



Roadmap for the DNS Load Balancing Service at CERN

Kristian Kouros

on behalf of the
DNS Load Balancing Team

HEPiX Autumn 2020 Workshop



Topics

1 INTRO

2 IMPLEMENTATION

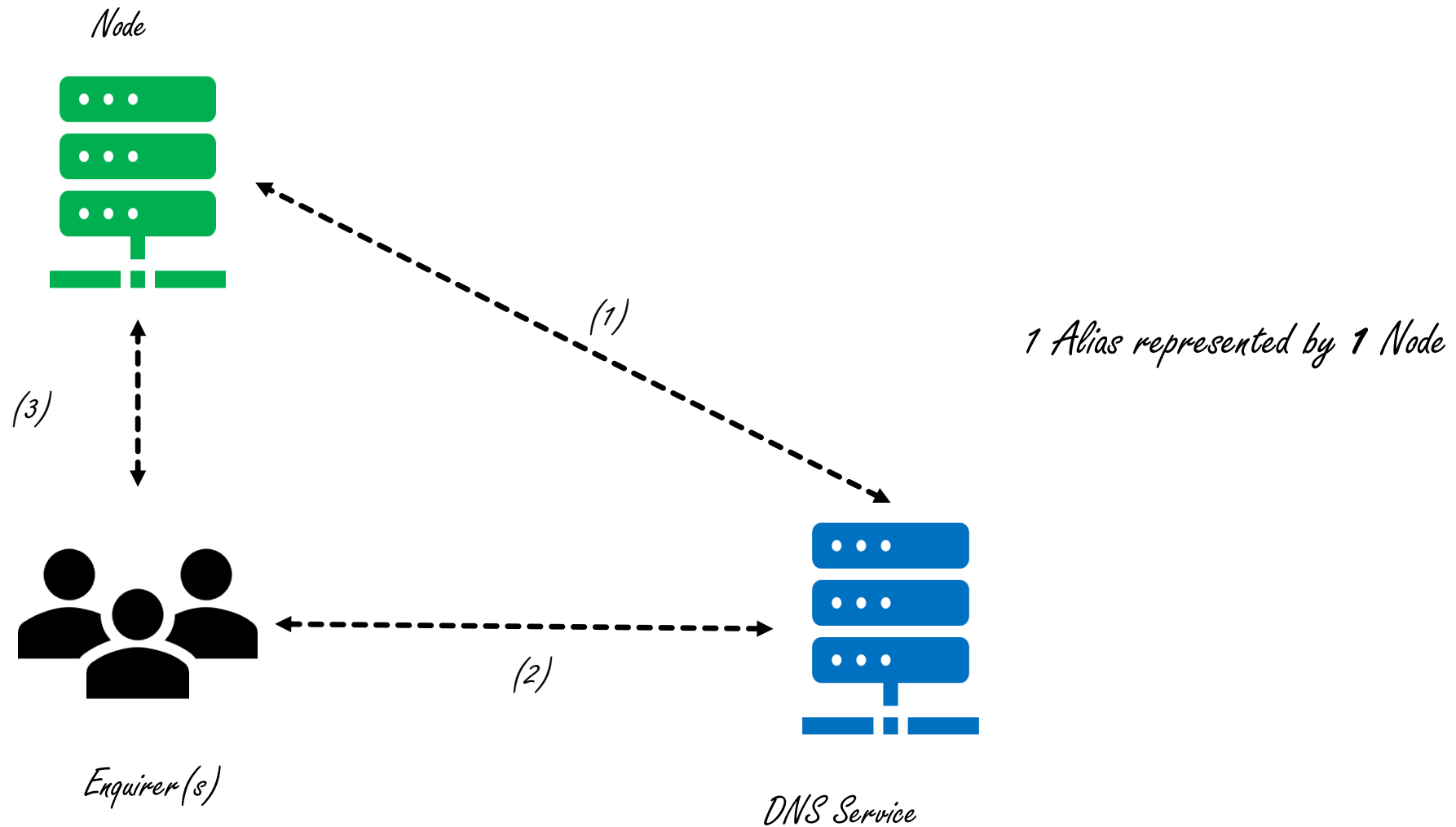
3 UPGRADES

Intro

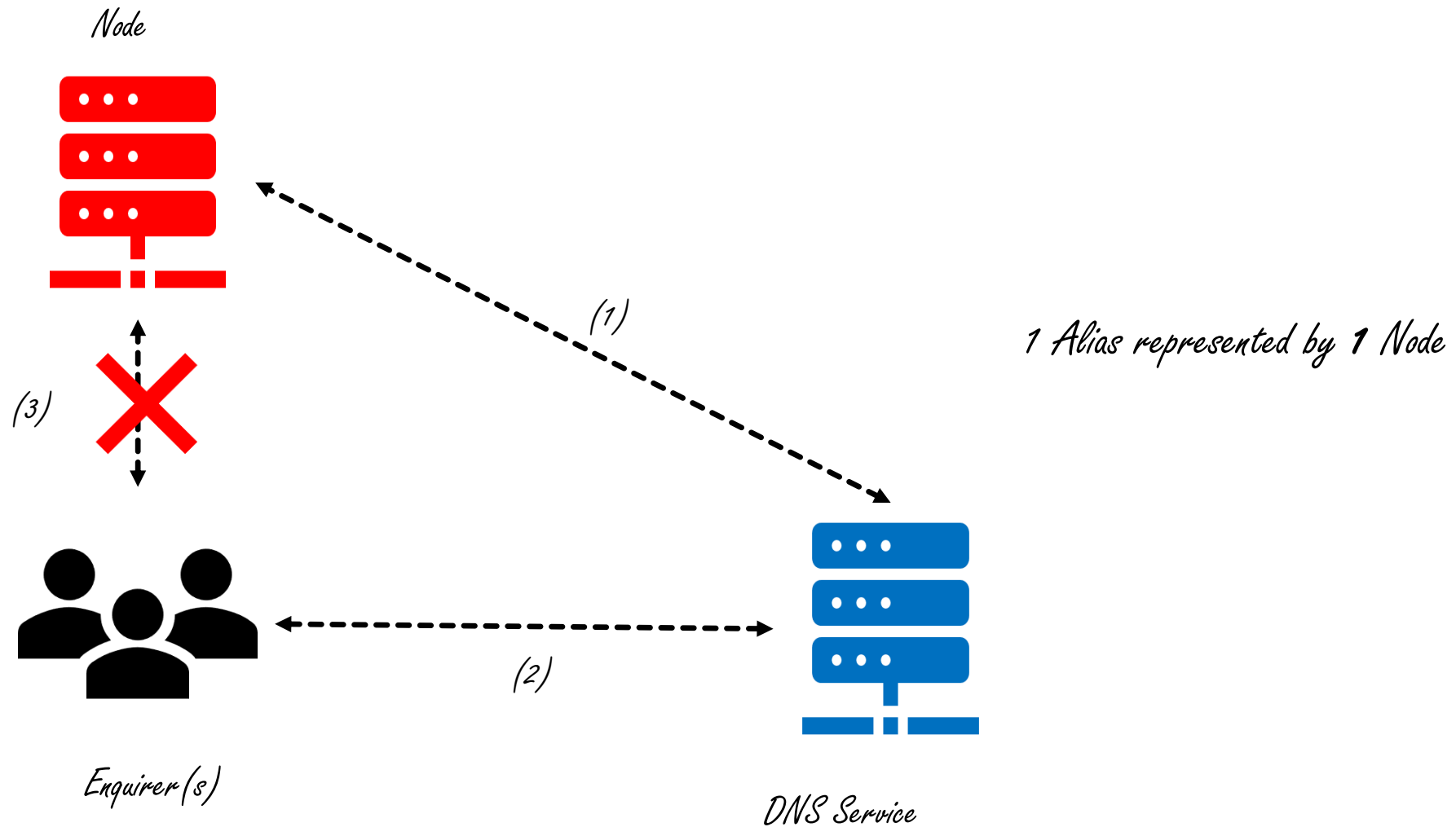
“There is nothing certain, but the uncertain.”

-Proverb-

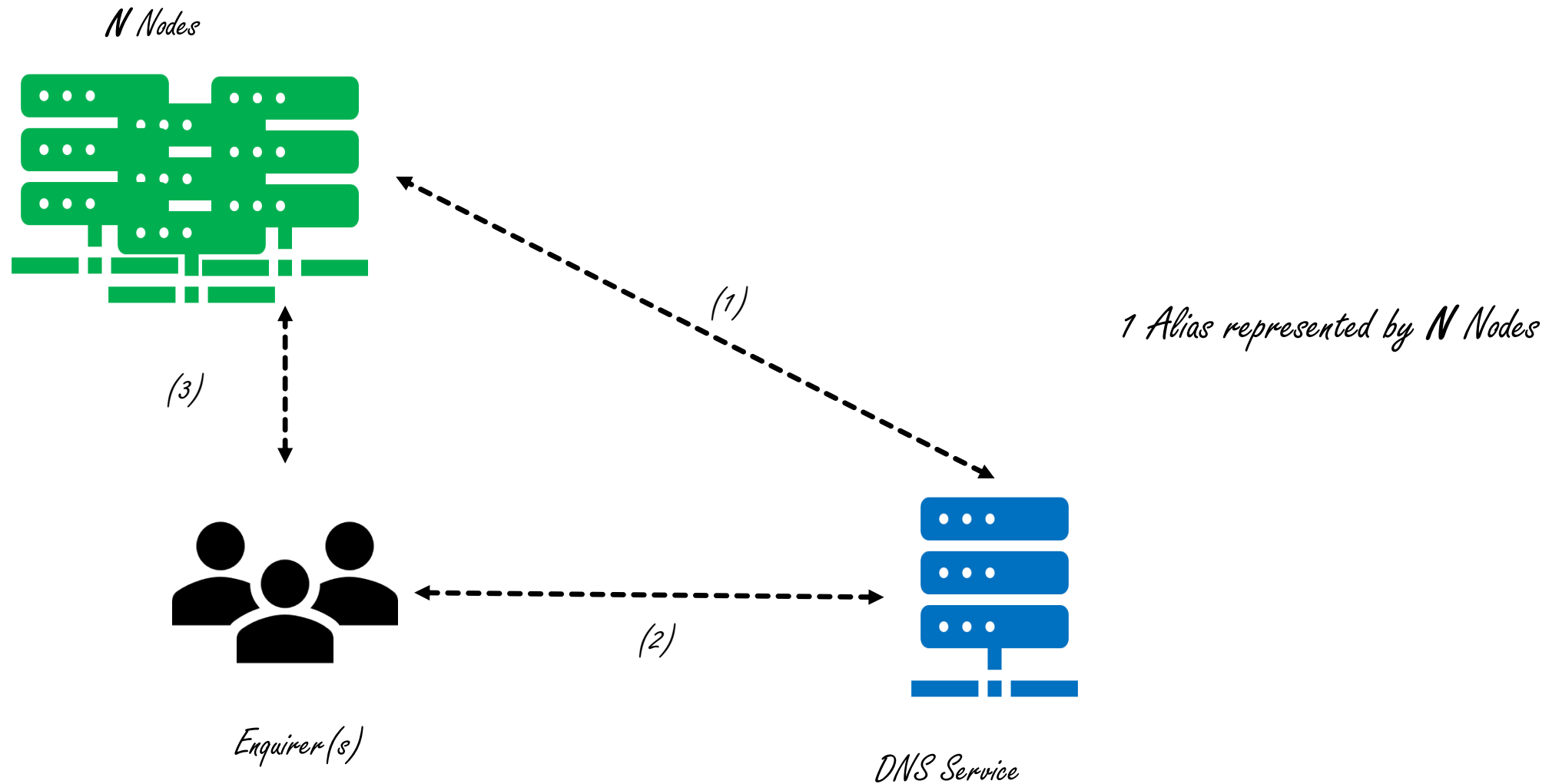
DNS LB 101



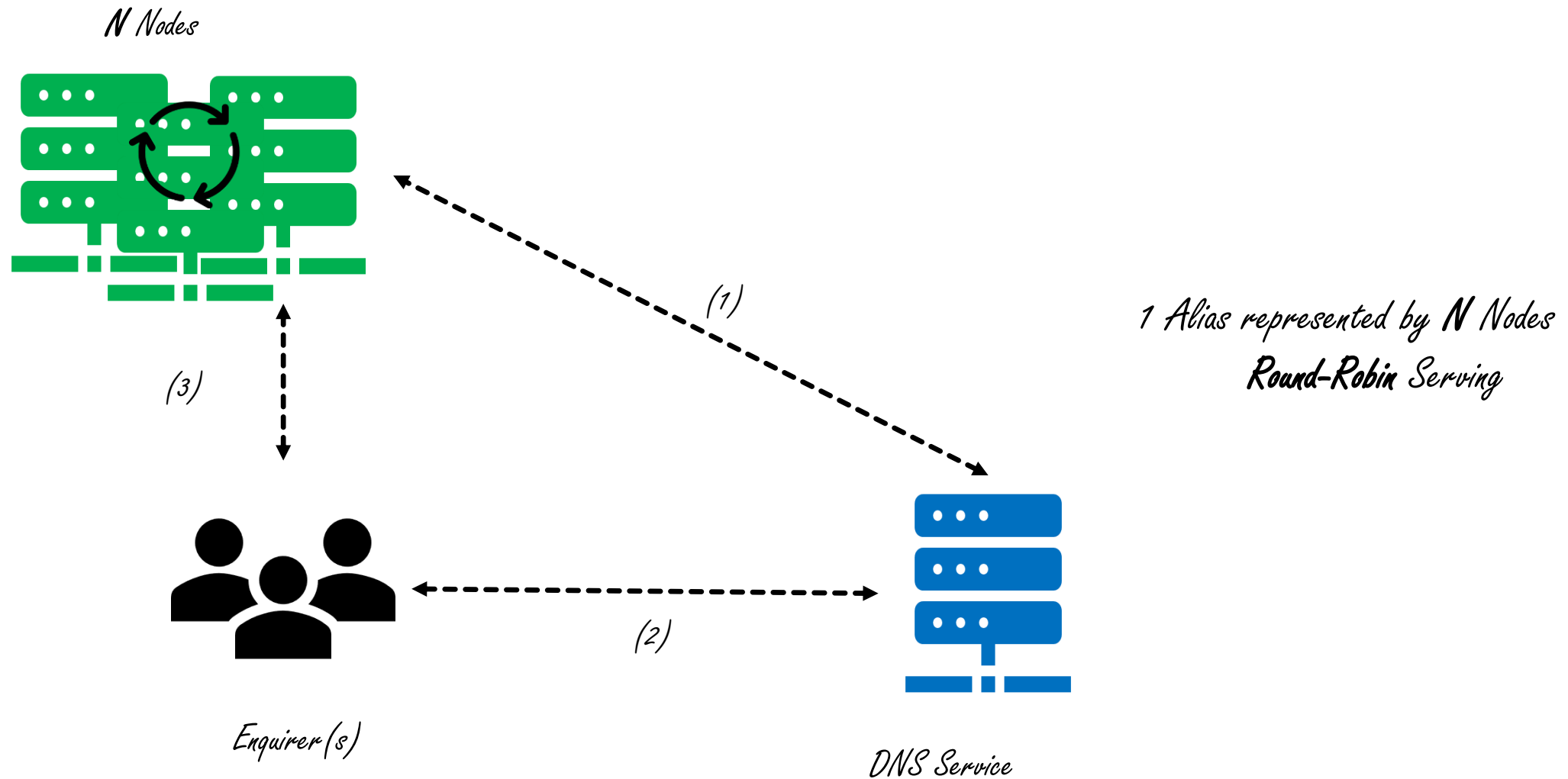
DNS LB 101



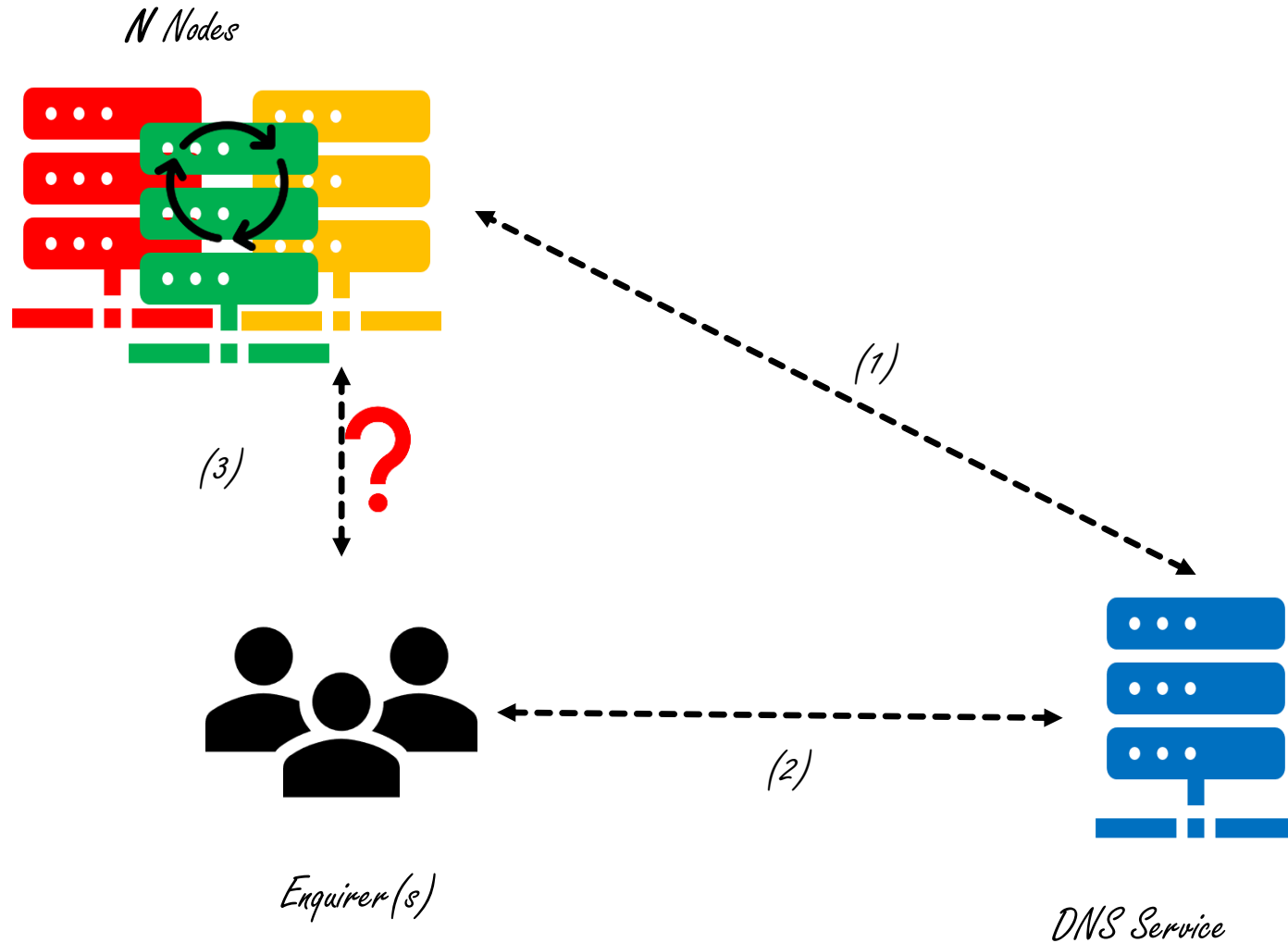
DNS LB 101



DNS LB 101



DNS LB 101

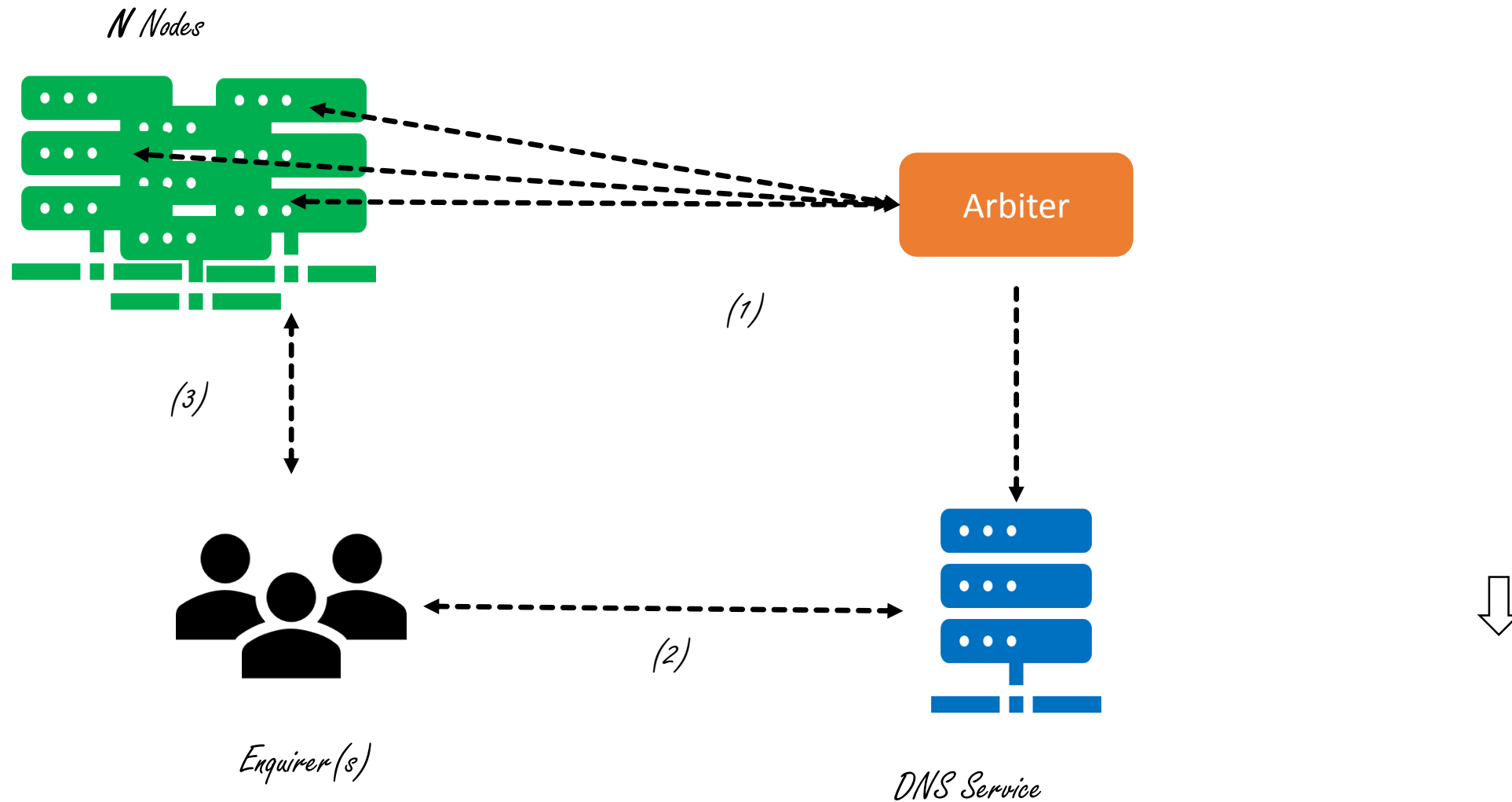


*1 Alias represented by N Nodes
Round-Robin Serving*

Agnostic of Node's Wellbeing

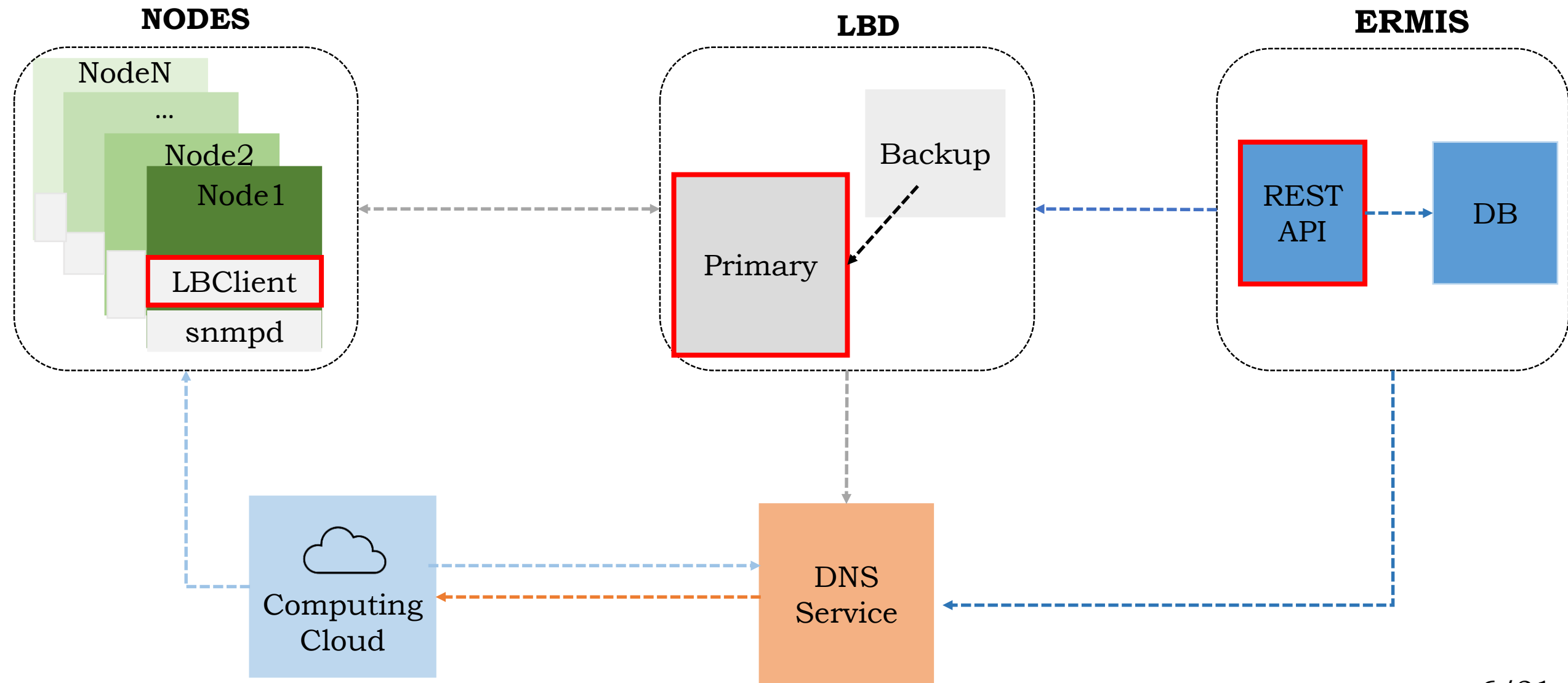
↓
DoS for 1 in N user

DNS LB 101



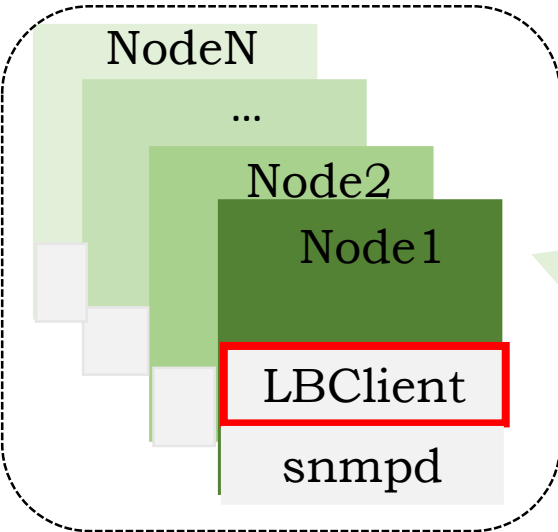
Our Implementation

System architecture



LBCClient

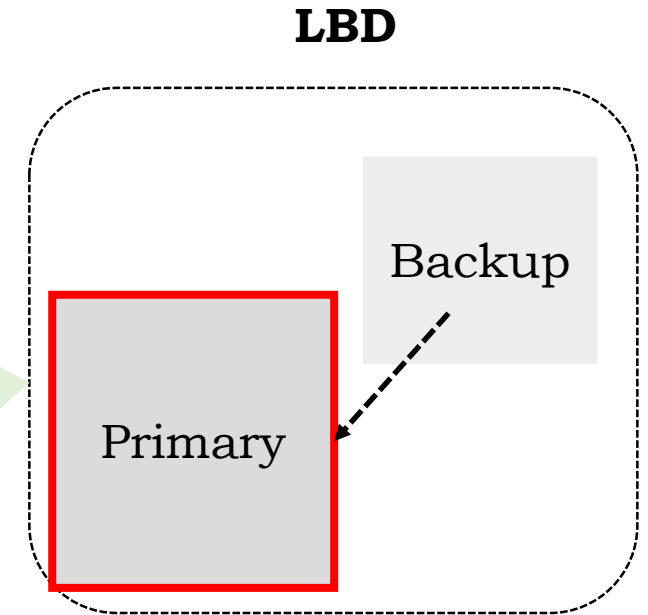
NODES



- Runs on **every** node that is behind an alias
- Metrics:
 - **Checks** -> tmpfull, nologin , daemon , etc
 - **Load** -> collectd, lemon, const
- Summarizes evaluation into an **int value**
- Best value: **The lowest positive**

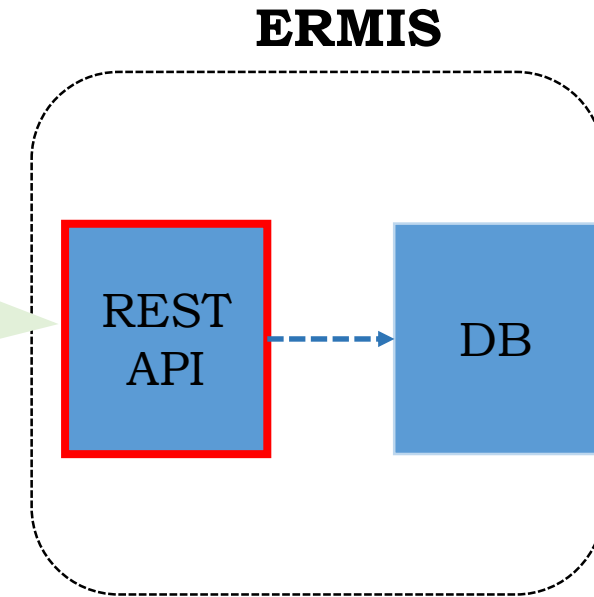
LB Daemon

- **Selects best nodes** per alias
- Backup performs **same tasks**
- **Checks Primary's heartbeat:**
If dead → take its place

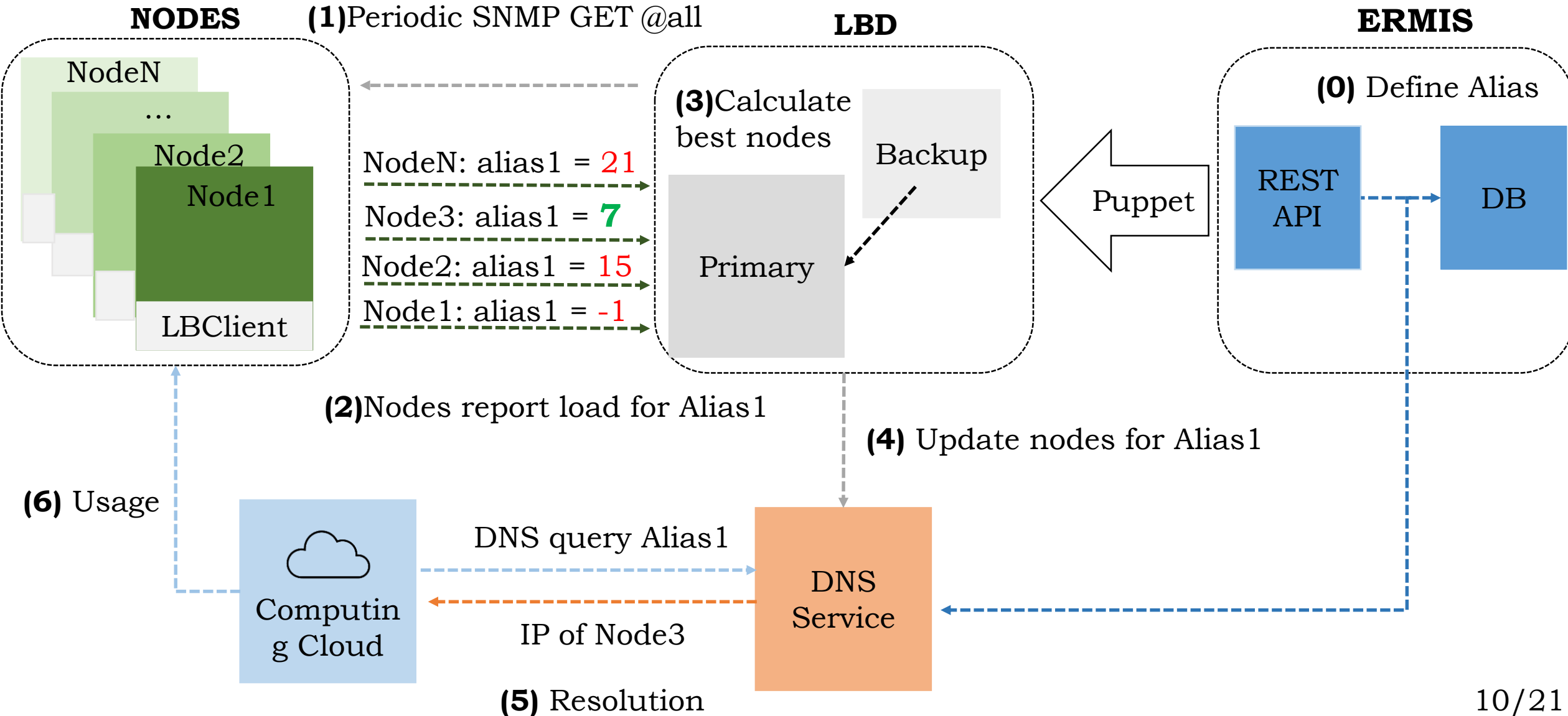


REST Service

- **Self-manage** aliases
- **UI & CLI** paired with **MySQL**
- What users can **CRUD** :
 - Alias
 - Hostgroup,
 - Visibility,
 - Number of best nodes,
 - Cnames,
 - Nodes
 - Etc.



Workflow



Upgrades

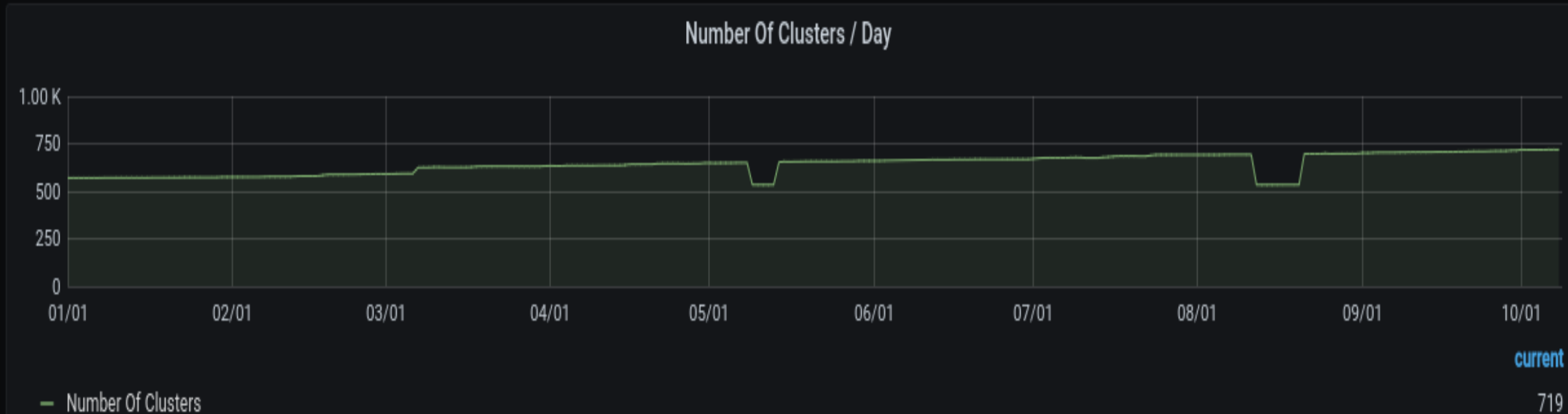
Let's rewind a bit

1. Stable service for +10 years
2. Each component in different language :
 - LBD → Perl
 - LBClient → C
 - Ermis → Python
3. Sequential processes
4. Number of aliases in increase
Example: From 2017 to 2019 ~50% increase

No. of aliases in 2020

☰ KPIs & SLIs / Services / Infrastructure / KPIs ☆ 🔗

▾ Load Balancing



The Issue:

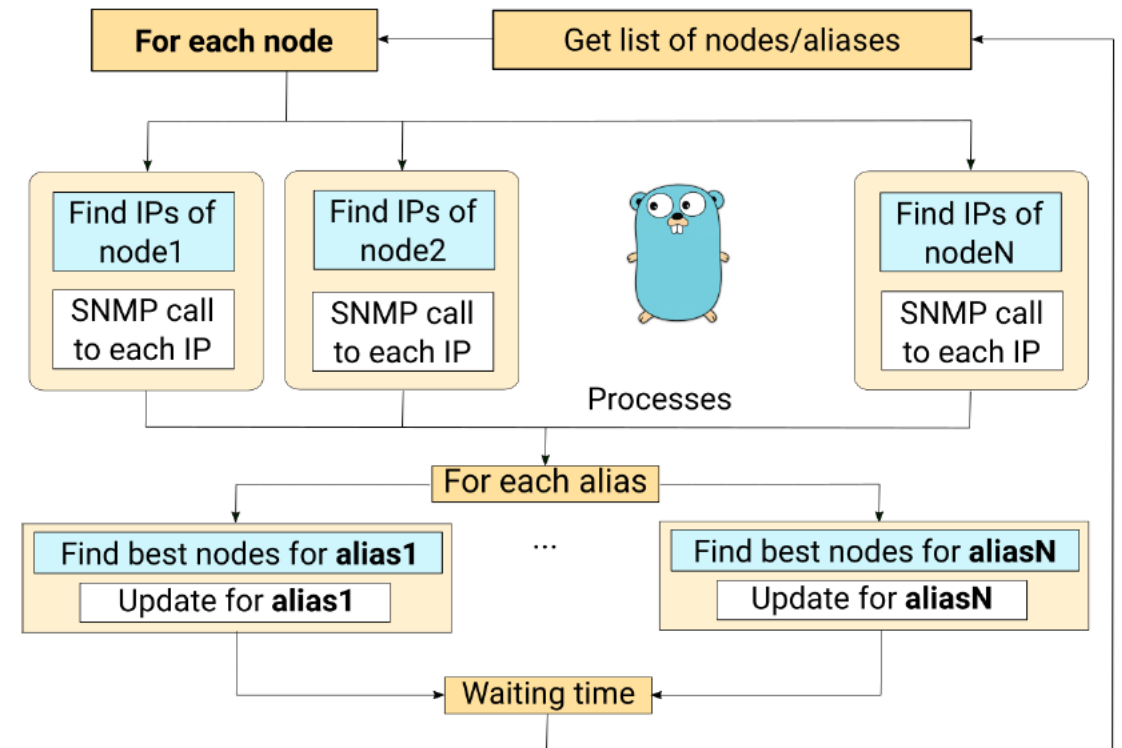
Sequential processes were limiting performance and alias capacity

The Solution:

1. Reimplement LBD and LBClient in Golang
 - Aliases evaluated in parallel
 - Nodes checked only once
2. Partition aliases & LBD/partition

The Result:

SNMP Querying time reduced from 5 minutes to less than 1 minute



Upgrade #2 : Node Control

The Issue:

There was no way to overwrite the nodes that LBD was receiving from PuppetDB.

The solution:

Introduce a new configuration element in Ermis UI & CLI for the user to define blacklisted and/or whitelisted nodes.



Upgrade #3 : Security

production

The Issue:

- Shibboleth SSO
- No 2FA

The Solution:

- Replace Shibboleth with OpenIDC
- Protect SSH connections with 2FA



Upgrade #4 : Ermis REST Back-end

development

The Issue:

- The Ermis REST API back-end is still in Django/Tastypie, different than the rest of the components
- High abstraction and undefined data types

The Solution:

- Reimplemented in Golang + Echo Framework + GORM

The result:

The Pros/Cons of Golang over Python. Performance not evaluated yet.

The Issue:

- Ermis REST Service, LBD Primary & Backup run on VMs
- Service availability not ideal

The Solution:

- Deploy in a Kubernetes cluster

Upgrade #6 : Node Alarms

The Issue:

- Users are not aware if the number of defined nodes per alias is maintained

The Solution:

- Introduce new feature for setting an alarm if number of nodes crosses the threshold



Evaluation: Push vs Pull

To do

The Issue:

LBD, even though in parallelization, it has to Pull data from a lot of nodes.

Alternative Scenario:

Evaluate if it would be better for the Nodes to Push instead.

Feedback or suggestions are welcomed

Summary

- Overview of CERN's open-source load-balancer for DNS (but not only!)
- Presentation of the undertaken upgrades to meet the new demands of the service
- Project : <https://github.com/cernops/golbclient>
<https://github.com/cernops/golbd>

A blue icon representing a group of three people.

lb-experts-public@cern.ch



kristian.kouros@cern.ch

Thank You !