



Client(s)

Gateway

Storage Cluster

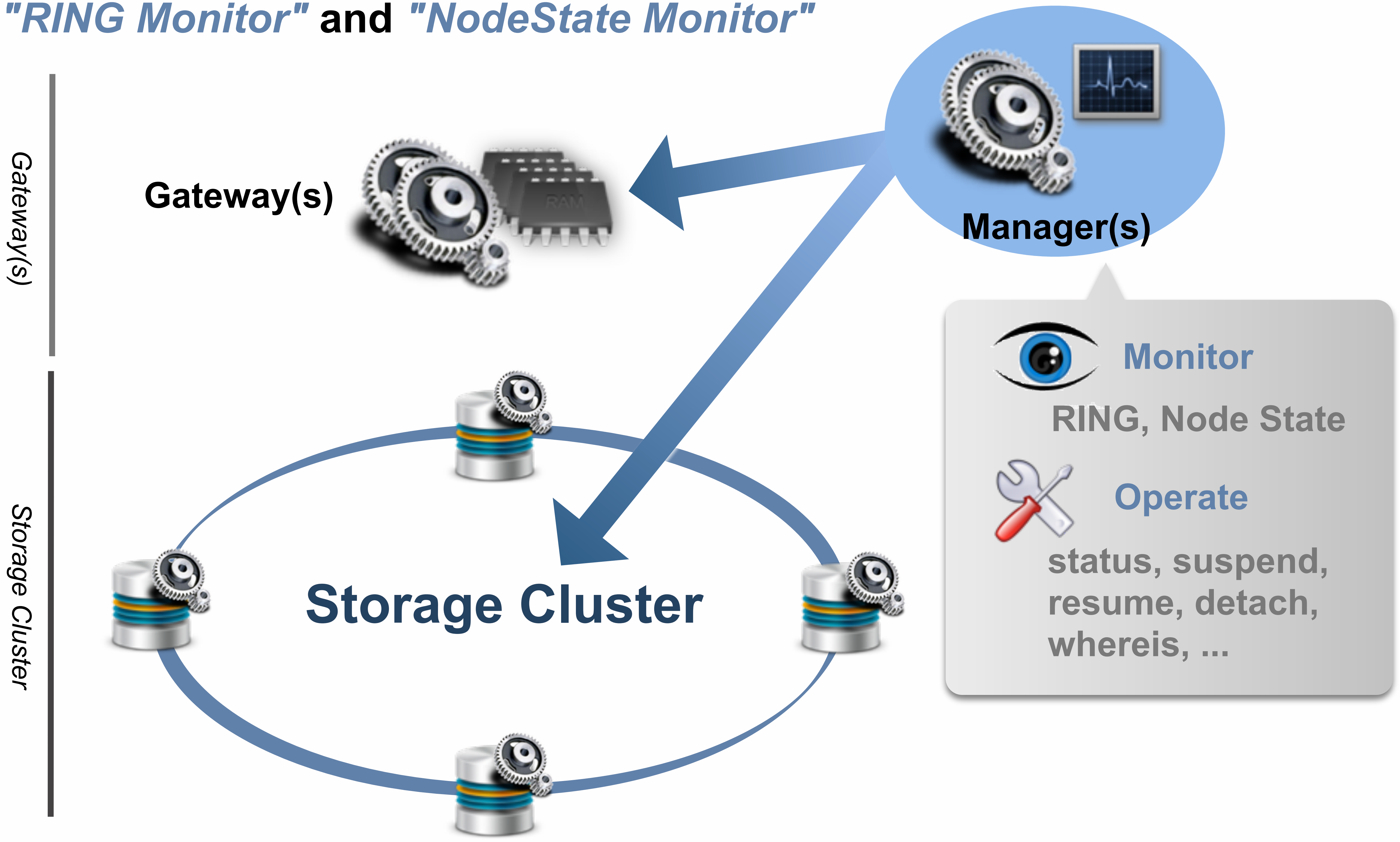
楽天 Rakuten

LeoFS Manager

LeoFS Overview - Manager

Operate LeoFS - Gateway and Storage Cluster

"RING Monitor" and *"NodeState Monitor"*



Gateway(s)

Storage Cluster

Gateway(s)

Manager(s)

Storage Cluster

Monitor

RING, Node State

Operate

status, suspend, resume, detach, whereis, ...

Brief Benchmark Report

Summary of the benchmark results

LeoFS kept in a stable performance through the benchmark

Bottleneck is Disk I/O

The cache mechanism contributed to reduce network traffic between Gateway and Storage

Brief Benchmark Report

1st Case:

Group of Value Ranges

Storage:5, Gateway:1, Manager:2
R:W = 9:1

source: https://github.com/leo-project/notes/tree/master/leofs/benchmark/leofs/20140605/tests/1m_r9w1_240min

2nd Case:

Group of Value Ranges

Storage:5, Gateway:1, Manager:2
R:W = 8:2

source: https://github.com/leo-project/notes/tree/master/leofs/benchmark/leofs/20140605/tests/1m_r8w2_120min

Brief Benchmark Report

Server Spec - Gateway:

CPU	Intel(R) Xeon(R) CPU X5650 @ 2.67GHz * 2 (12 cores / 24 threads)
Memory	96GB
Disk	HDD - 240GB RAID0
Network	10G-Ether

Server Spec - Storage x5:

CPU	Intel(R) Xeon(R) CPU X5650 @ 2.67GHz * 2 (12 cores / 24 threads)
Memory	96GB
Disk	HDD - 240GB RAID0 (System)
	HDD - 2TB RAID0 (Data)
Network	10G-Ether

Brief Benchmark Report - 1st Case (R:W=9:1)

Environment:

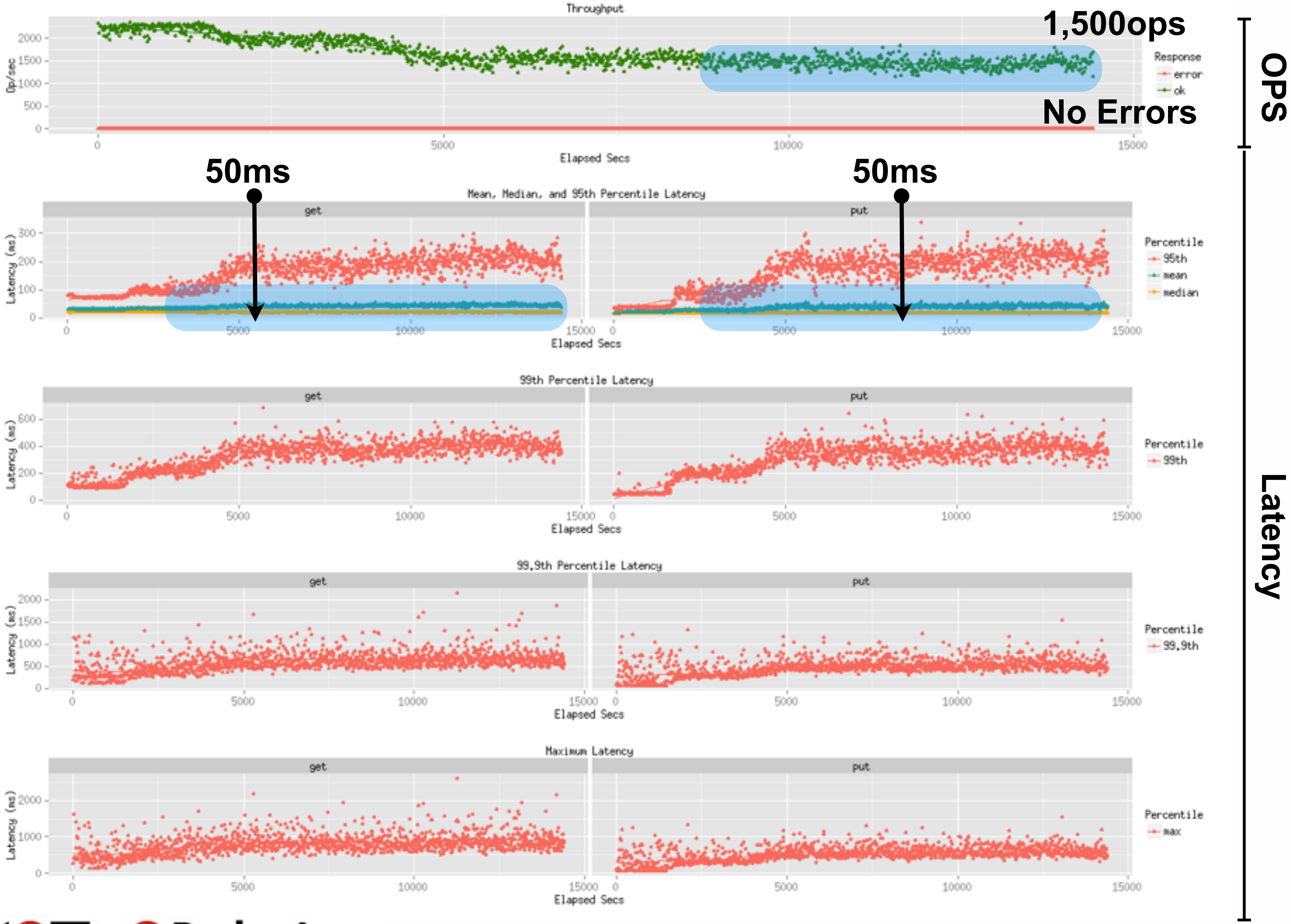
Network	10Gbps
OS	CentOS release 6.5 (Final)
Erlang	OTP R16B03-1
LeoFS	v1.0.2

System Consistency Level: [N:3, W:2, R:1, D:2]

Benchmark Configuration:

<table>
<tr><td>Duration</td><td>4.0h</td></tr>
<tr><td><b>R:W</b></td><td><b>9:1</b></td></tr>
<tr><td># of Concurrent Processes</td><td>64</td></tr>
<tr><td># of Keys</td><td>100,000</td></tr>
<tr><td>Value Size</td><td>
<table>
<tr><th colspan="2">Range (byte)</th><th>Percentage</th></tr>
<tr><td>1024</td><td>10240</td><td>24.00%</td></tr>
<tr><td>10241</td><td>102400</td><td>30.00%</td></tr>
<tr><td>10241</td><td>819200</td><td>30.00%</td></tr>
<tr><td>819201</td><td>1572864</td><td>16.00%</td></tr>
</table>
</td></tr>
</table>

Brief Benchmark Report - 1st Case (R:W=9:1)



1,500ops

No Errors

OPS

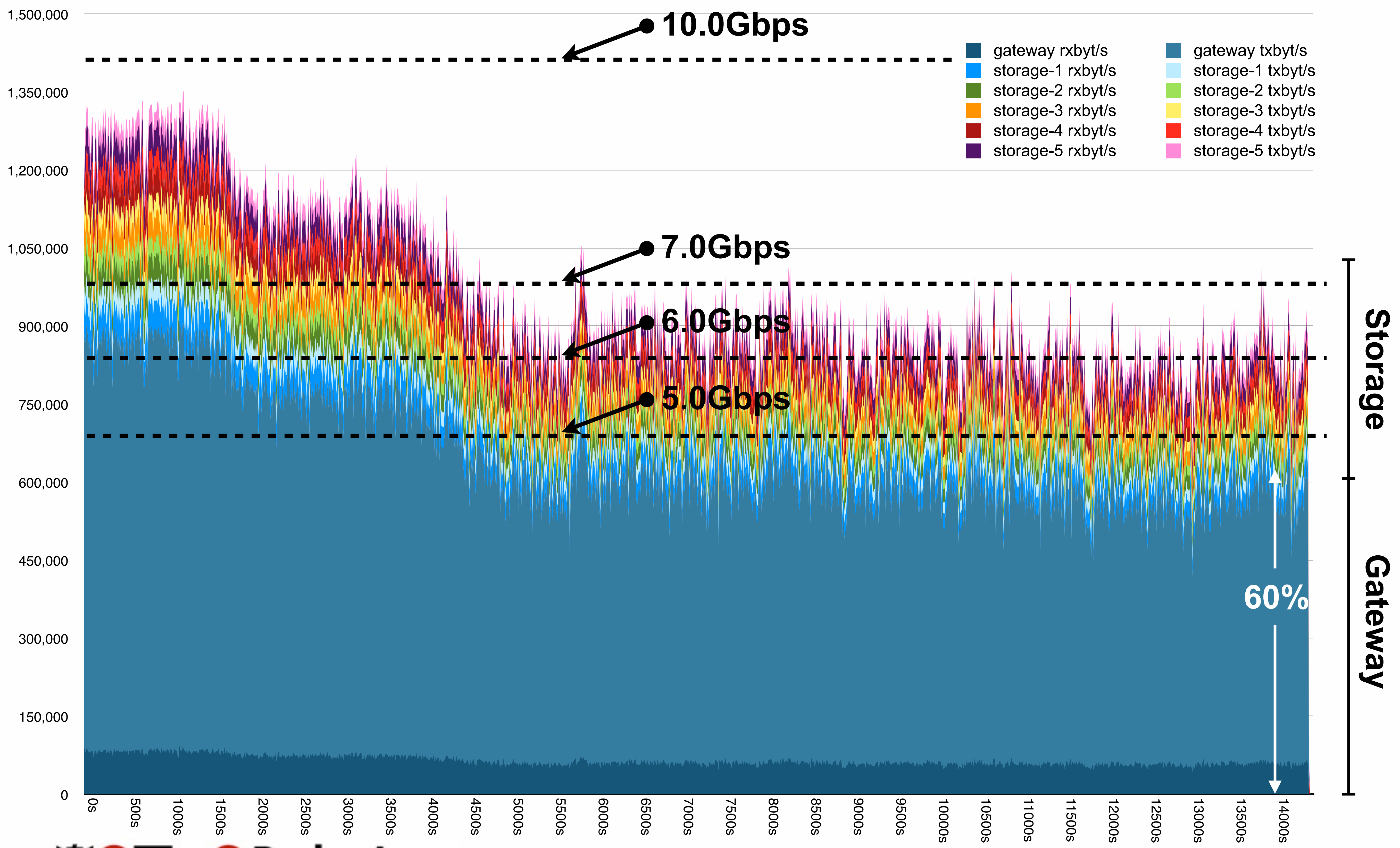
50ms

50ms

Latency

source: https://github.com/leo-project/notes/tree/master/leofs/benchmark/leofs/20140601/tests/1m_r9w1_240min

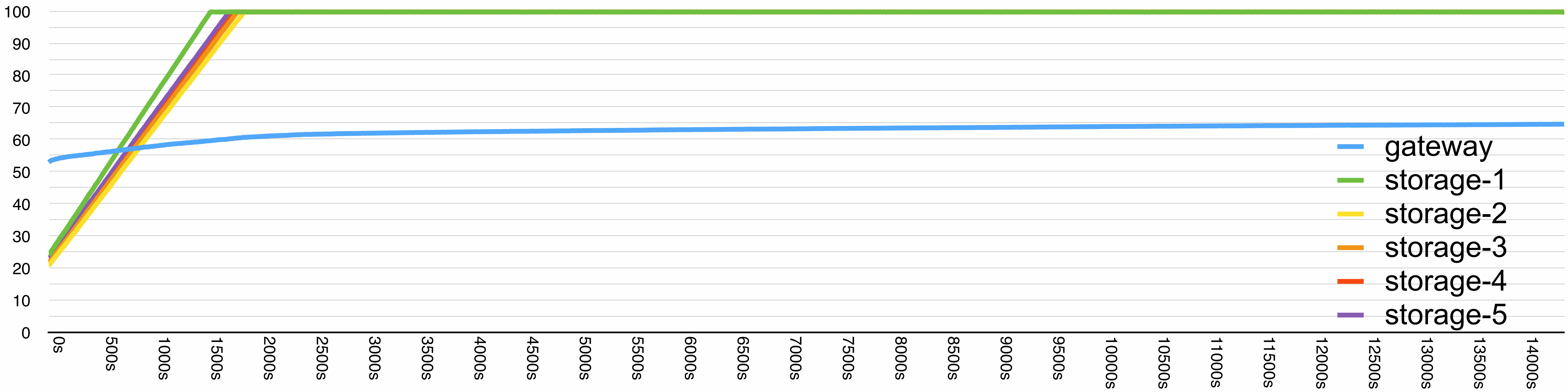
Brief Benchmark Report - 1st Case / Network Traffic



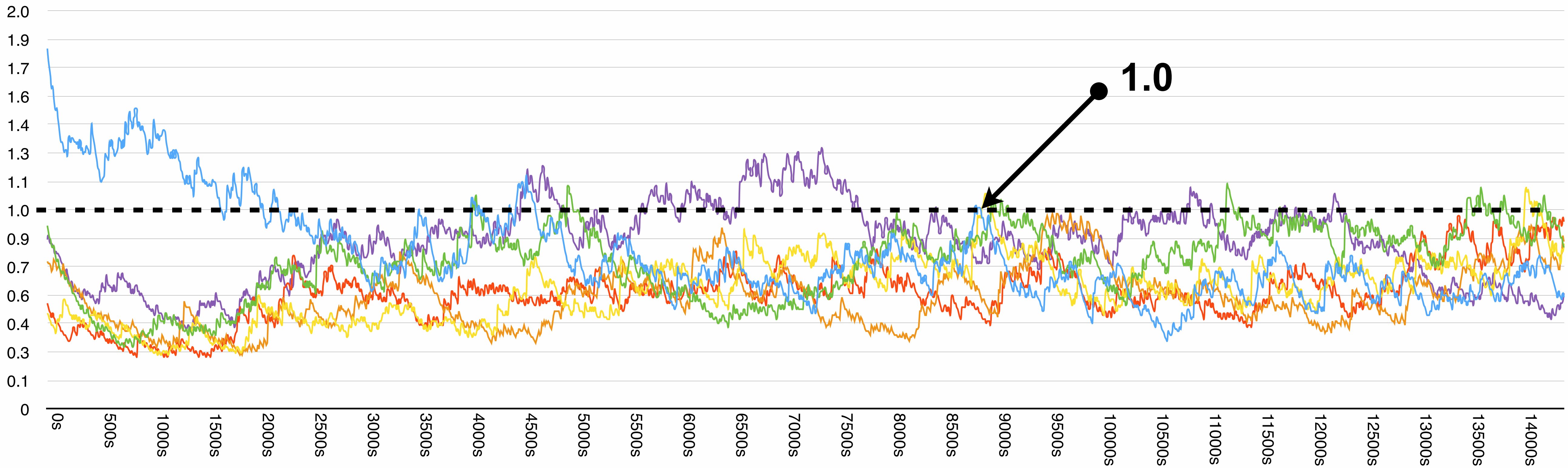
楽天 Rakuten

Brief Benchmark Report - 1st Case / Memory and CPU

Memory Usage



CPU Load 5min



Brief Benchmark Report - 2nd Case (R:W=8:2)

Environment:

Network	10Gbps
OS	CentOS release 6.5 (Final)
Erlang	OTP R16B03-1
LeoFS	v1.0.2

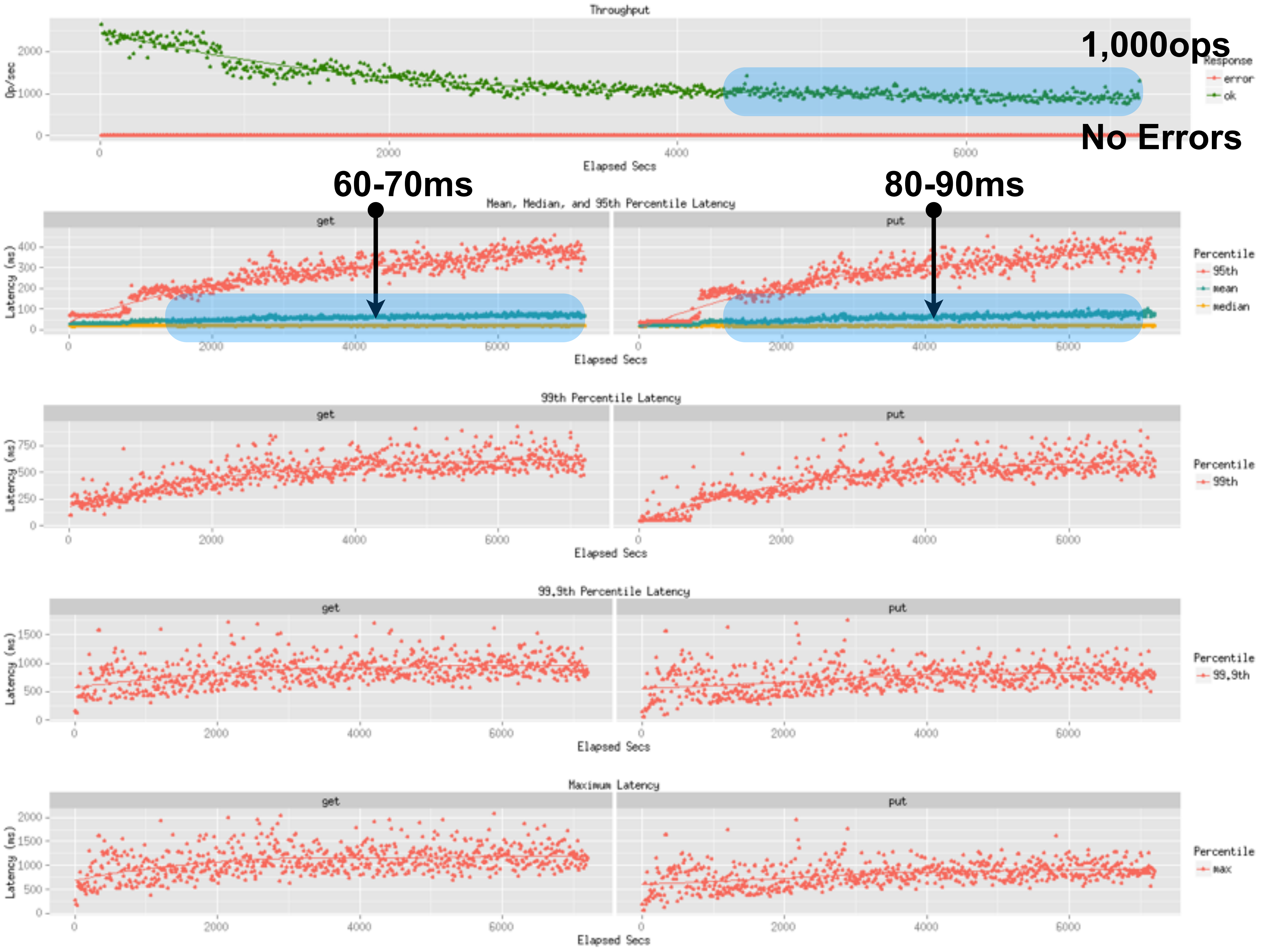
System Consistency Level: [N:3, W:2, R:1, D:2]

Benchmark Configuration:

<table>
<tr><td>Duration</td><td>2.0h</td></tr>
<tr><td>R:W</td><td>8:2</td></tr>
<tr><td># of Concurrent Processes</td><td>64</td></tr>
<tr><td># of Keys</td><td>100,000</td></tr>
<tr><td>Value Size</td><td>
<table>
<tr><th colspan="2">Range (byte)</th><th>Percentage</th></tr>
<tr><td>1024</td><td>10240</td><td>24.00%</td></tr>
<tr><td>10241</td><td>102400</td><td>30.00%</td></tr>
<tr><td>10241</td><td>819200</td><td>30.00%</td></tr>
<tr><td>819201</td><td>1572864</td><td>16.00%</td></tr>
</table>
</td></tr>
</table>

楽天 Rakuten

Brief Benchmark Report - 2nd Case (R:W=8:2)



1,000ops

No Errors

60-70ms

80-90ms

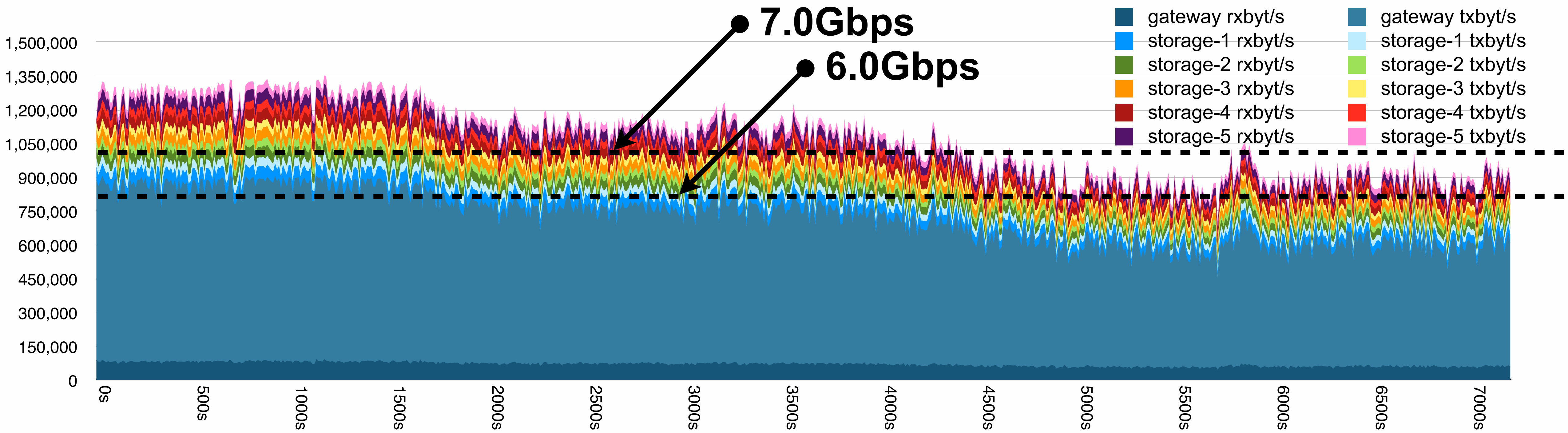
OPS

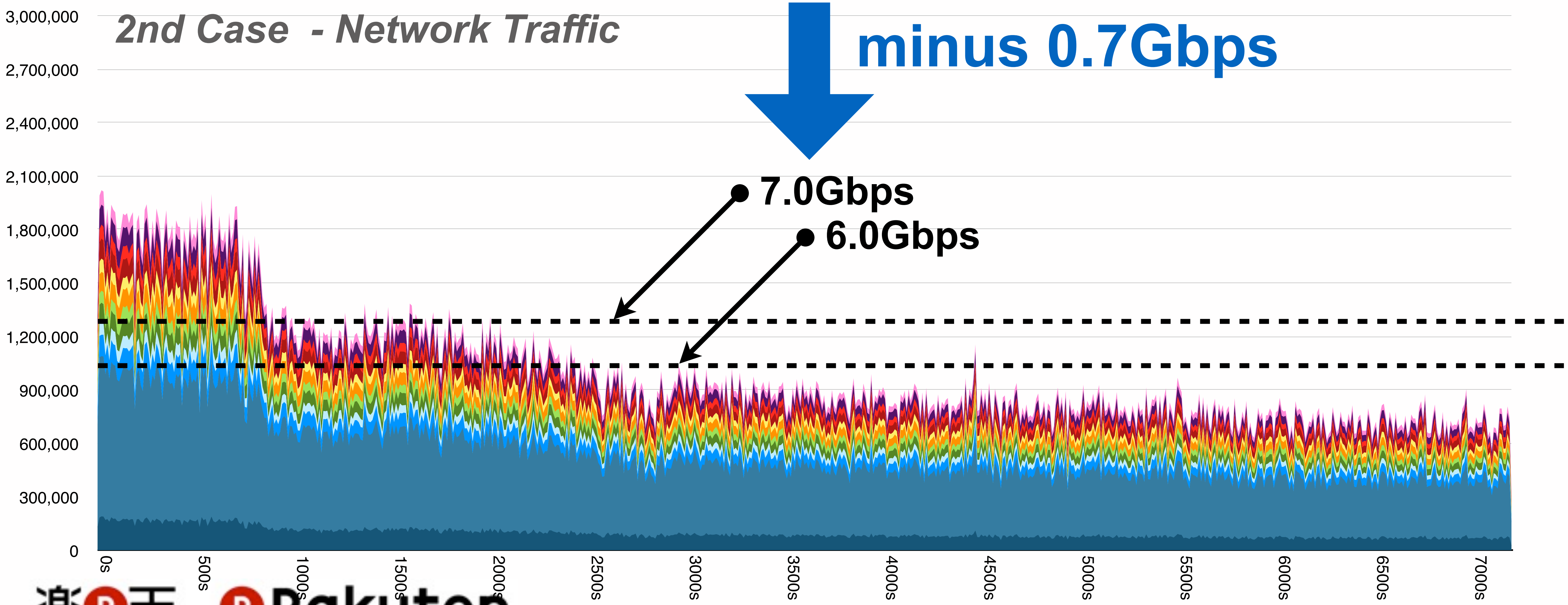
Latency

Compare 1st case with 2nd case

Brief Benchmark Report

1st Case - Network Traffic





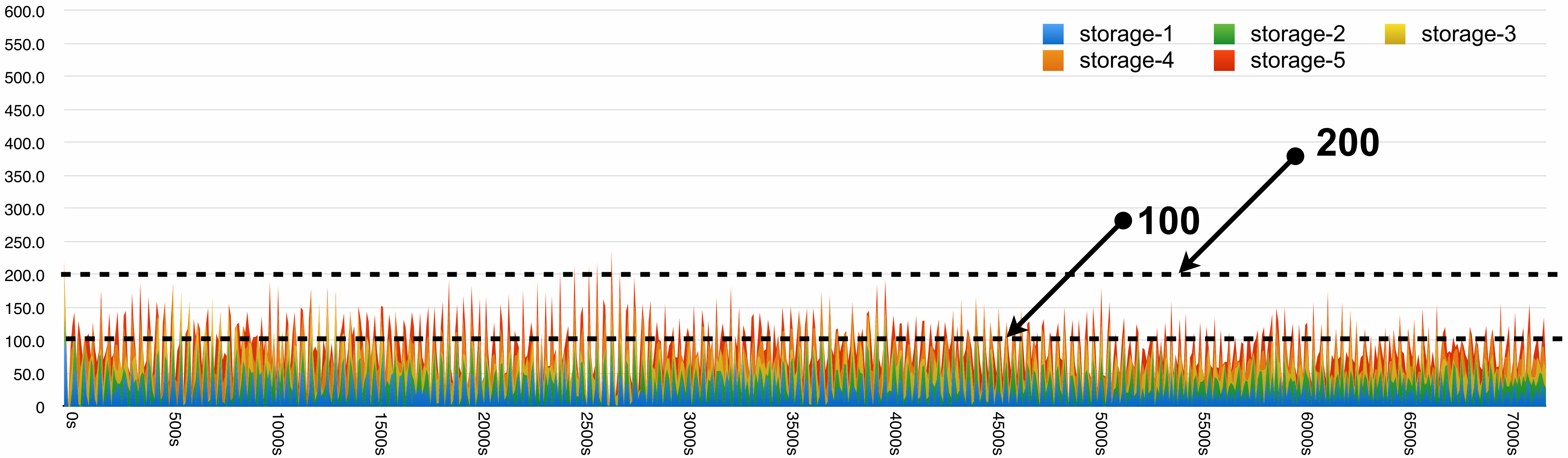
2nd Case - Network Traffic

minus 0.7Gbps

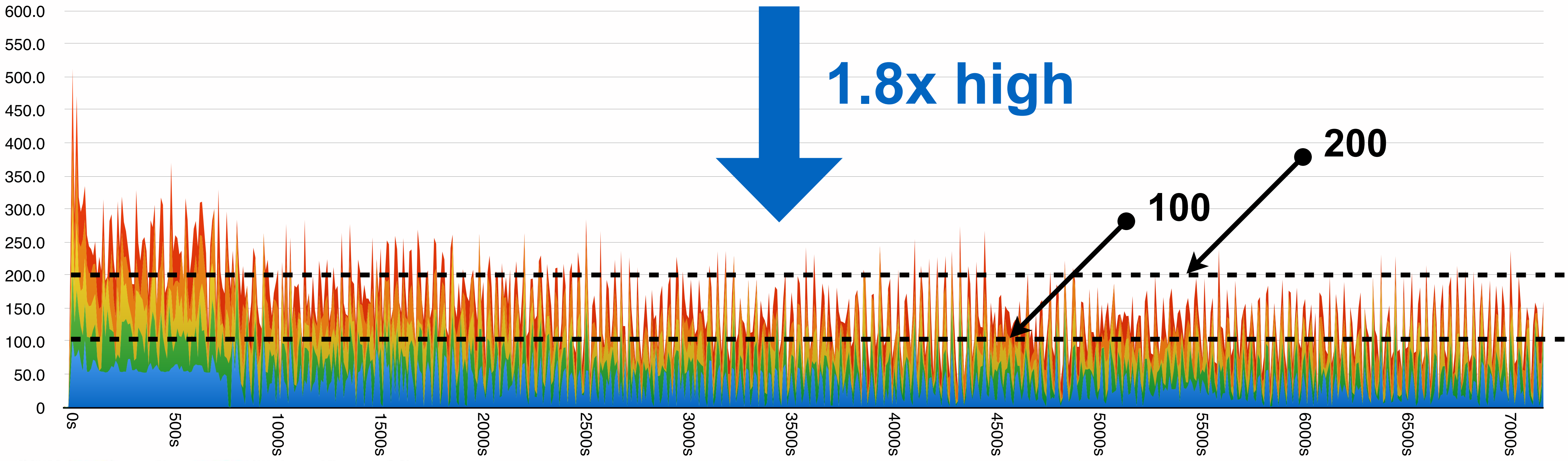
楽天 Rakuten

Brief Benchmark Report

1st Case - Disk util%



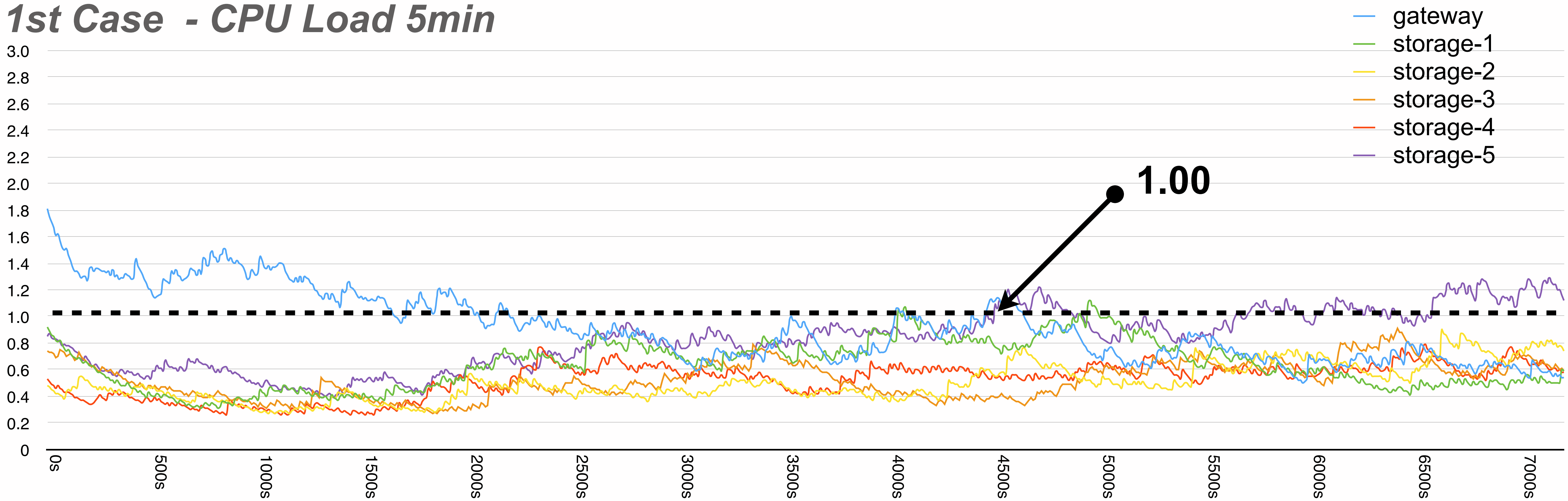
2nd Case - Disk util%



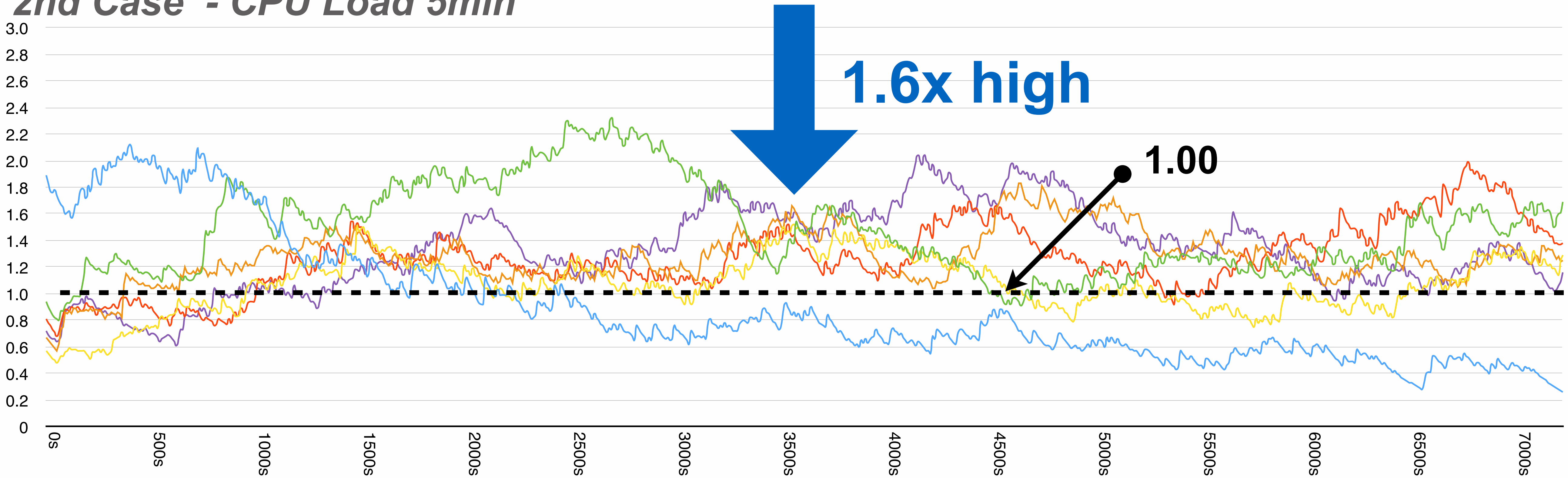
楽天 Rakuten

Brief Benchmark Report

1st Case - CPU Load 5min



2nd Case - CPU Load 5min



Brief Benchmark Report

Conclusion:

LeoFS kept in a stable performance through the benchmark

Bottleneck is Disk I/O

The cache mechanism contributed to reduce network traffic between Gateway and Storage

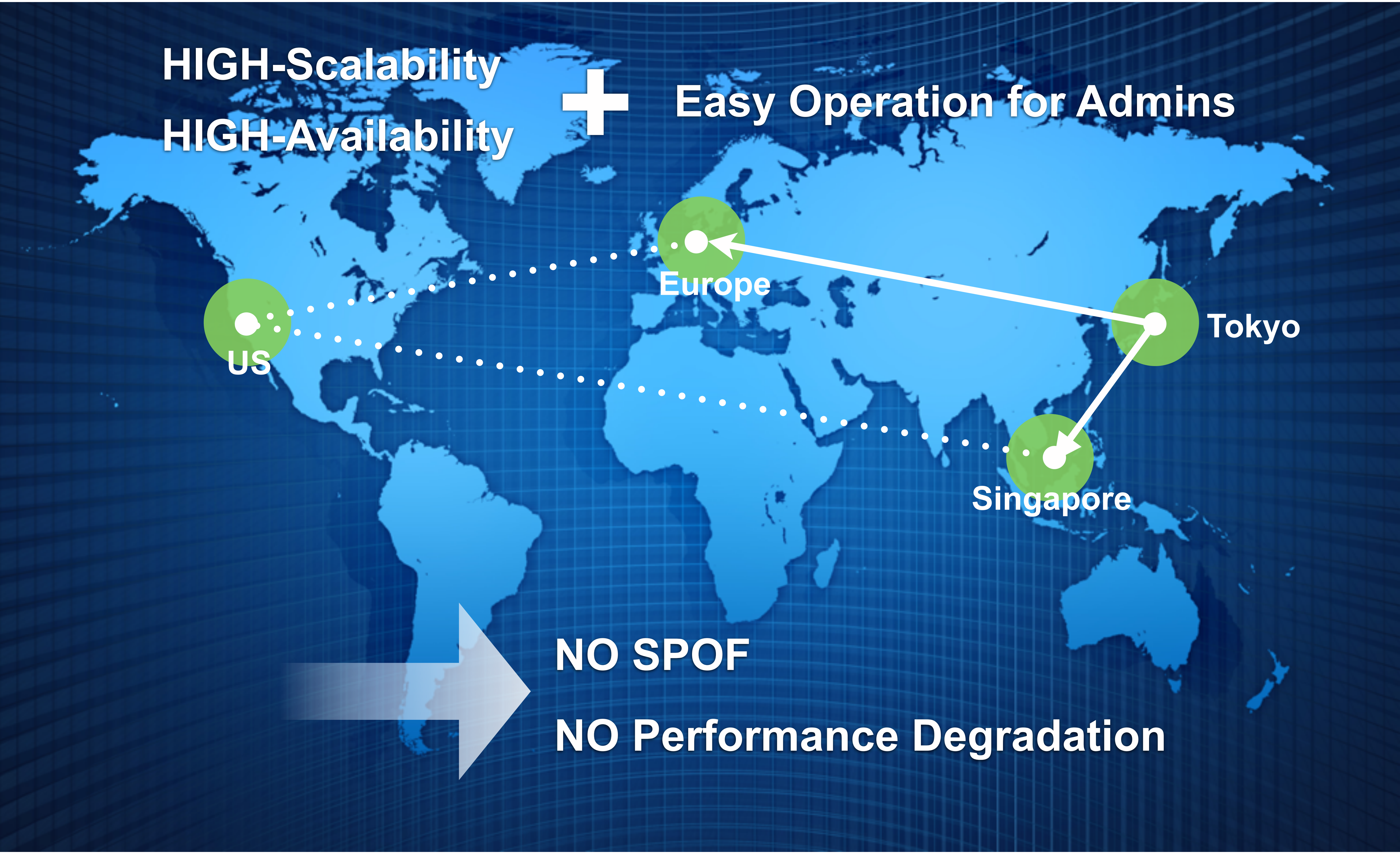
Multi Data Center Replication

Multi Data Center Replication

HIGH-Scalability

HIGH-Availability

+ Easy Operation for Admins



NO SPOF

NO Performance Degradation

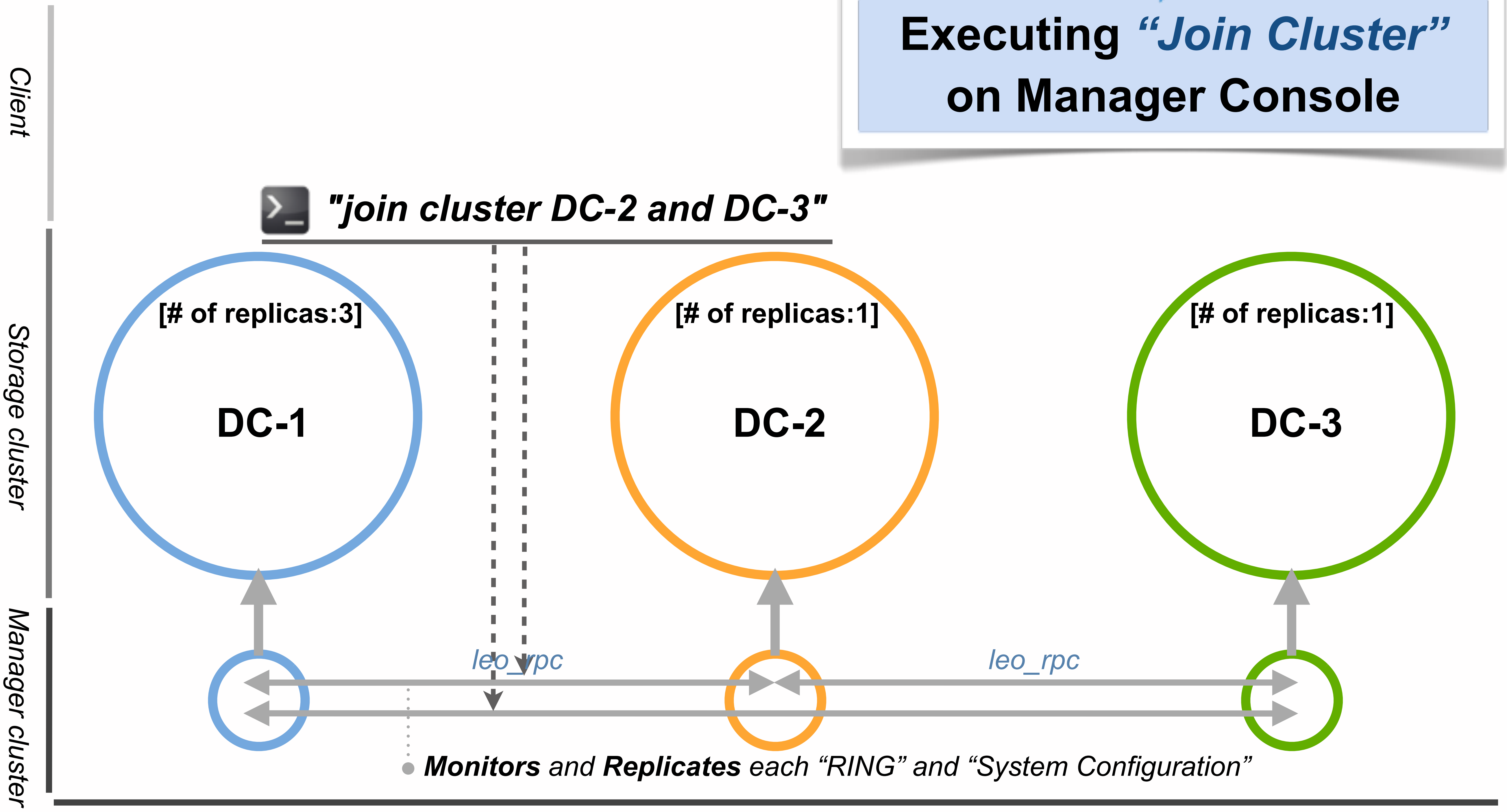
Designed it as simple as possible

1. Easy Operation to build **multi clusters**.
2. **Asynchronous data replication** between clusters

   Stacked data is transferred to remote cluster(s)
3. **Eventual consistency**

Preparing the MDC Replication

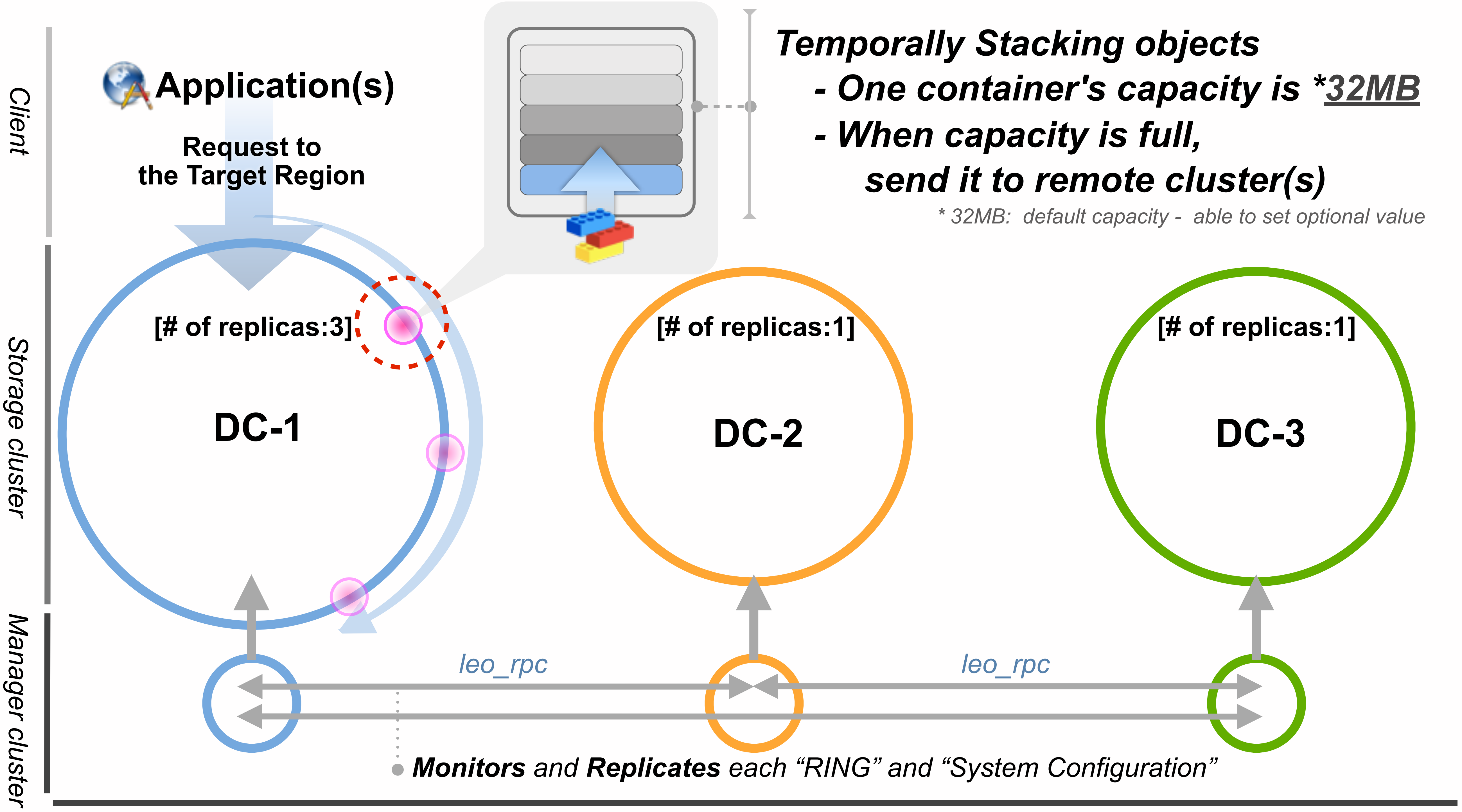
Executing *"Join Cluster"* on Manager Console



"Leo Storage Platform"

Multi Data Center Replication

Stacking objects



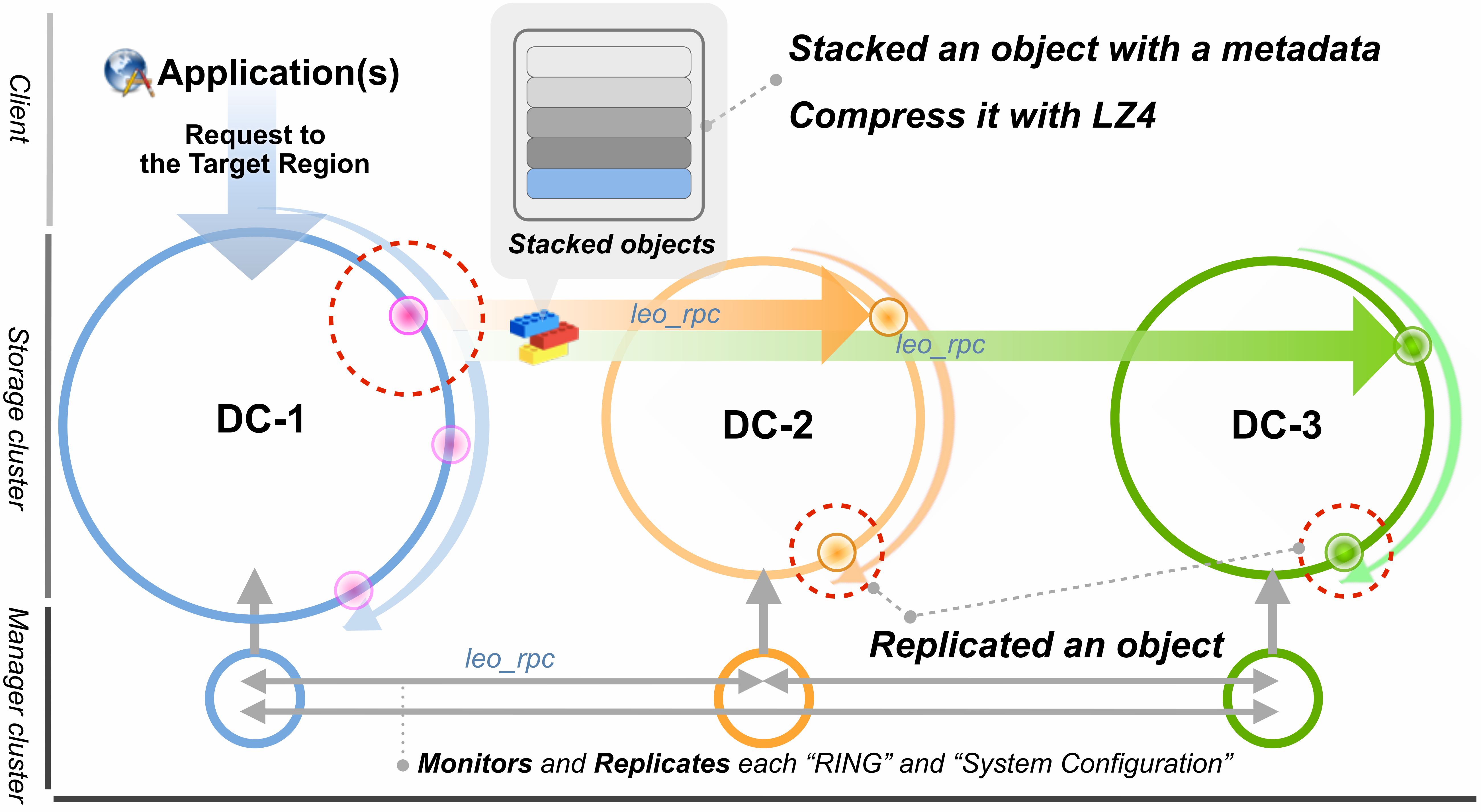
Temporally Stacking objects
- ***One container's capacity is *<u>32MB</u>***
- ***When capacity is full, send it to remote cluster(s)***

** 32MB: default capacity - able to set optional value*

***Monitors** and **Replicates** each "RING" and "System Configuration"*

"Leo Storage Platform"

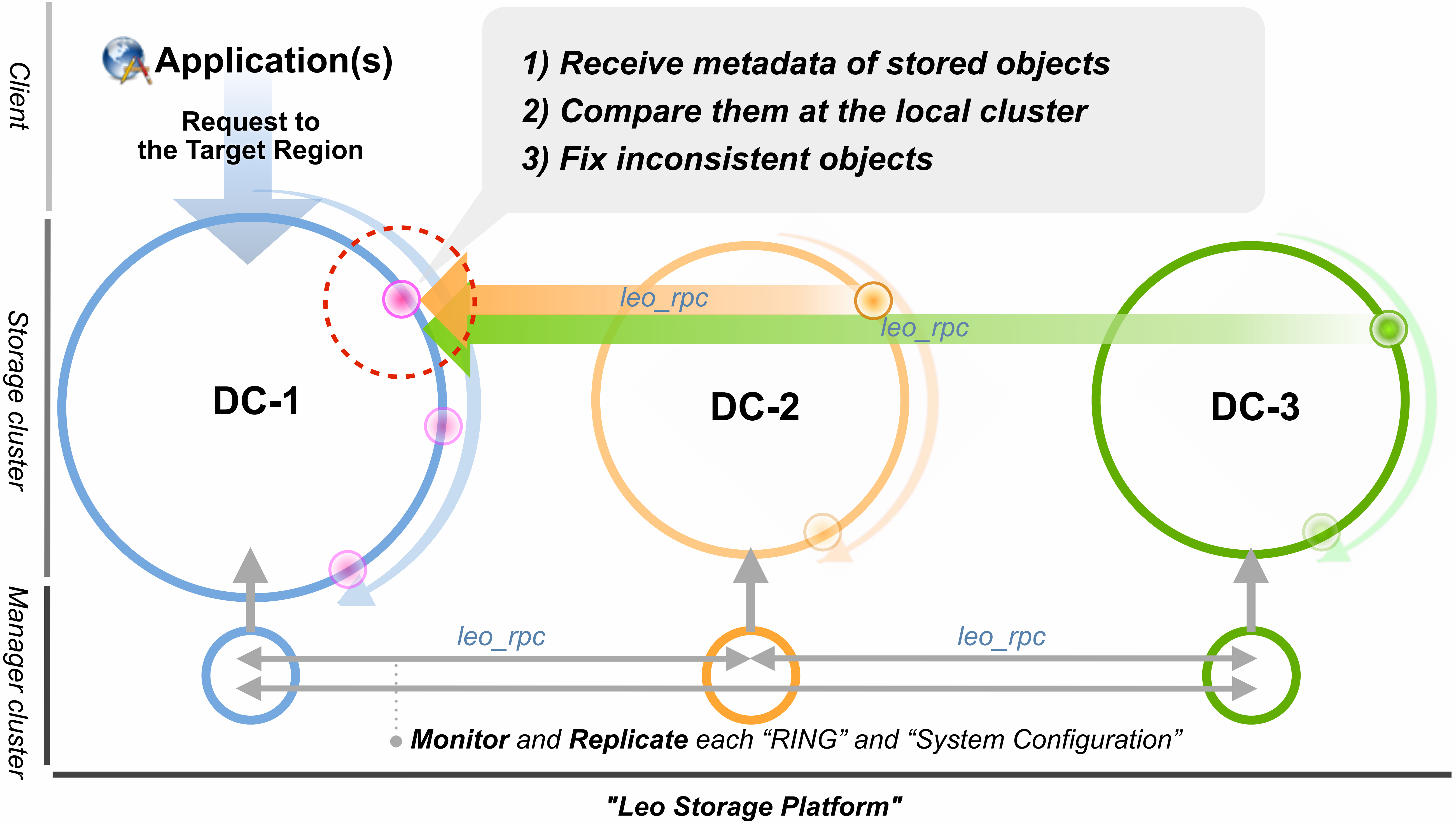
Transferring stacked objects



"Leo Storage Platform"

Multi Data Center Replication

Investigating stored objects



1) Receive metadata of stored objects

2) Compare them at the local cluster

3) Fix inconsistent objects

***Monitor** and **Replicate** each "RING" and "System Configuration"*

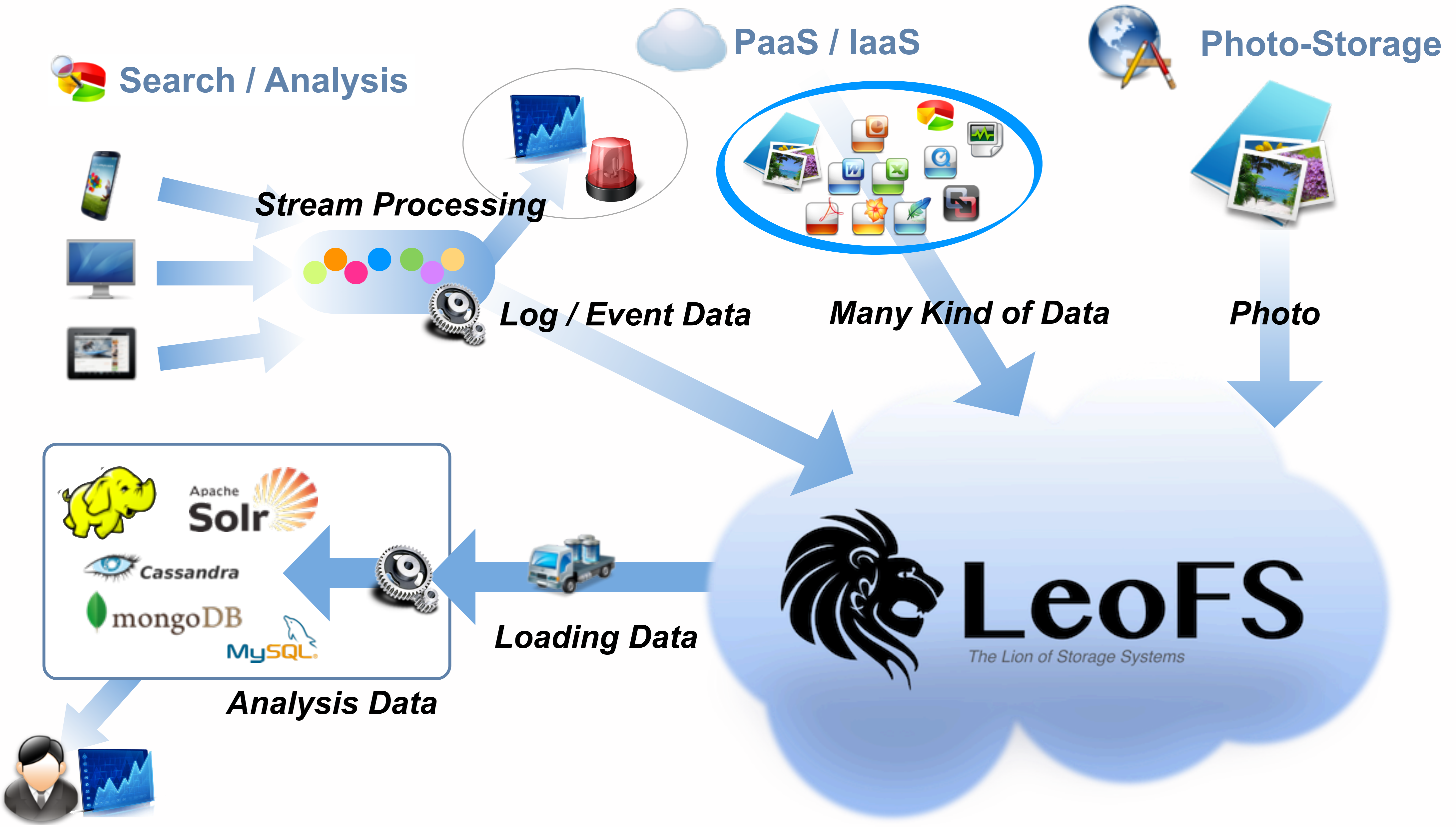
"Leo Storage Platform"

NFS Support

Future Plans

NFS Support

Data-HUB: Centralize unstructured data in LeoFS



楽天 Rakuten

LeoFS Administration at Rakuten

Presented by Masahiro Sanjo
Rakuten Institute of Technology

LeoFS Administration at Rakuten

Storage Platform

File Sharing Service

Others

Portal Site

Photo Storage

Background Storage of OpenStack

Storage Platform

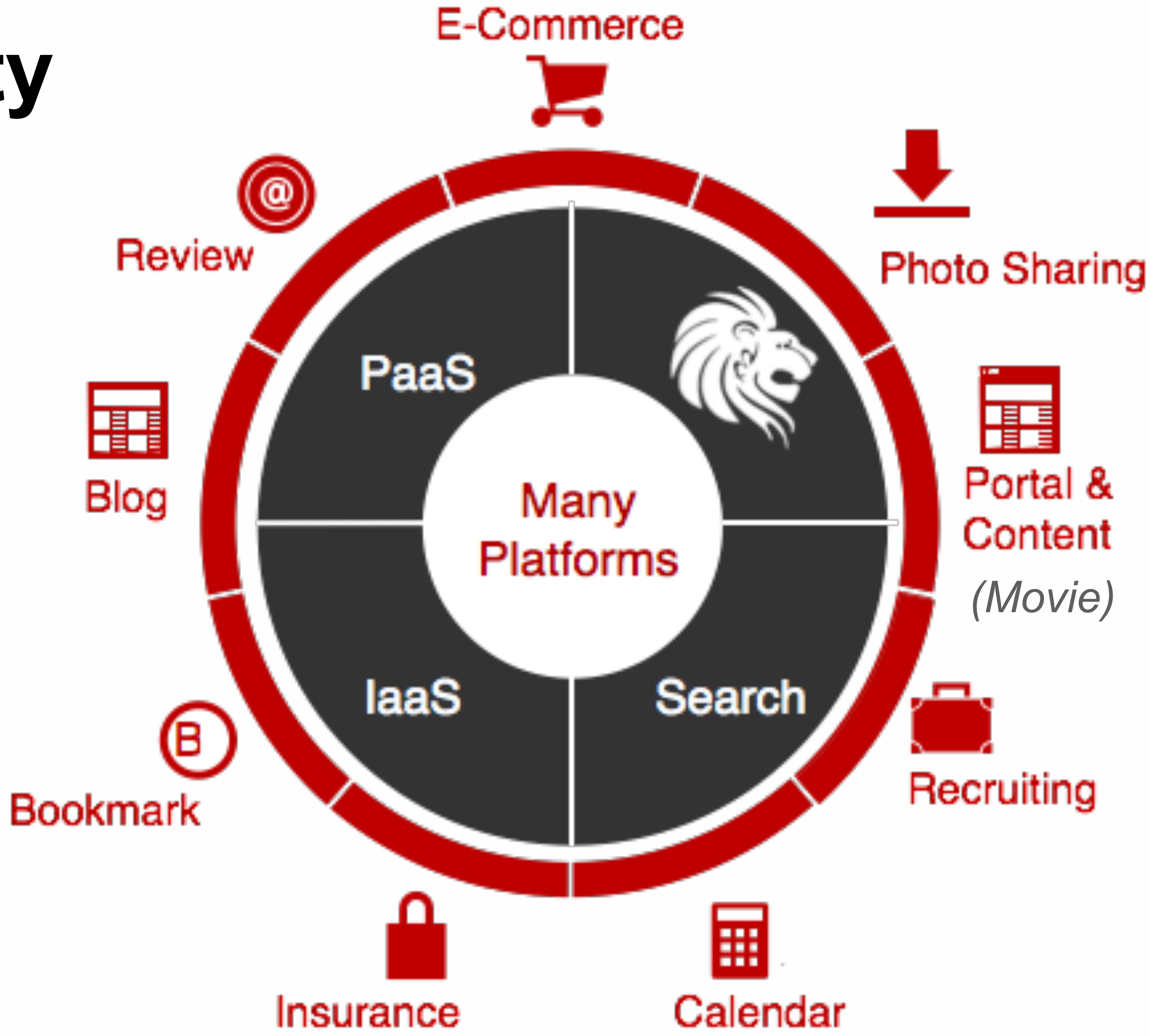
Storage Platform - Scaling the Storage Platform

Reduce Costs

High Reliability

Easy to Scale

S3-API



楽天 Rakuten

Storage Platform - Scaling the Storage Platform

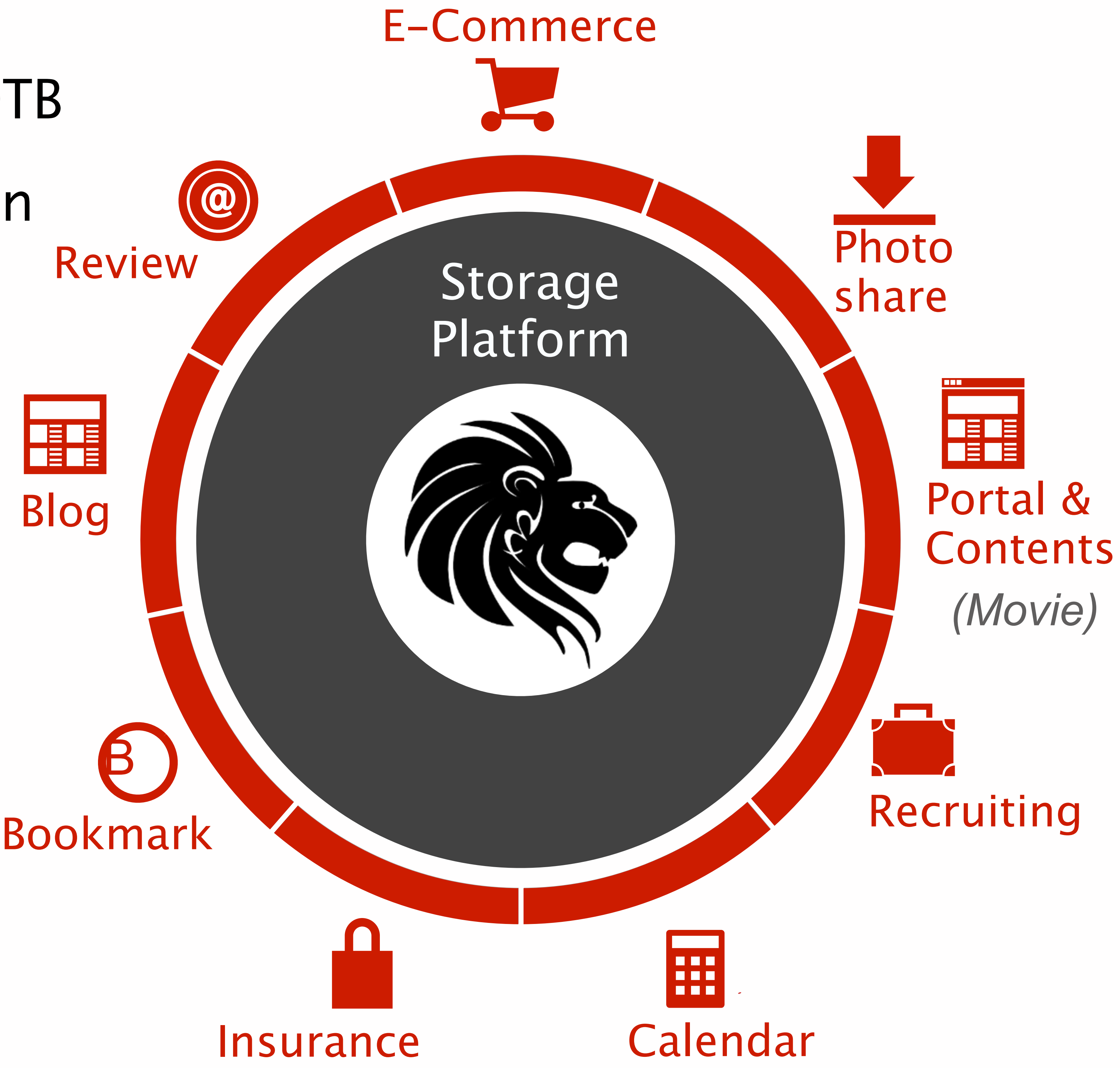
Using Various Services

Total Usage: 450TB/600TB

# of Files: 600Million

Daily Growth: 100GB

Daily Reqs: 13Million



楽天 Rakuten

Storage Platform - System Layout

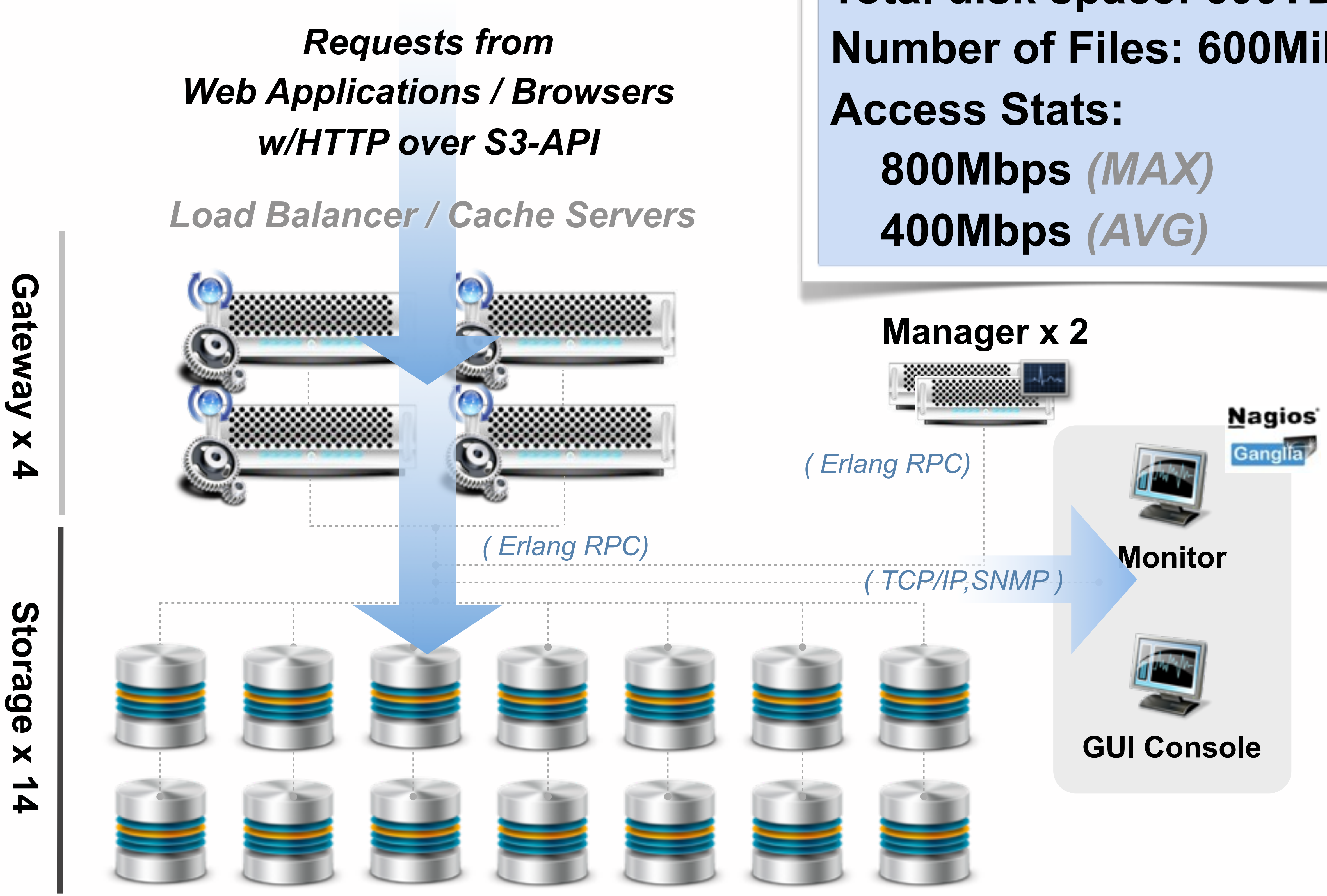
Total disk space: 600TB

Number of Files: 600Million

Access Stats:

800Mbps *(MAX)*

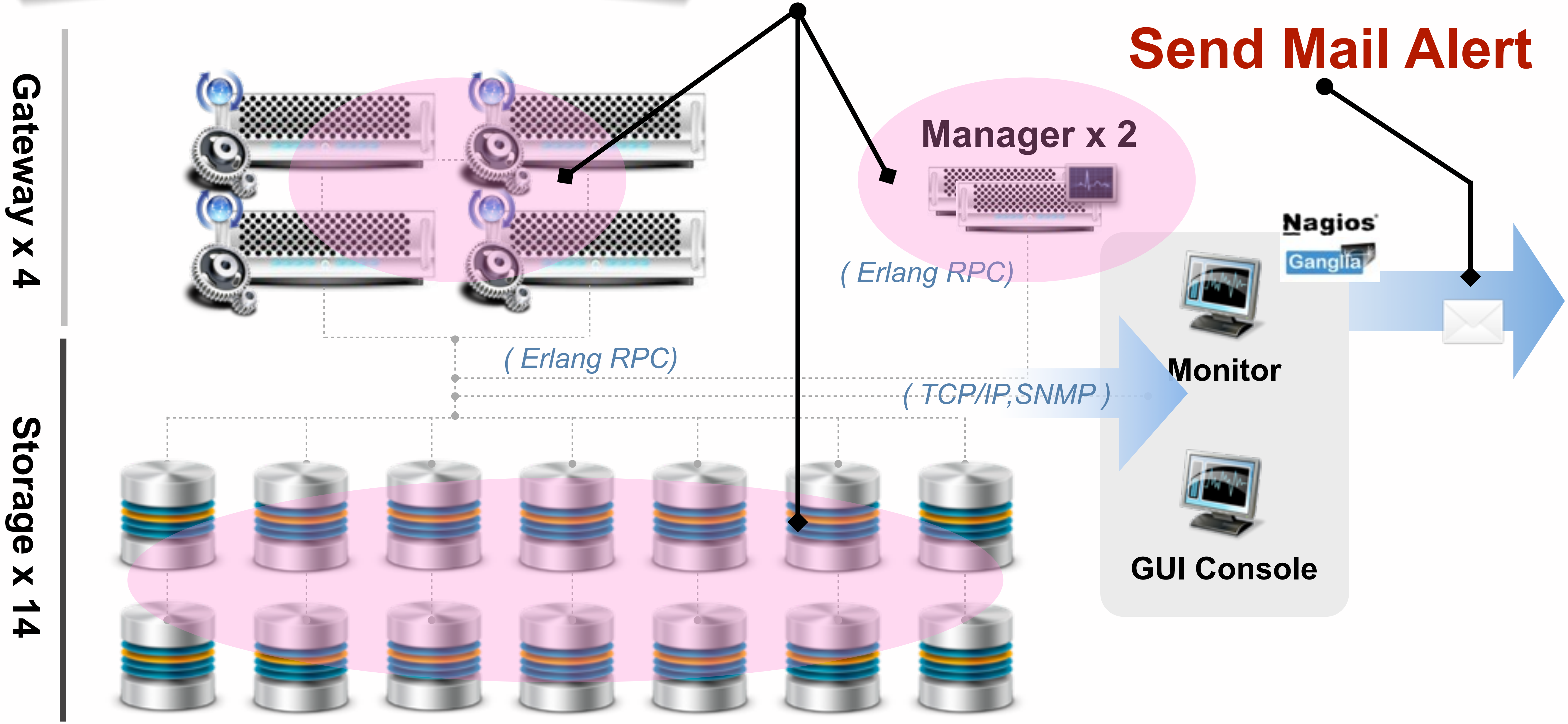
400Mbps *(AVG)*



楽天 Rakuten

Storage Platform - Monitor

Status Collection *(Ganglia)*
Status Check *(Nagios)*
Port + Threshold Check



Ganglia Agent

Send Mail Alert

Gateway x 4

Storage x 14

Manager x 2

(Erlang RPC)

(Erlang RPC)

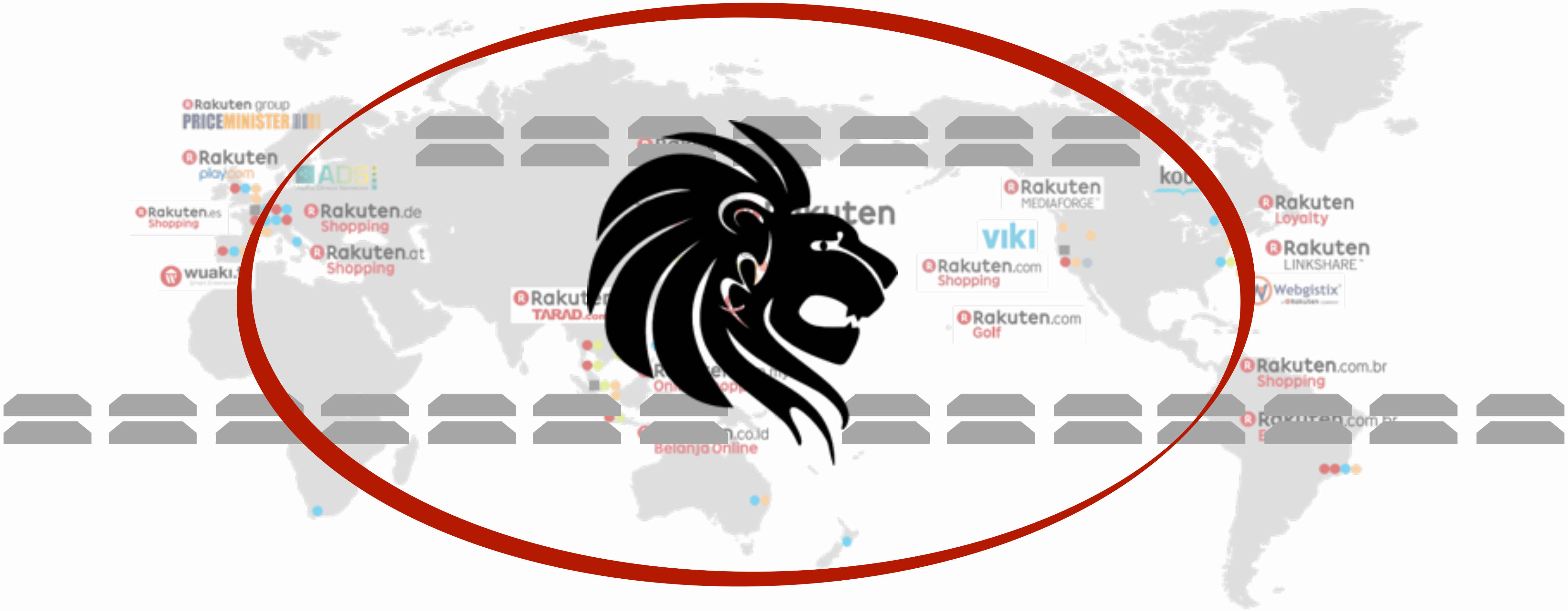
(TCP/IP,SNMP)

Monitor

GUI Console

Storage Platform - Spreading Globally

Covering All Services with Multi DC Replication



File Sharing Service

+

https://owncloud.com/

File Sharing Service - Required Targets

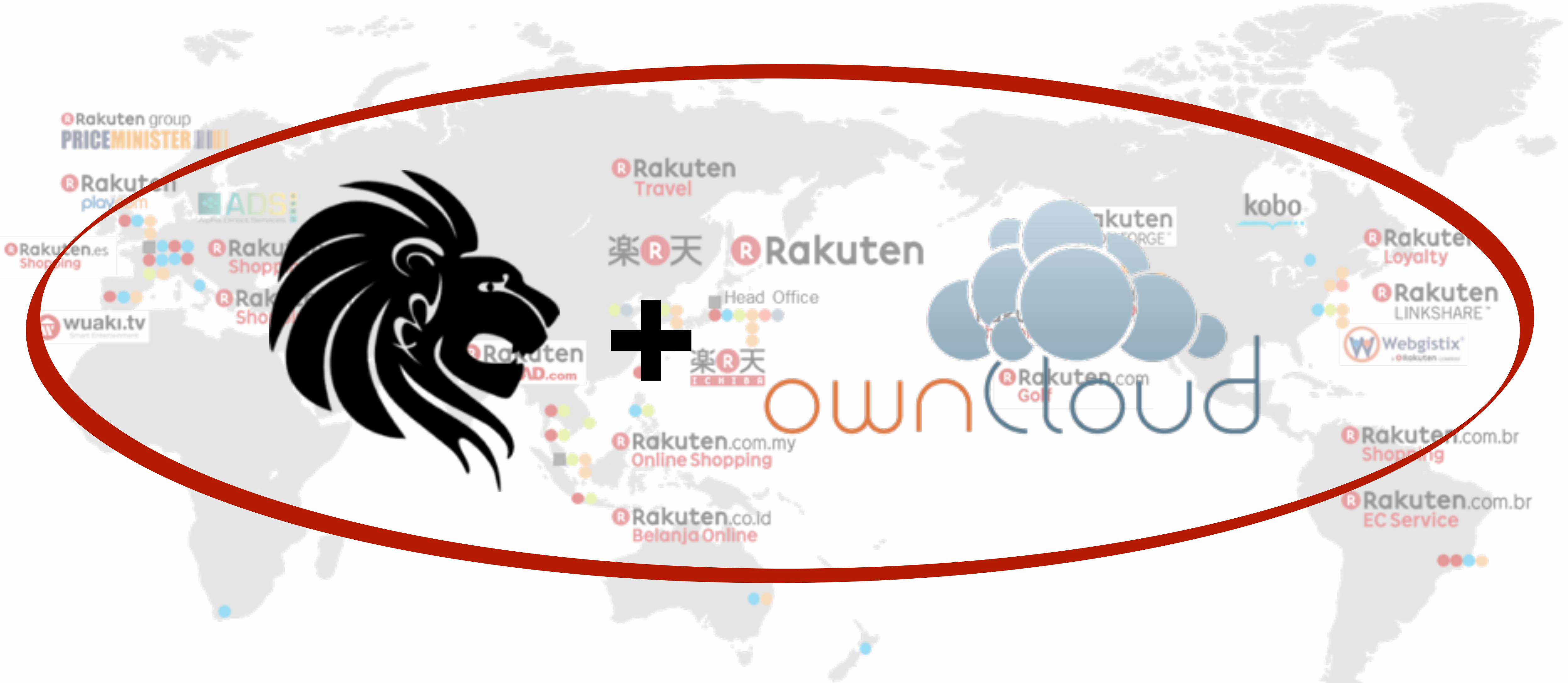
Reduce Costs

Handle Confidential Files

Store Large Files

Scale Easily

File Sharing Service - Usage



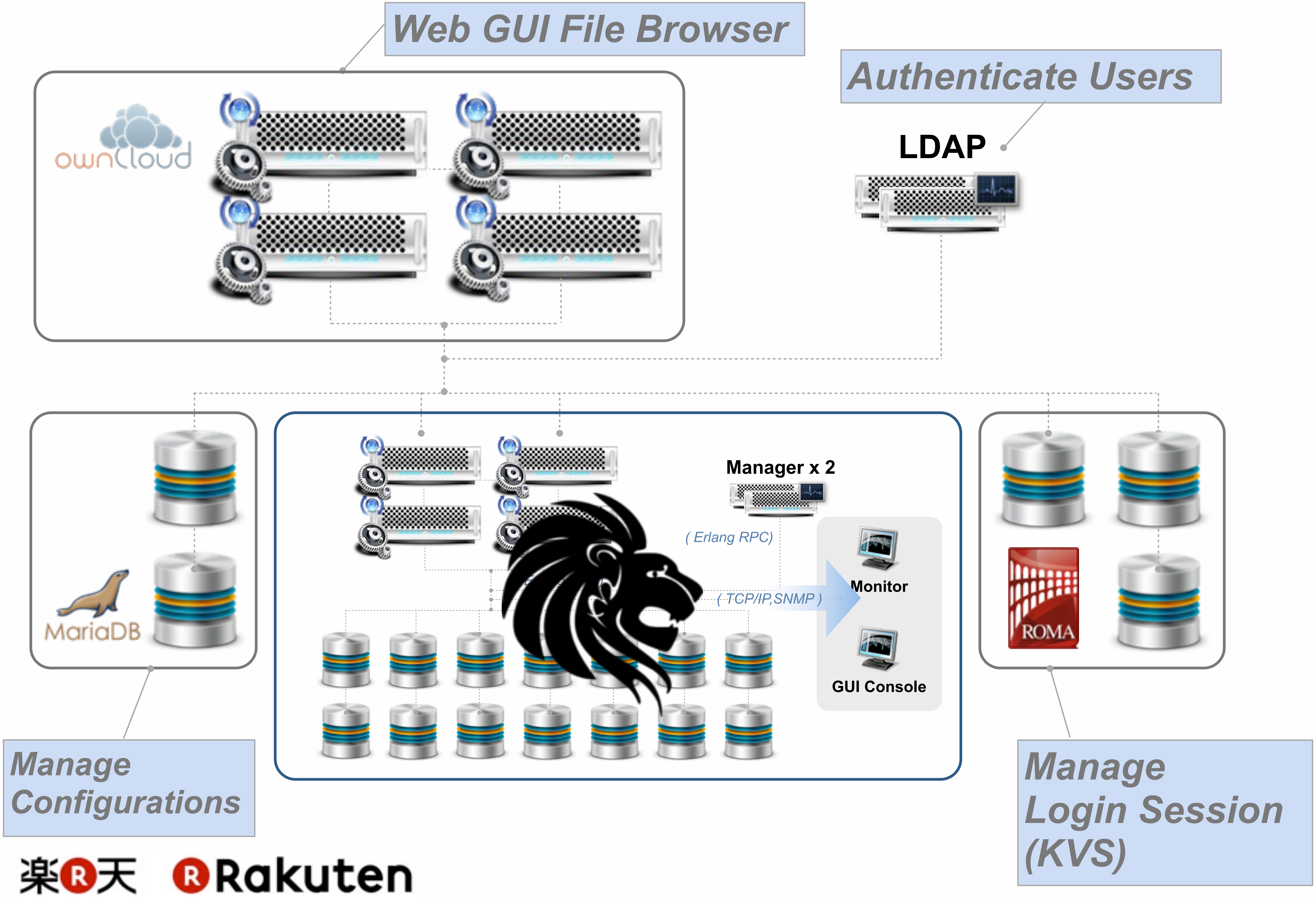
Share Docs and Videos with Group Companies

Over 20 Companies, Over 10 Countries

Over 4,000 Users, Over 10,000 Teams

楽R天 Rakuten

File Sharing Service - System Layout

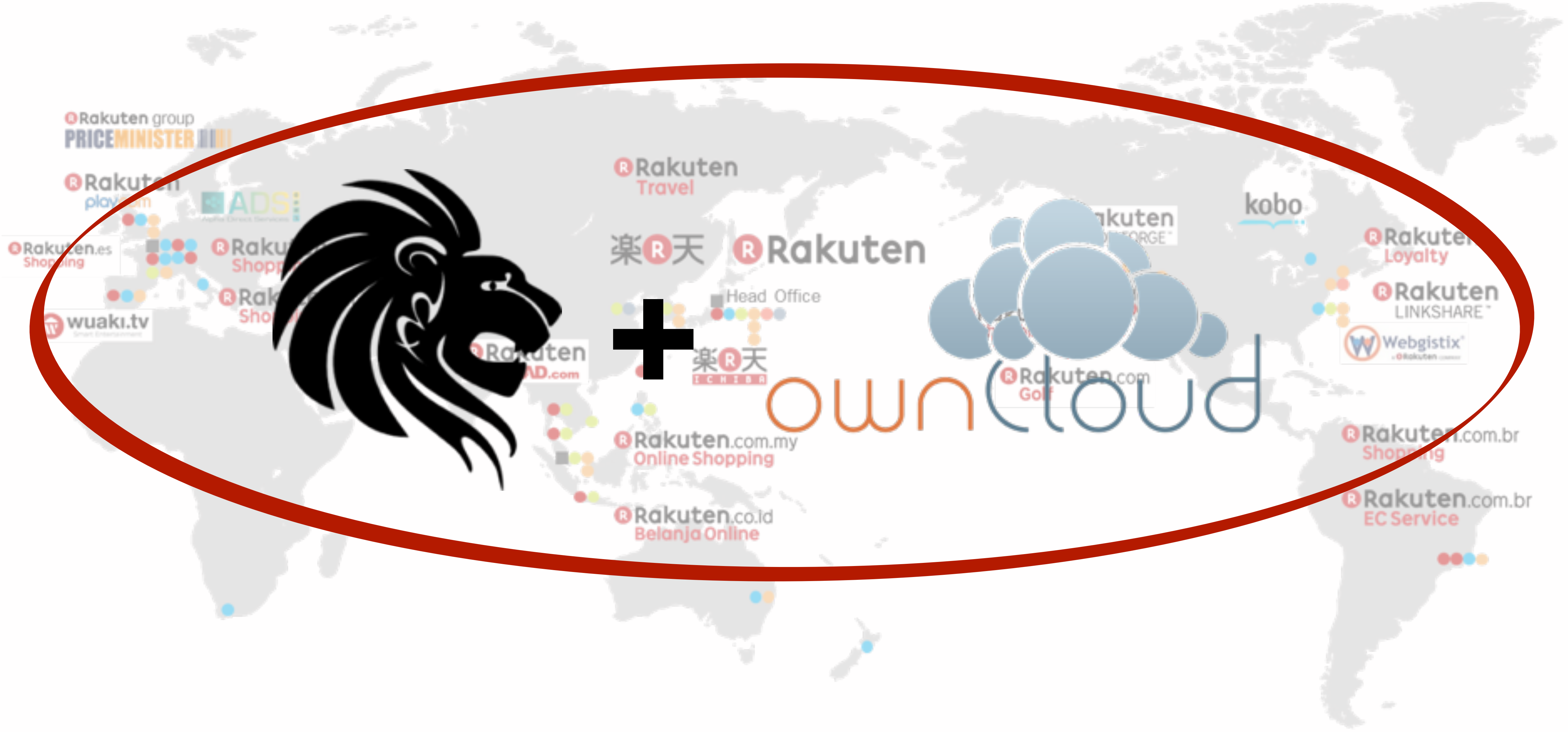


楽Ⓡ天 Ⓡ Rakuten

File Sharing Service - Future Plans

Cover 25 Countries/Regions

Over 20,000 Users



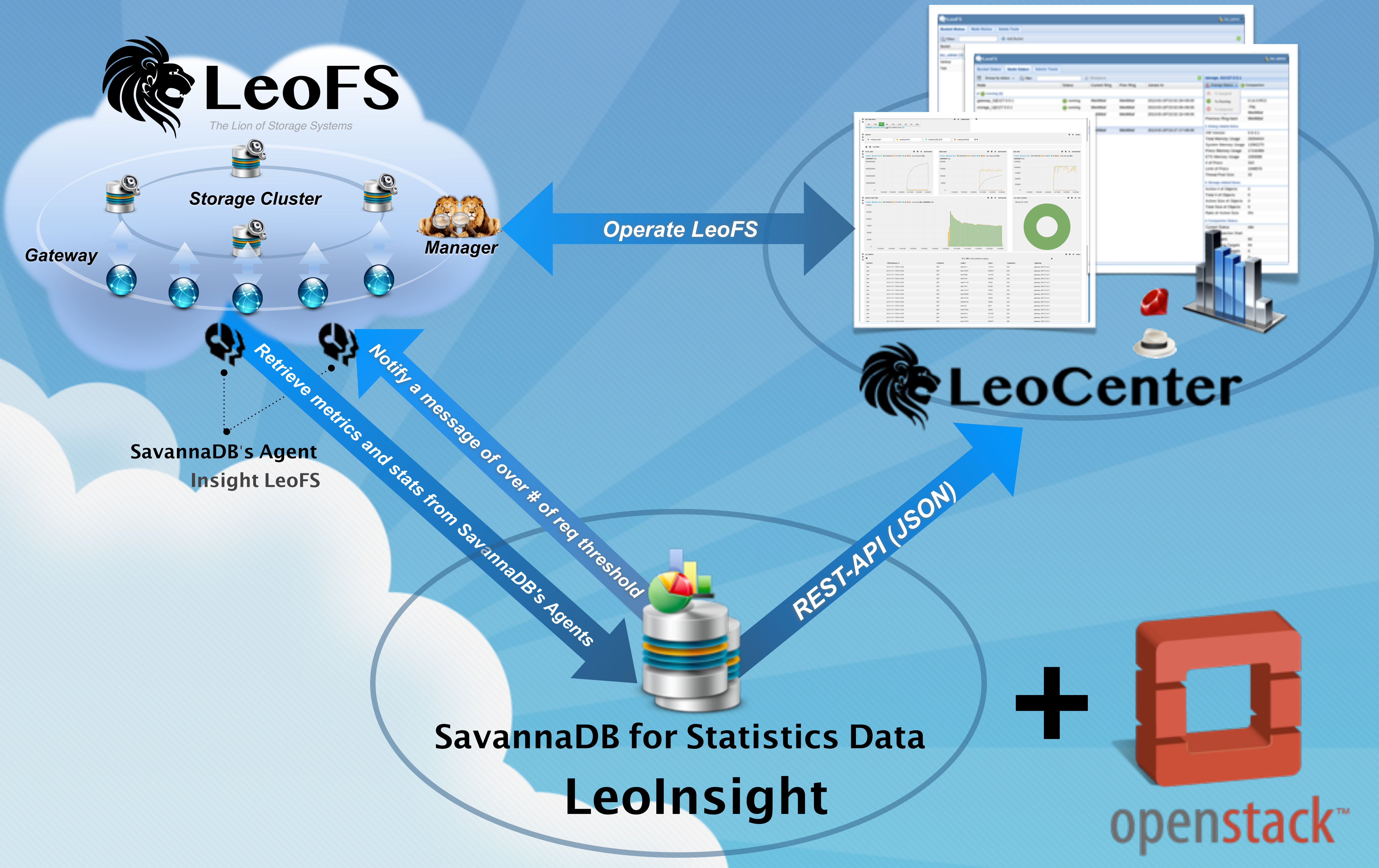
楽天 Rakuten

Empowering the Services and the Users Through the Cloud Storage

楽天 Rakuten

Future Plans

Future Plans



LeoFS

The Lion of Storage Systems

Storage Cluster

Gateway

Manager

Operate LeoFS

LeoCenter

SavannaDB's Agent

Insight LeoFS

Retrieve metrics and stats from SavannaDB's Agents

Notify a message of over # of req threshold

REST-API (JSON)

SavannaDB for Statistics Data

LeoInsight

+

openstack™

Set Sail for "Cloud Storage"

Website: leo-project.net
Twitter: @LeoFastStorage