Notifications workflows using the CERN IT central messaging infrastructure

ZHECHKA TOTEVA (CERN/IT), DARKO LUKIC (UNIV. OF NOVI SAD), LIONEL CONS (CERN/IT)

CHEP2018 – 10/07/2018

Outline

- Motivation for the CERNMegabus project
  - Roger and different scenarios to propagate Roger state change
  - Need for improvements
  - CERNMegabus architecture and project goals
- CERNMegabus use cases and project evolution
  - EOS and CASTOR
  - DNS Load Balancing
  - CERN Computer Centre power cut management
- Future plans

Background

FOREMAN

HAPROXY

puppet

CASTOR

Roger

43 000
Puppet managed machines

RabbitMQ
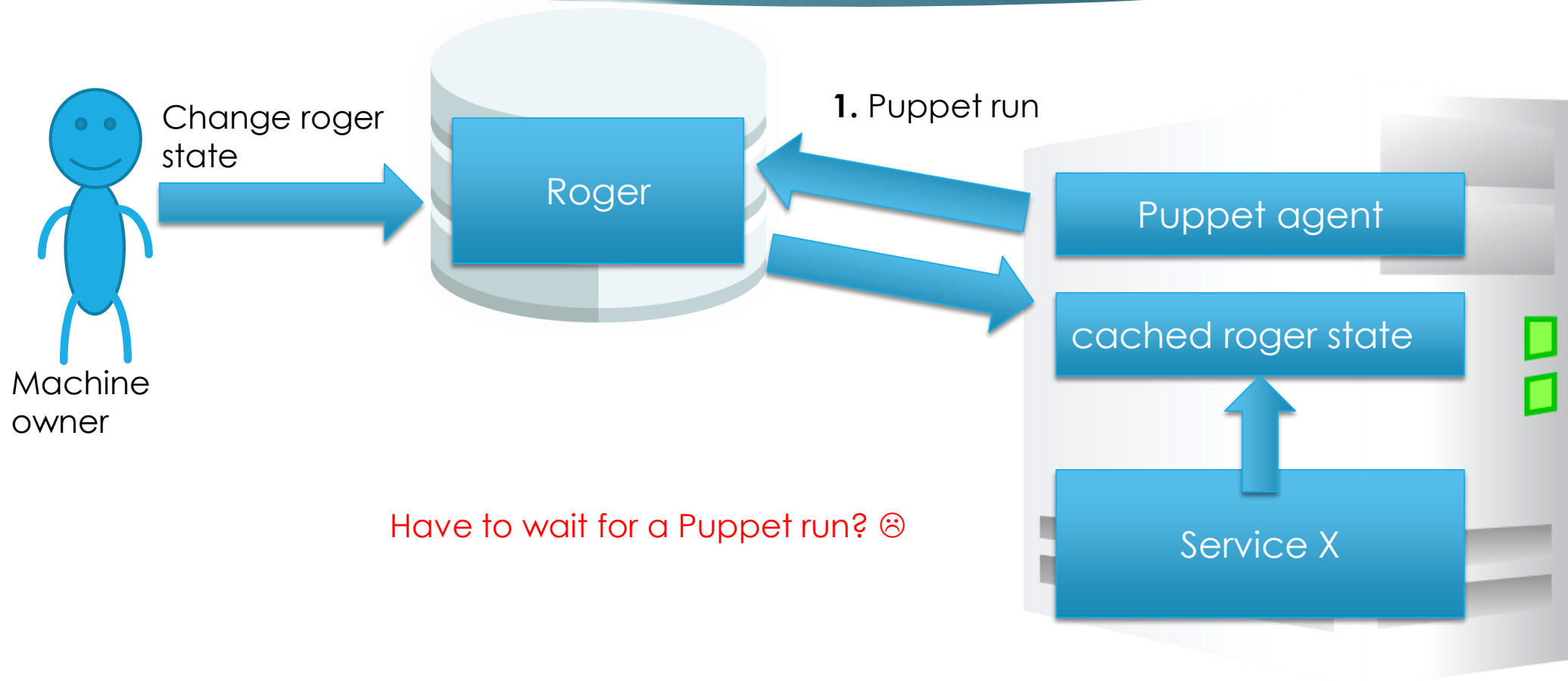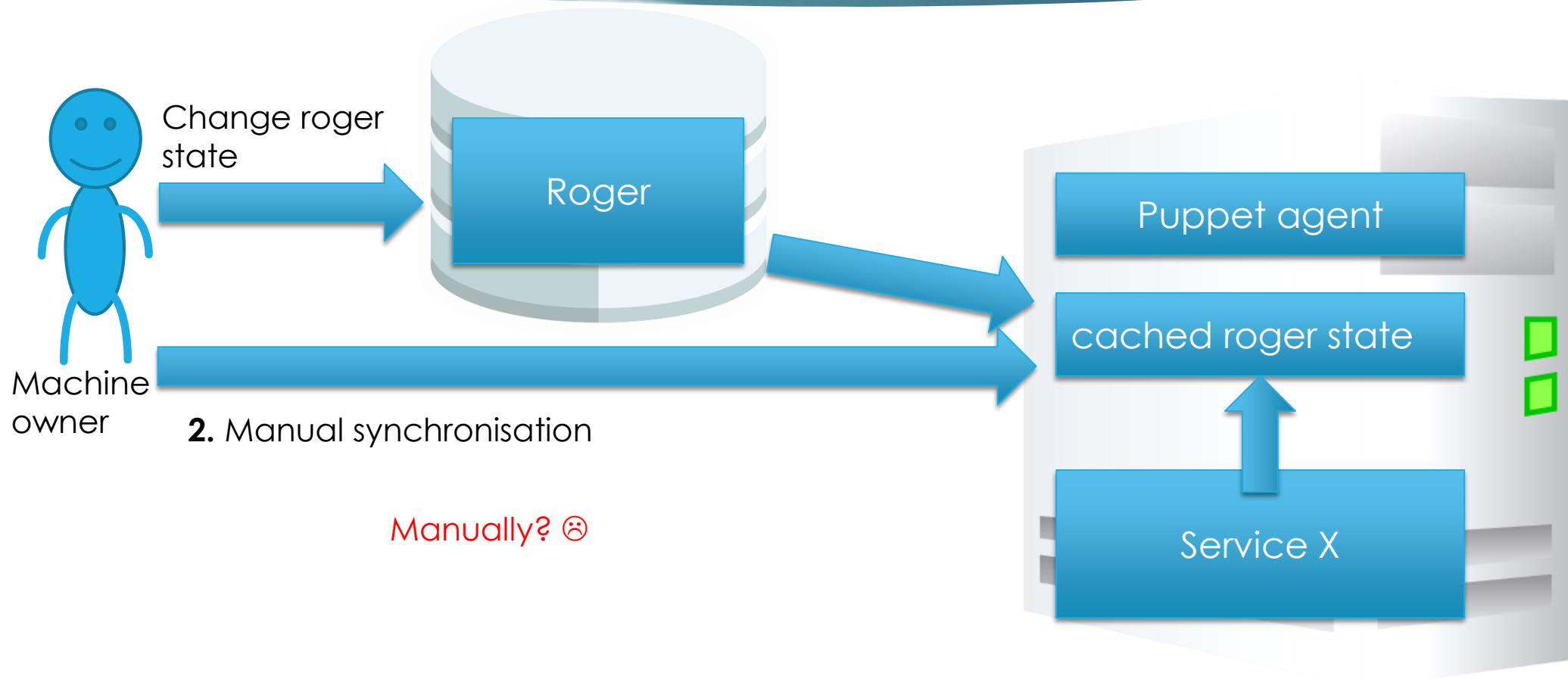
ActiveMQ

EOS

DNS Load Balancing

Roger

Roger

▶ Is a tool that stores and manages the overall application state of a machine

▶ Example states: production, disabled, intervention, draining

```
-bash-4.2$ ai-dump `hostname`
+---------------+-------------------------------------------------------+
|      Hostname: | aiadm72.cern.ch                                      |
|      Hardware: | virtual, 2 cores, 3.37 GiB memory, - swap, 2 disks   |
|     Hostgroup: | aiadm/nodes/login                                    |
|       Comment: | aiadm machines are for administrating nodes in the computer |
|               | centre                                               |
|   Environment: | qa                                                   |
|   Responsible: | ai-config-team@cern.ch                              |
| FE Responsible: | Configuration Management                            |
|            OS: | CentOS 7.5.1804 x86_64 (3.10.0-862.2.3.el7.x86_64)  |
|    VM Project: | IT Configuration Management Services                |
|    VM Flavour: | m2.medium                                           |
|    Avail zone: | cern-geneva-c                                       |
|     LANDBsets: | it_cc_lxadm_with_ssh                                |
|    LB aliases: | aiadm7.cern.ch, aiadm7-testing.cern.ch, aiadm.cern.ch, … |
|  CNAME aliases: | -                                                   |
|          IPv4: | 137.xxx.xxx.xxx (ITS) (S513-C-VM32)                 |
|          IPv6: | 2001:1xxx:xxx:xx::xxx (ITS) (S513-C-VM32)           |
|     App state: | production                                          |
|    Alarm mask: | Hardware(N) OS(N) App(N) NoContact(N)               |
|               |                                                     |
|   Last report: | 4 minutes ago                                       |
+---------------+-------------------------------------------------------+
```
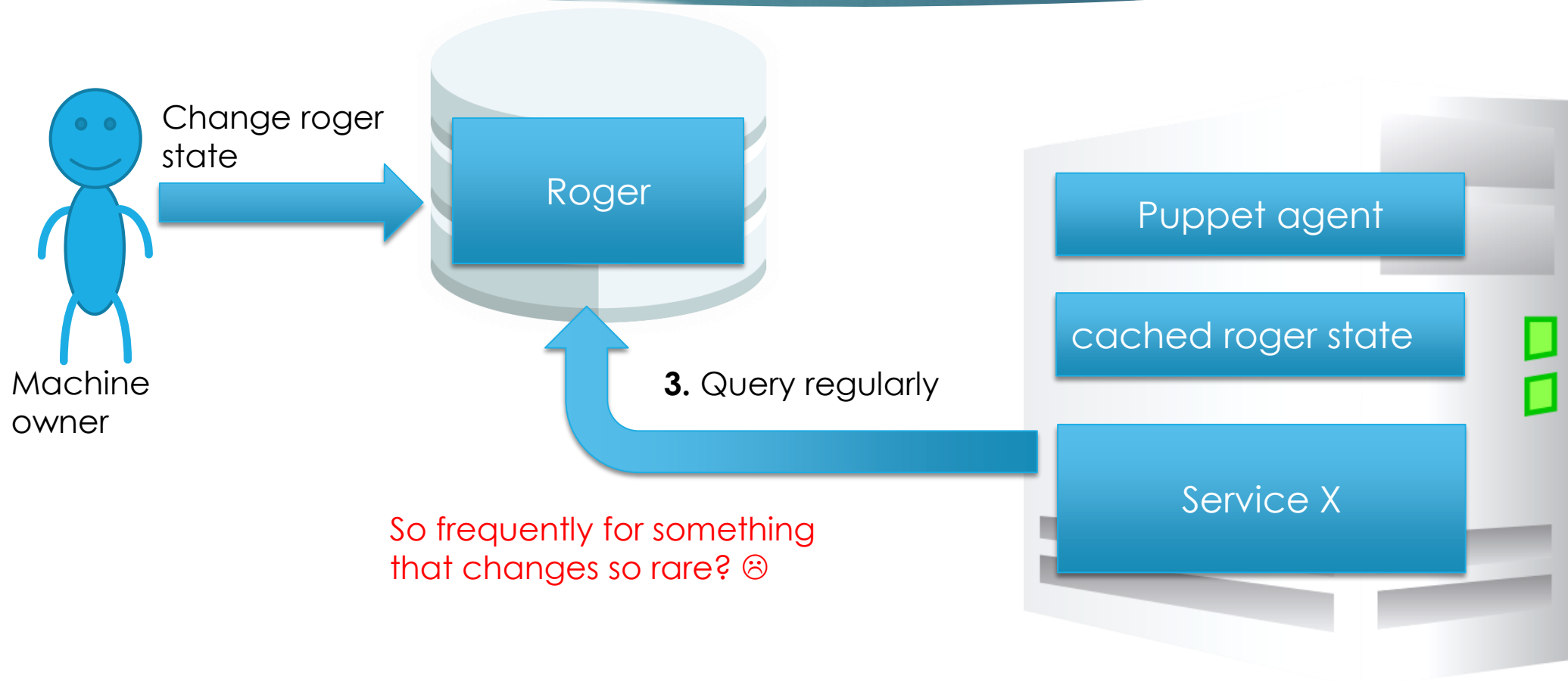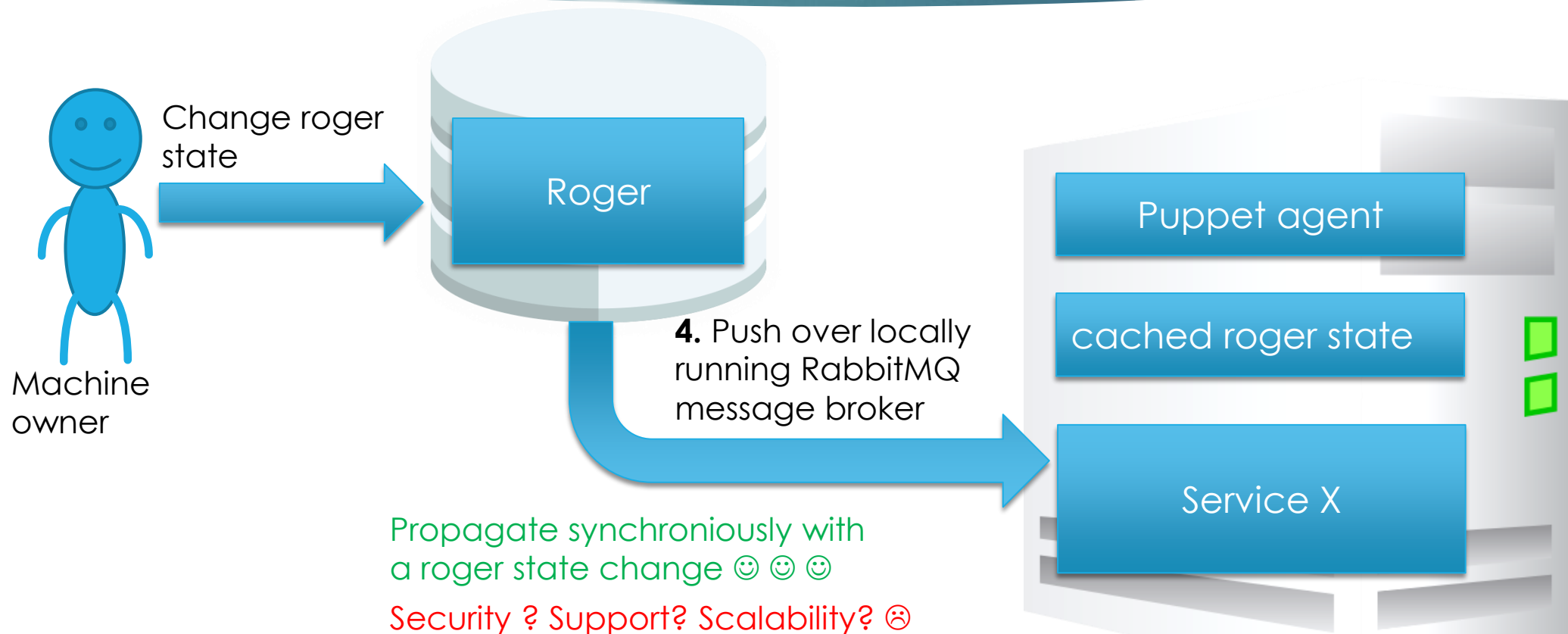
Roger state and Puppet

Machine owner

Change roger state

Roger

1. Puppet run

Puppet agent

cached roger state

Service X

Have to wait for a Puppet run? ☹

Roger state and manual synchronisation

Change roger state

Roger

Machine owner

2. Manual synchronisation

Manually? ☹

Puppet agent

cached roger state

Service X

Roger state and querying Roger server

Change roger state

Machine owner

Roger

Puppet agent

cached roger state

3. Query regularly

Service X

So frequently for something that changes so rare? ☹

Roger state and RabbitMQ

Change roger state

Machine owner

Roger

4. Push over locally running RabbitMQ message broker

Propagate synchroniously with a roger state change ☺ ☺ ☺

Security ? Support? Scalability? ☹

Puppet agent

cached roger state

Service X

Need for improvements

- **Speed up** the propagation of a **roger state change** on the machine
- **Without** unnecessarily **querying** the **roger server**
- **Without** extra lines of **private code**
- By using **supported, large-scale proven, messaging infrastructure** with a flexible **authentication and authorisation schema**
  - **Central IT ActiveMQ message brokers**

Solution: CERNMegabus architecture

Change service "A"

2. Publish message →

Central CERN IT ActiveMQ brokers

1. Subscribe for message ←

3. New message →

Affected service "B"

4. Execute reactive actions

External program
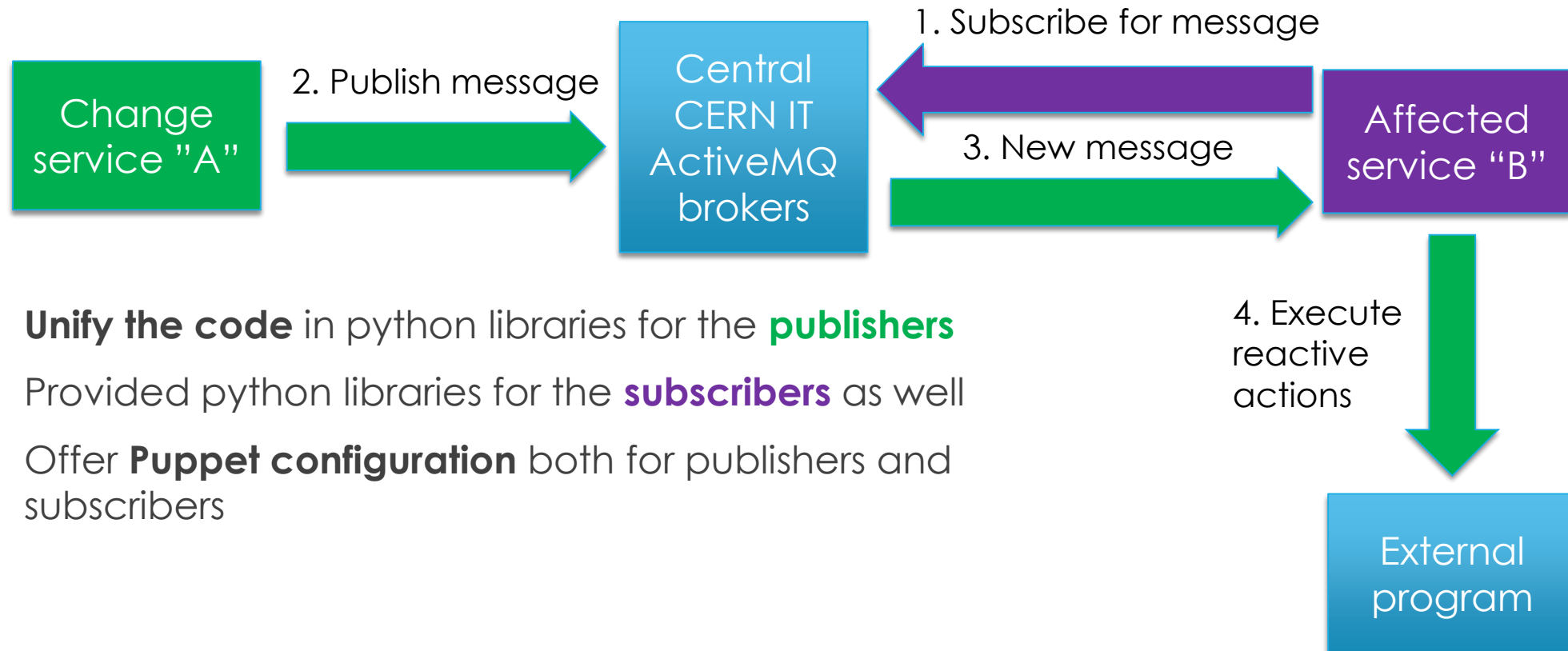
- ▶ **Unify the code** in python libraries for the **publishers**
- ▶ Provided python libraries for the **subscribers** as well
- ▶ Offer **Puppet configuration** both for publishers and subscribers

CERNMegabus use cases and project evolution

- Read/write vs ReadOnly mode of **CASTOR tapes** and **EOS disks**
- Presence in a **DNS Load Balancing(LB)** alias
- **CERN Computer centre (CC) power cut** management
- **Alarms** handled by the CERN IT **monitoring infrastructure**

Start small: EOS and CASTOR use cases

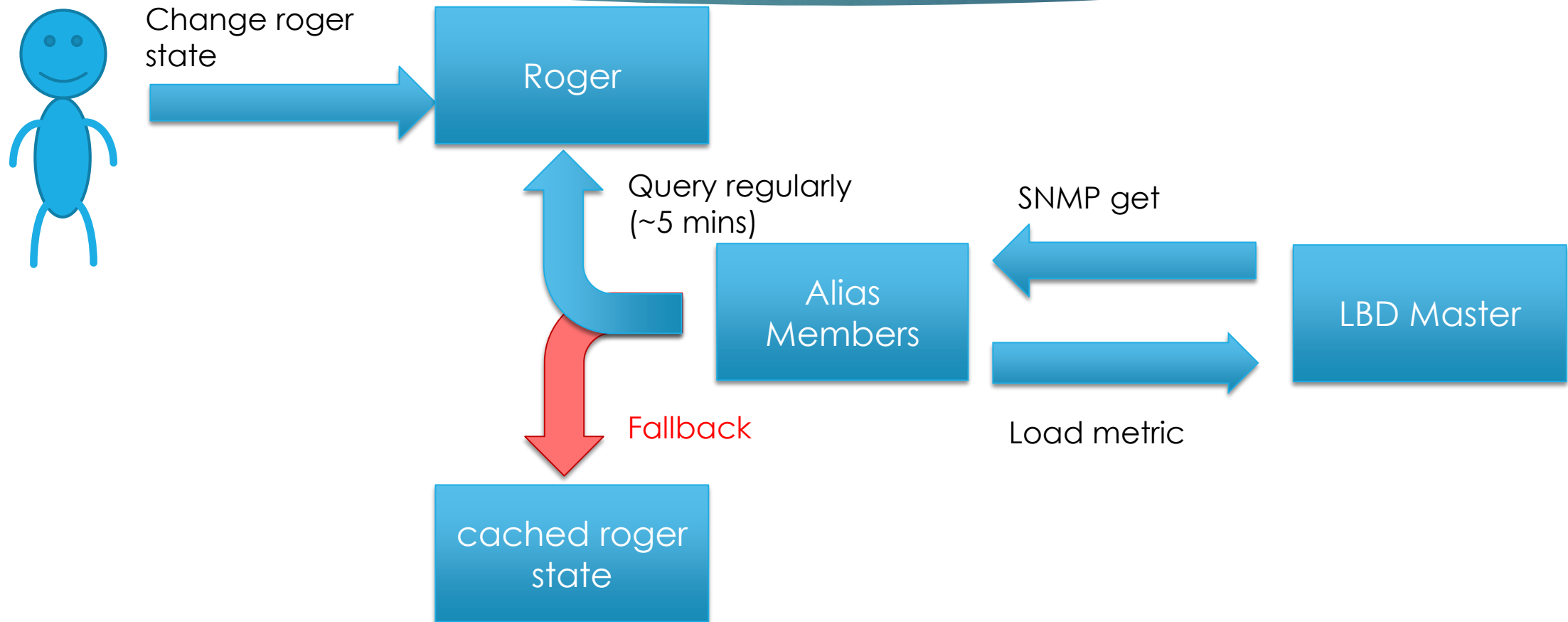**Replaced** local RabbitMQ with central IT ActiveMQ message brokers

- ▶ Organise **ActiveMQ topic**

- ▶ **Local user name-**based authentication schema

- ▶ Changed the python library – from PIKA to STOMP.PY

- ▶ Use Puppet to configure both consumer and publisher

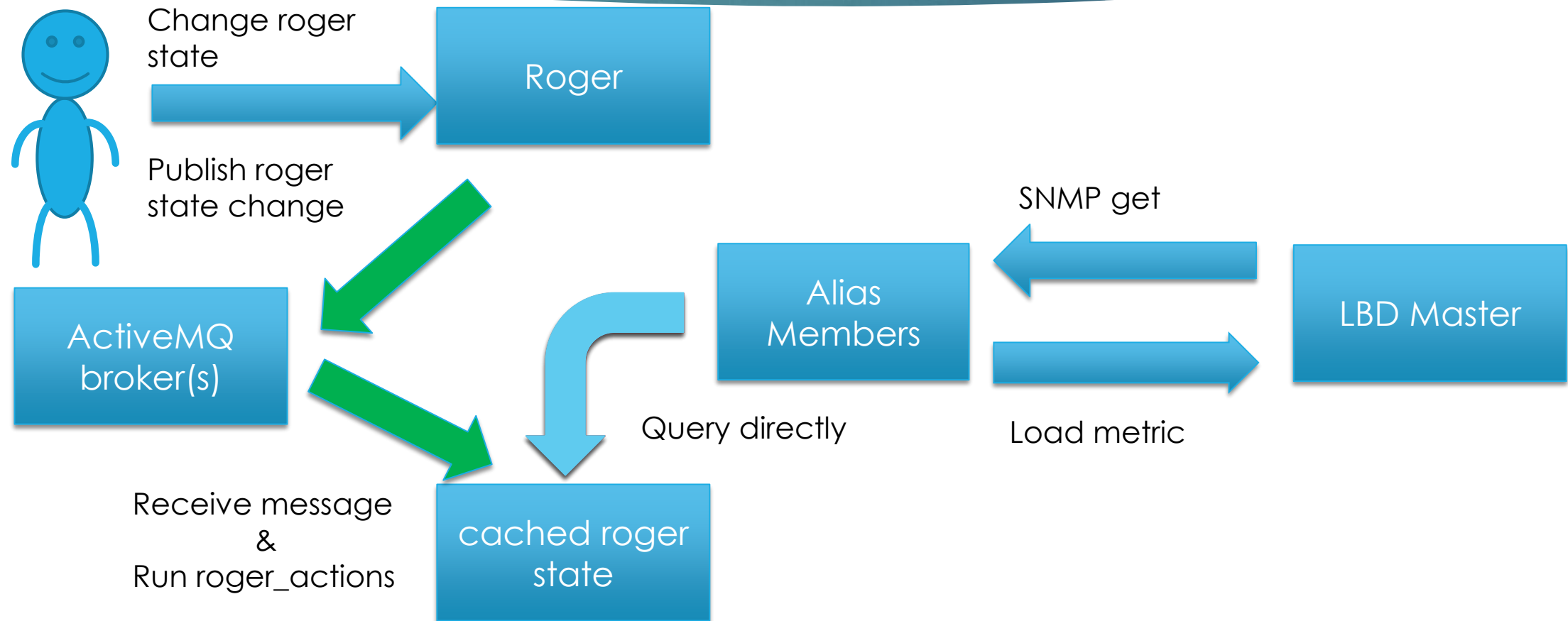EOS and CASTOR using ::cernmegabus::client::consumer (Dec 2017)

```
::cernmegabus::client::consumer {'eos':
    server              => 'agileinf-mb.cern.ch',
    port                => 61213,
    user                => 'rogercons',
    pass_key            => 'roger_key',
    destination         => 'roger.notification.hostgroup',
    destination_type    => 'topic',
    top_hostgroup       => 'eos',
    hostgroup_selector  => "eos/${instance_without_eos}/storage",
    host_selector       => '',
    owner               => 'root',
}
```

EOS

DNS LB: Before CERNMegabus

Change roger state → **Roger**

Query regularly (~5 mins)

Fallback

cached roger state

Alias Members

SNMP get

Load metric

LBD Master

DNS LB: After CERNMegabus

Change roger state

Roger

Publish roger state change

SNMP get

ActiveMQ broker(s)

Alias Members

LBD Master

Query directly

Load metric

Receive message & Run roger_actions

cached roger state

More challenges: DNS LB use case

- **Orchestration issue**: more listeners than in the EOS/CASTOR use cases (2000 vs 20)
  - "**Publish to one and Listens to all**" vs "**Publish to all and Listens to one**"
  - Offer both orchestration models to publishers and consumers
- Use **stompclt** to configure the consumer
  - listen to ActiveMQ message broker for a roger state change
  - update the **cached roger state**
- Use **Puppet** to configure the **stompclt** configuration file
- Use the existing **roger_action** script when a message is received
  - A positive side effect: trigger **the custom defined roger actions**
- **Use certificates for authentication**

CERN CC Power cut management

- Challenges
  - **20 minutes** on UPS
  - Specific format of the **UPS monitoring data**
- Requirements
  - Formalise **complex algorithms** for decision making **if there is a power cut**
  - **Propagate** the power cut event **to all** machines in the CC
  - Handle the event depending on a **predefined recipe**

UPS monitoring data

Decision making algorithm
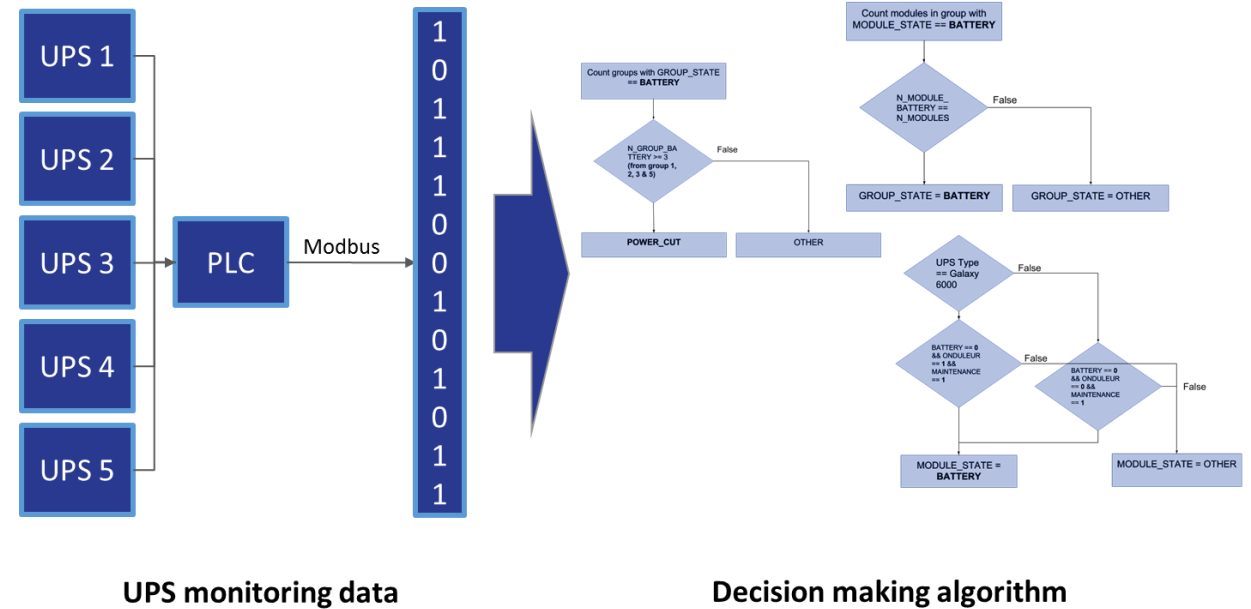
CERN CC Power cut tests
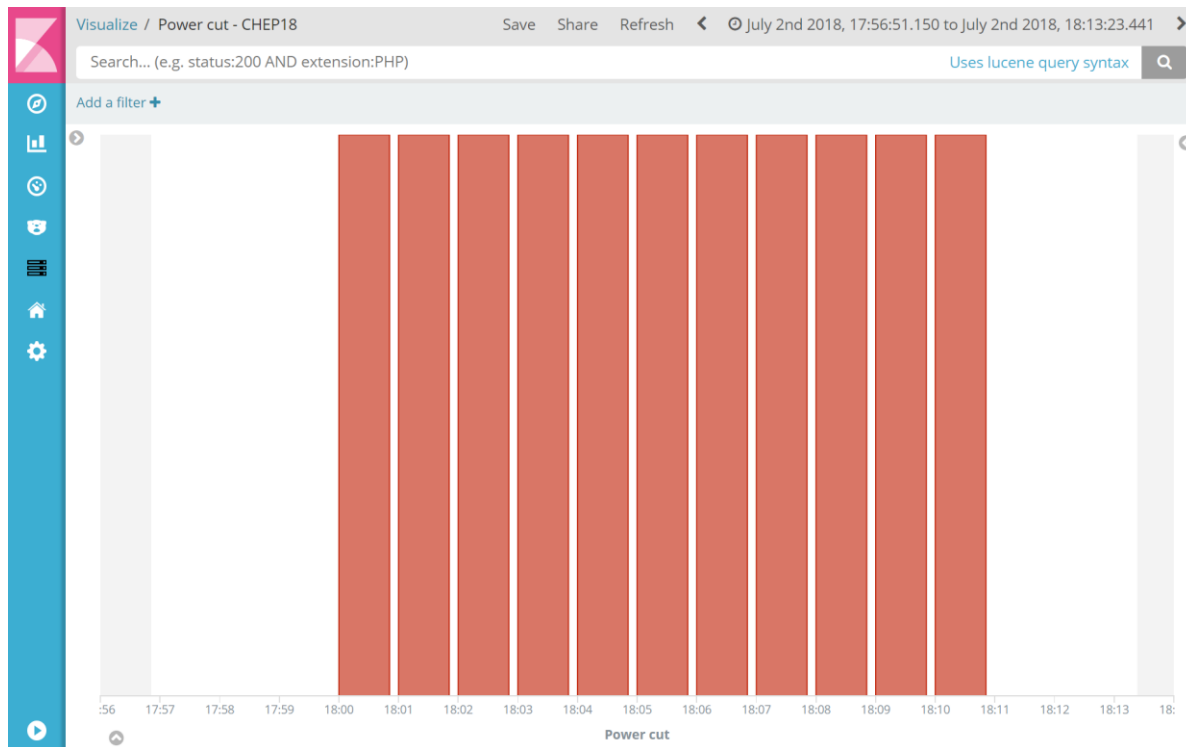
- During mid-annual power cut test on the 2nd of July, 2018
  - Detected power cut
  - Notified the subscribed machines
  - Shutdown the machines, which had been predefined to be shutdown
  - Detected the power back
  - Notified the machines, which had been predefined to wait

Future plans

▶ **Install CERNMegabus client on all machines in the CERN CC**

▶ **Release CERN CC Power Cut management in production**

▶ Use DNS LB client with roger state criterion

▶ Configure stompclt configuration files with CERNMegabus Stompclt module

▶ **Assist colleagues to define and realise their CERNMegabus use cases**

Thank you!

Questions?

See you at the poster sessions at 16:30

Securing and sharing Elasticsearch resources with ReadonlyREST

Concurrent Adaptative Load Balancing at (@CERN)

Thanks

**THANKS our customers: CASTOR, EOS, DNS LB,
THANKS our collaborators: CERN/IT-CF, CERN/EN-EL
THANKS all my colleagues from IT-CM**

CERNMegabus Puppet module – general use

▶ New Puppet resource **::cernmegabus::plugins::roger**

    ▶ CASTOR and EOS did not need hundreds of lines of private code

    ▶ On-boarded the **Puppet master HAproxy** configuration with CERNMegabus

▶ New predefined plugin to **update the cached d roger state**

```
include ::cernmegabus::plugins::teigiclt_roger_actions
```

    ▶ we satisfy the needs for alarms handling by the **monitoring infrastructure**

    ▶ The later will be included in base

▶ Provide easy way to re-write a **stompclt configuration file** with **Puppet**

```
::cernmegabus::plugins::roger { 'protect
tapes':
  on_change_param_name  => 'appstate',
  on_change_command         =>
'modifydiskserver -s $(echo ${NEW_APPSTATE}
| sed -e "s/quiesce/readonly/g")
${HOSTNAME}',
}
```

CASTOR

```
 ::cernmegabus::plugins::roger{'disable-aips-
via-roger':    on_change_param_name =>
'appstate',    on_change_param_from =>
'production',    filters            =>
{'hostgroup' =>
"punch/puppet/ps/v4/%/${::hostgroup_3}"},
on_change_command    => '/usr/bin/haproxyctl
disable all ${HOSTNAME}',  }
```

HAProxy

CERNMegabus Puppet module – computer center power cut management

▶ Already implemented the predefined Puppet standard client action

```
include ::cernmegabus::plugins::ccpco
```

    ▶ Decided to be "send an email" (and/not sink to disk) during the test phase

▶ Possibility to use a predefined action

```
class{ 'cernmegabus::plugins::ccpco':
    standard_action      => 'shutdown',
}
```

▶ Or even custom action (both on power loss and on power restore events)

```
class{ 'cernmegabus::plugins::ccpco':
    custom_action         => '/bin/backupdata.sh',
    power_back_action => '/bin/restoredata.sh',
}
```

Authentication and Authorisation challenges

- Roger notification and the CC power cut management use messages with **public content**

- BUT the **publishers must be verified**

- Discussing if the CC power cut management needs also to exchange **signed messages** for extra validation

- Unknown need of the future use cases