Rakuten

LeoFS

The Lion of storage systems

Rakuten. Inc, Yosuke Hara Mar 21, 2013

LeoFS

The Lion of storage systems

http://www.leofs.org

LeoFS v0.14.0 was released!

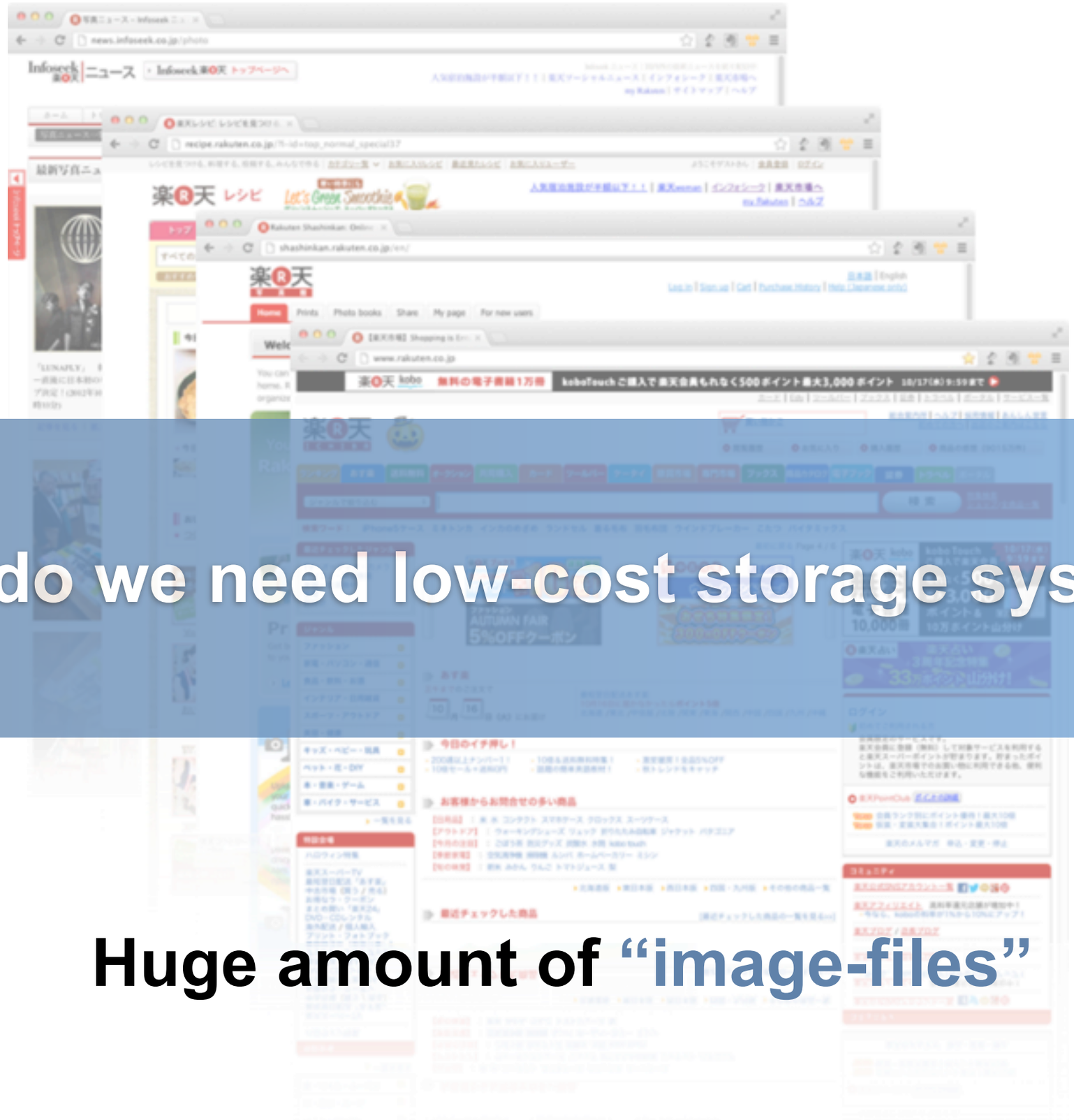
Rakuten

Motivation

Rakuten

2010

“We need to store and manage huge amount of Files at low-cost”



Why do we need low-cost storage system?

Huge amount of “image-files”

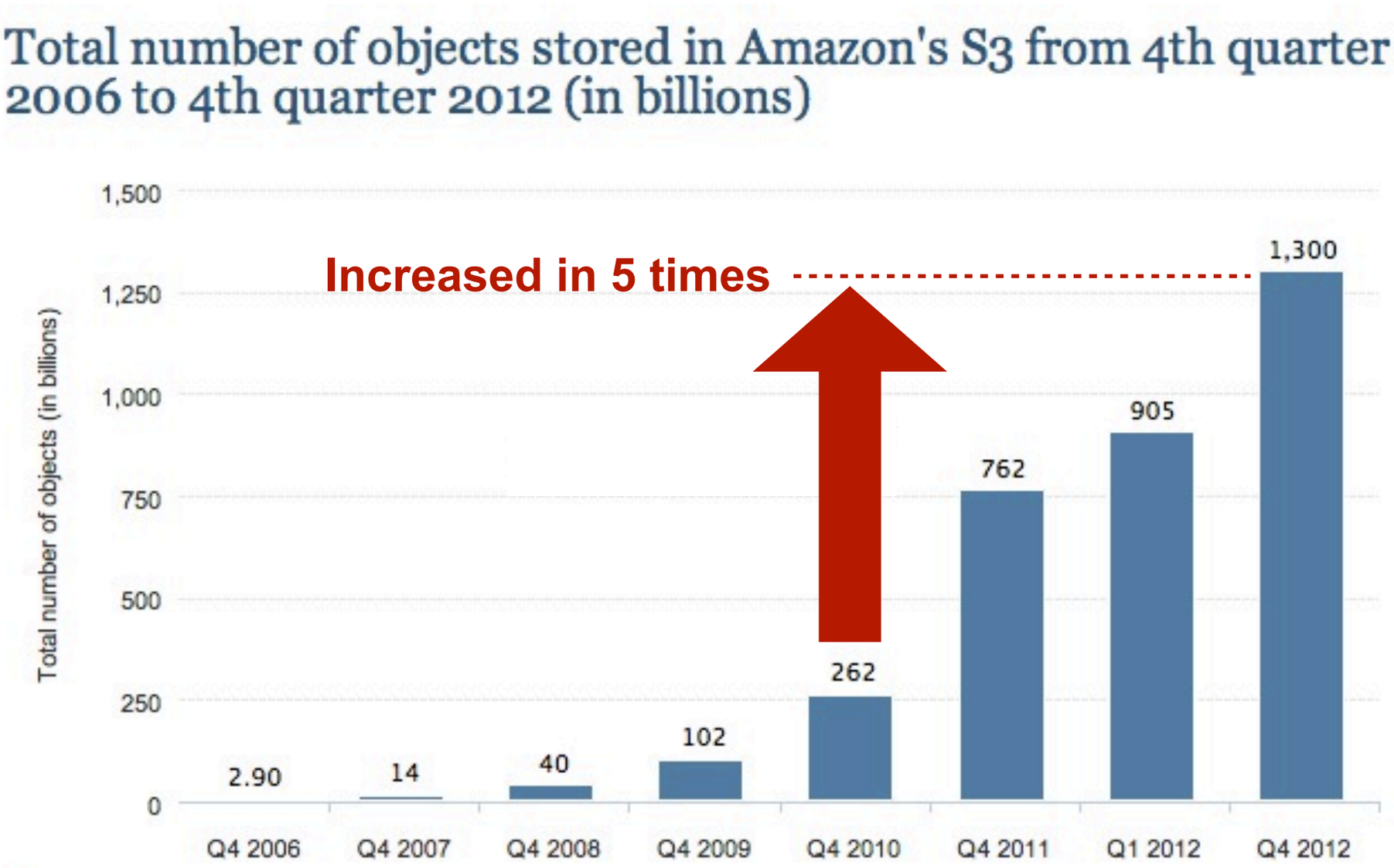
Motivation

“Need to move
from expensive storage to something”

Problems:

1. **Low ROI**
2. **Possibility of SPOF**
3. **Storage Expansion is difficult during increasing data**

Face Same Situation



Rakuten

Introducing LeoFS

Rakuten

What kind of storage we need for web-services?

1. ONE-Huge Storage

2. Non-Stop Storage

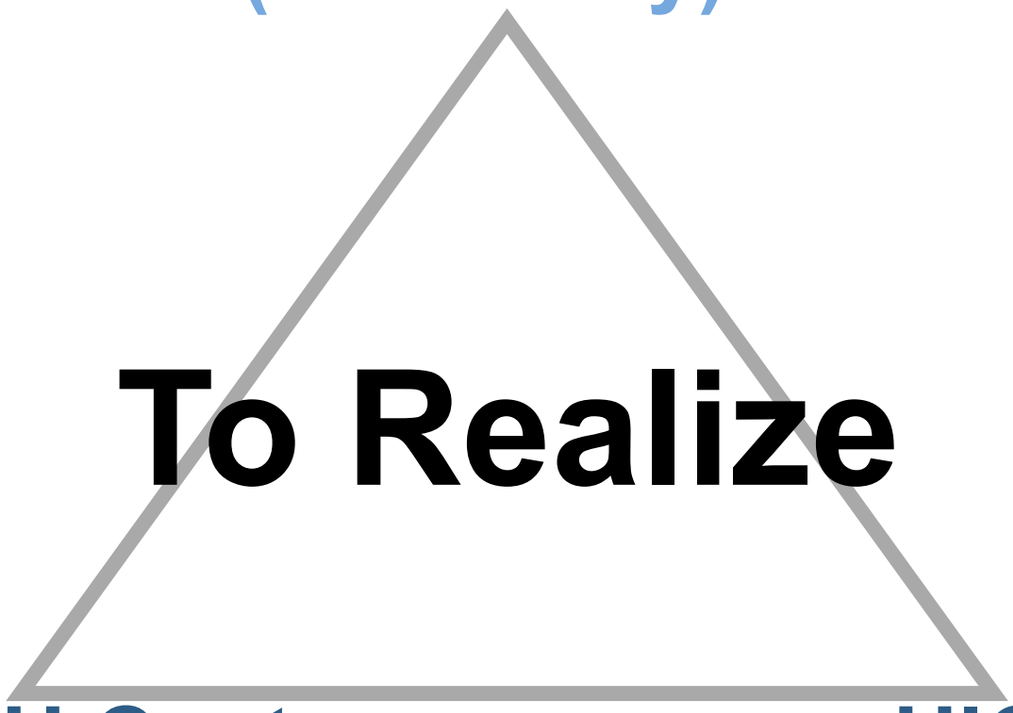
3. Specialized in the Web

Not FUSE But REST-API over HTTP

Rakuten

What should we realize?

HIGH Availability
(Reliability)



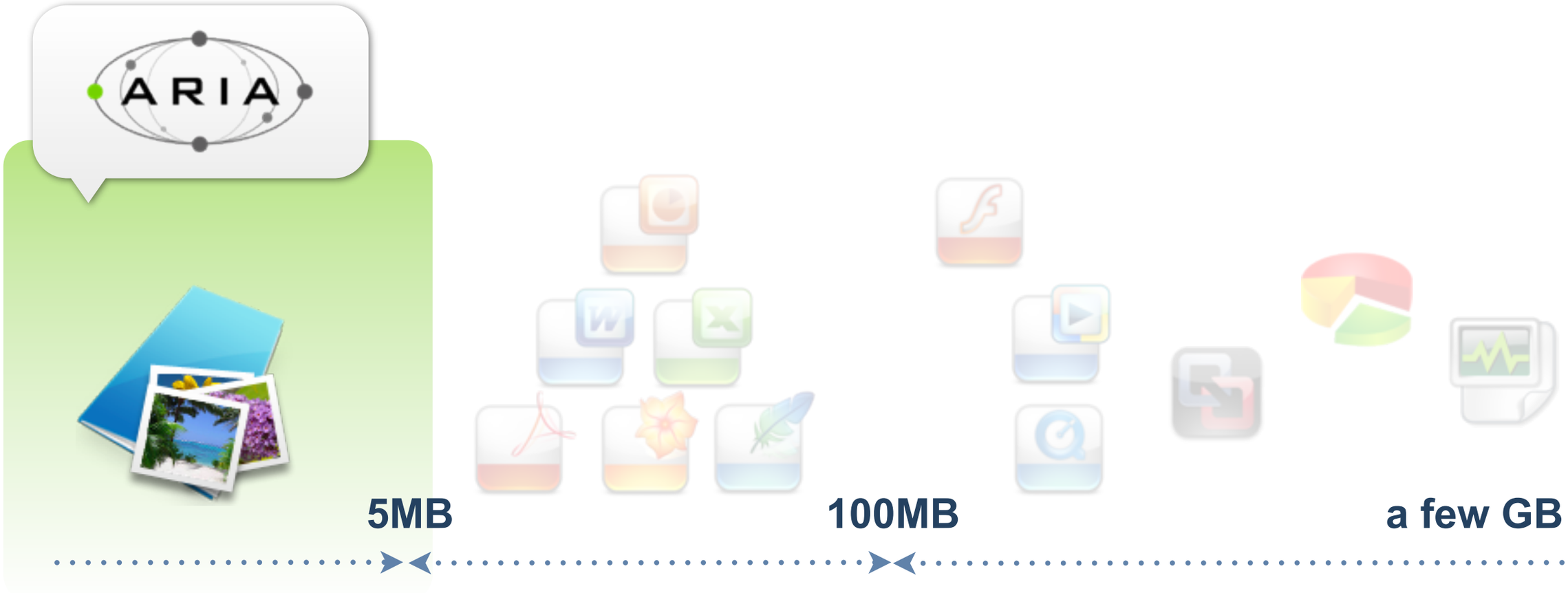
To Realize

HIGH Cost Performance Ratio

HIGH Scalability

Rakuten

From Photo Storage To Cloud Storage

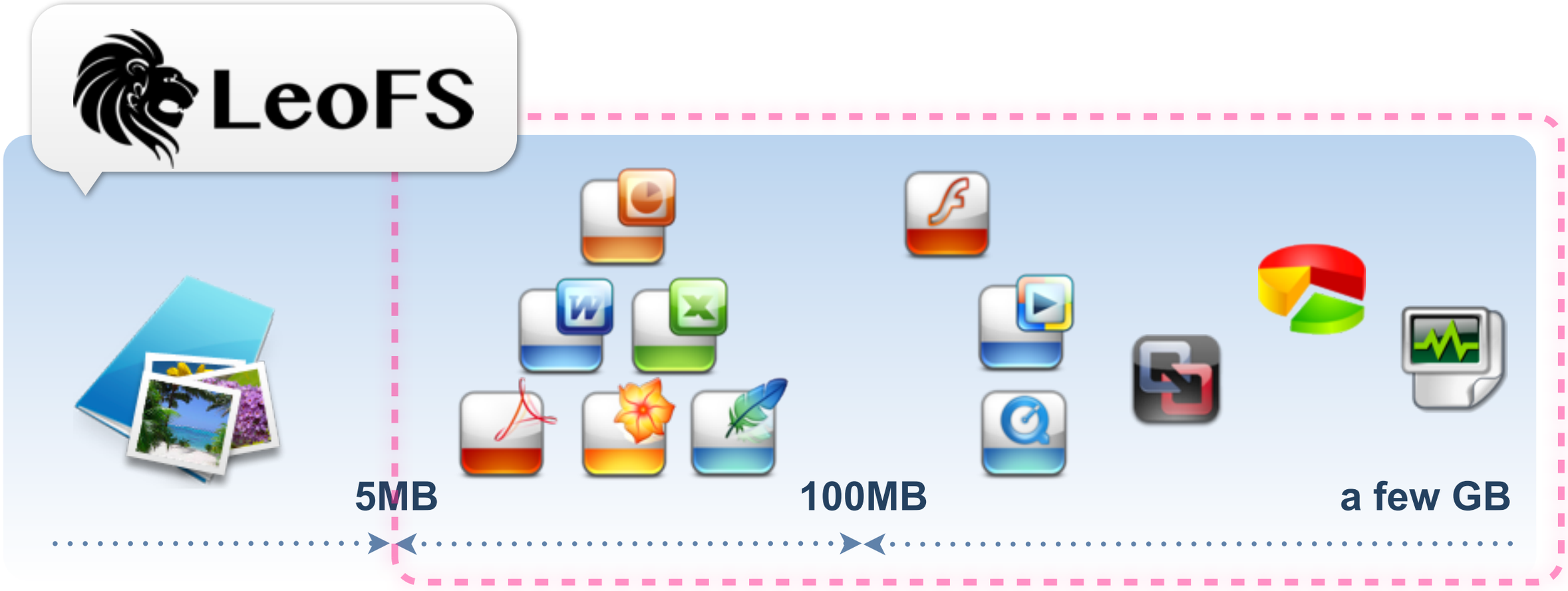


1st step as Cloud Storage

→ Specialize in “Photo”

Rakuten

From Photo Storage To Cloud Storage

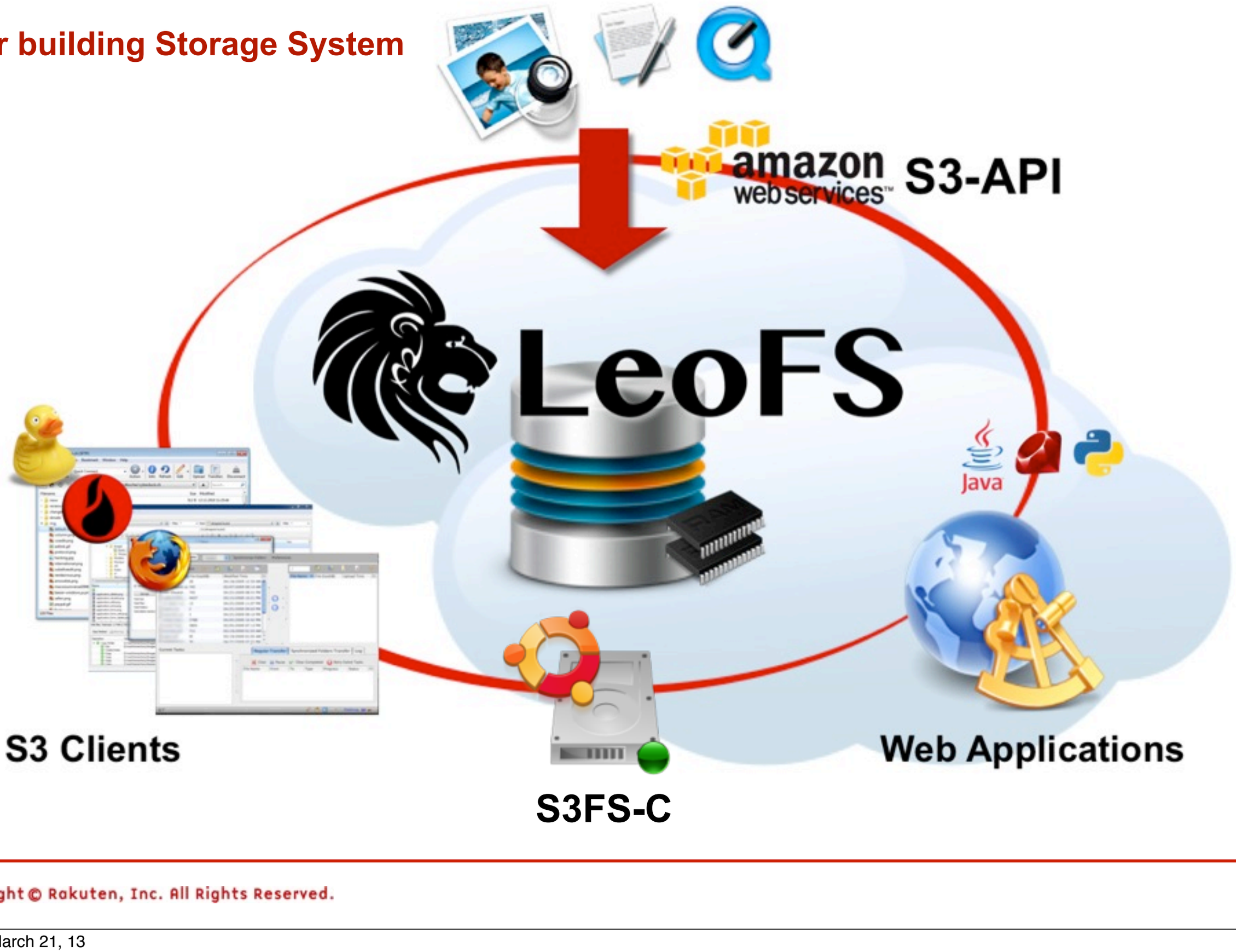


Aim to "Storage Platform" in the Cloud

→ Able to store various unstructured-data

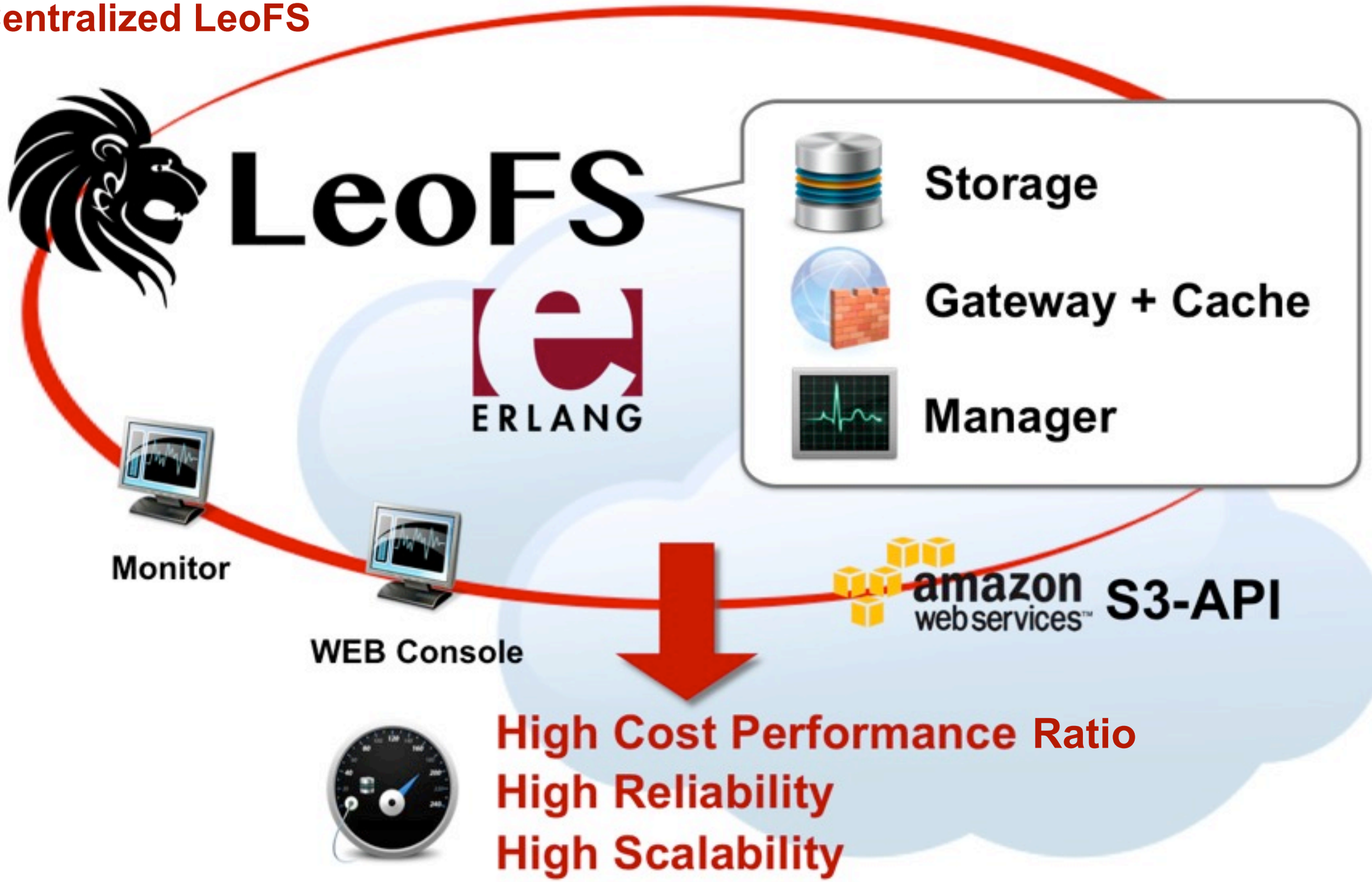
Rakuten

For building Storage System



Rakuten

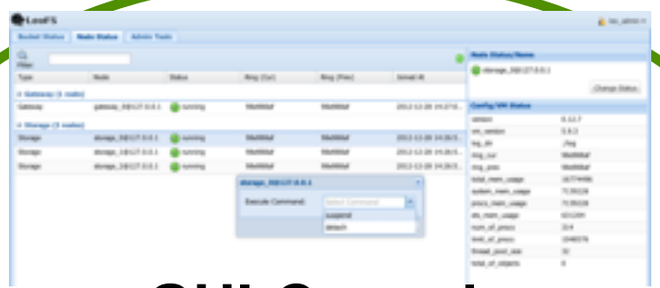
In Centralized LeoFS



High Cost Performance Ratio
High Reliability
High Scalability

Rakuten

To Realize “Storage Platform”



GUI-Console

LeoTamer

QoS

LeoDenebola

Rakuten

Overview

Rakuten

2010

ERLANG

Concurrency

Distribution

Fault tolerance

Using in Telecom, Banking, e-commerce, Instant messaging,...

“Robust” + “Scalable”
Storage System

LeoFS Overview

Gateway (Stateless Proxy)

HTTP Request/Response Handling

+

w/Object Cache

Manager

System Management

Monitor Storage/Gateway

RoutingTable Monitor

NodeState Monitor

Keep running + Keep Consistency

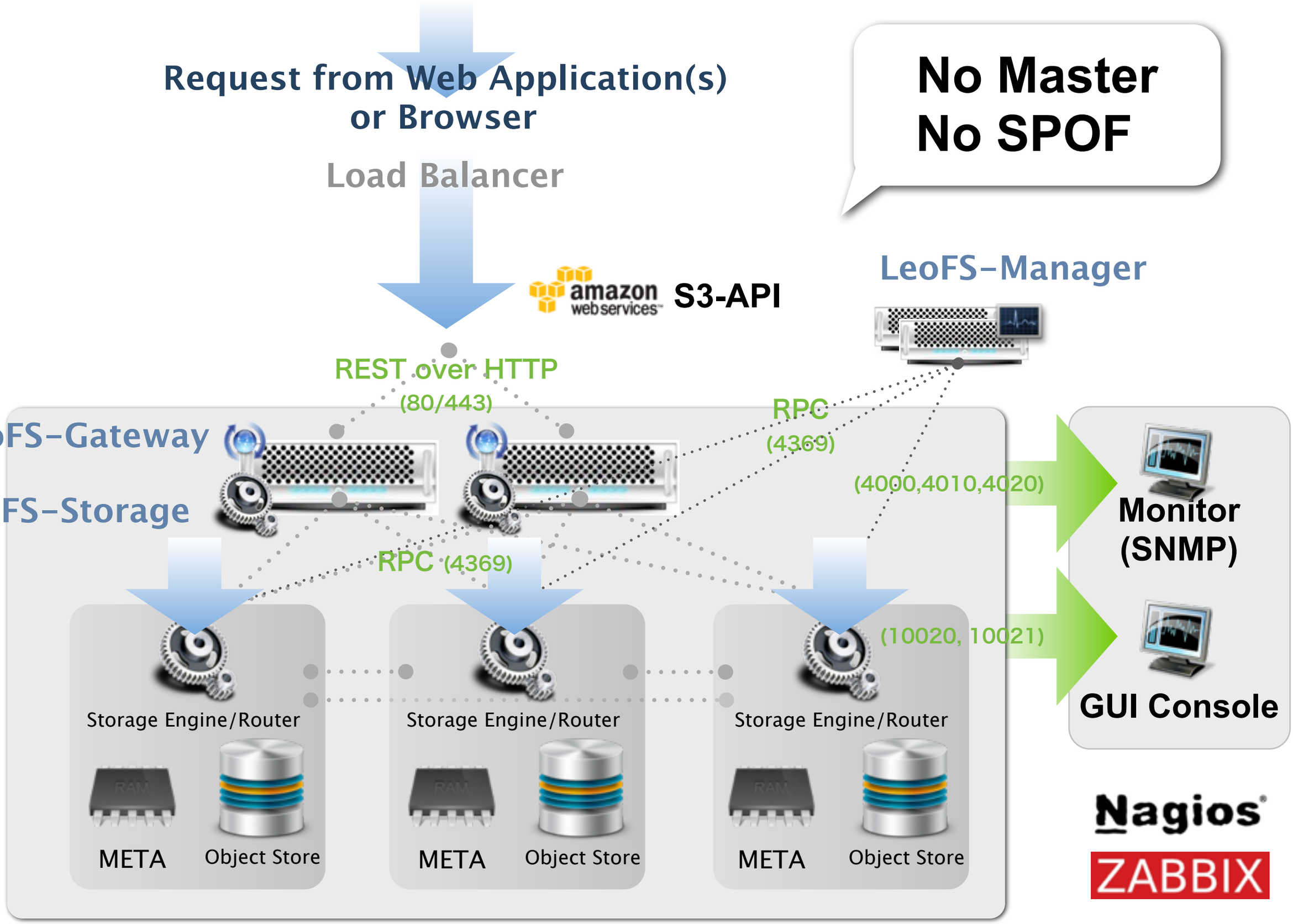
Storage

Object Storage, Meta data Storage

+

Replicator/Recoverer, Queue

LeoFS Overview

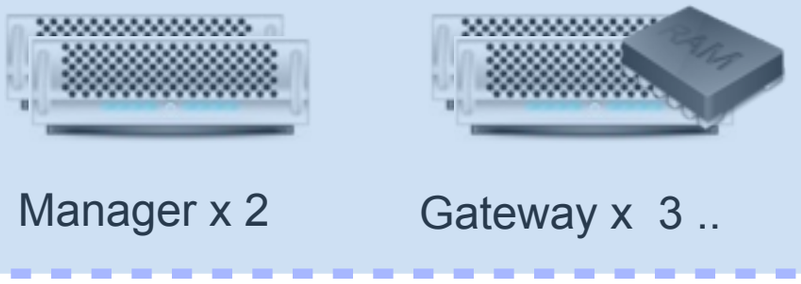


Rakuten

1 – Minimum for Development

Manager x 1 Gateway x 1 Storage x 1

20 – 50TB Storage System (# of replicas = 3)



Manager x 2 Gateway x 3 ..

10TB .. 20TB / server

Storage x 8 - 15

50 – 300TB Storage System (# of replicas = 3)

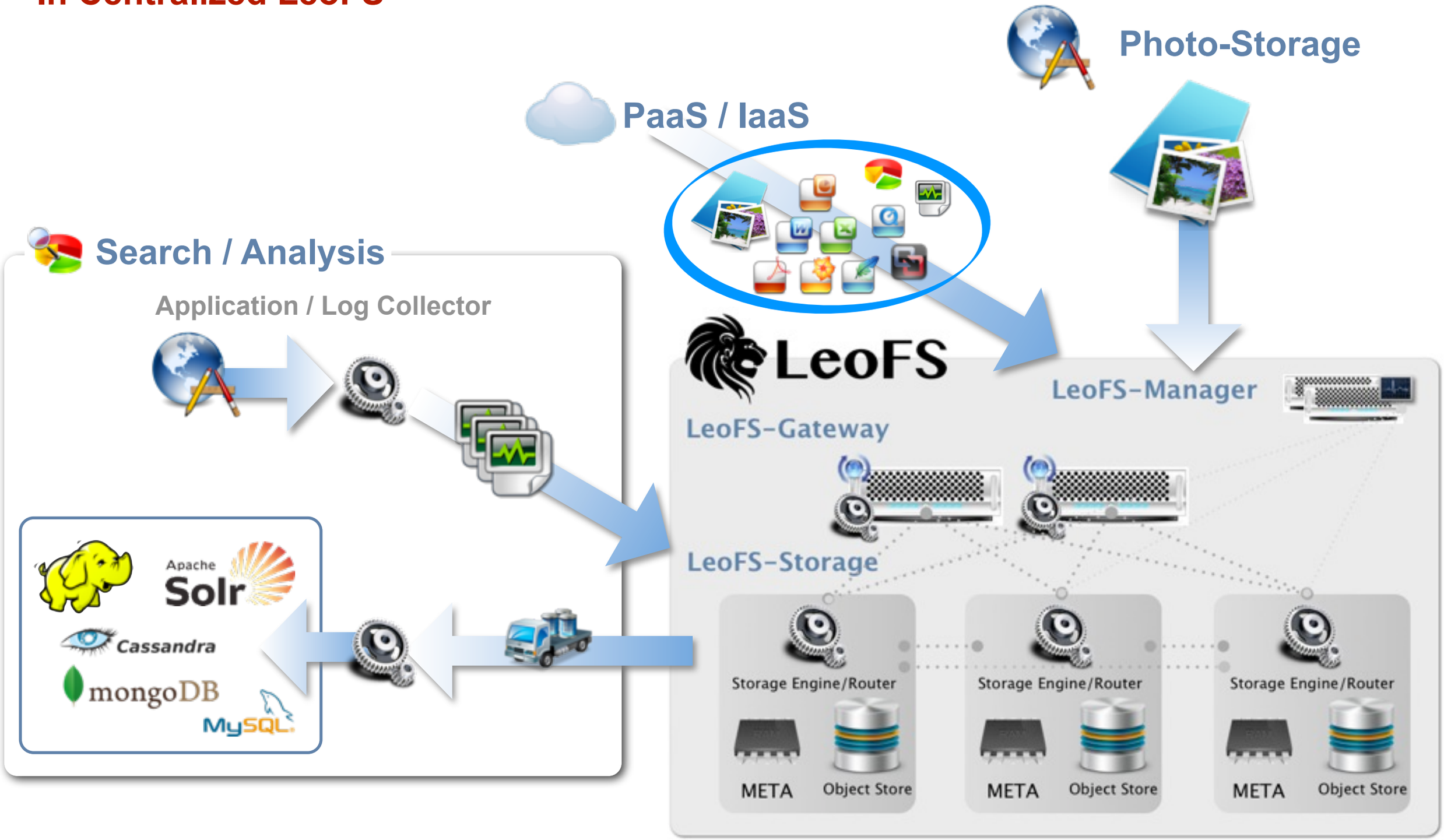
Manager x 2 Gateway x 4 ..

10TB .. 20TB / server

Storage x 45 - 90

Rakuten

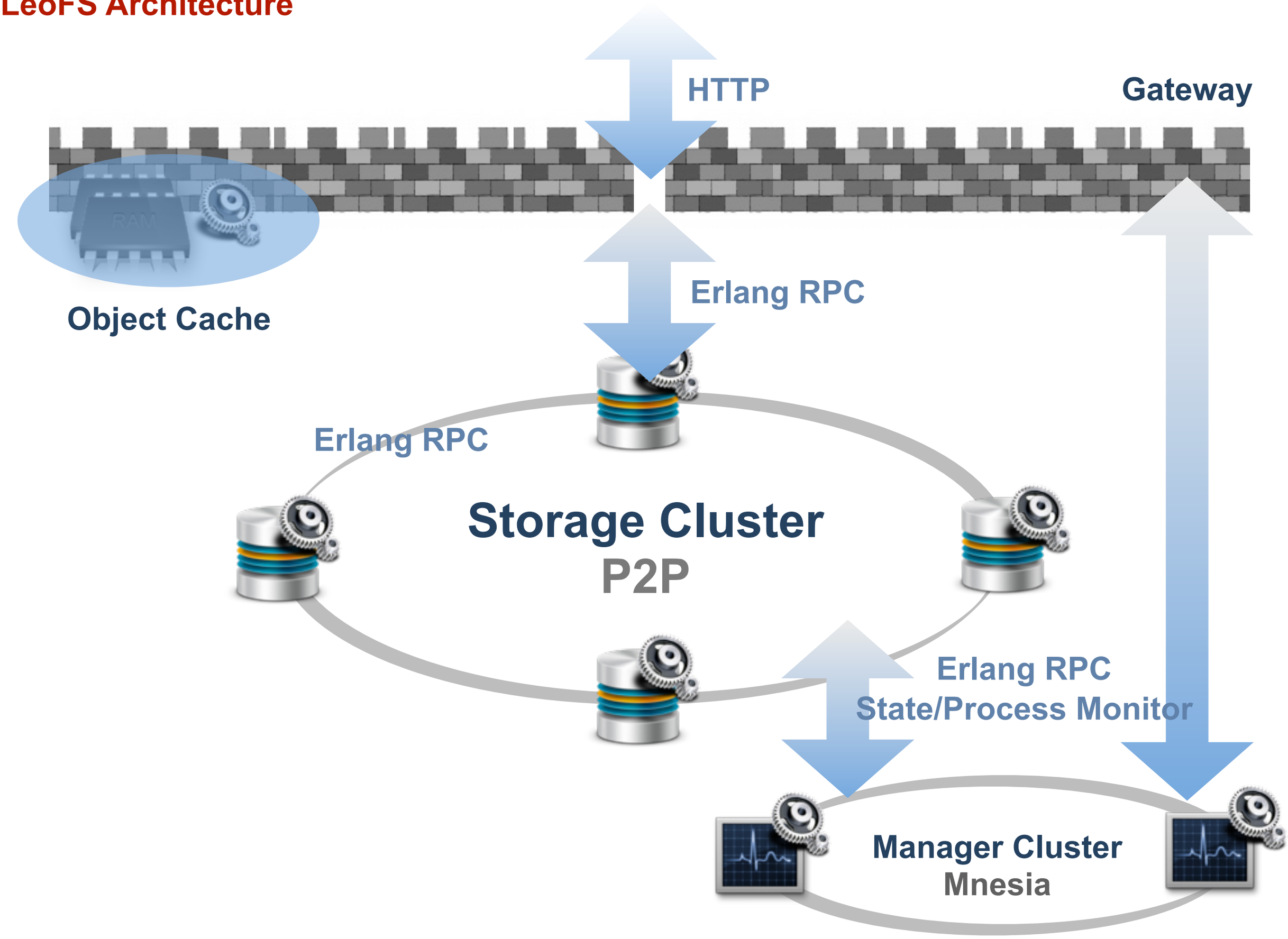
In Centralized LeoFS



Rakuten

Inside LeoFS

LeoFS Architecture



Rakuten

LeoFS Gateway

Rakuten

“Stateless Proxy”

“Cowboy”

From Applications

amazon web services S3-API

Gateway

RAM

Object Cache

[LRU, Slab allocator, Skip graph]

Consistent Hashing

Horizontal Distribution

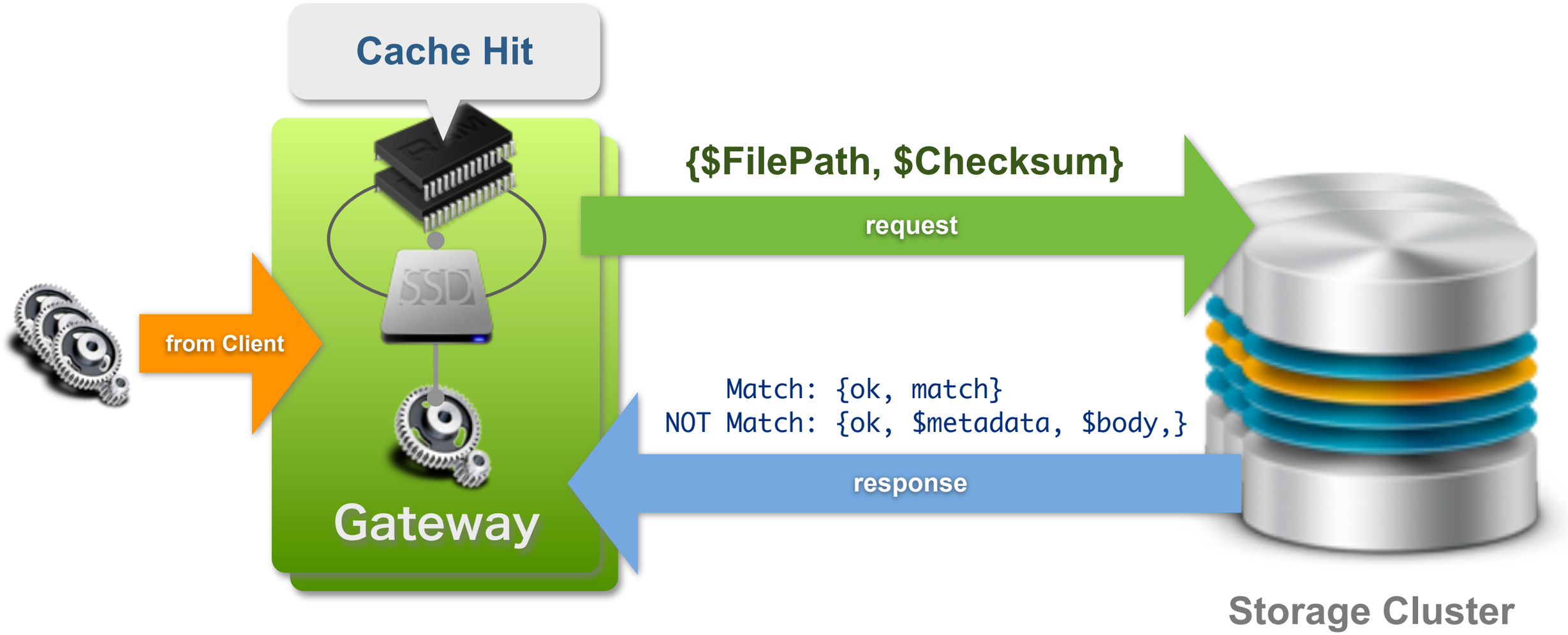
Storage Nodes

Replicate when using RPC

*Cowboy: Erlang light-weight HTTP-Server - http://http://www.ninenines.eu/

Rakuten

Object Cache Behavior



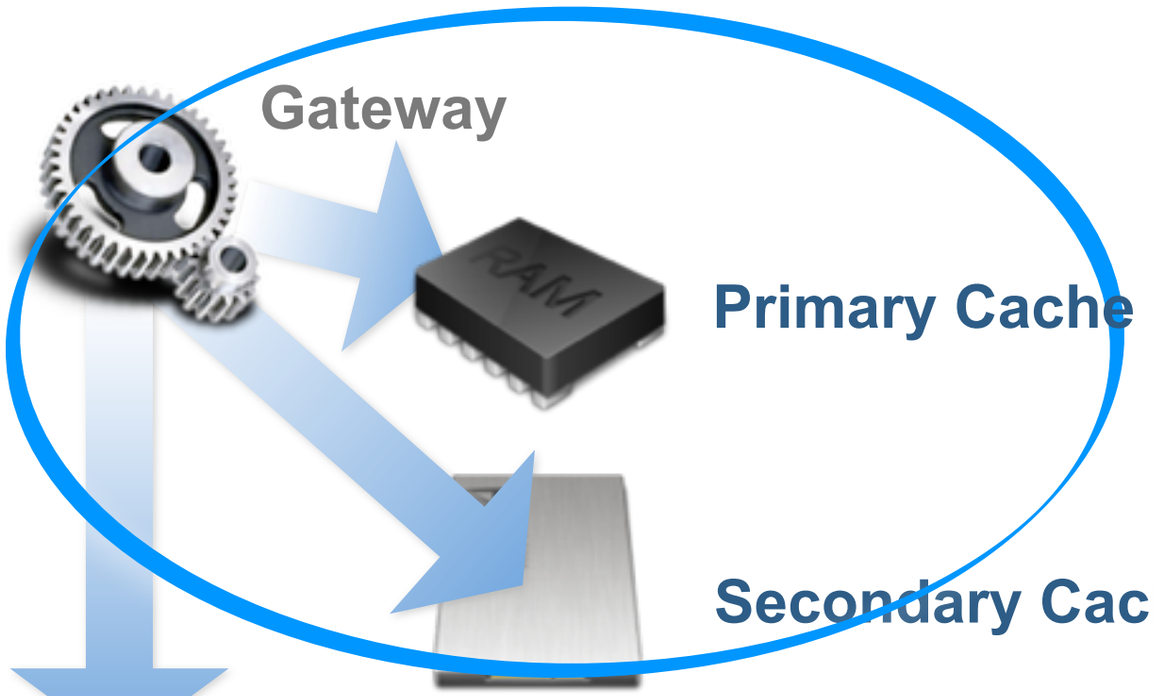
Object Cache always keep an object consistency

Rakuten

Hierarchical Object Cache (Using RAM, SSD)

HIGH-Performance (Low latency)

High I/O efficiency



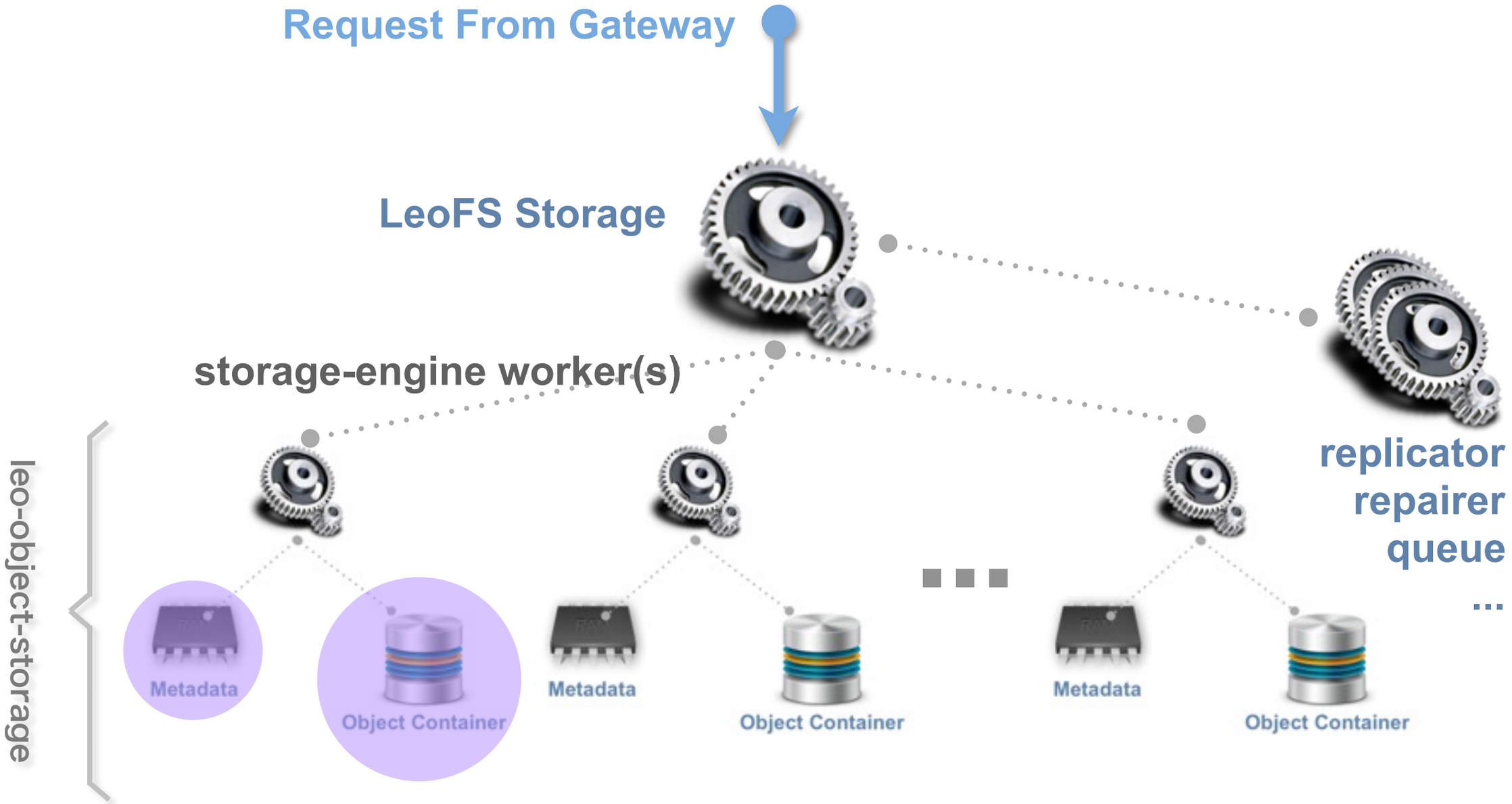
Storage Cluster

Reduce traffic between Gateway and Storage

Rakuten

LeoFS Storage

LeoFS Storage Engine



Metadata : Keeps an in-memory index of all data
Object Container : Log structured (append-only) file

Rakuten

Why do we need the original storage engine?

1. Input and Retrieve various data
2. Controllable “Compaction”

LeoFS Storage Engine - Data Structure

Rakuten

for Retrieve an File (Object)

for Sync

<Metadata>

{VNodeId, Key} | KeySize | Custom Meta Size | File Size | Offset | Version | Time-stamp | Checksum

<Needle>

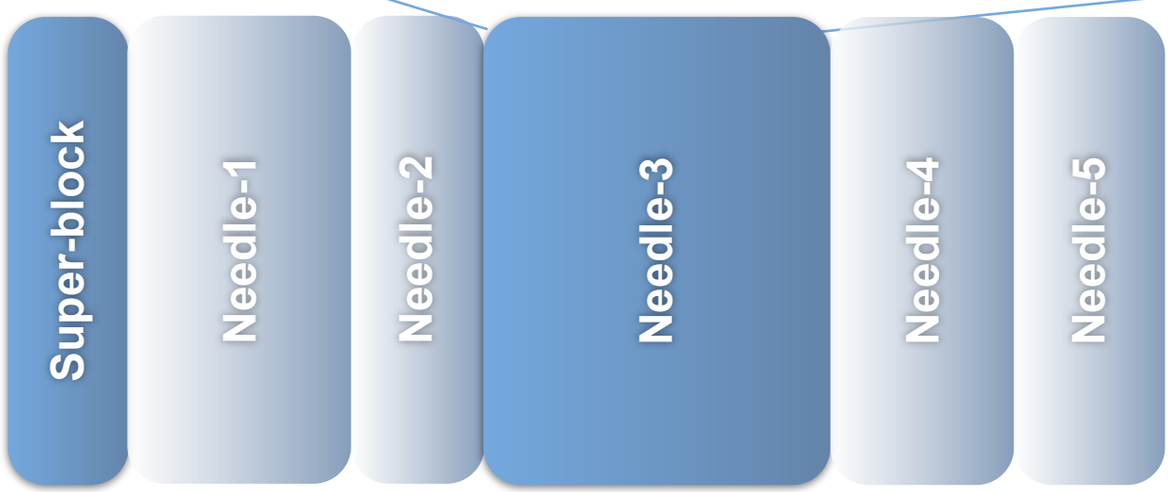
Checksum | KeySize | User-Meta Size | DataSize | Offset | Version | Time-stamp | {VNodeId, Key} | User-Meta | Actual File | Footer

Header (Metadata - Fixed length)

Body (Variable Length)

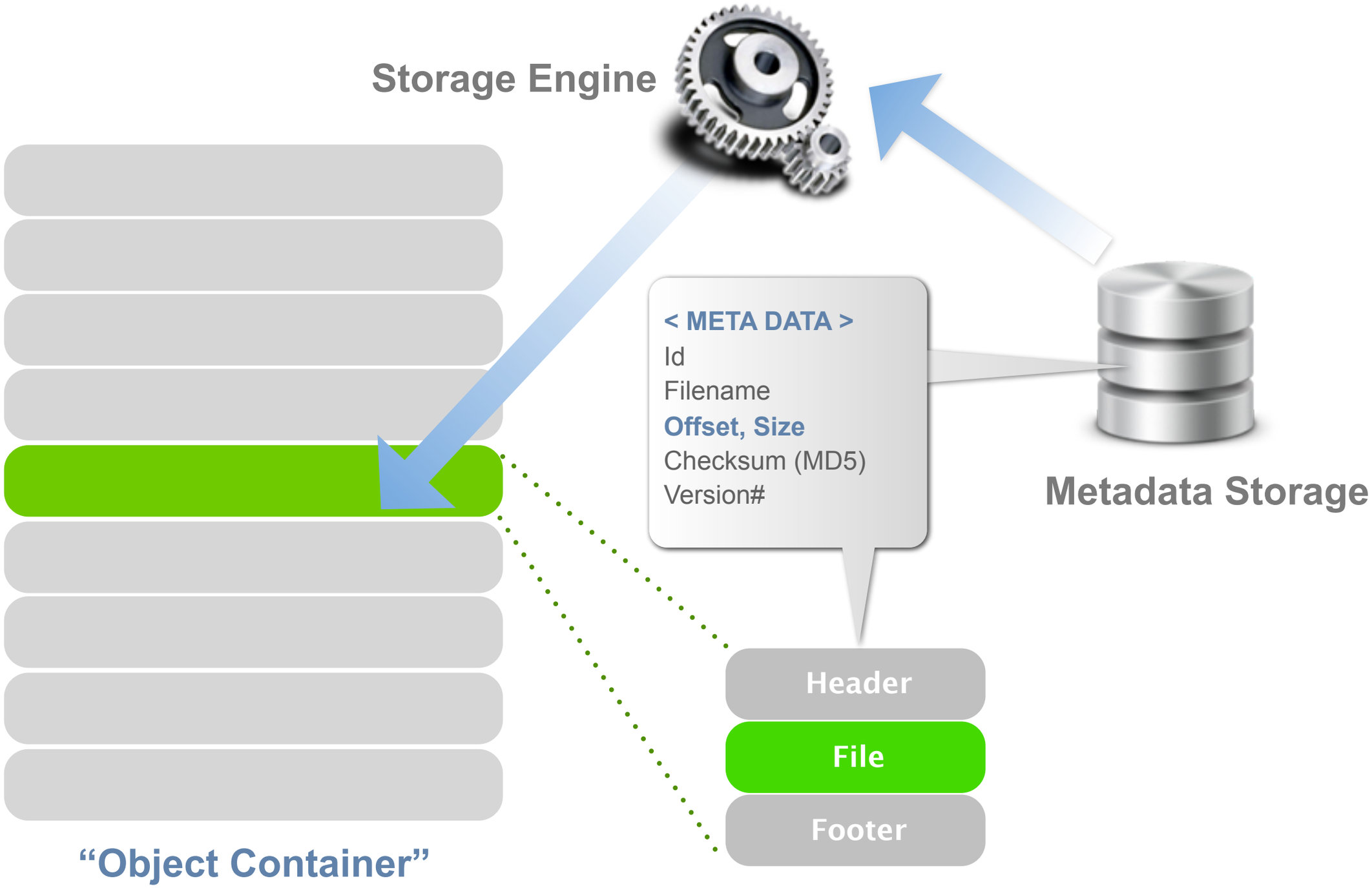
Footer (8B)

<Object Container>



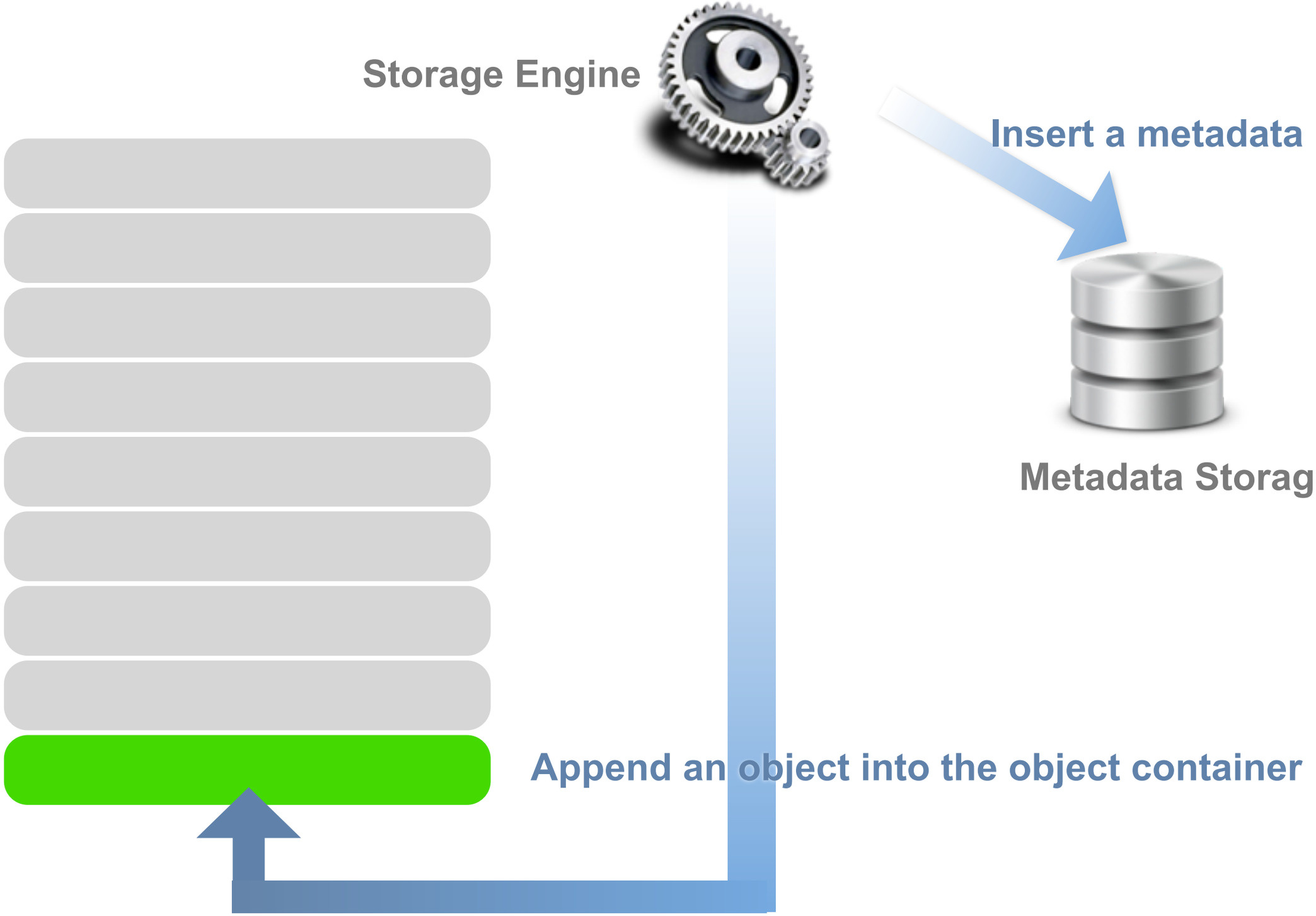
LeoFS Storage Engine - Retrieve an object from the storage

Rakuten



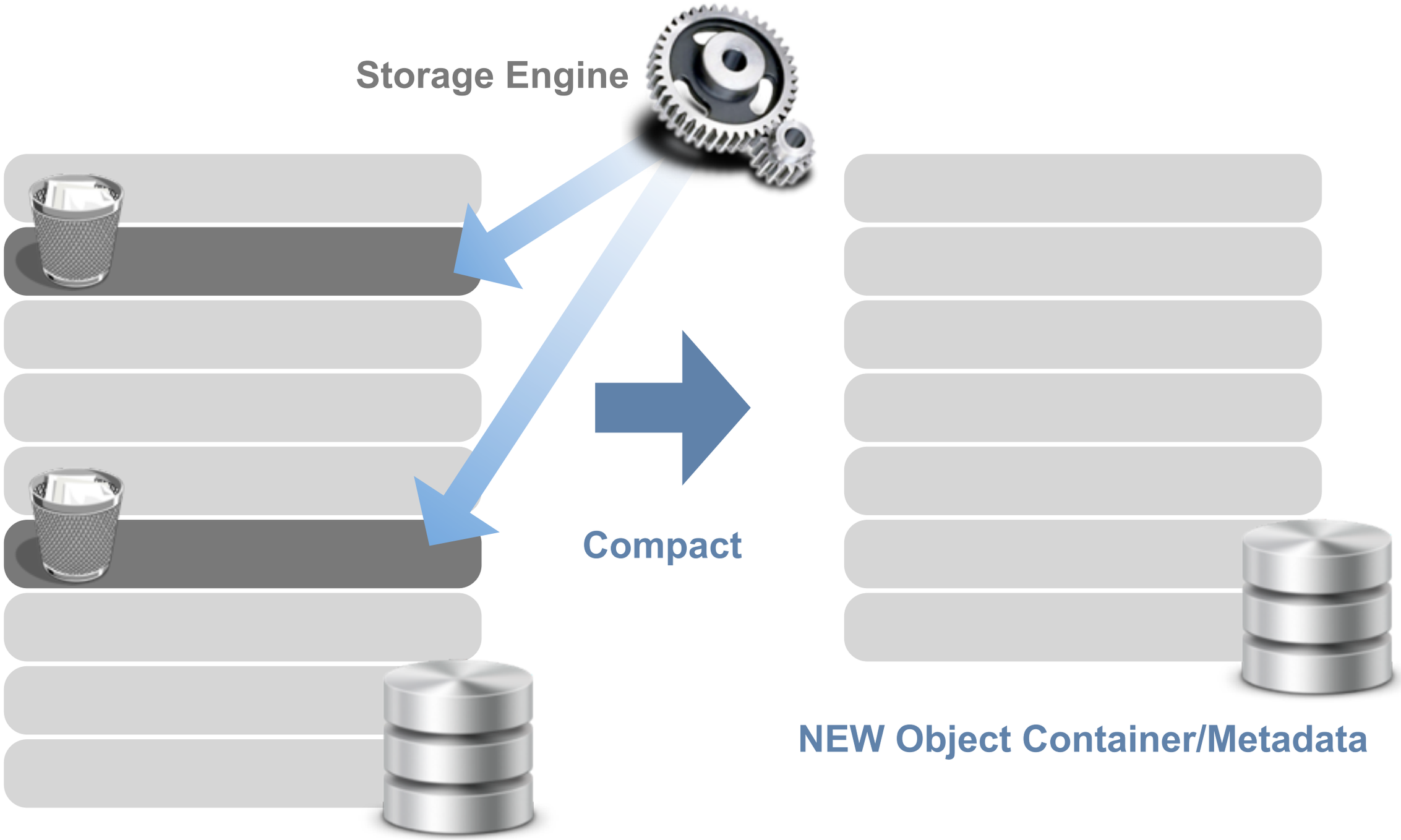
LeoFS Storage Engine - Insert an object into the storage

Rakuten

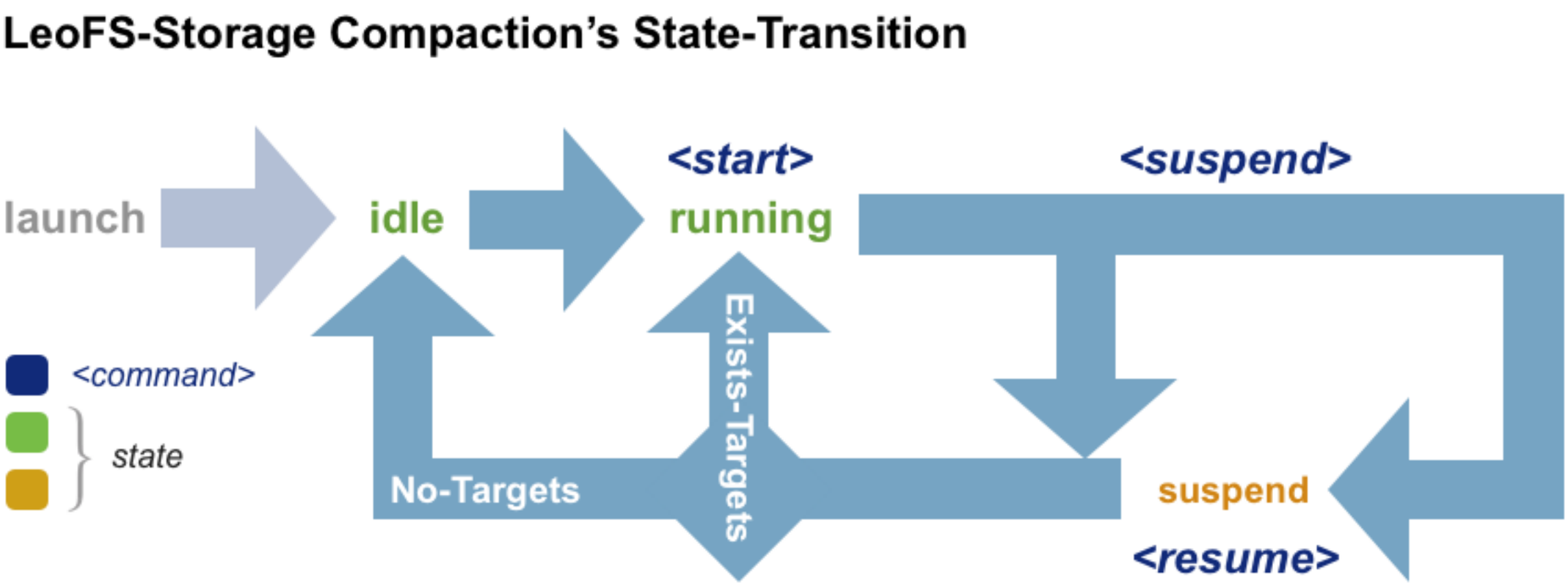


LeoFS Storage Engine - Remove unnecessary objects from the storage

Rakuten



Compaction Manager = “FSM”



“Phased Compaction”

Plan to support “auto-compaction”

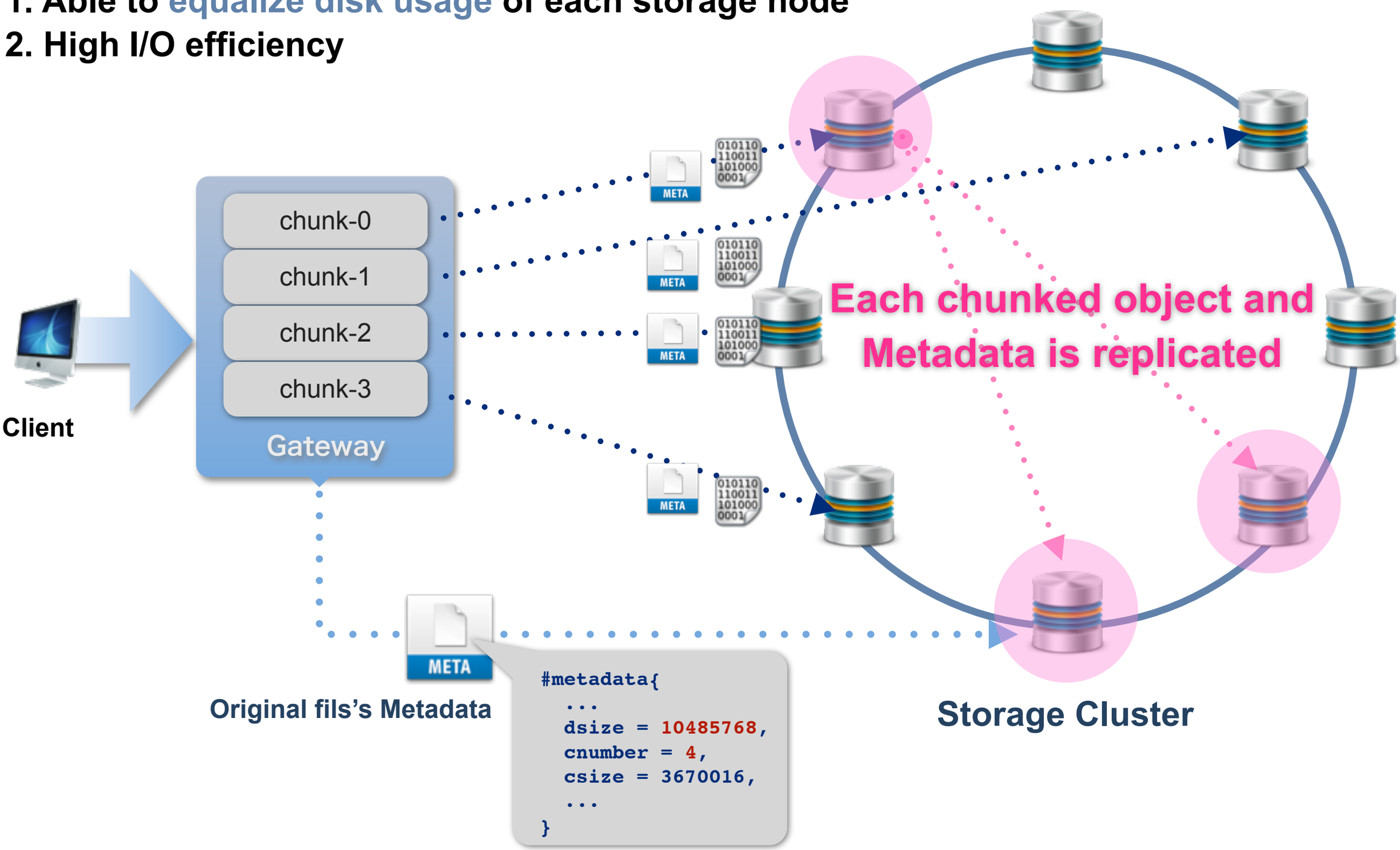
with v0.14.2

Rakuten

Large object Support

Rakuten

1. Able to equalize disk usage of each storage node
2. High I/O efficiency

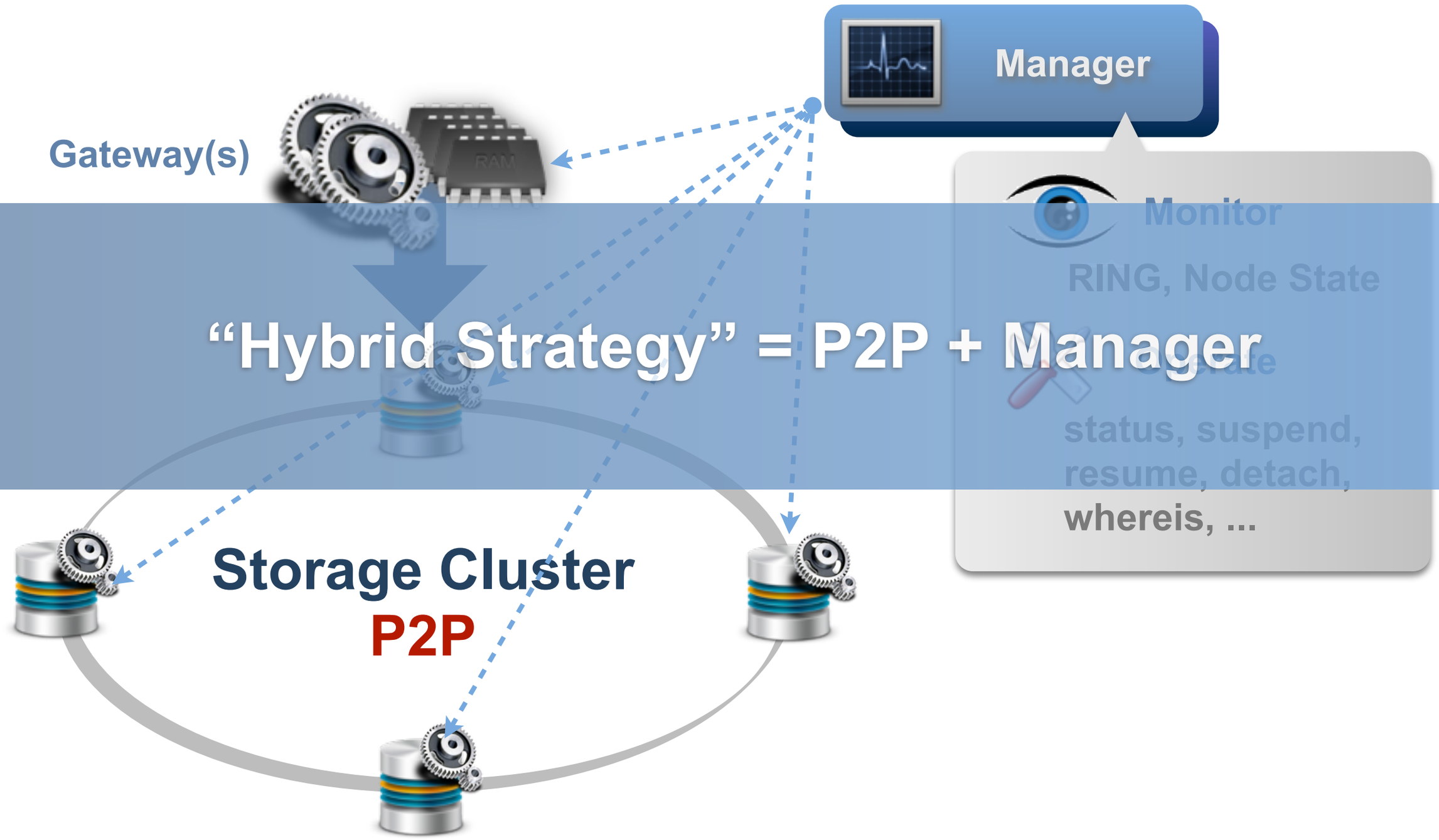


Rakuten

LeoFS Manager

Rakuten

For Keep High-Availability and Easy System Operation



“Hybrid Strategy” = P2P + Manager

Integrated Storage System

Request from Web Application(s)
or Browser

Load Balancer

amazon web services S3-API

REST over HTTP
(80/443)

LeoFS-Manager

RPC
(4369)

(4000,4010,4020)

LeoFS-Gateway

Monitor
(SNMP)

LeoFS-Storage

RPC (4369)

(10020, 10021)

GUI Console

Storage Engine/Router

RAM

META

Object Store

Storage Engine/Router

RAM

META

Object Store

Storage Engine/Router

RAM

META

Object Store

Brief benchmark report

Benchmark

LeoFS v0.12.7

Rakuten

Benchmark-er [2]

amazon web services S3-API

# of concurrent = 100

[Condition]

```
# of Replica           = 3
# of Successful WRITE  = 2
# of Successful READ   = 1
```

REST over HTTP
(80/443)

LeoFS-Gateway [1]

LeoFS-Manager

RPC (4369)

RPC
(4369)

LeoFS-Storage [5]

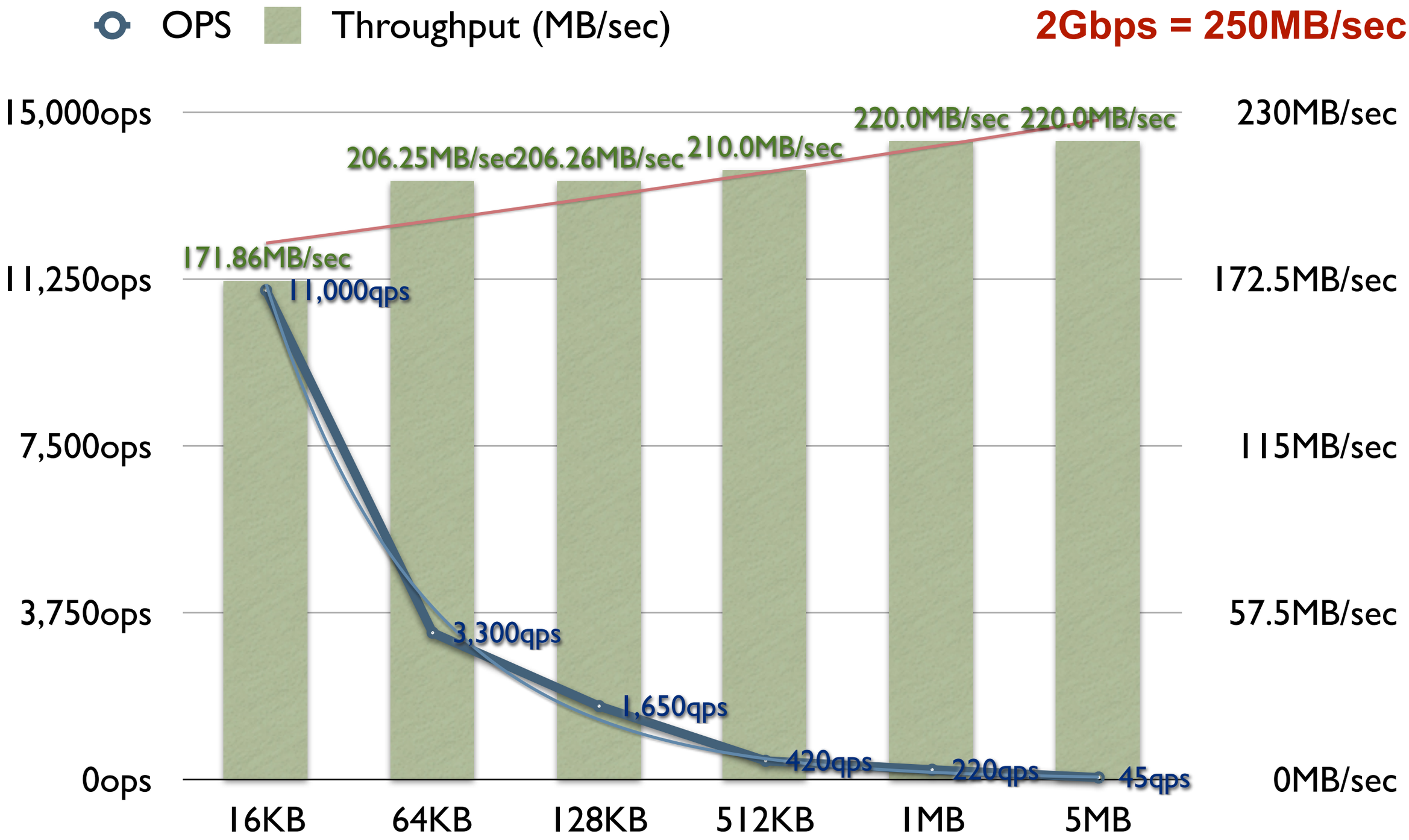
Rakuten

Item	Value
Network	2Gbps (bonding 1Gbps*2)
OS	Ubuntu 12.04 LTS Server
CPU	XEON 2.2GHz (2Core/4Thread)
HDD	RAID-0 / HDD x 6
RAM	16GB
Erlang's version	R15B03-1
At the start of # of objects	1,000,000
# of replicas	3
# of successful write	2

Benchmark-1

Rakuten

READ : WRITE = 8 : 2



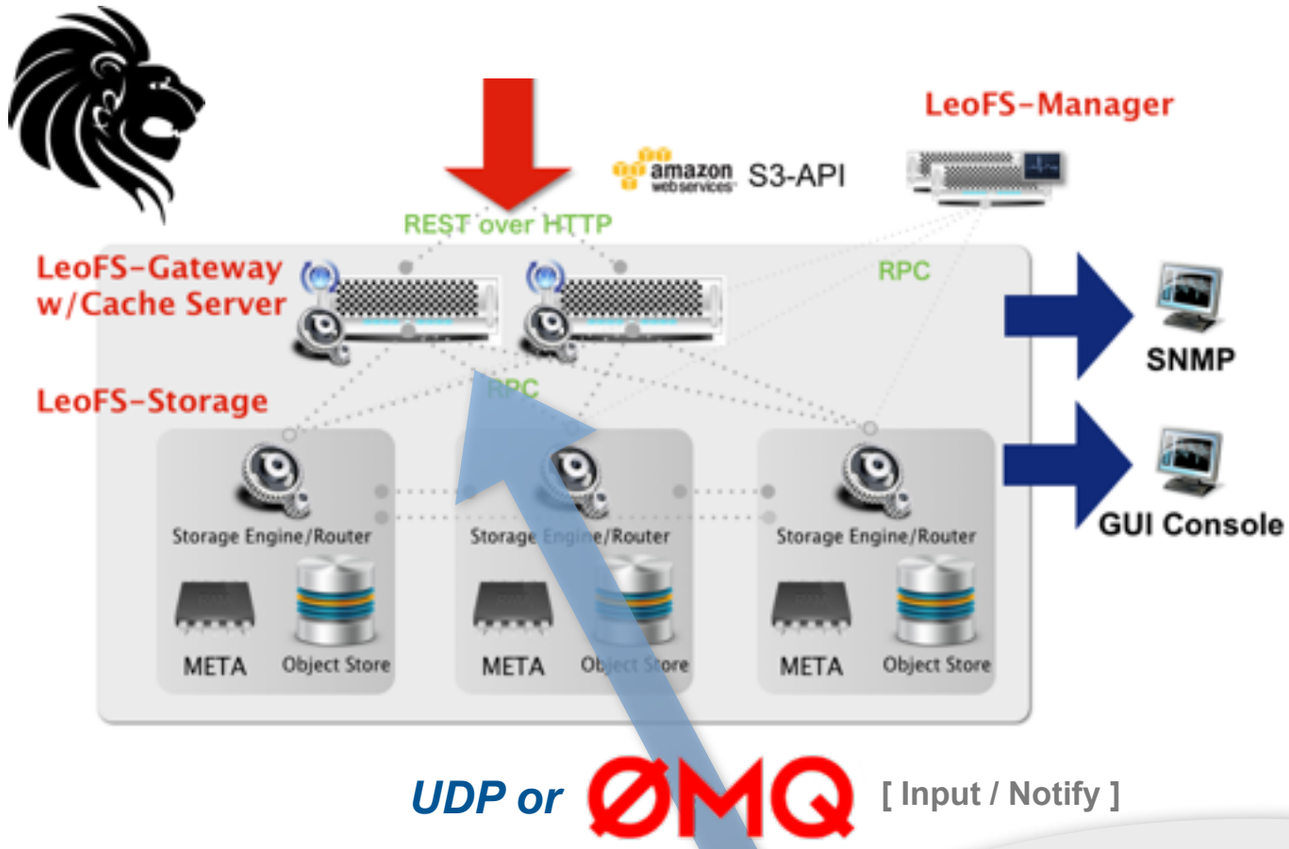
Rakuten

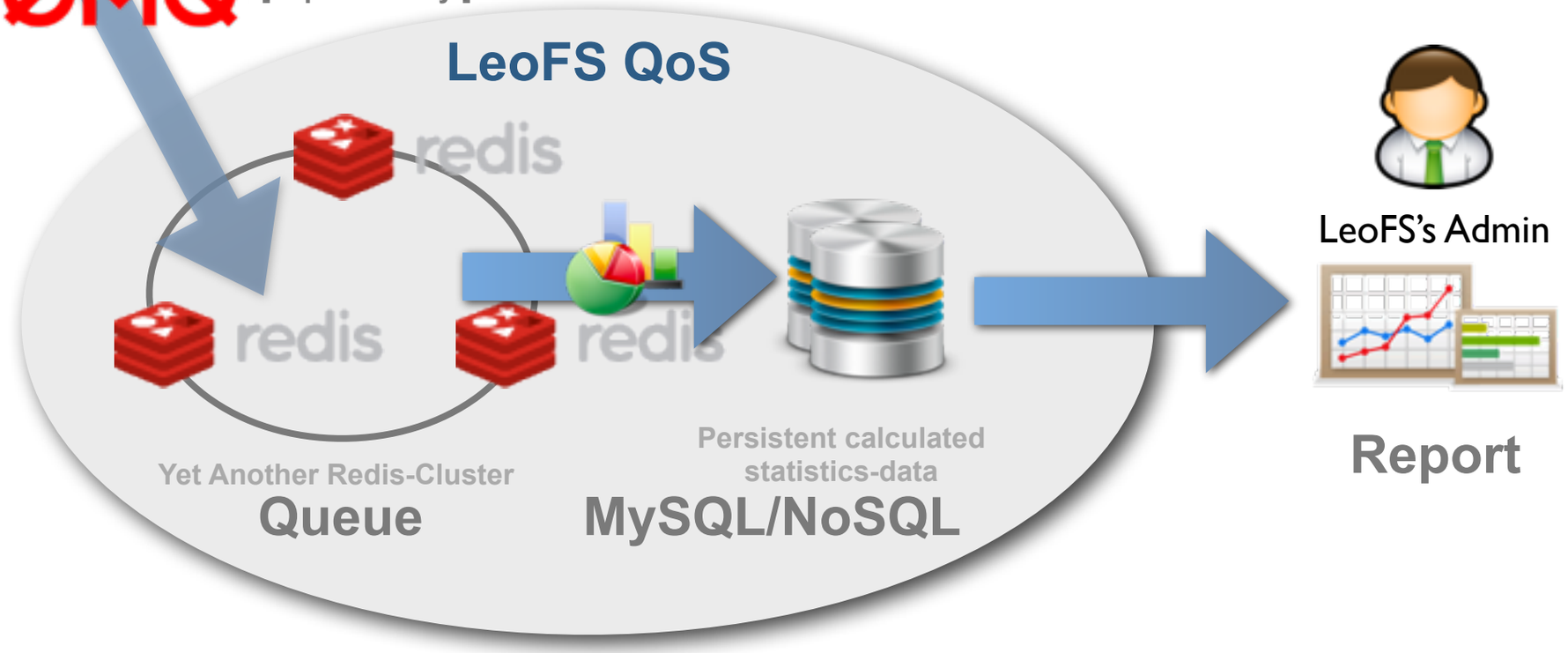
Demo

Future Works

Future Works - LeoFS QoS

Rakuten





[Purpose]

1. **Able to control request from clients to LeoFS**
2. **Able to store/see LeoFS's traffic data**

* The **quality of service** (**QoS**) refers to several related aspects of telephony and computer networks that allow the transport of traffic with special requirements.

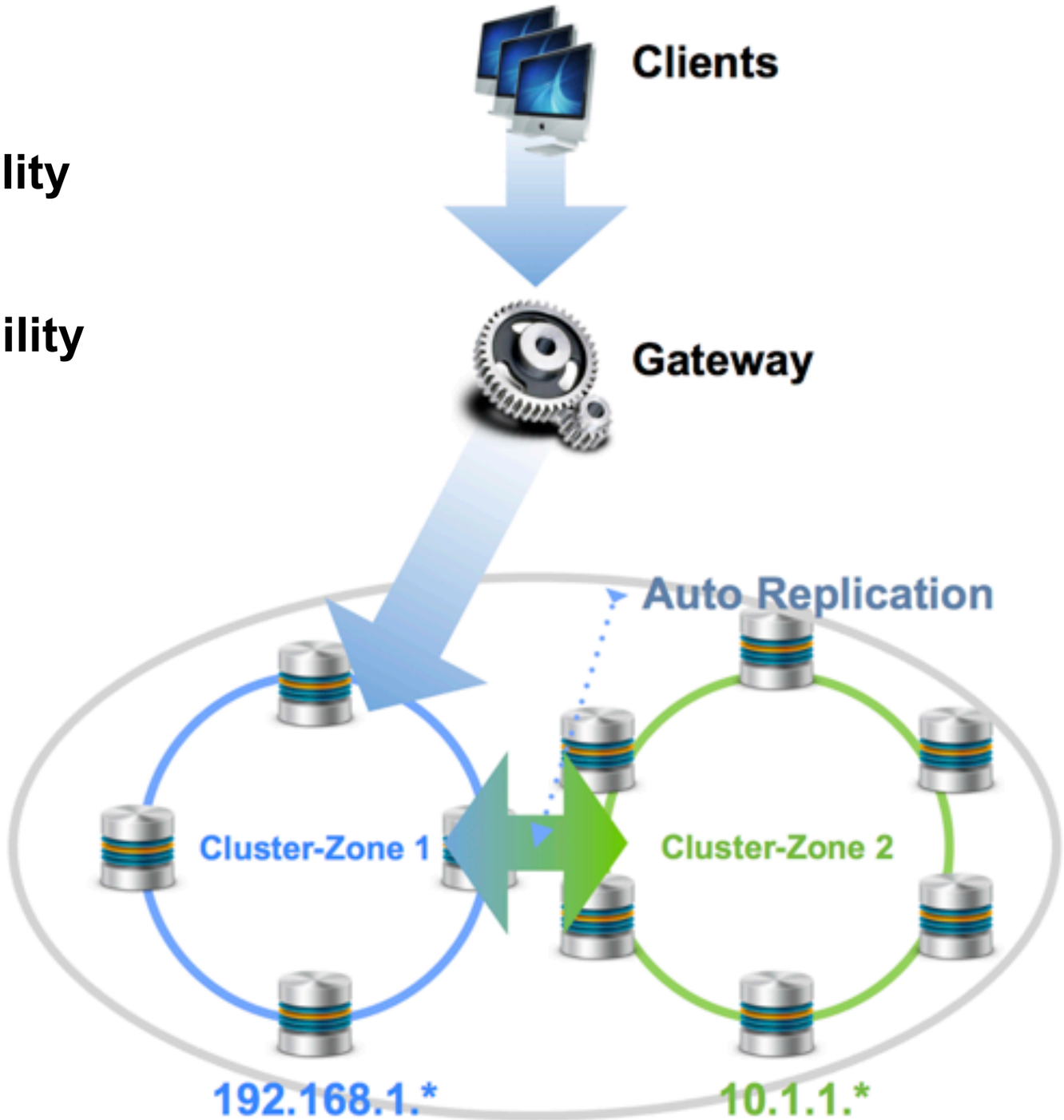
Future Works - Multi-Datacenter Replication

Rakuten

[Purpose]

HIGH-Scalability

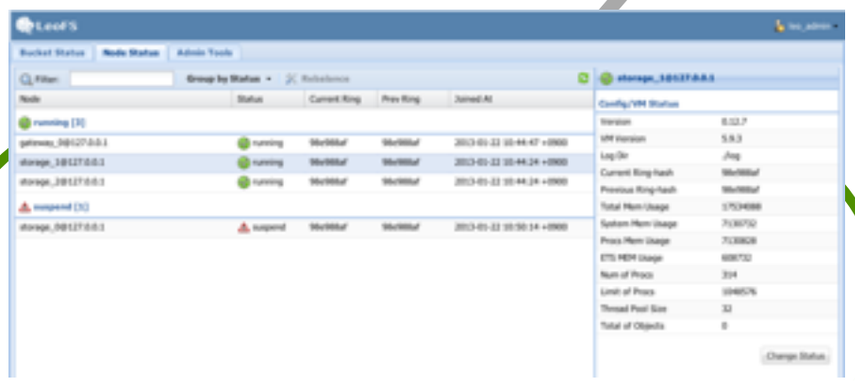
HIGH-Availability



Rakuten

Wrap Up

Wrap Up



LeoFS GUI-Console

LeoFS QoS

Rakuten

Q & A


Thank you for your time
LeoFS - http://www.leofs.org