# Towards FIT-aware scheduling policies for Cloud Computing

Vinicius Vielmo Cogo

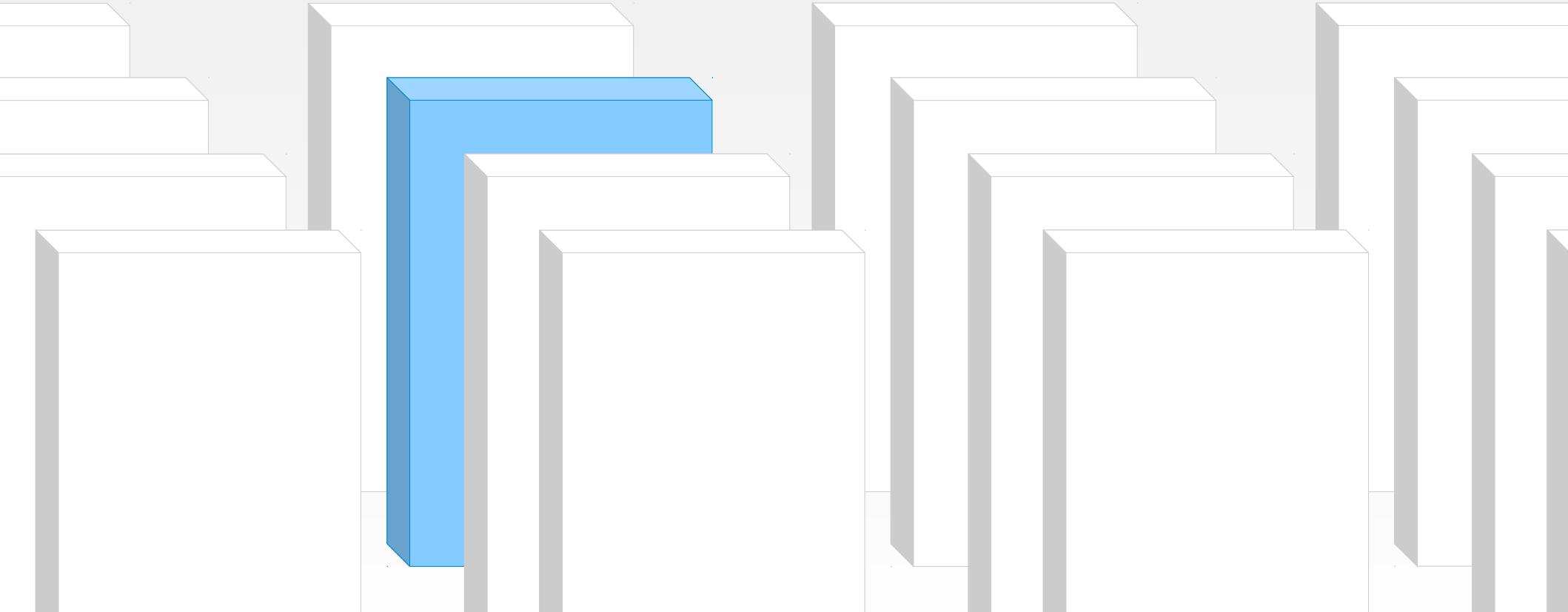
Marcelo Pasin

# The Motivation:

**Cloud Computing managers deal with large amounts of globally distributed resources.**

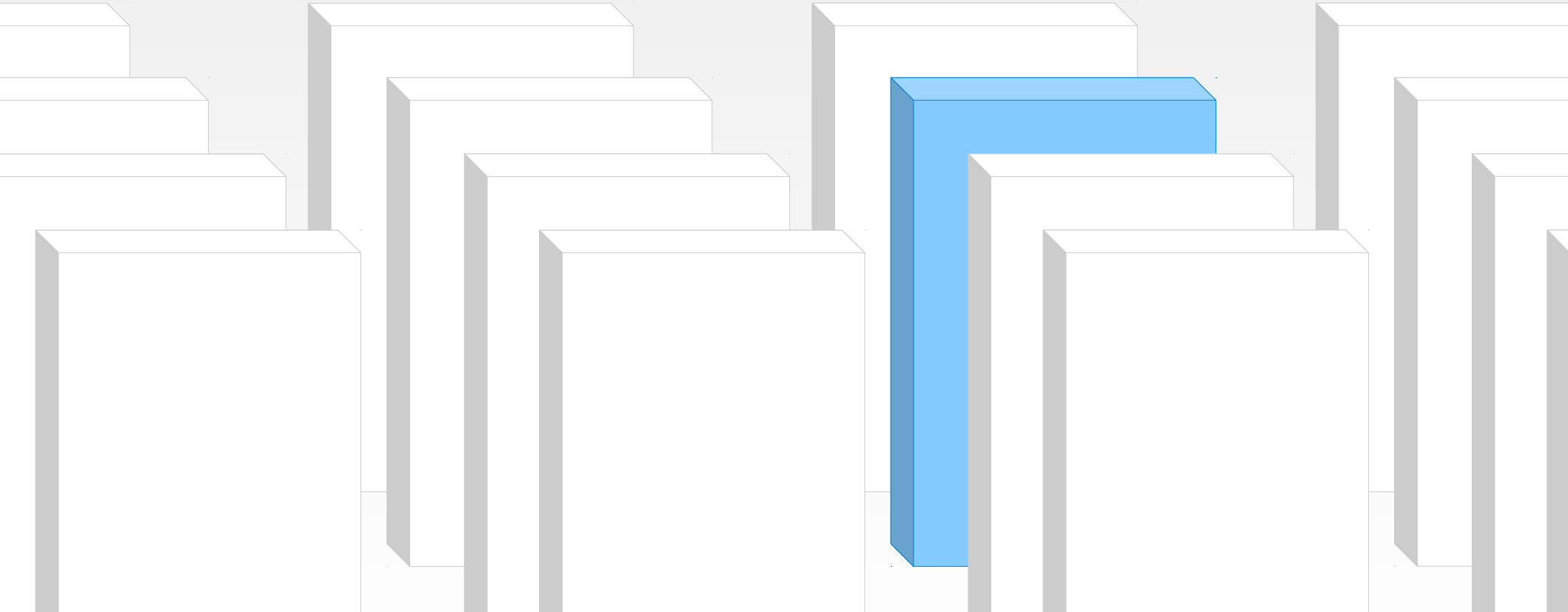
# The Problem:

**Scheduling policies are mainly focused on performance and economy.**



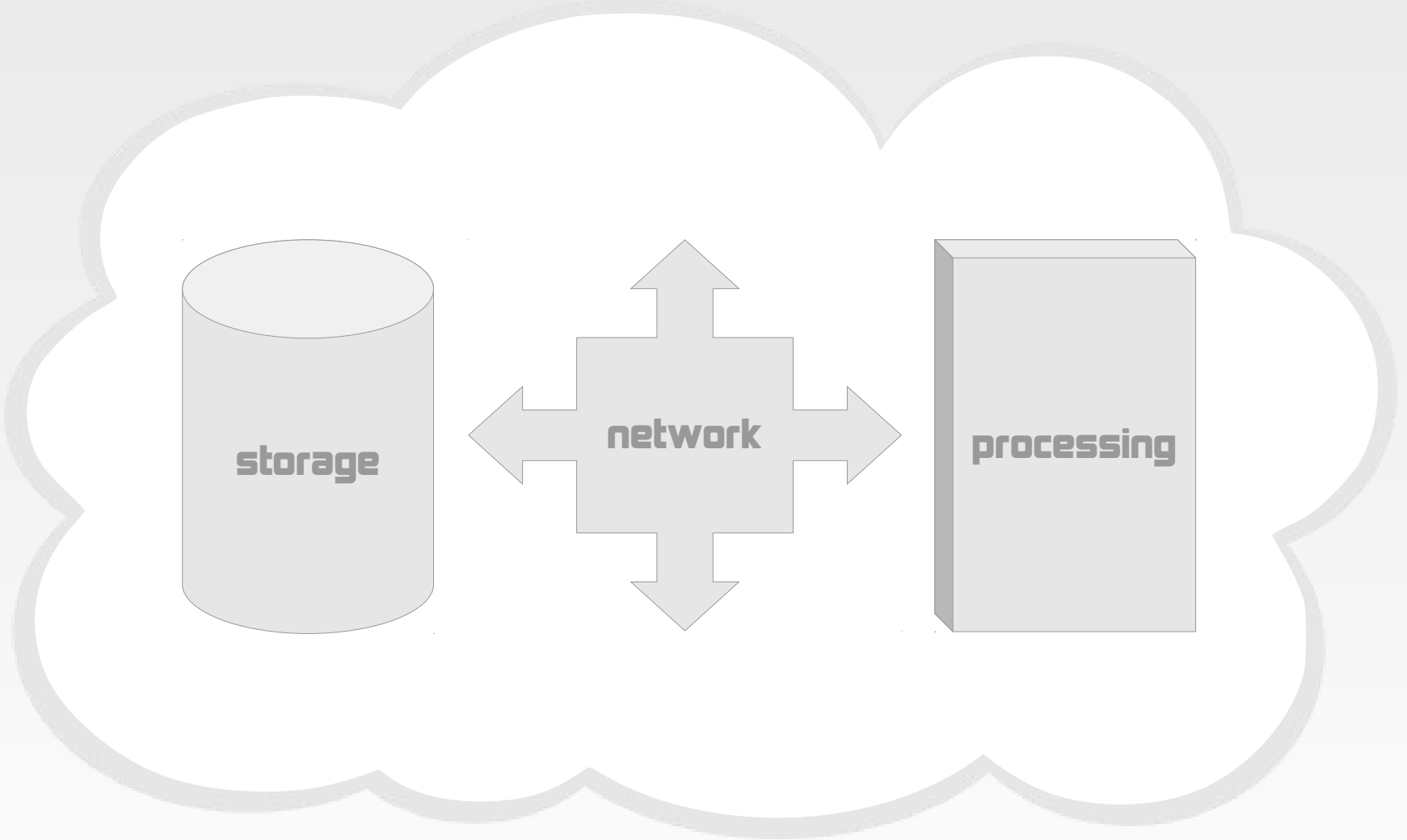
# Main Goal:

**Choose the best resource for a Fault and Intrusion Tolerant system.**



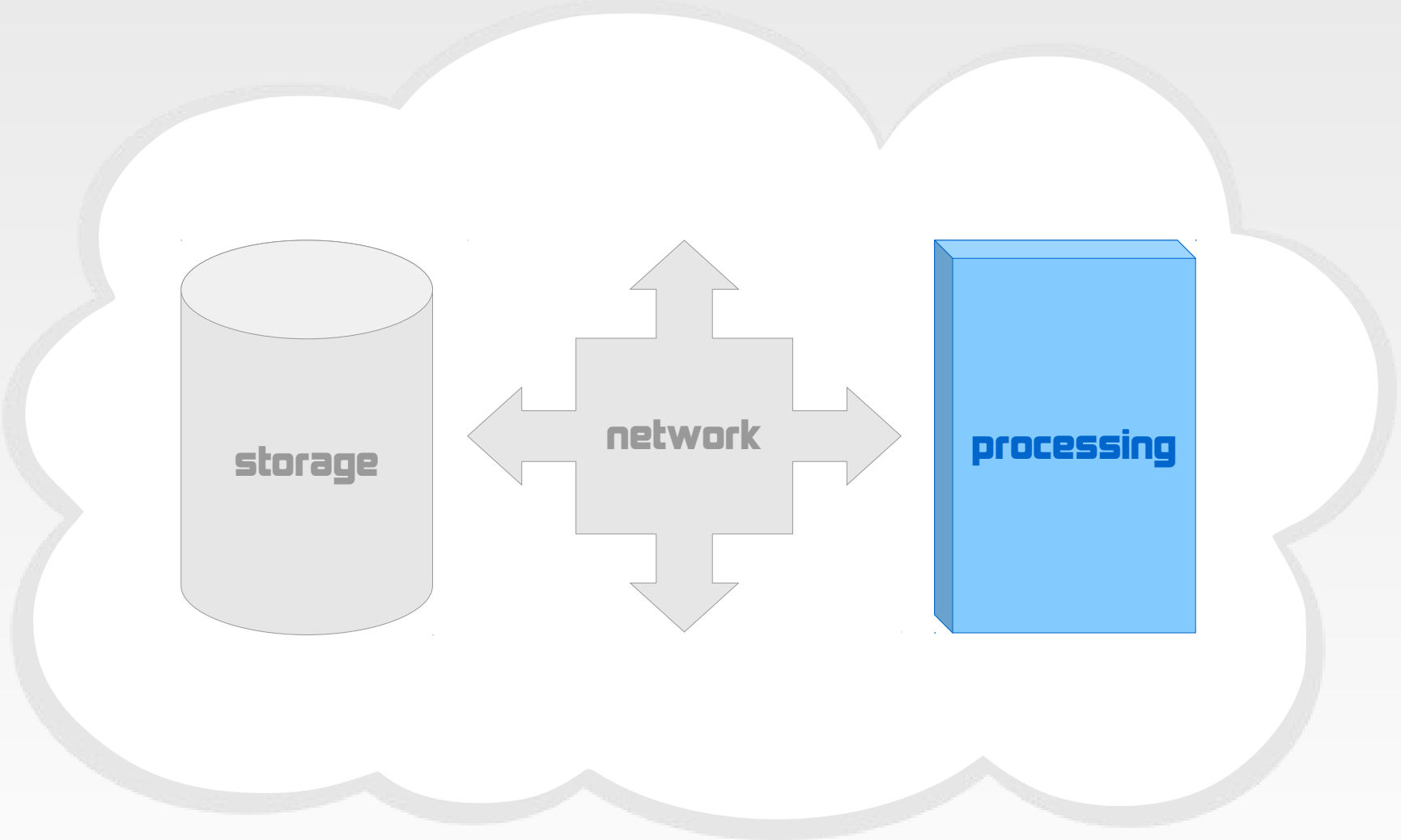
# Background:

- There are basically 3 types of resources in Cloud Computing (IaaS)



# Background:

- This work is focused on Processing resources



# Outline:

1 – CloudFIT Project

2 – Intrusion-Tolerance Mechanisms

3 – Resource Allocation Algorithm

4 – Requirements for Scheduling

5 – Future Work

6 – Open Questions

# 1 - CloudFIT Project:

**CloudFIT: Fault and Intrusion Tolerance for Clouds**

**- Goal:**

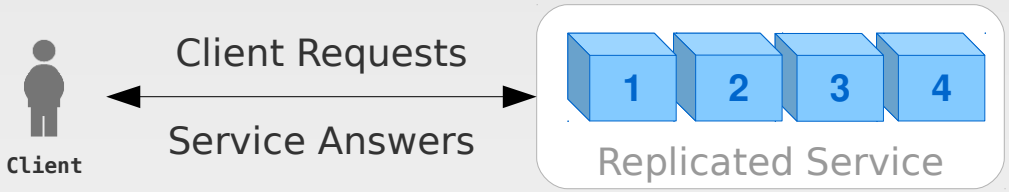
Create an infrastructure for FIT services in a cloud environment

**- Components:**

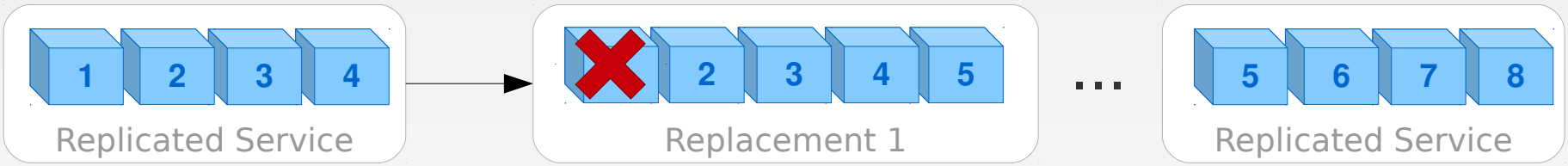
- Cloud Resource Manager (OpenNebula)
- Hosts with Virtual Machine Monitors (e.g.: Xen)
- Service that uses the bft-SMaRt library for State Machine Replication
- And others ...

# 2 - Intrusion-Tolerance Mechanisms:

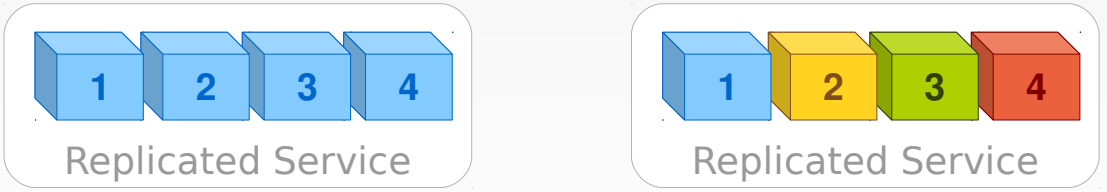
## State Machine Replication ...




## Proactive Recovery ...




## Diversity ...



# 2 - Intrusion-Tolerance Mechanisms:

## State Machine Replication ...

... to increase integrity and availability

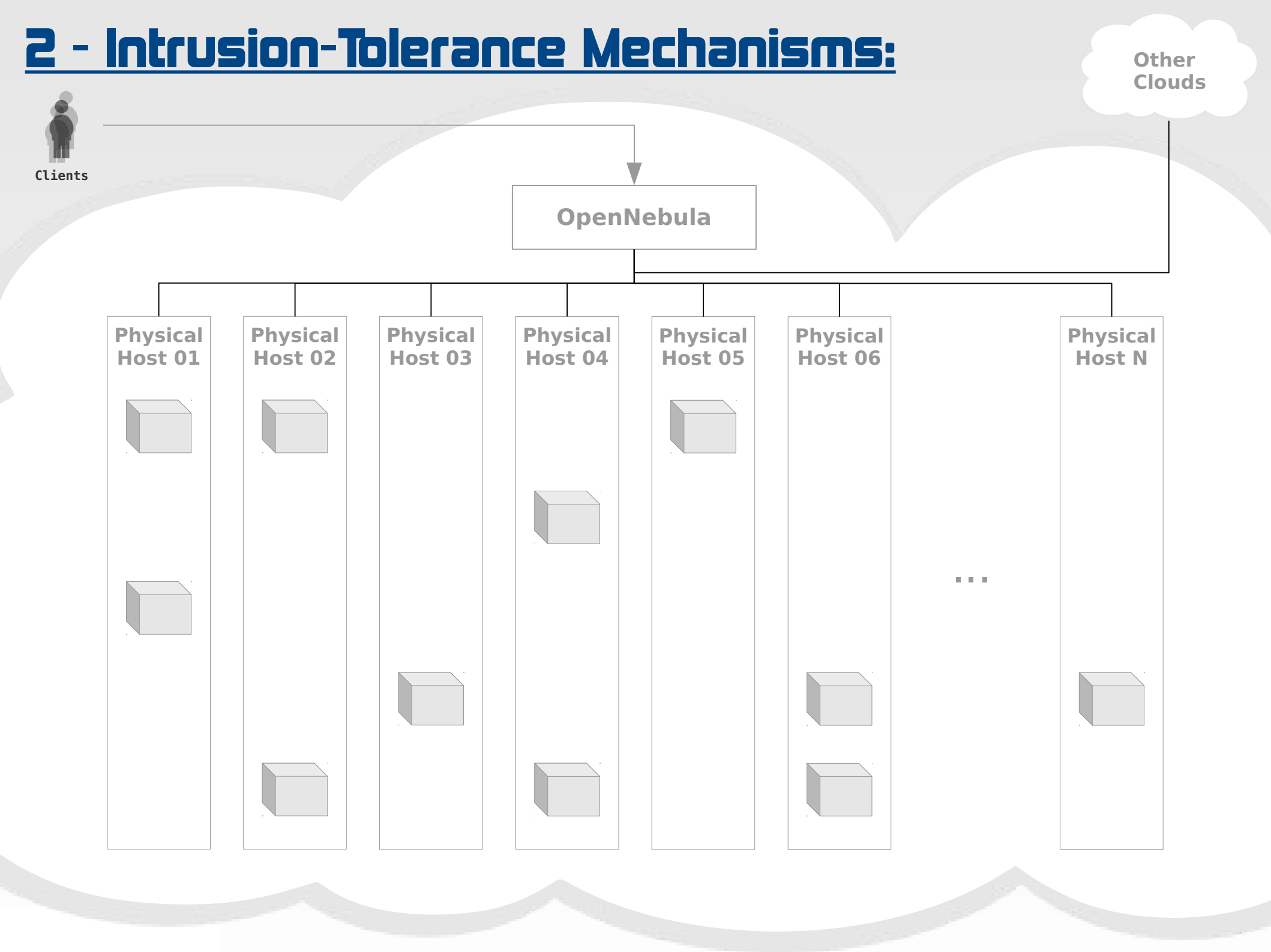
## Proactive Recovery ...

... to tolerate any number of intrusions over the system's lifetime

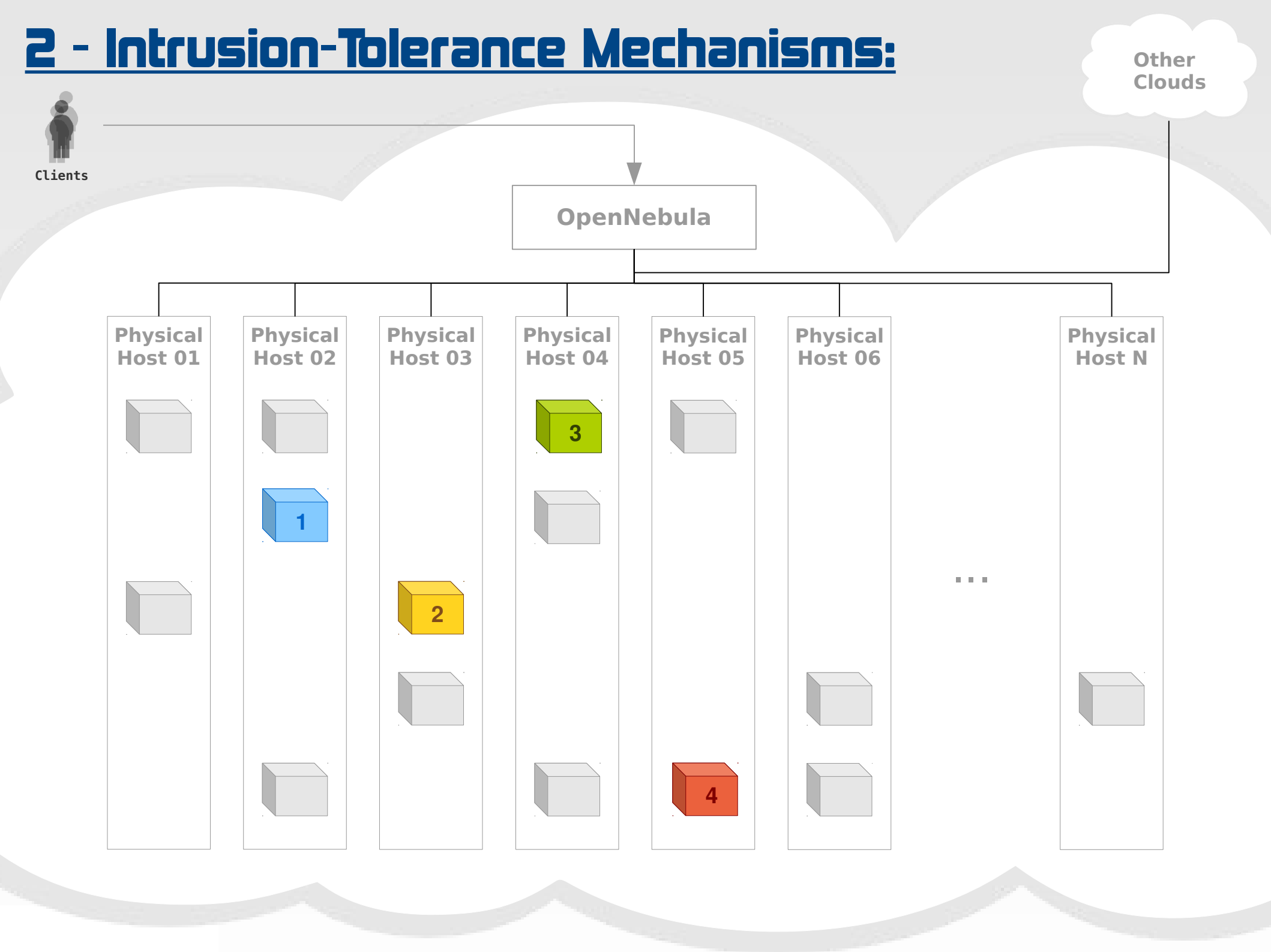
## Diversity ...

... to increase the difficulty of attacks and independence between faults

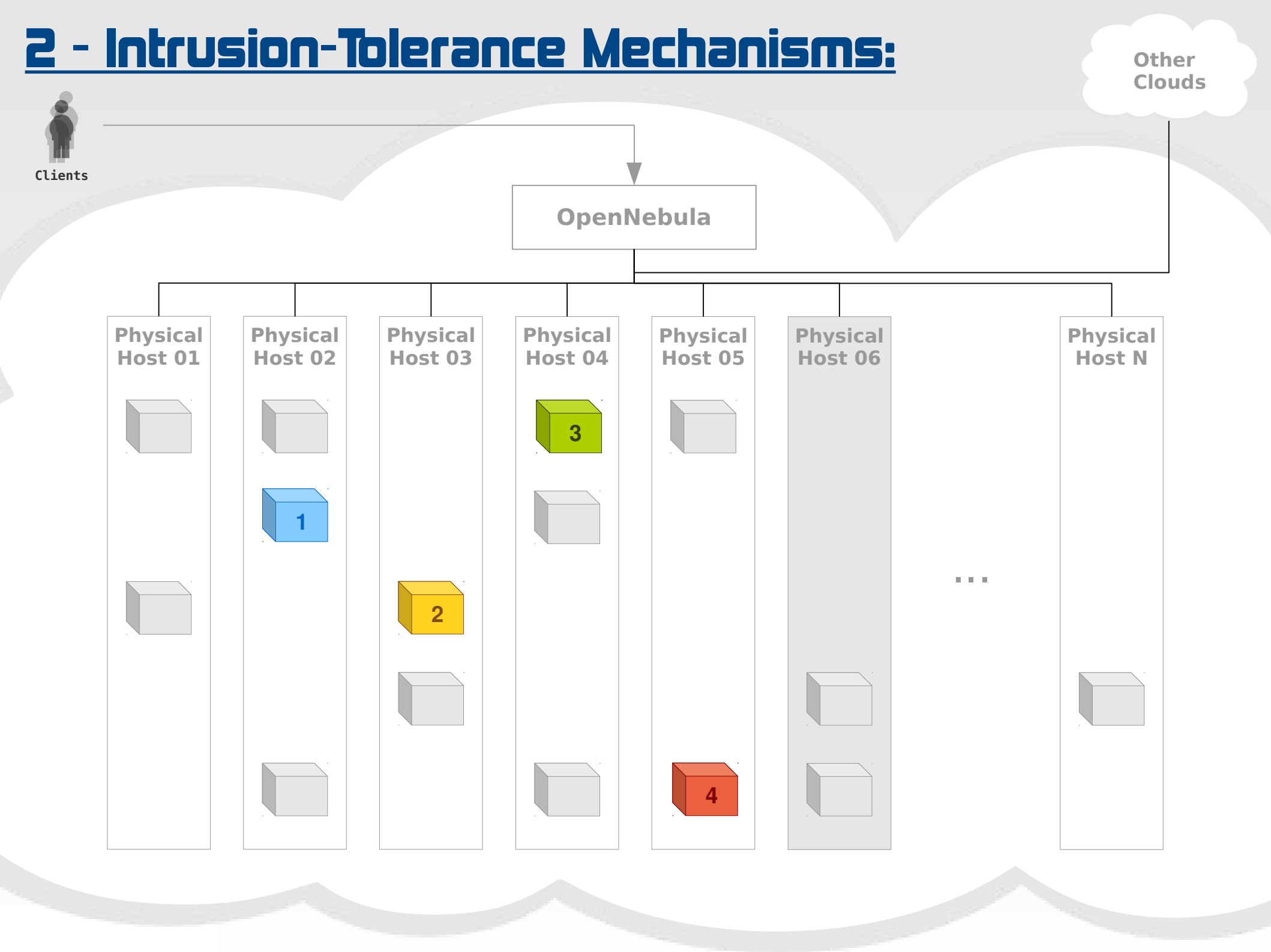
# 2 - Intrusion-Tolerance Mechanisms:



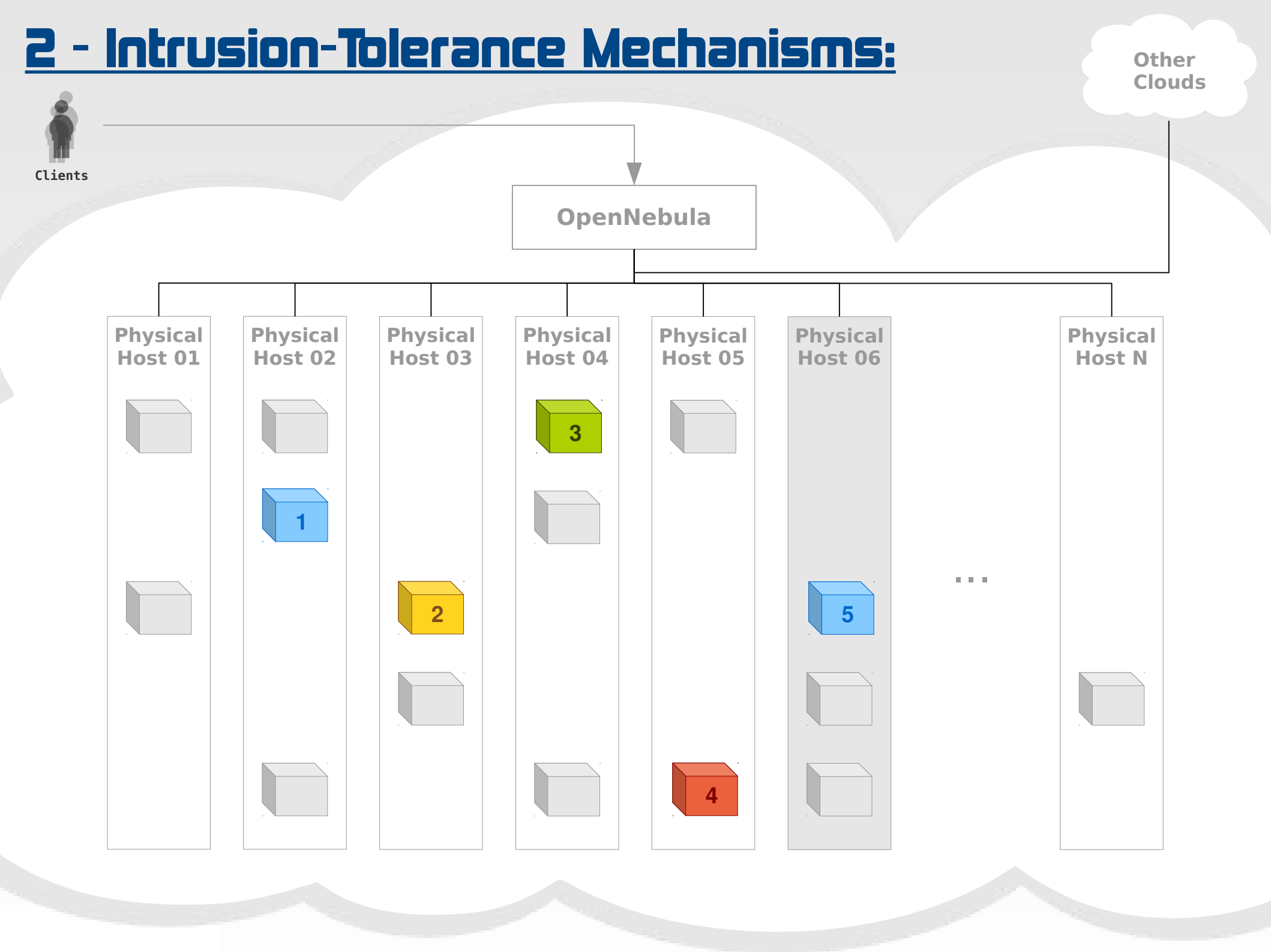
# 2 - Intrusion-Tolerance Mechanisms:



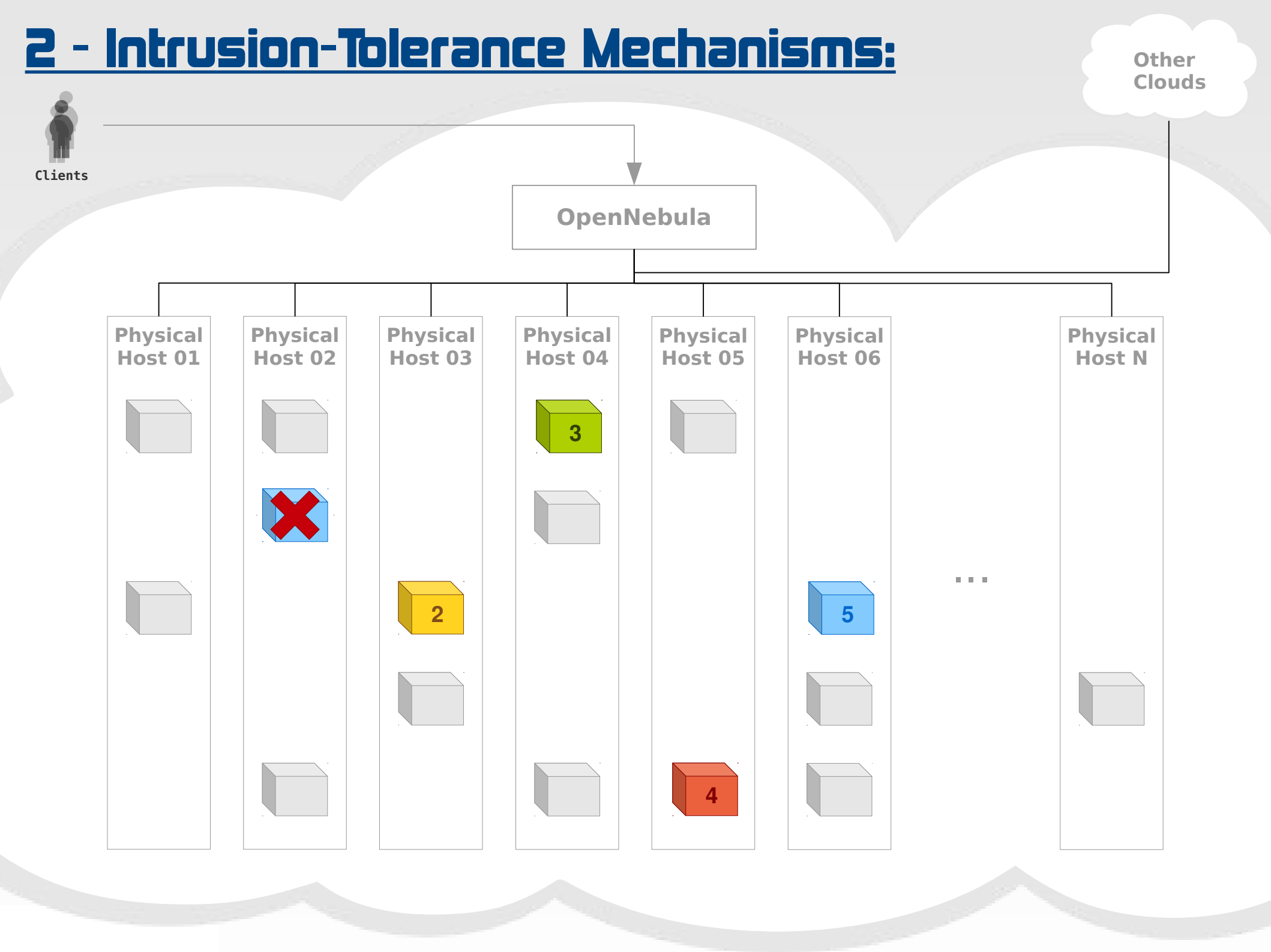
# 2 - Intrusion-Tolerance Mechanisms:



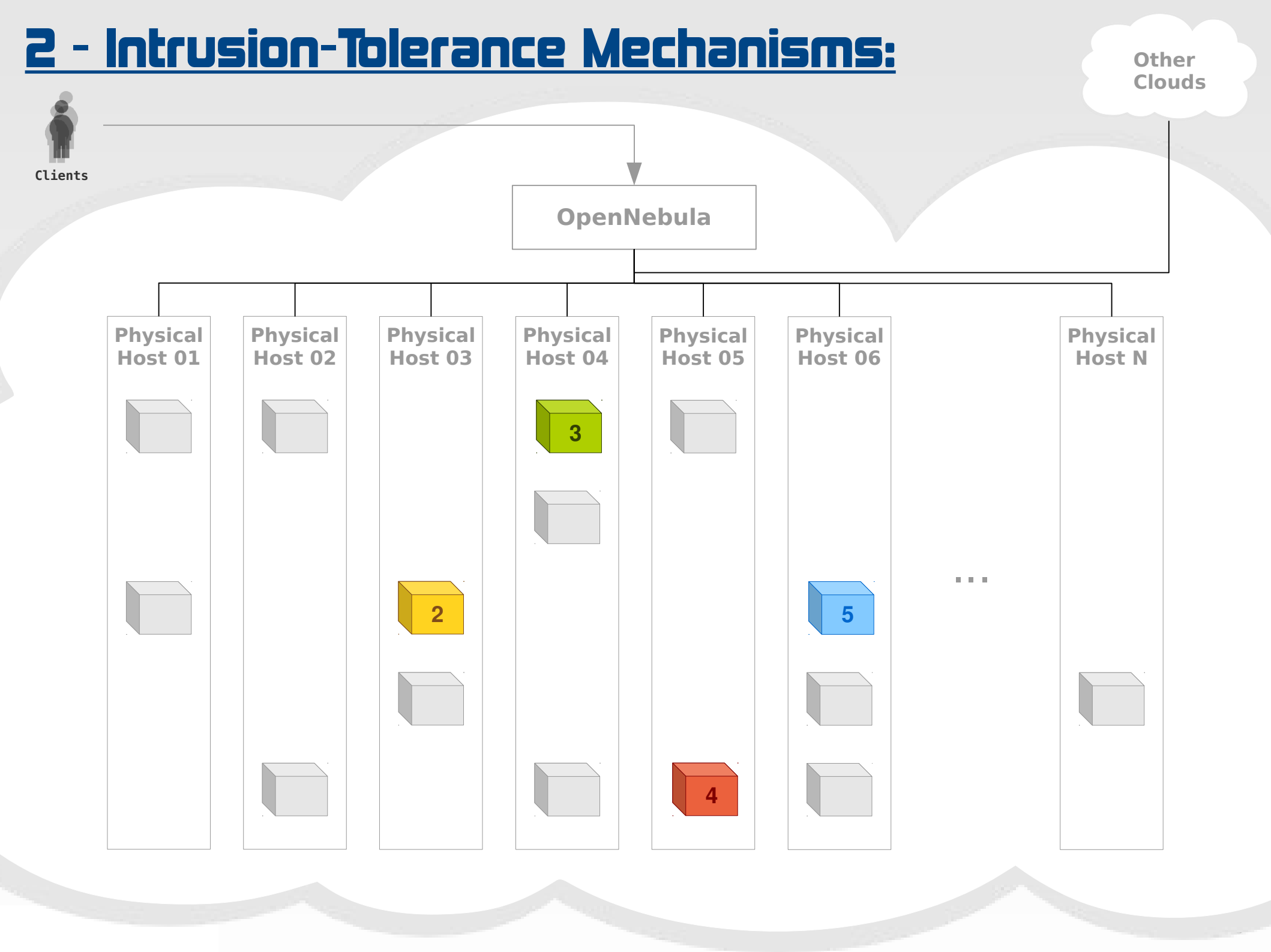
# 2 - Intrusion-Tolerance Mechanisms:



# 2 - Intrusion-Tolerance Mechanisms:



# 2 - Intrusion-Tolerance Mechanisms:



# 3 - Resource Allocation Algorithm:

**- Choose one host to allocate a Virtual Machine**

**-** Algorithms are normally based on **two steps**, or a combination of them:

- **REQUIREMENTS** (boolean expression)
- **RANK** (numeric expression)

- **OpenNebula** uses a matchmaking algorithm:
  1. Filter out the hosts based on **REQUIREMENTS**
  2. Rank the remaining hosts based on **RANK**
  3. Choose the **highest ranked host** to allocate the VM

# 3 - Resource Allocation Algorithm:

Clients

OpenNebula

**Requirement:** Hypervisor=Xen
**Rank:** Total of Memory

Module | Module | Information Manager | Module | Module

| HOSTNAME | HYPERVISOR | RUNNING_VMS | CPUSPEED | TOTALMEMORY |
|---|---|---|---|---|
| host01 | Xen | 5 | 3.2 | 32 |
| host02 | Xen | 2 | 3.2 | 16 |
| host03 | KVM | 0 | 2.3 | 8 |
| host04 | VMWare | 0 | 1.8 | 8 |
| host05 | VMWare | 3 | 2.5 | 8 |
| host06 | KVM | 6 | 2.7 | 16 |
| host07 | KVM | 1 | 3.0 | 32 |
| host08 | VMWare | 2 | 1.8 | 32 |
| host09 | VMWare | 3 | 2.7 | 16 |
| host10 | Xen | 4 | 2.3 | 4 |
| ... | ... | ... | ... | ... |
| hostN | Xen | 6 | 3.2 | 8 |

# 4 - Requirements for Scheduling:

**4.1 - Diversity** Requirements

**4.2 - Proactive Recovery** Requirements

**4.3 - State Machine Replication** Requirements

# 4.1 - The Various Diversities:

## Diversity of ...

### ... Application
N-Version programming

### ... Supporting Software
Manifold Operating Systems, Server or Daemon implementation

### ... Hardware
Distinct Architecture, CPU Model or Speed

### ... Administrative Domain
Different Hostname, Cluster or Cloud Provider

### ... Location
Varied GPS coordinates, Geopolitical structures

# 4.1 - Diversity of Administrative Domain:

**- Replicas should not be placed in the same physical host, rack or cluster**

| | | |
|---|---|---|
| HOSTNAME != ”some-host” | REQUIREMENT | ✓ |
| RACK != ”some-rack-id” | REQUIREMENT | ✗ |
| CLUSTER != ”some-cluster” | REQUIREMENT | ✓ |

**- Replicas should not be placed in the same cloud provider**

| | | |
|---|---|---|
| CLOUD_PROVIDER != ”some-cloud-provider”* | REQUIREMENT | ✗ |

**- Other measures:**

| | | |
|---|---|---|
| “- ROUND_TRIP_TIME to X.X.X.X” | RANK | ✗ |
| “- NETWORK_LATENCY to X.X.X.X” | RANK | ✗ |
| “- NUM_HOPS to X.X.X.X” | RANK | ✗ |

* Cloud Providers: Private, Amazon, Rackspace, Salesforce, ElasticHosts, GoGrid, SliceHost, etc.

# 4.1 - Diversity of Physical Hardware:

## - Replicas should be placed in distinct architectures

| | | |
|---|---|---|
| ARCH != “some-arch” | REQUIREMENT | ✓ |

## - Replicas should use different processors

| | | |
|---|---|---|
| MODELNAME != “some-processor-model” | REQUIREMENT | ✓ |
| “CPUSPEED” | RANK | ✓ |

## - Replicas should use different hardware components

| | | |
|---|---|---|
| TPM != “some-tpm-implementation” * | REQUIREMENT | ✗ |
| NETWORK_CARD != “some-network-card” | REQUIREMENT | ✗ |
| VIDEO_CARD != “some-video-card” | REQUIREMENT | ✗ |

* Trusted Platform Module: Atmel, Broadcom, Infineon, Intel, Sinosun, STMicroeletronics, Winbond, Toshiba, etc.

# 4.1 - Diversity of Supporting Software:

**- Replicas should be placed in different hypervisors**

| | | |
|---|---|---|
| HYPERVISOR != “some-hypervisor” | REQUIREMENT |  |

**- Replicas must use different Wormholes**

| | | |
|---|---|---|
| WORMHOLE != “some-wormhole” | REQUIREMENT |  |

**- Replicas should use different Operating System**

| | | |
|---|---|---|
| Service Notion or Service Descripton Language | PLUG-IN |  |

**- Replicas should use different Server implementation**

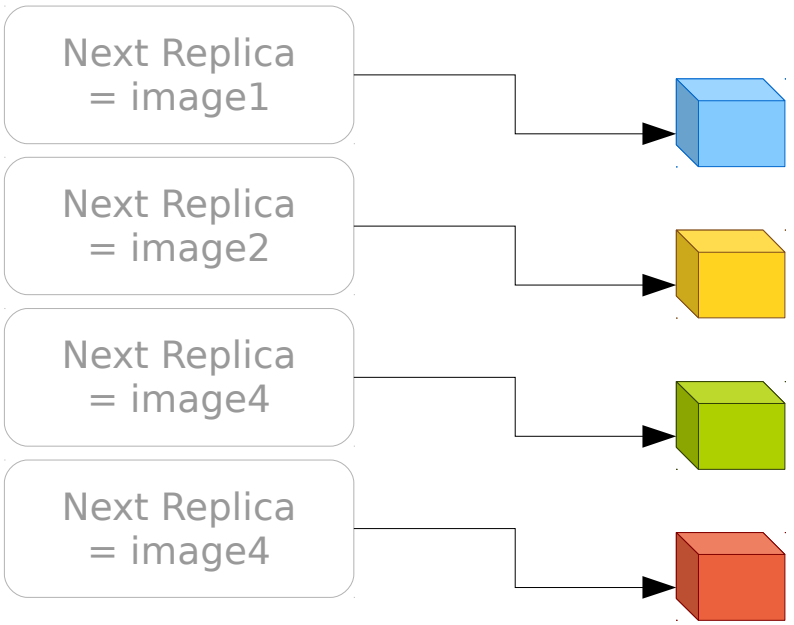
| | | |
|---|---|---|
| Service Notion or Service Descripton Language | PLUG-IN |  |

# 4.1 - Diversity of Application:

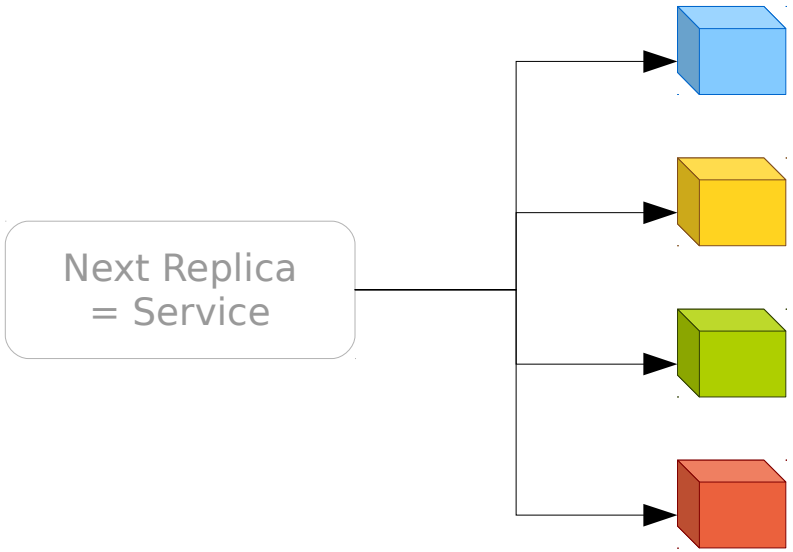
**- Replicas should use different application versions/implementations**

| Service Notion or Service Descripton Language | PLUG-IN |
|---|---|

**The service notion consists in deploy any service's replica instead of a specific image, on each replica replacement**




Without Service Notion




With Service Notion

# 4.1 - Diversity of Location:

**- Recoveries should change replica location**

| | | |
|---|---|---|
| “DISTANCE to +XX.XXXXXX -YY.YYYYYY” | RANK | |
| “- DISTANCE to +XX.XXXXXX -YY.YYYYYY” | RANK | |
| COUNTRY != “some-country” | REQUIREMENT | |

**- Service replicas should not be illegal in the location or should be allocated under a determined legislation (reputation or legal issues)**

| | | |
|---|---|---|
| CITY != “some-city” | REQUIREMENT | |
| STATE != “some-state” | REQUIREMENT | |
| COUNTRY != “some-country” | REQUIREMENT | |
| CONTINENT != “some-continent” | REQUIREMENT | |
| GROUP != “some-group” | REQUIREMENT | |

# 4.2 - Proactive Recovery Requirements:

**- Replicas must be added and removed in a controlled way**


**- Haizea is a batch scheduler that provides 3 types of lease**

| | |
|---|---|
| START_TIME="+00:00:30” | Advanced Reservation |
| START_TIME=”2008-11-04 11:00:00” | Advanced Reservation |
| START_TIME=”best_effort” | Best-effort Provisioning |
| START_TIME=”now” | Immediate Provisioning |

**- The important metrics are:**

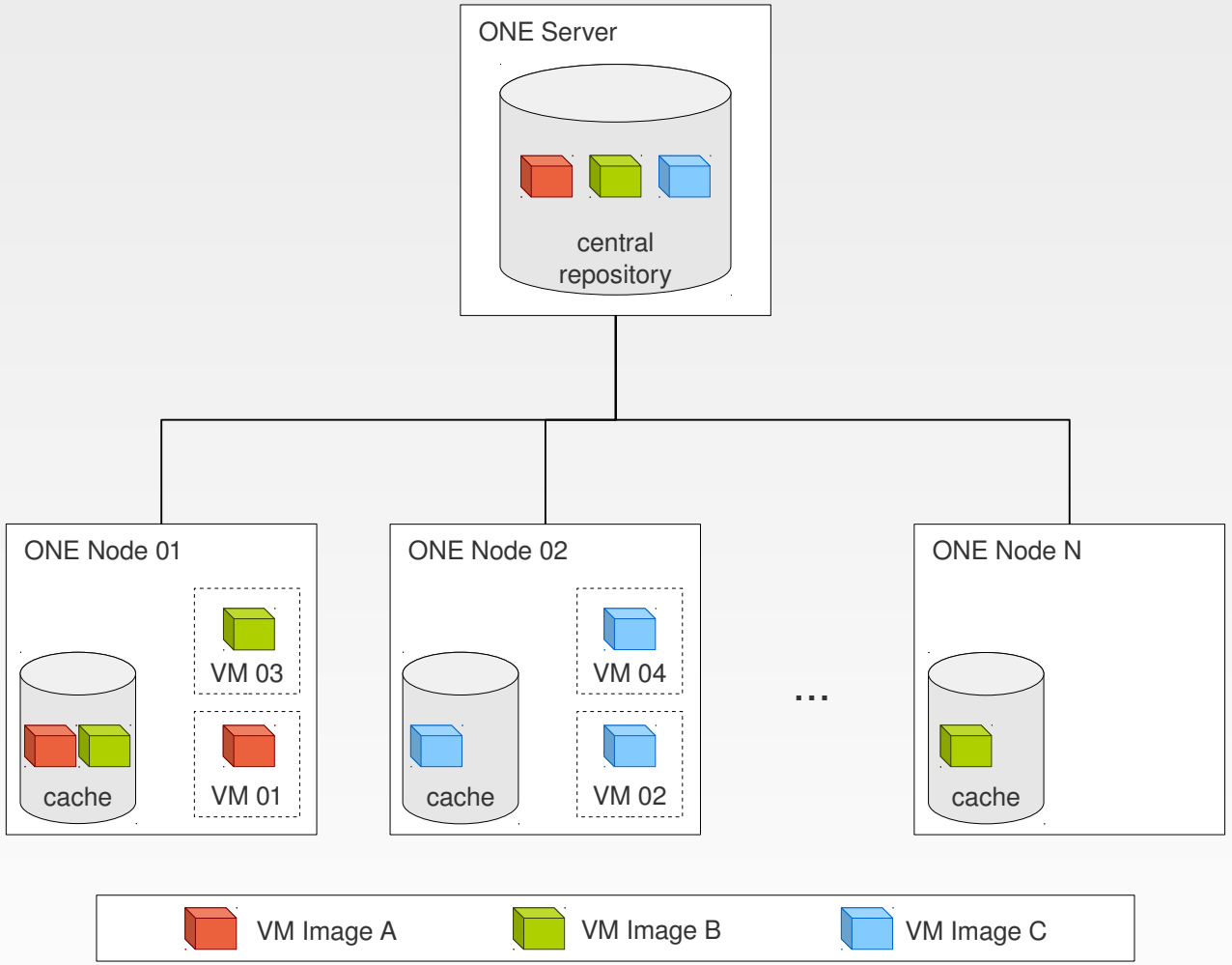
| | |
|---|---|
| START_TIME=”2008-11-04 11:00:00” | ✓ |
| END_TIME=”2008-11-04 12:00:00” | ✓ |
| DURATION=”00:00:30” | ✓ |
| PREEMPTIBLE=”yes” | ✓ |
| PRIORITY=”10” | ✗ |

**- The scheduling can not block the resource allocation properties**

# 4.3 - State Machine Replication Requirements:

**- Replicas could reuse the VM image from a local repository**

| Image Cache on Physical Hosts | PLUG-IN |
|---|---|



# 5 - Future Work:

- Delineating the influence of each metric in scheduling policies

- Specifying which metrics will be implemented

- Implementing and testing

- Organizing the metrics in two-level of scheduling: Cloud-of-clouds and Single Cloud

# 6 - Open Questions:

## - How independent are the faults in diversities cited here?

For diversity in Software of Support (**Operating Systems**) there is a new paper:

Miguel Garcia, Alysson Bessani, Ilir Gashi, Nuno Neves, and Rafael Obelheiro.
OS Diversity for Intrusion Tolerance: Myth or Reality?
In Proc. of the DSN'11: International Conference on Dependable Systems and Networks, Hong Kong, China, July 2011.

## - How guarantee that a Virtual Machine will be ready in an Advanced Reservation Lease for proactive recovery?

A cache for images, placed in hosts, helps but doesn't solve everything.


http://cloudfit.di.fc.ul.pt/

OpenNebula.org

The Open Source Toolkit for Cloud Computing

http://opennebula.org

TClouds

http://www.tclouds-project.eu/