Infrastructure as Code & Monitoring at scale

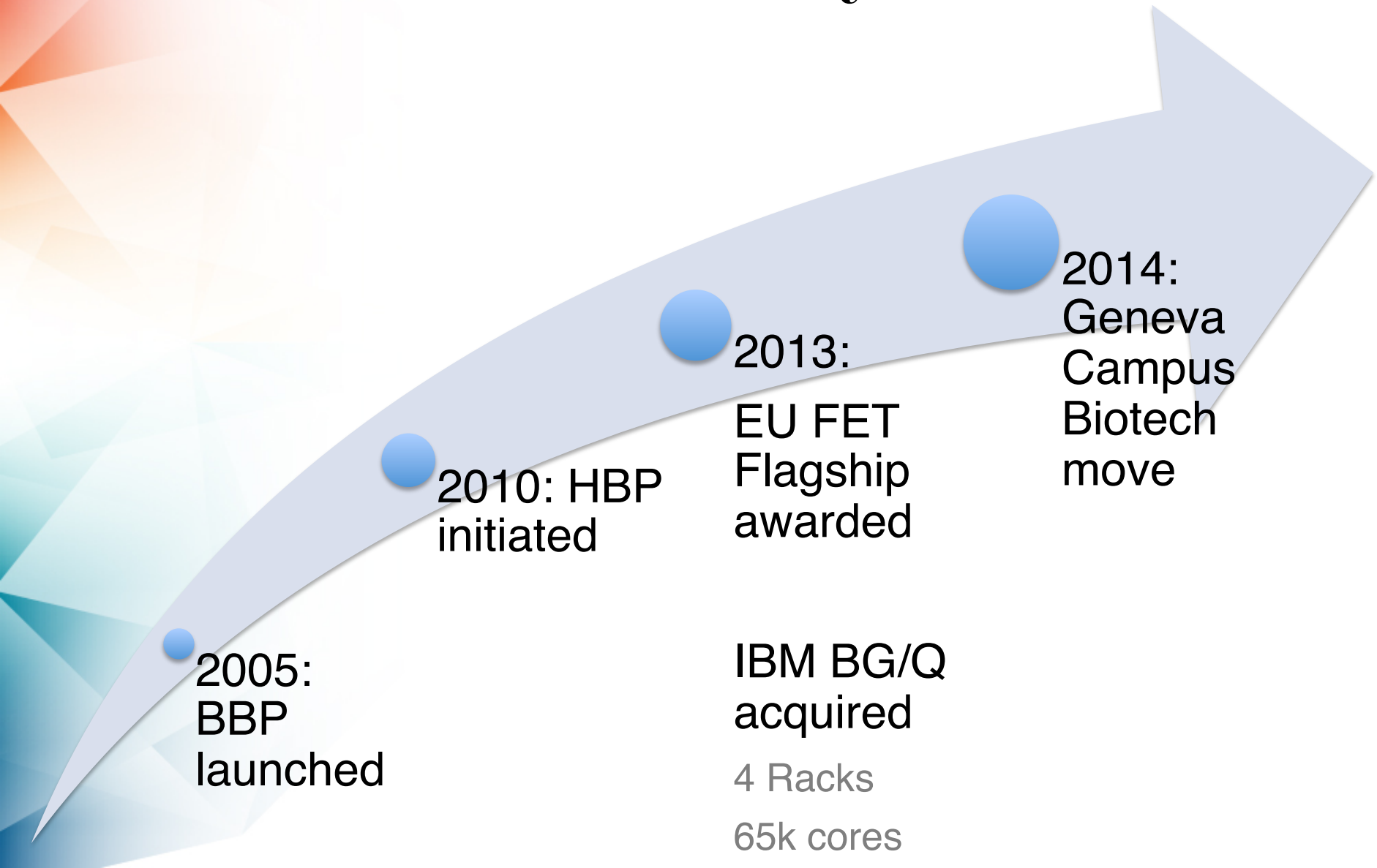
Alexandre Beche
EPFL / BBP

<alexandre.beche@epfl.ch>

Outlines

- Project overview
- Core services team
- Infrastructure as code
- Monitoring at scale

History


2005:
BBP
launched
2010: HBP
initiated
2013:
EU FET
Flagship
awarded
IBM BG/Q
acquired
4 Racks
65k cores
2014:
Geneva
Campus
Biotech
move

BBP / HBP

Blue Brain Project

- Swiss federally funded project
- Unit within EPFL
- Over 100 employees
- Working on Campus Biotech (GVA)

BBP / HBP

Blue Brain Project

- Swiss federally funded project
- Unit within EPFL
- Over 100 employees
- Working on Campus Biotech (GVA)

Human Brain Project

- 135 partners / 26 countries
- Funded under EU – FET grant
- Development of 6 ICT platforms
  - Neuroinformatics / Brain simulation / High Performance Computing / Medical informatics / Neuromorphic computing / Neurorobotics

Core Services team

- Currently 4 staff maintaining:
  - Infrastructure:
    - Storage: Netapp, CEPH, GSS/AFM
    - Compute: HPC clusters
    - Virtualization: ESX / OpenStack
  - Services:
    - Puppet configuration management
    - Gerrit code review workflow
    - Jenkins continuous integration
    - Monitoring framework

Mission Statement

"Build and operate scalable and elastic computing infrastructure and ensure the operation, integrity and longevity of the systems and the project"

Core Principles

- **Scalability**
  - hardware virtualisation
- **Elasticity**
  - extend into clouds
- **Longevity**
  - scriptable, versioned and documented infrastructure
  - no manual installations

OpenStack

Controller nodes (2 regions)
- HAProxy
- Clustered RabbitMQ
- Active/active DB
- Horizon / Keystone only on primary regions

Compute nodes (2 regions)
- KVM hypervisor
- Neutron OpenVSwitch
- VMs stored on CEPH

CEPH (2 regions)
- 36 OSDs (4 TB 7.2K rpm)
- Journals on SSD
- 130TB raw capacity

- Icehouse running on RHEL 6
- 100% Puppet managed
- 2 regions – Geneva / Lugano
- 5 tenants
- About 150 VMs

EPFL network constraints

- Network managed by EPFL
- DNS modification is a slow process
  - Better to be bulk operation

EPFL network constraints

- Network managed by EPFL
- DNS modification is a slow process
  - Better to be bulk operation

- Impact on our infrastructure:
  - Limited set of ip / hostname
  - Per-tenant network + 1 for user VMs
  - User cannot chose their ip / hostname
  - Single routable interface per VM

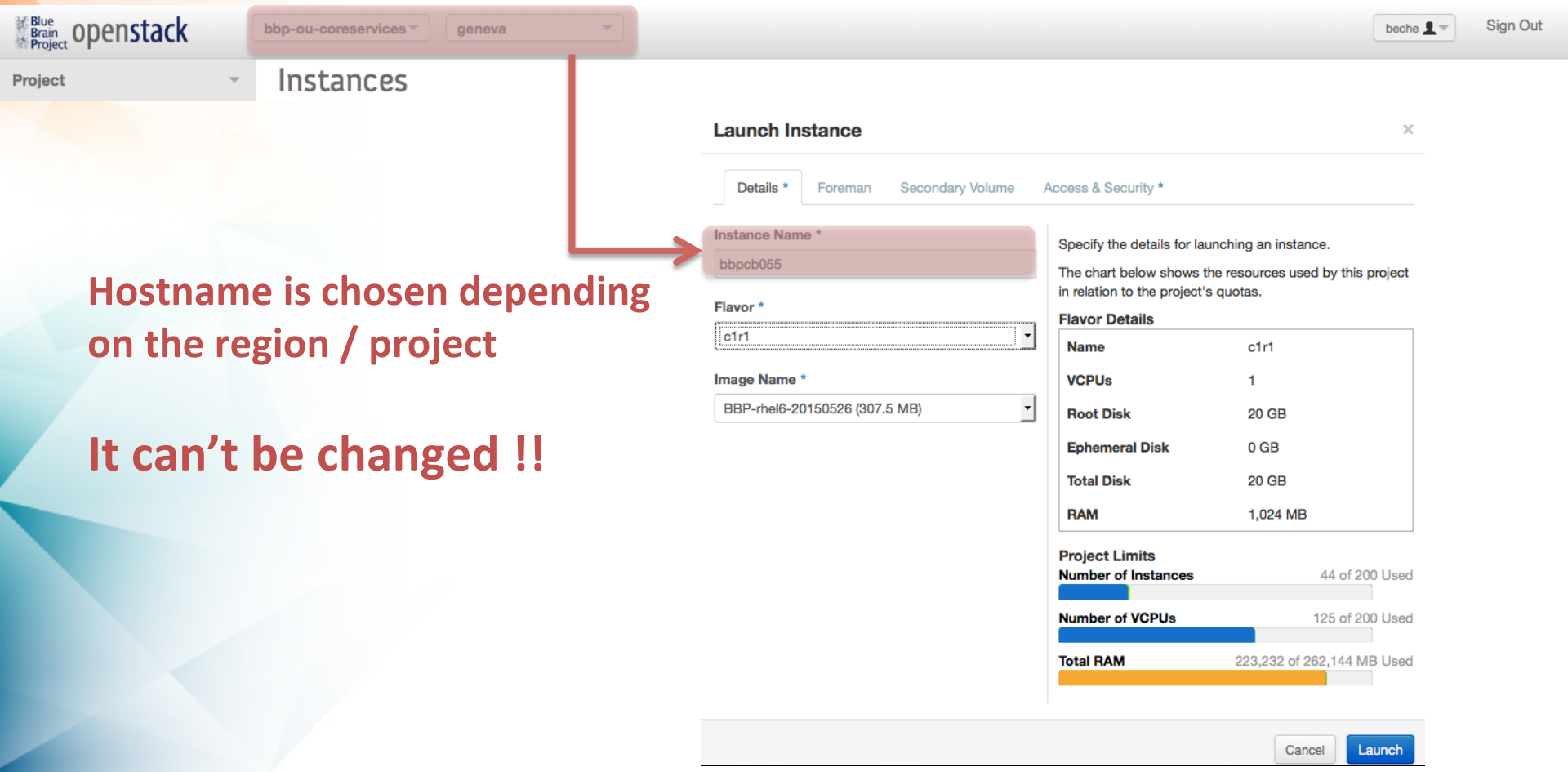
Auto naming algorithm

- Find the correct network:
  - From OS project → Find network UUID
  - From network UUID → Find CIDR

Auto naming algorithm

- Find the correct network:
  - From OS project → Find network UUID
  - From network UUID → Find CIDR
- Find an hostname that:
  - Has a record in EPFL DNS
  - Does not exist in OpenStack
  - Does not exist in Foreman

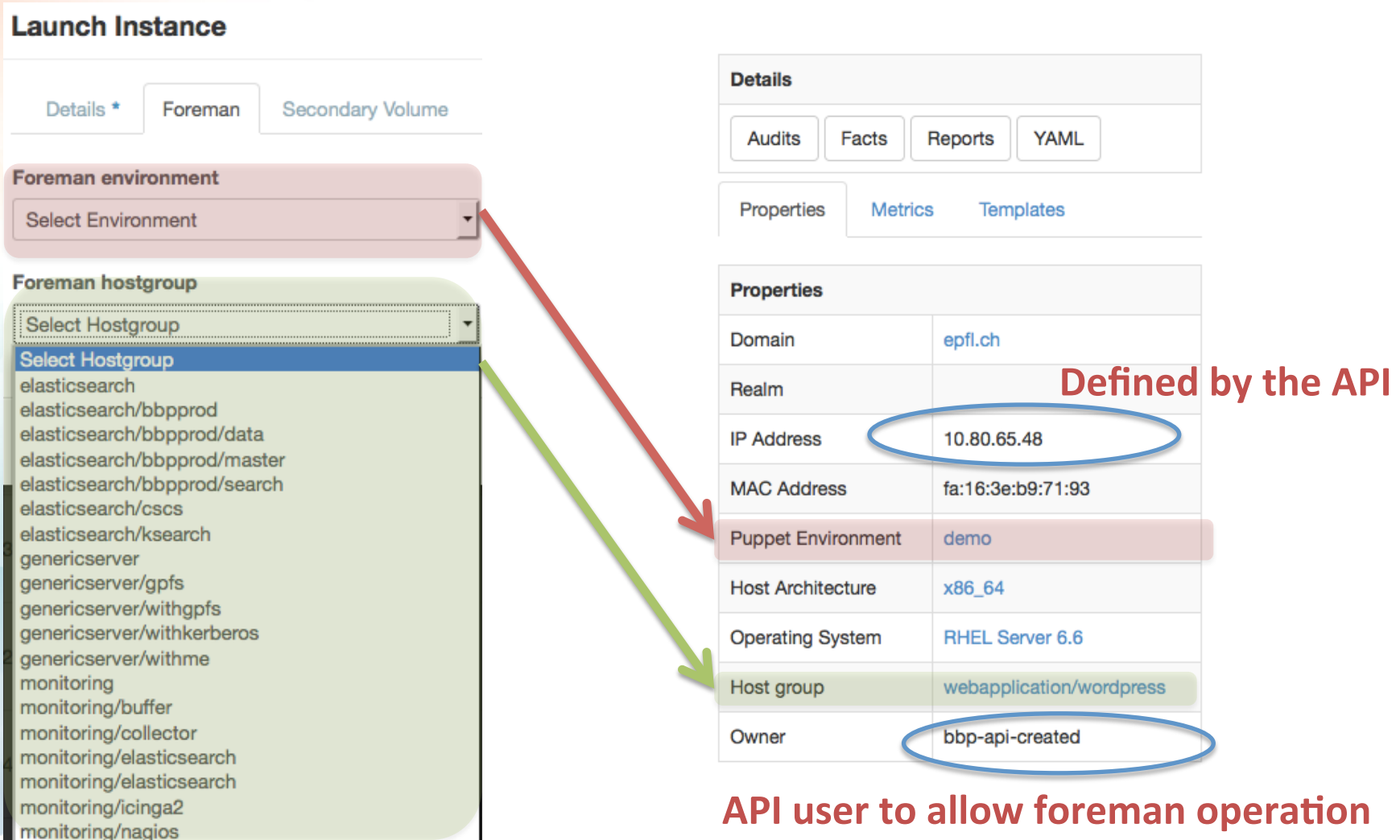
Auto naming in action



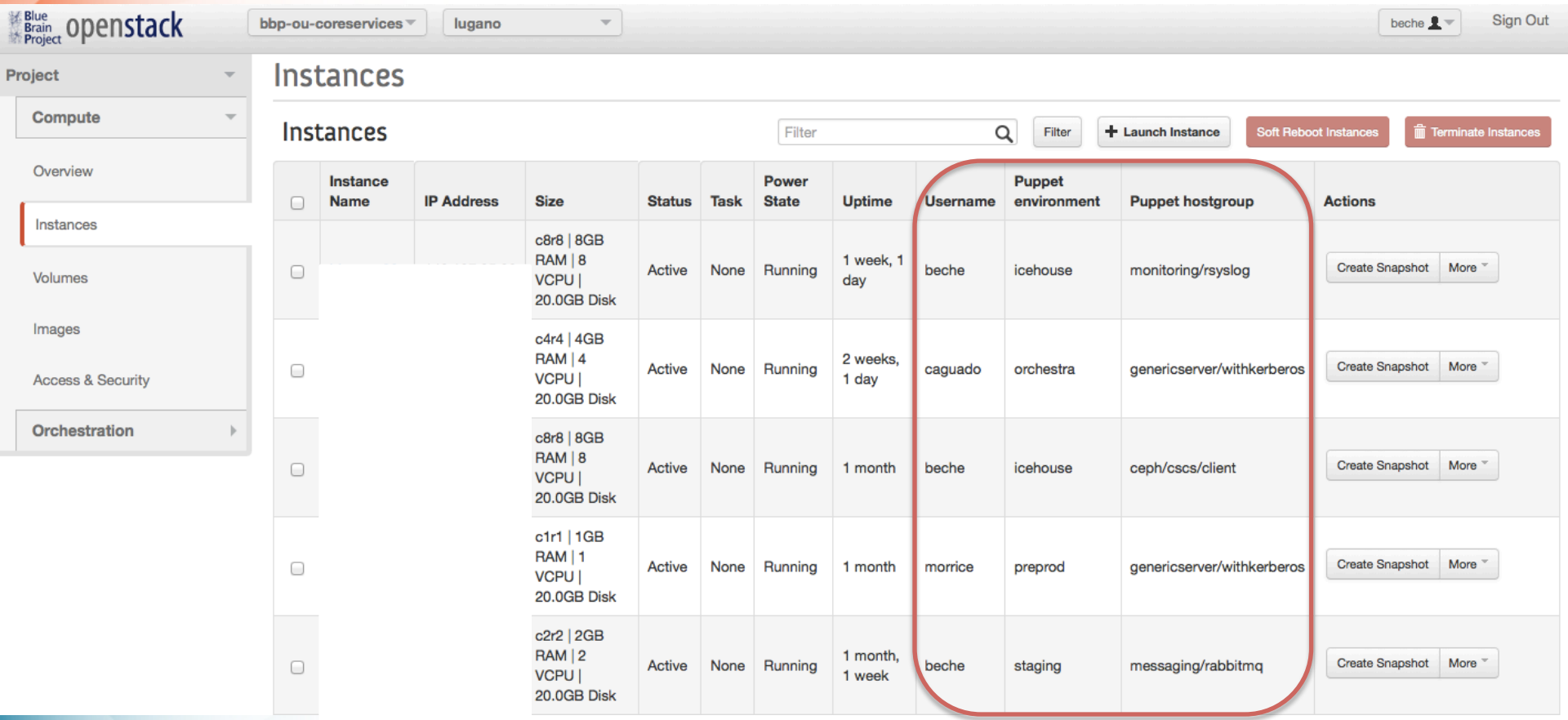
Hostname is chosen depending on the region / project

It can’t be changed !!

Puppet integration



Puppet integration



Custom fields from foreman and keystone

Puppet model

- 3 layers model for the code

Modules

90 modules

Module→ how you install a sotware

Puppet model

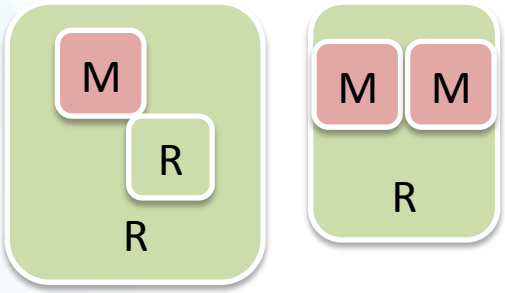
- 3 layers model for the code

Modules

Recipes

90 modules
171 recipes

Recipe → how you install a service



Recipes organization:

::apps
::env
::repo
::security
…

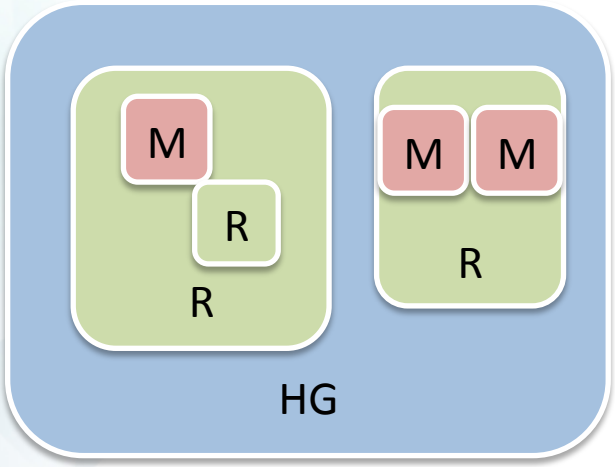
Puppet model

- 3 layers model for the code

Modules | Recipes | Hostgroup

90 modules
171 recipes
111 hostgroups

Hostgroup → services installed on a fqdn



Recipes organization:

::apps
::env
::repo
::security
…

Puppet model

- 3 layers model for the code

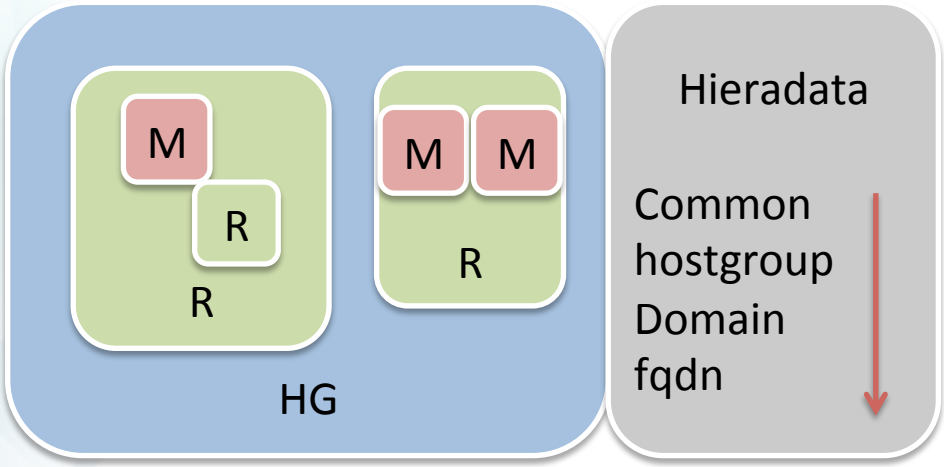
Modules | Recipes | Hostgroup

90 modules
171 recipes
111 hostgroups

- Hieradata for variables

Hieradata

Hieradata → Parameterization



Recipes organization:

::apps
::env
::repo
::security
…

Puppet example

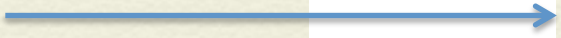
Host group definition

```
class role::webapplication::wordpress {

  include ::role::web application

  include ::env::krb5

  include ::apps::wordpress
}
```

Puppet example

Host group definition

```
class role::webapplication::wordpress {

  include ::role::web application

  include ::env::krb5

  include ::apps::wordpress
}
```



Kerberos recipe

```
class env::krb5 (
  $kstart       = false,
  $kstart_users = [],
) {


  include repo::epel

  include ::krb5

  $krb5_realms = hiera('krb5::realms')
  create_resources(krb5::realm, $krb5_realms)

  K5login <| tag == k5login |>

  ...

  package { $packages:
    ensure  => present,
    require => Class['repo::epel'],
  }
}
```

Include Kerberos module

Provided by krb module

Puppet example

Host group definition

```
class role::webapplication::wordpress {

  include ::role::web application

  include ::env::krb5

  include ::apps::wordpress
}
```

Kerberos recipe

```
class env::krb5 (
  $kstart       = false,
  $kstart_users = [],
) {


  include repo::epel

  include ::krb5

  $krb5_realms = hiera('krb5::realms')
  create_resources(krb5::realm, $krb5_realms)

  K5login <| tag == k5login |>

  ...

  package { $packages:
    ensure  => present,
    require => Class['repo::epel'],
  }
}
```

Include kerberos module

Provided by krb module

Wordpress recipe

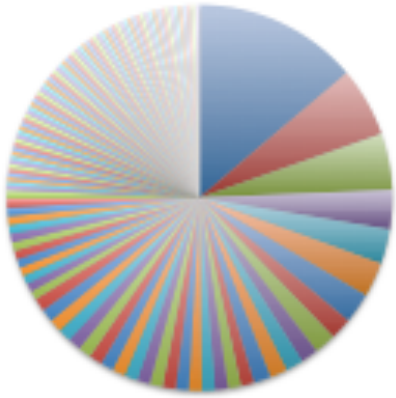
```
class apps::wordpress (
  $dbname       = 'wordpress',
  $dbuser       = 'wordpress',
  $dbpass       = 'notverysecure',
  $dbhost       = 'localhost',
  $wptitle      = 'wordpress',
  $wpuser       = 'admin',
  $wppass       = 'admin',
  $wpmail       = 'admin@example.com',
) {
  include ::apps::apache
  include ::php::mod_php5
  include ::mysql::server

   class { 'selinux':
    mode => 'disabled'
  }

  $php_packages = [ 'php-mysql' ]

  package {$php_packages:
    ensure => present,
    notify => Service['httpd'],
  }

  php::ini { '/etc/php.ini':
    date_timezone      => 'Europe/Zurich',
    short_open_tag     => 'On',
    max_execution_time => '300',
  }

…

}
```

hieradata/hostgroup/webapplication/wordpress.yaml

krb5::params::generate_host_keytab: true
apps::wordpress::dbname: ’wp-db’
apps::wordpress::dbpass: ’wp-db-pass'
apps::wordpress::wptitle: ’wp-title'
apps::wordpress::wpuser: ’wp-user’

Puppet environment

- High diversity in host group



Host group diversity

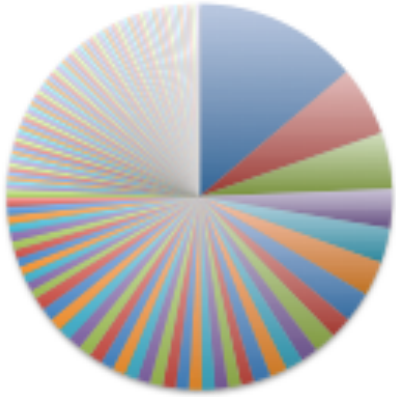
330 Servers

111 Host groups

Puppet environment

- High diversity in host group

- 3 main environments:
  - Devel, staging & preprod



Host group diversity

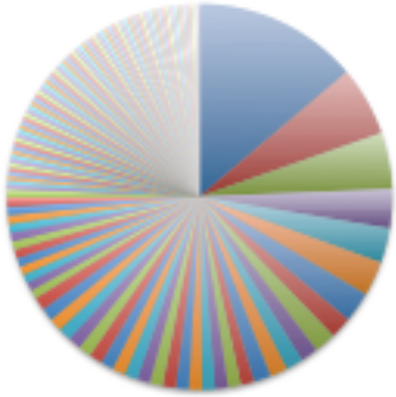
330 Servers
111 Host groups
3 Environments

Puppet environment

- High diversity in host group

- 3 main environments:
  - Devel, staging & preprod

- Git branch per environment
  - No push privileges on preprod
  - Only code review mechanism
    - Gerrit



Host group diversity

330 Servers
111 Host groups
3 Environments

Code review

1. Working on a topic branch
2. Committing the change (locally)
3. Submitting for review

```
git checkout origin/devel -b fix/elasticsearch-firewall
Branch fix/elasticsearch-firewall set up to track remote branch devel from origin.
Switched to a new branch 'fix/elasticsearch-firewall'
```

Code review

1. Working on a topic branch
2. Committing the change (locally)
3. Submitting for review

```
git diff
diff --git a/recipes/apps/manifests/elasticsearch/search.pp b/recipes/apps/manifests/elasticsearch/search.pp
index 0c7c858..6440d59 100644
--- a/recipes/apps/manifests/elasticsearch/search.pp
+++ b/recipes/apps/manifests/elasticsearch/search.pp
@@ -41,6 +41,15 @@ class apps::elasticsearch::search {
      dport  => '5601',
      action => 'accept',
    }
+
+   @firewall { '5601-v6 allow kibana4':
+     state    => 'NEW',
+     proto    => 'tcp',
+     dport    => '5601',
+     action   => 'accept',
+     provider => 'ip6tables'
+   }
+
  }
```

Code review

1. Working on a topic branch
2. Committing the change (locally)
3. Submitting for review

```
git commit -sam "Opening firewall for kibana4 on ipv6"
[fix/elasticsearch-firewall 308211b] Opening firewall for kibana4 on ipv6
 1 file changed, 9 insertions(+)

git review devel
remote: Resolving deltas: 100% (6/6)
remote: Processing changes: (-)
remote: Processing changes: new: 1, refs: 1, done
remote:
remote: New Changes:
remote:   https://abcd.epfl.ch/xxx/13217 Opening firewall for kibana4 on ipv6
remote:
To ssh://beche@abcd.epfl.ch/xxx/puppet-modules
 * [new branch]      HEAD -> refs/publish/devel/fix/elasticsearch-firewall
```

Code review

All My Projects People Documentation

Changes Drafts Draft Comments Watched Changes Starred Changes Groups

Search term Search Alexandre Beche

Change 13217 - **Needs Code-Review**

Reply... Code-Review+2 Patch Sets (1/1) Download

```
Opening firewall for kibana4 on ipv6

Change-Id: Ib01e74b60bce577dbb37ba08a55f2d2caefe17eb
Signed-off-by: Alexandre Beche <alexandre.beche@epfl.ch>
```

Owner Alexandre Beche
Reviewers BBP CI x Add...
Project puppet-modules
Branch devel
Topic fix/elasticsearch-firewall
Strategy Merge if Necessary
Updated 27 seconds ago

Cherry Pick Rebase Abandon Follow-Up

Code-Review
Verified +1 BBP CI

Automatic linting

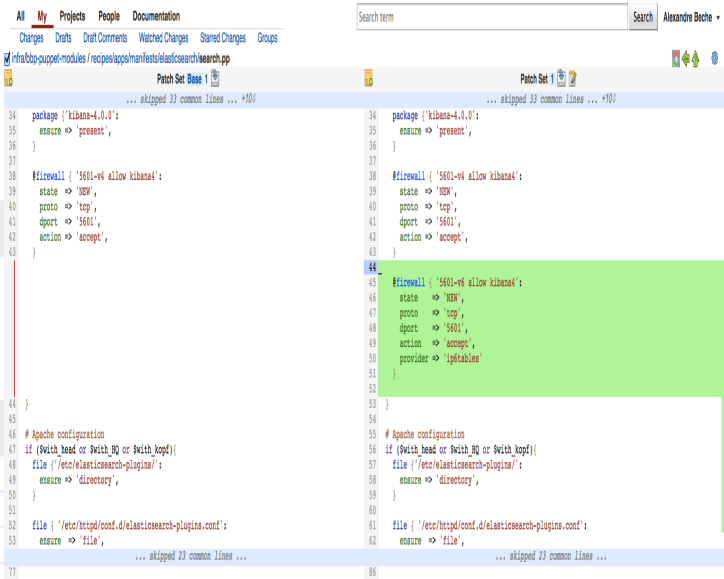
Author	Alexandre Beche <alexandre.beche@epfl.ch>	Jun 12, 2015 9:55 AM
Committer	Alexandre Beche <alexandre.beche@epfl.ch>	Jun 12, 2015 9:55 AM
Commit	308211b65d97301f17a5c77e66e83c44682dcfb0	(browse)
Parent(s)	d34814272c2f3ef968c1b341dc48faada46774f2	(browse)
Change-Id	Ib01e74b60bce577dbb37ba08a55f2d2caefe17eb	

Files Open All Diff against: Base Edit

File Path	Comments	Size
Commit Message		
recipes/apps/manifests/elasticsearch/search.pp		9
		+9, -0

History Expand All

Alexandre Beche	Uploaded patch set 1.
BBP CI	Patch Set 1: Build Started https://bbpcode.epfl.ch/ci/job/infra.bbp-puppet-modules.gerrit/4285/
BBP CI	Patch Set 1: Verified+1 Build Successful https://bbpcode.epfl.ch/ci/job/infra.bbp-puppet-modules.gerrit/4285/ : SUCCESS



Puppet synchronization

- Why?
  - People are only allowed to commit change in GIT
  - Puppet master needs to be aware

Puppet synchronization

- Why?
  - People are only allowed to commit change in GIT
  - Puppet master needs to be aware
- How?
  - Recurrent cron on puppet master:
    1. Clone git bare repository
    2. Checkout branch with modification

Puppet synchronization

- Why?
  - People are only allowed to commit change in GIT
  - Puppet master needs to be aware
- How?
  - Recurrent cron on puppet master:
    1. Clone git bare repository
    2. Checkout branch with modification
    3. On modified branch:
       - Find hieradata modification (including gpg files)
       - Load hieradata into redis if change (env transaction)
       - Update environment and host group in foreman

Kerberos self-service

- How do you get access to the VM?
  - Public key : owner only
  - Kerberos : valid ktab required

Kerberos self-service

- How do you get access to the VM?
  - Public key : owner only
  - Kerberos : valid ktab required
- Kerberos self-service api with basic validation:
  - Machine in puppet authorized
  - Hosts can only request it for their fqdn

From localhost to production in 5 minutes

1. Puppet code is committed…
2. … and reviewed
3. Puppet-aware VM is spawned…
4. … with automatic registration in master
5. VM got contextualized
   - Foreman env / hg apply at boot time
6. Kerberos ktab automatically generated

Monitoring at scale

Mission:

Delivers a generic monitoring workflow and infrastructure to provide dashboards, reports, notification system and analytic tools.

Wide range of sources

- Physical & virtual nodes
- Network equipment
- Application

Different data types

- Metrics
- Status
- Logs

Single bucket
collect / store everything

Notification

- Email

Dashboards

- Sys-admin
- Executive

Analytics tool

- Drilldown the raw data

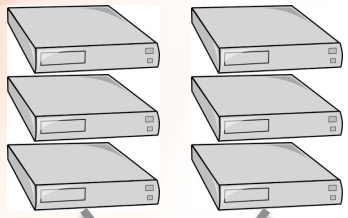
APIs

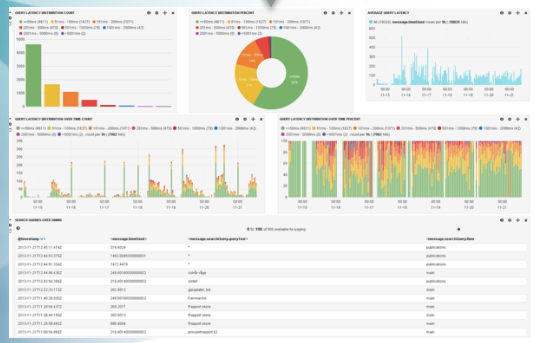
- Interface to anything

TARGETS BUCKET TOOLS

Data sources
&
challenges

Logs

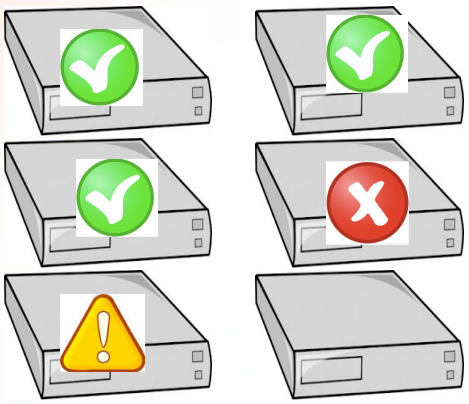




```
2015-01-23 09:26:46,343 3785 DEBUG openstack_auth.backend Beginning
user authentication for user "beche".
2015-01-23 09:26:46,344 3785 DEBUG keystoneclient.session REQ: curl -i
-X POST https://bbpopenstack.epfl.ch:5000/v2.0/tokens -H "Content-
Type: application/json" -H "Accept: application/json" -H "User-Agent:
pytho
n-keystoneclient" -d '{"auth": {"passwordCredentials": {"username":
"beche", "password": "beche"}}}'
RESP BODY: {"access": {"token": {"issued_at":
"2015-01-23T09:26:47.697197", "expires": "2015-01-23T10:26:47Z", "id":
"158ee4e1d25c4717825ca40589b2859a"}, "serviceCatalog": [], "user":
{"username": "beche", "roles
_links": [], "id": "93639e67f4dd40e3a98e8fddf9aa803e", "roles": [],
"name": "beche"}, "metadata": {"is_admin": 0, "roles": []}}}
RESP BODY: {"tenants_links": [], "tenants": [{"description": "tenant
for beche", "enabled": true, "id": "4f2c64a4d8954ae98aac98045901cb79",
"name": "beche"}, {"description": "tenant for bbp-ou-coreservices",
"ena
bled": true, "id": "56afe0635fcc418b884e223ef252567d", "name": "bbp-ou-
coreservices"}]}
```

- Central store containing logs
- Metadata + free text
- Big security concerns

Status



Host service

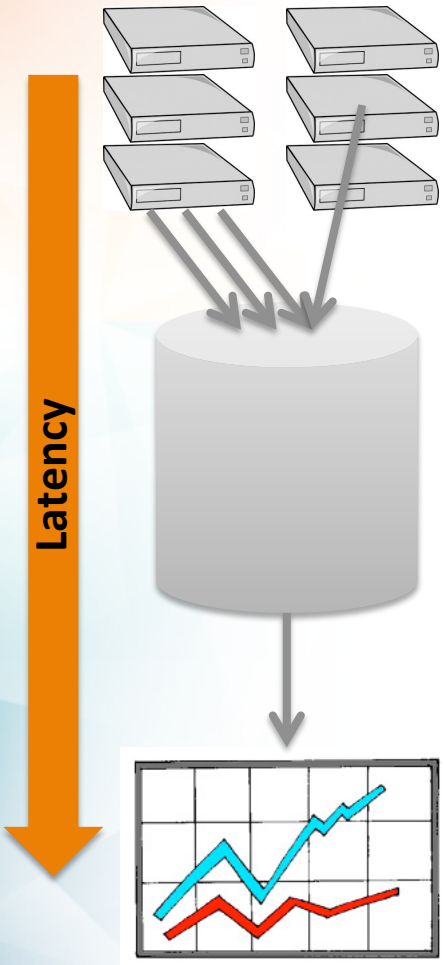
State
- OK
- Problem
- Unknown

Severity
- Warning
- …
- Critical

/dev/hda1 Free Space	CRITICAL
Current Users	OK
HTTP	WARNING
PING	OK
Total Processes	OK

- Health of the infrastructure
  - Snapshot at a given time
- Transition can be recorded
  - Availability algorithm

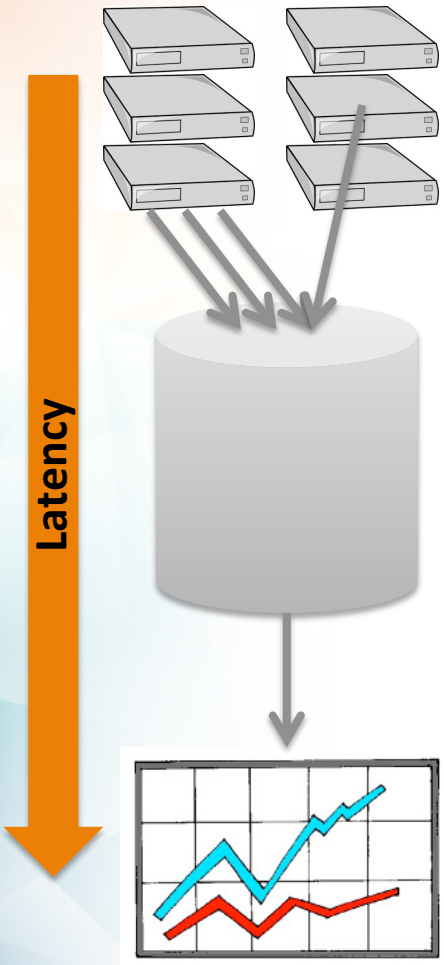
Metrics



- Cardinality
  - How many metrics you collect
- Resolution
  - How often you collect
- Scale
  - How many host are reporting

- Latency
  - Expected time for the metrics to be available

Metrics

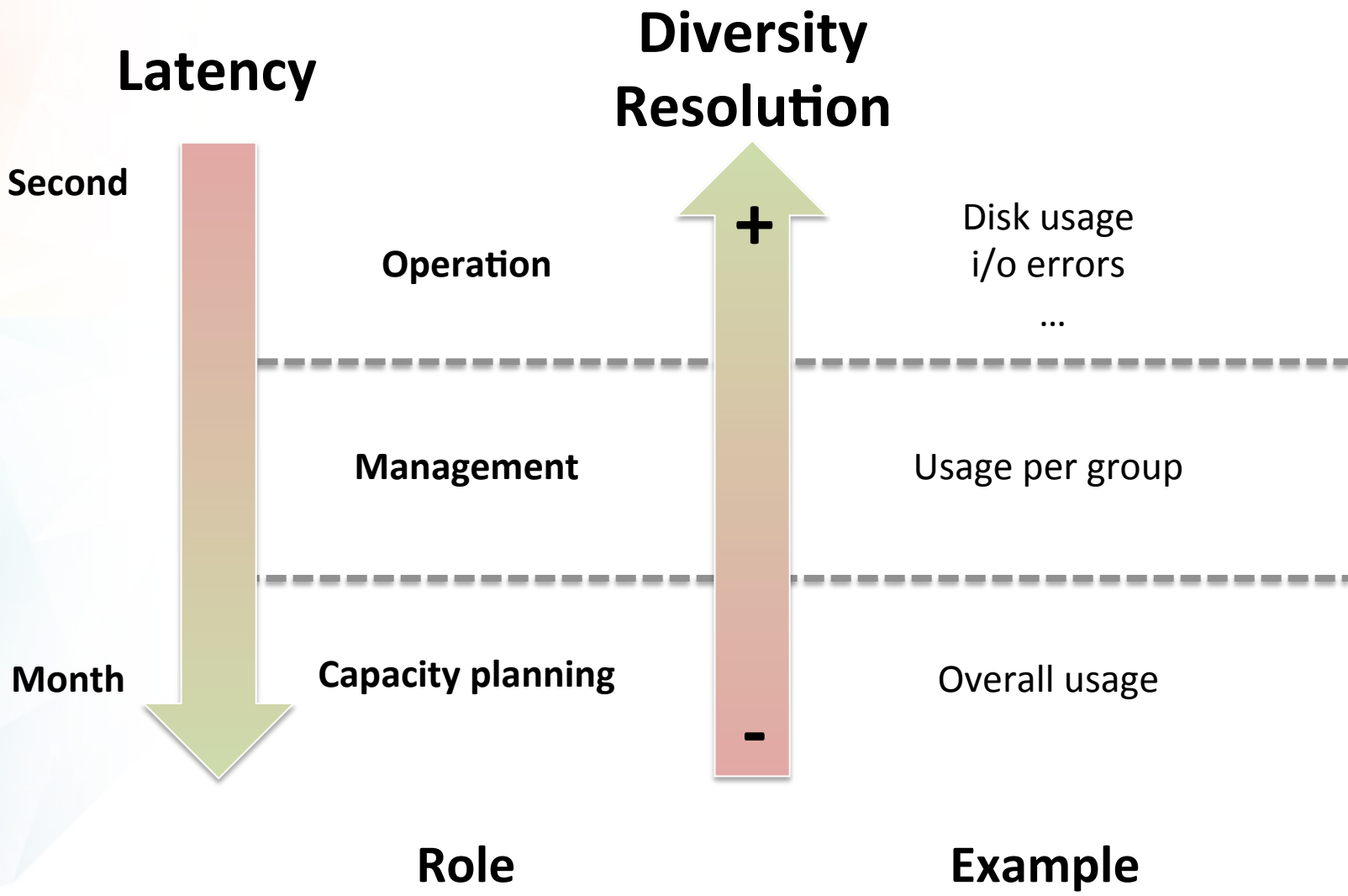


- Cardinality **~ 250**
  - How many metrics you collect
- Resolution **10 s**
  - How often you collect
- Scale **~100**
  - How many host are reporting

2’500 / s

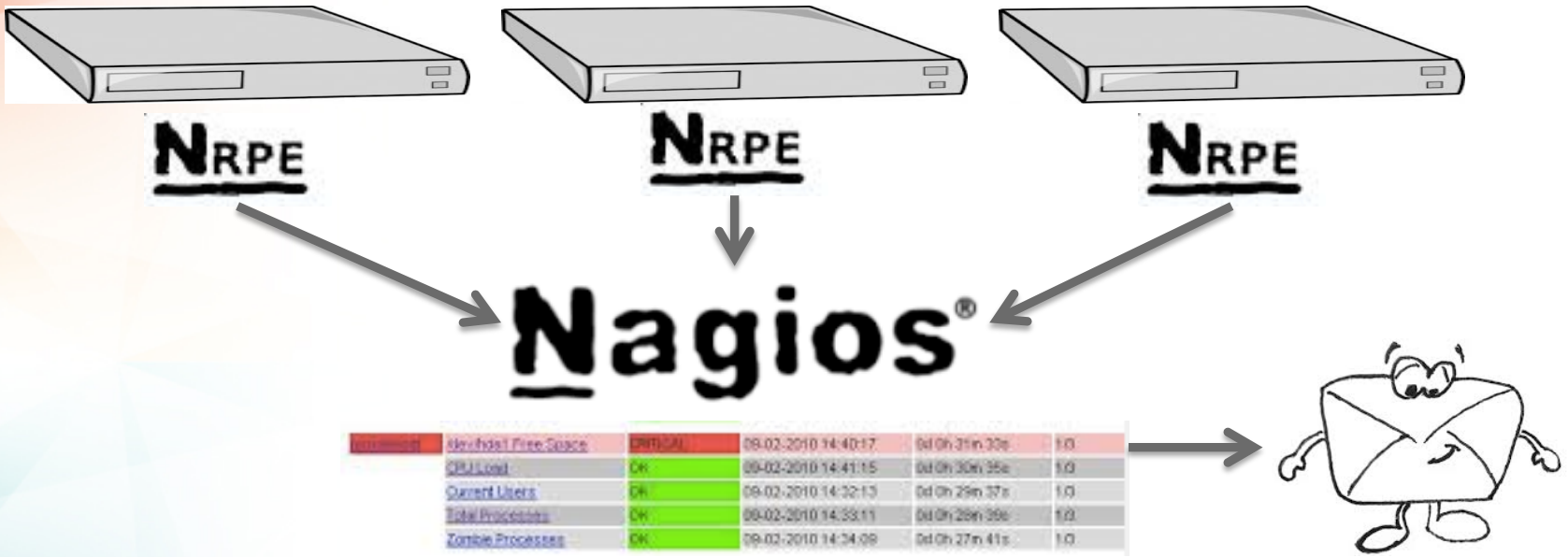
- Latency
  - Expected time for the metrics to be available

Metrics usage

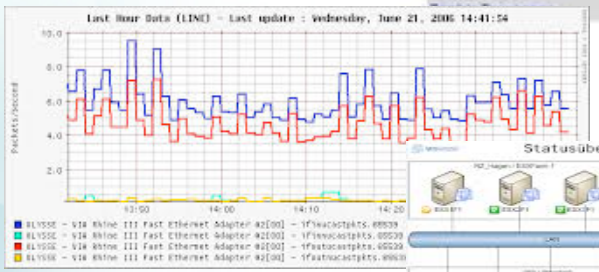


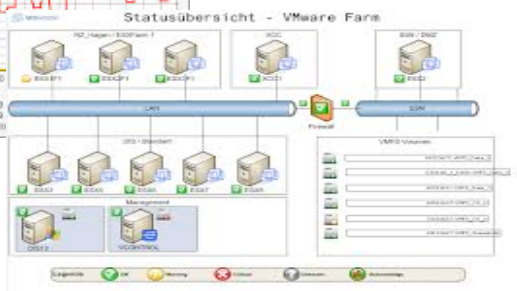
Building a monitoring framework

Nagios based infrastructure



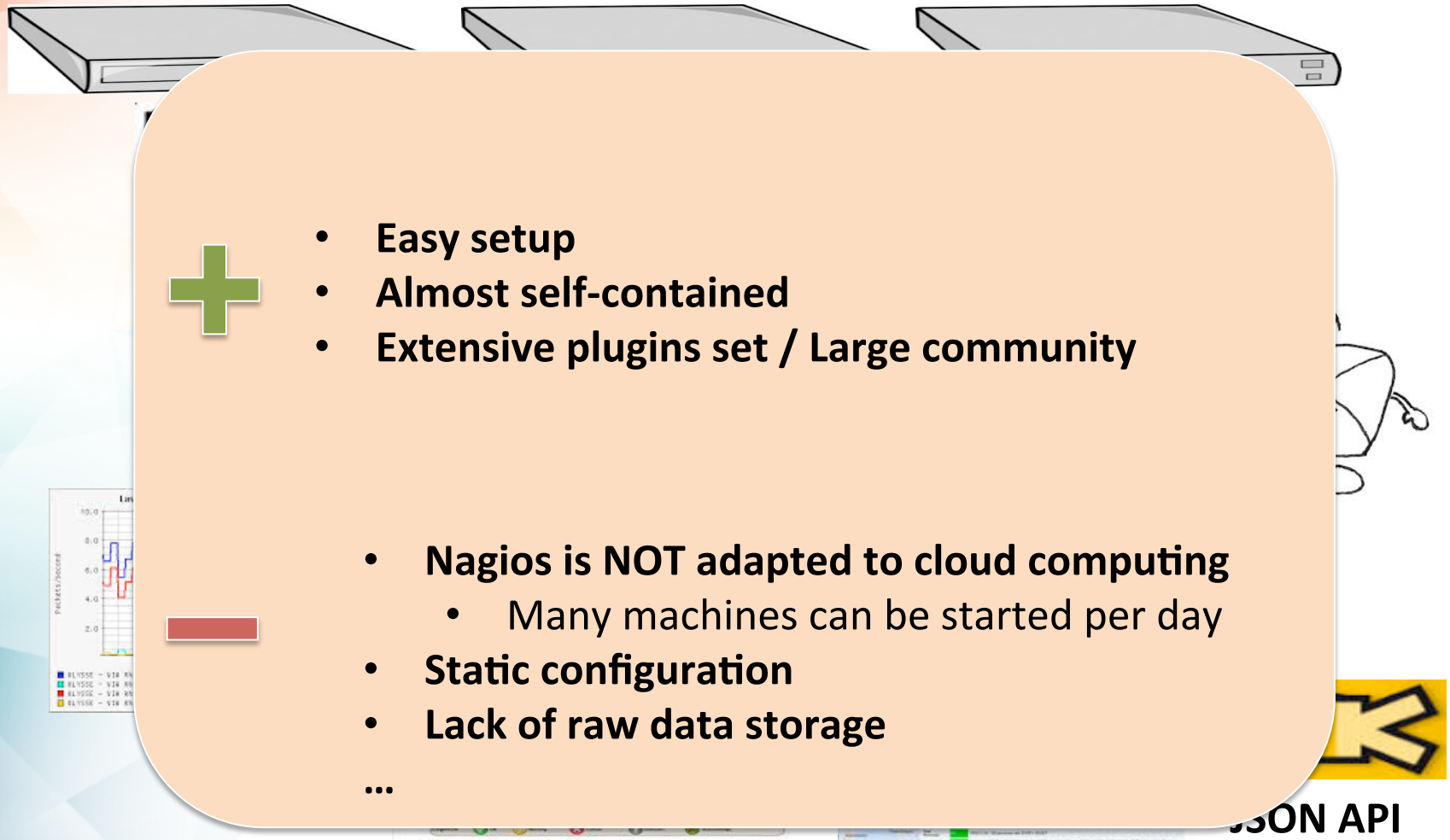
Nagios based infrastructure





JSON API

Nagios based infrastructure



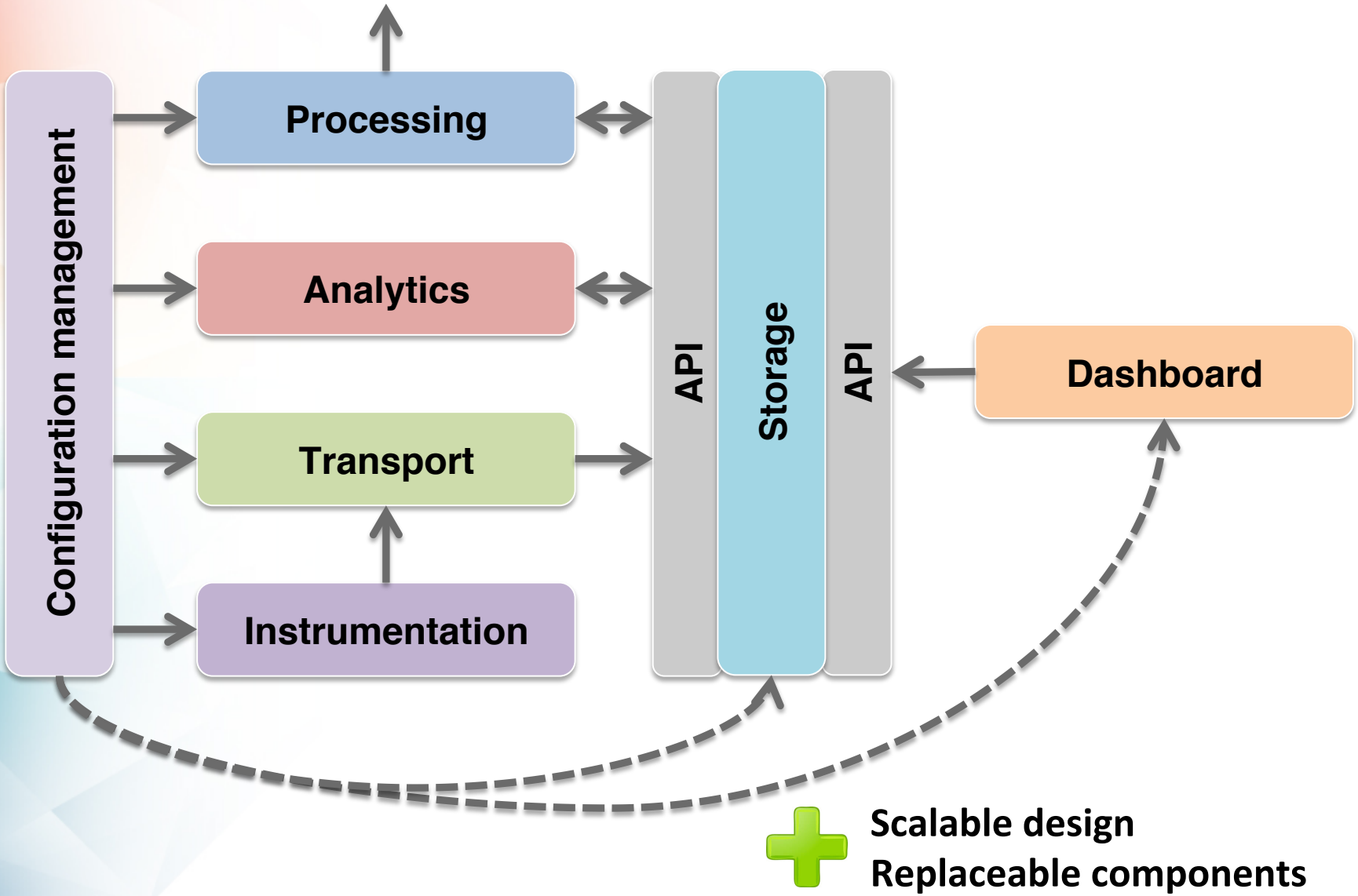
- **Easy setup**
- **Almost self-contained**
- **Extensive plugins set / Large community**

- **Nagios is NOT adapted to cloud computing**
  - Many machines can be started per day
- **Static configuration**
- **Lack of raw data storage**

…

JSON API

Framework building blocks



Framework consideration

- Footprint should be limited (5% max)
  - Limited impact on main activity
- Different from accounting
  - We need statistically significant datasets
  - Data loss is acceptable (but we aim to limit it)
- Tool chain must be made of building blocks
  - Open-source technologies move really fast
  - API based communication

Server instrumentation


```
class os::server {
  ...
  include ::env::icinga2
  include ::apps::collectd::publisher
  include ::rsyslog::client
  ...
}
```

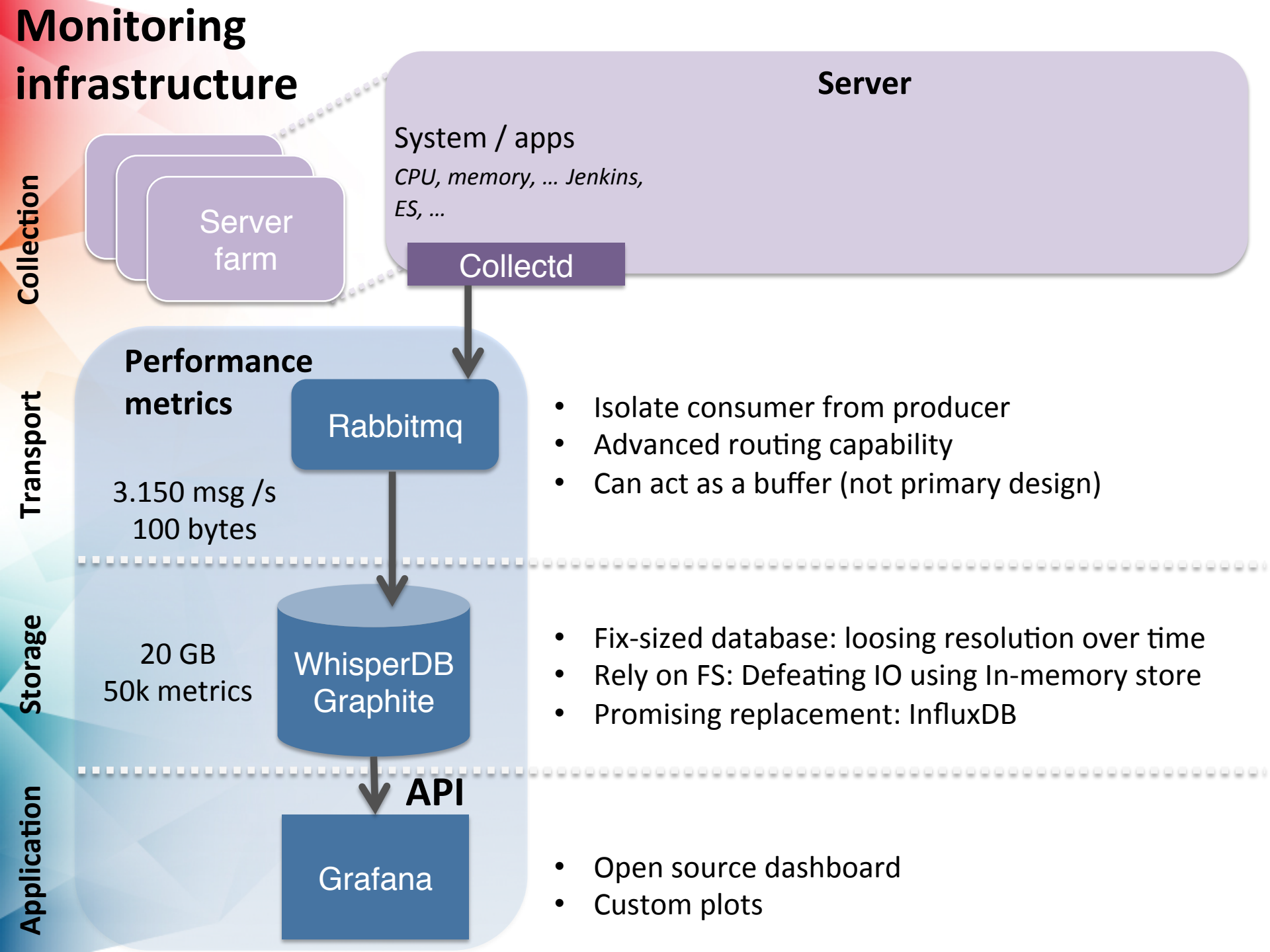
```
class apps::collectd::publisher {

  include ::apps::collectd

  $default_plugins = [
    'cpu',
    'memory',
    'interface',
    'load',
    ...
  ]

  collectd::plugin { $default_plugins: }

  $amqp_channel   = hiera('apps::collectd::publish','monitors')
  $hiera_prefix   = "apps::collectd::amqp::${amqp_channel}"

  collectd::config::plugin::multipart { 'collectd_amqp_publish':
    plugin      => 'amqp',
    settings    => "
      <Publish "publish_${amqp_host}">
        Host        hiera("${hiera_prefix}::host")
        Port        hiera("${hiera_prefix}::port")
        VHost       hiera("${hiera_prefix}::vhost")
        User        hiera("${hiera_prefix}::user")
        Password    hiera("${hiera_prefix}::password")
        Exchange    hiera("${hiera_prefix}::exchange")
        RoutingKey  hiera("${hiera_prefix}::routingkey")
      </Publish>
    "
  }
```

Plugins installation

Pipeline configuration

Push monitoring

Monitoring infrastructure

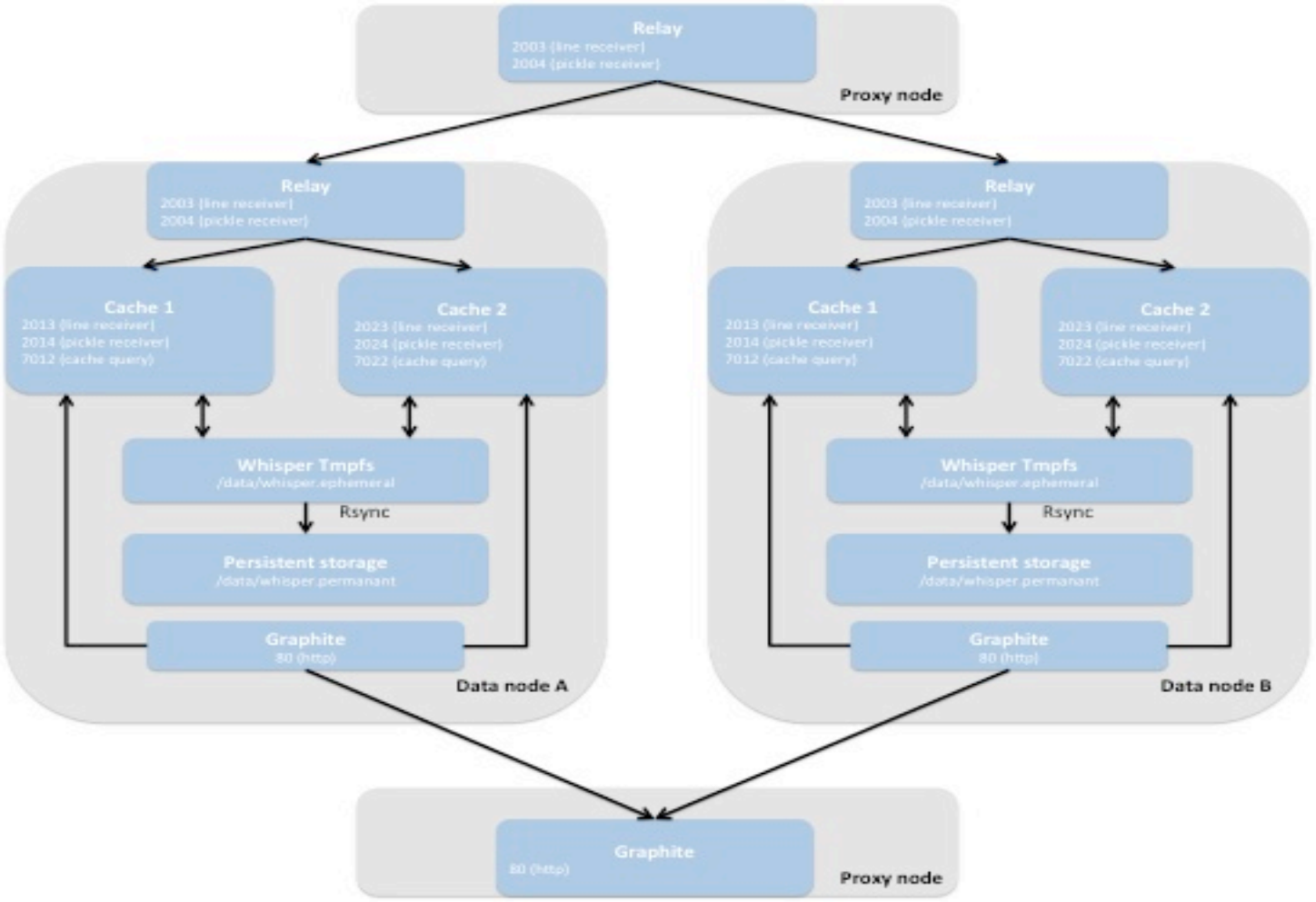


- Isolate consumer from producer
- Advanced routing capability
- Can act as a buffer (not primary design)

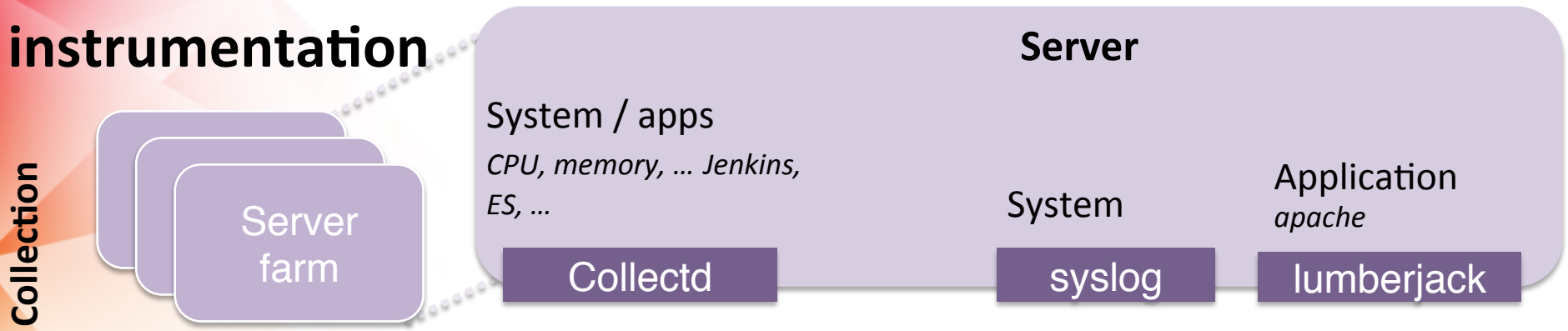
- Fix-sized database: loosing resolution over time
- Rely on FS: Defeating IO using In-memory store
- Promising replacement: InfluxDB

- Open source dashboard
- Custom plots

Defeating IO limitations



Server instrumentation




```
class os::server {
  ...
  include ::env::icinga2
  include ::apps::collectd::publisher
  include ::rsyslog::client
  ...
}
```

```
class rsyslog::client (
  $server       = hiera('rsyslog::server::name','rsyslog'),
  $port         = hiera('rsyslog::server::port','514'),
  $tcp          = hiera('rsyslog::server::tcp',true),
  $spool_size   = hiera('rsyslog::spool_size','1g'),
  {

  ...

rsyslog::addconf { '99-centralsyslog.conf':
  ensure  => present,
  content => template("${module_name}/centralsyslog.conf.erb");
}
}
```

Rsyslog config

Push monitoring

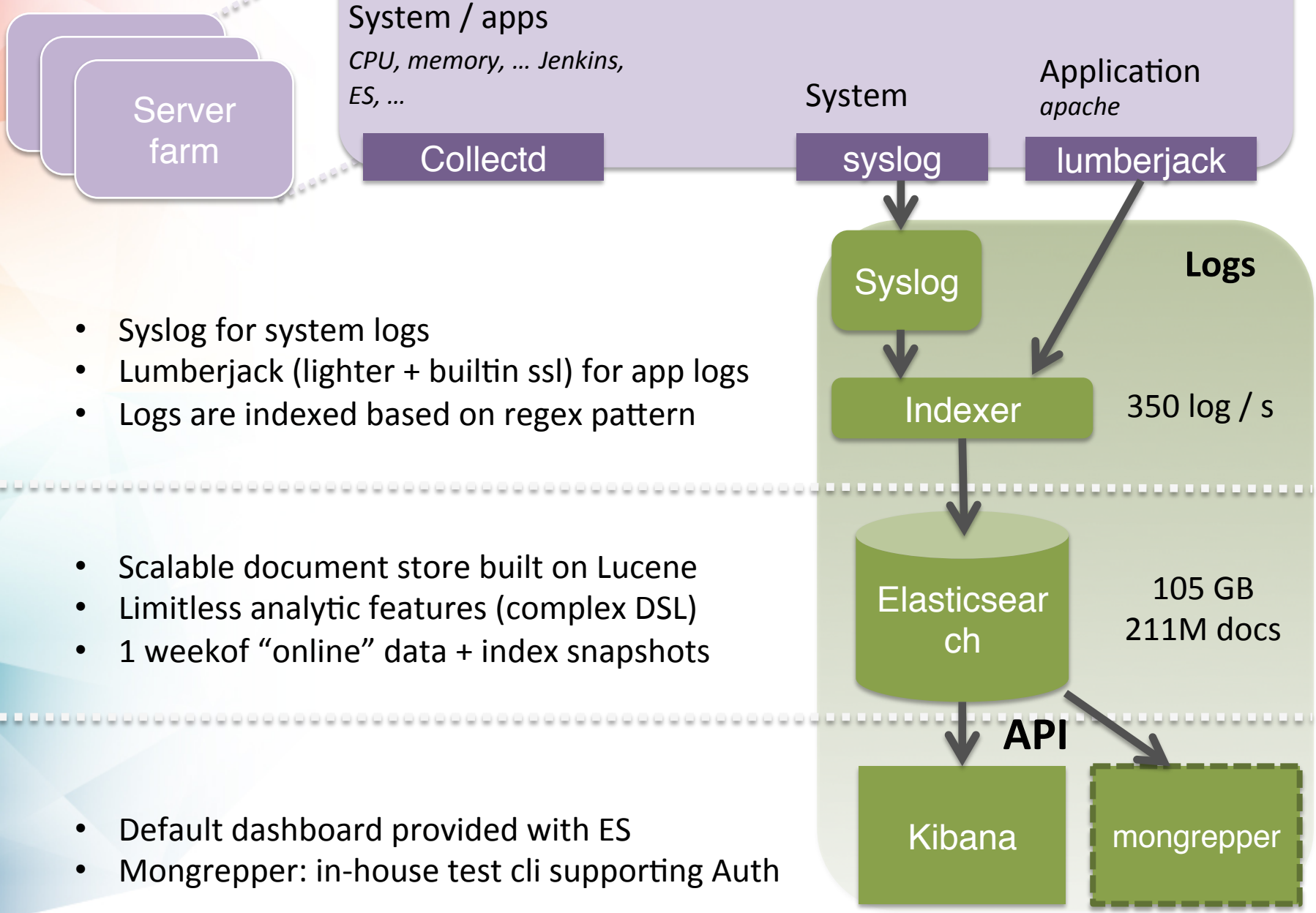
Monitoring infrastructure

Collection

Transport

Storage

Application



- Syslog for system logs
- Lumberjack (lighter + builtin ssl) for app logs
- Logs are indexed based on regex pattern

- Scalable document store built on Lucene
- Limitless analytic features (complex DSL)
- 1 weekof “online” data + index snapshots

- Default dashboard provided with ES
- Mongrepper: in-house test cli supporting Auth

Server instrumentation

Exported resource

Machine details

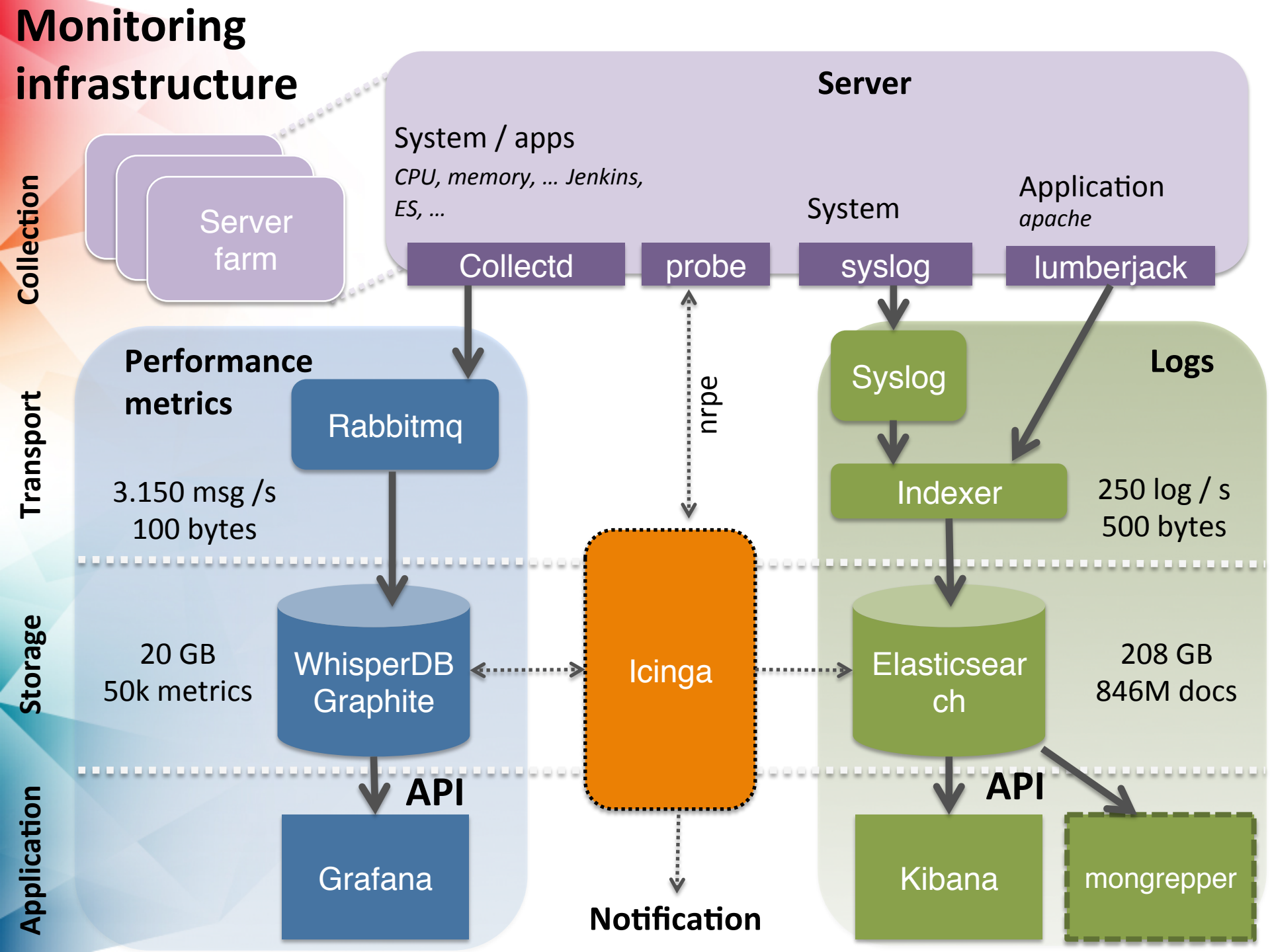
Foreman aware

```
class os::server {
  ...
  include ::env::icinga2
  include ::apps::collectd::publisher
  include ::rsyslog::client
  ...
}
```

```
class env::icinga2 {

  @@icinga2::object::host { $::fqdn:
    tag                 => $::environment,
    display_name        => $::fqdn,
    ipv4_address        => $::ipaddress_eth0,
    groups => [],
      vars => {
      os                => $::kernel,
      distro            => $::operatingsystem,
      virtual_machine   => $::is_virtual,
      foreman_environment => $::environment,
      foreman_hostgroup => $::hostgroup,
    },
    target_dir          => '/etc/icinga2/objects/hosts',
    target_file_name    => "${fqdn}.conf",
  }

  ...

}
```

Pull monitoring

Monitoring infrastructure



PoC: Security on ES

- Problematic:
  - ElasticSearch does not have any (OpenSource) AuthN/AuthZ mechanism
  - User wants to access logs of their machines

PoC: Security on ES

- Problematic:
  - ElasticSearch does not have any (OpenSource) AuthN/AuthZ mechanism
  - User wants to access logs of their machines
- Workaround:
  - Filtered aliases per server
  - Proxy access to these aliases using httpd
  - ***Using mod_rewrite to avoid query-crafting***
  - Delegate authN/AuthZ to apache
    - AuthN : kerberos
    - AuthZ : ldap

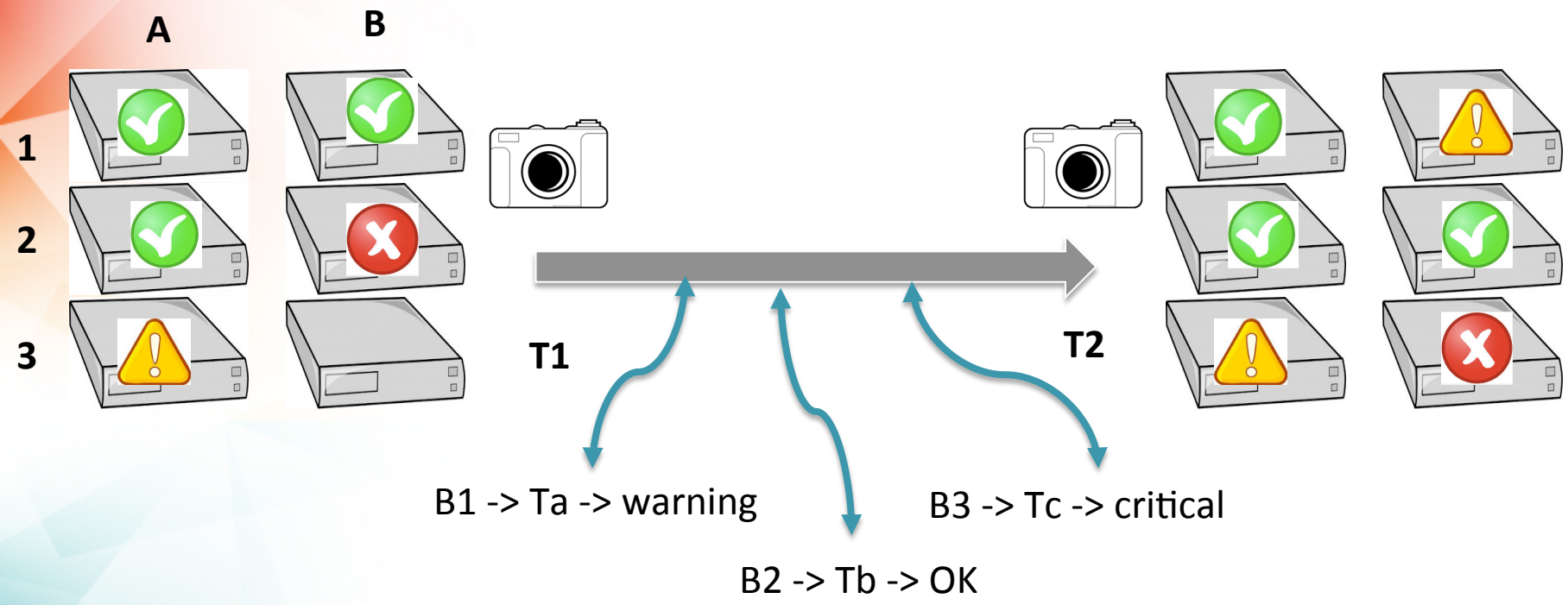
Mongrepper (logs)

```
./mongrepper logs --help
usage: mongrepper logs [-h] [-r REGION] [-H HOST] [-l LIMIT] [-s SORT] [-p]
optional arguments:
  -h, -help                    show this help message and exit
  -r REGION, --region REGION   Choose a region to grep for logs
  -H HOST, --host HOST         Filter data for the given host
  -l LIMIT, --limit LIMIT      Limit the number of returned document
  -s SORT, --sort SORT         Sort the data
  -p, --prettyprint            Properly format the response
```

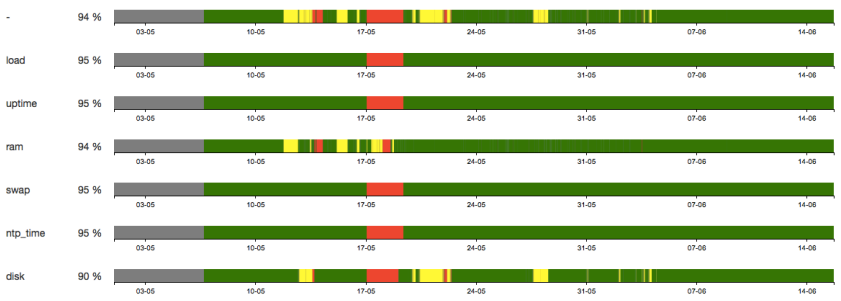
Live demo api

Advanced use cases & Visualizations

Availability computation



- Transition replaces state
- History over a time period
- Algorithm for availability



State / Availability APIs

- Internal state of services:
  - Lightweight API in go
- More advanced API for availability computation / History
- All the APIs are foreman aware

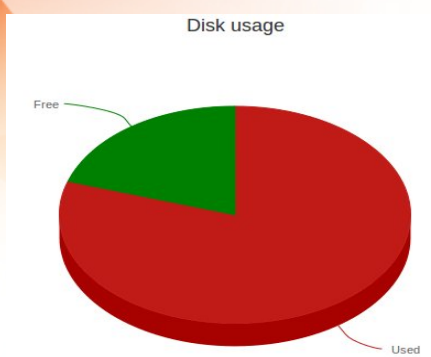
Live demo

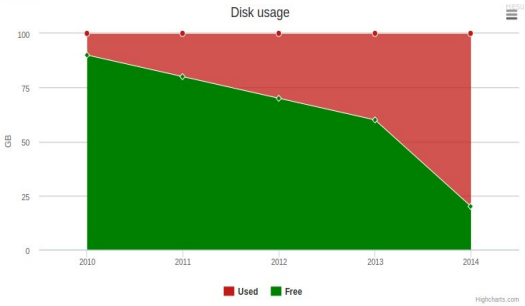
Event stream & collector

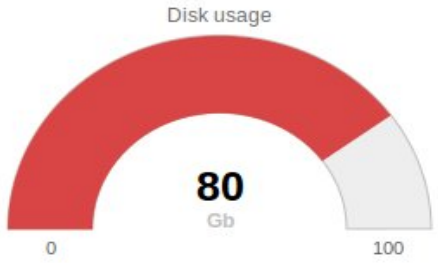
- History can be only computed if transition are recorded
- Notification script reports status change into a RabbitMQ queue
- A collector listen on this queue and store in ElasticSearch new state
- Mongrepper can query the state history

Live demo

Asking the good question



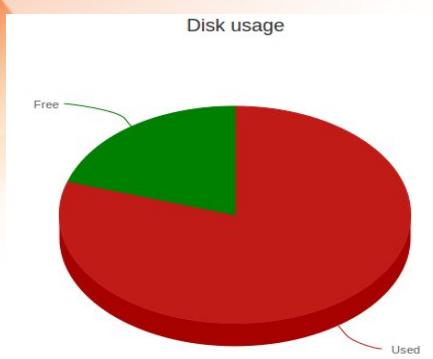


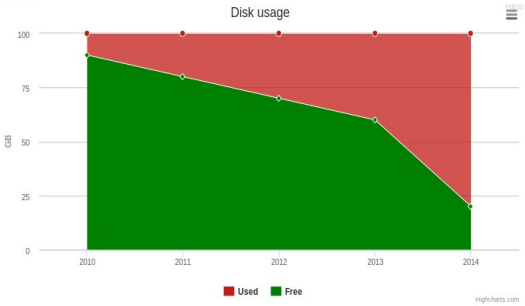


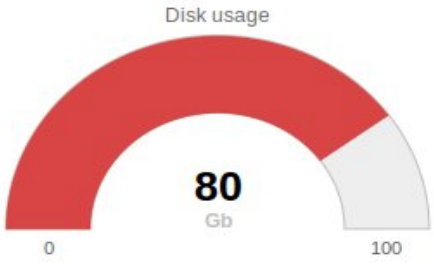
State

Is my disk full ?

Asking the good question





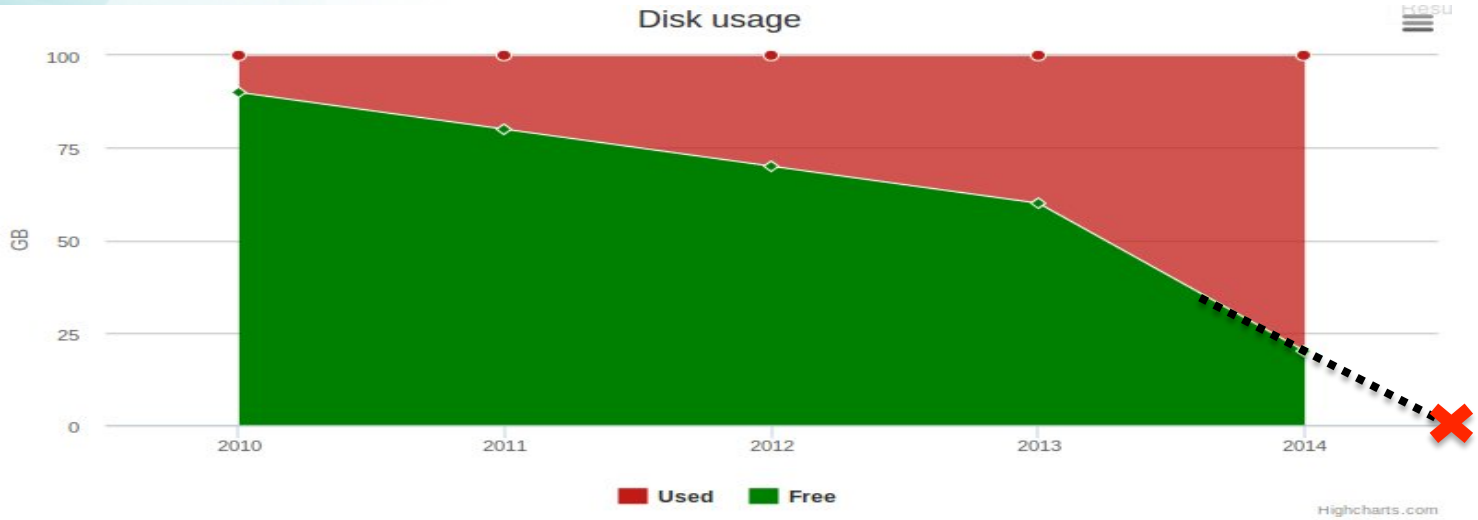


State

Is my disk full ?

Prediction

When my disk will be full ?

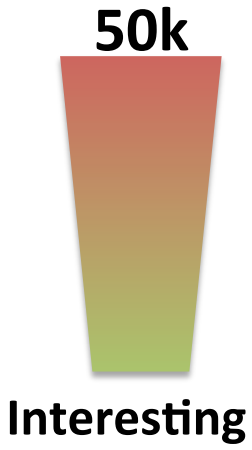


Data are good derivative are better

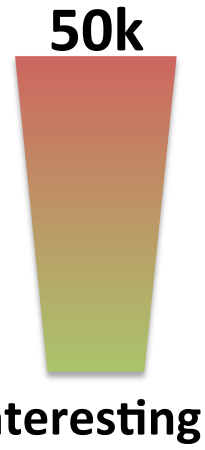
- *Derivative enables prediction*
  1. Smooth the time series
     1. Down sampling, moving average
  2. Derive the time series
  3. Extrapolate the time serie by applying the derivative

Anomaly detection (skyline)

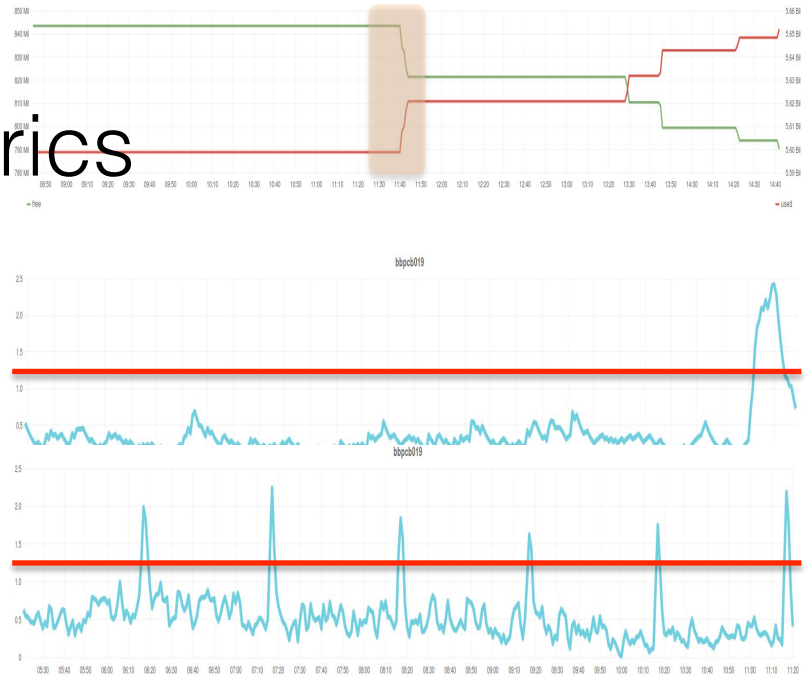
- Impossible to analyze all metrics
  - Automatic filtering required



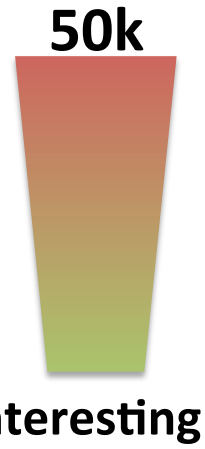
Anomaly detection (skyline)



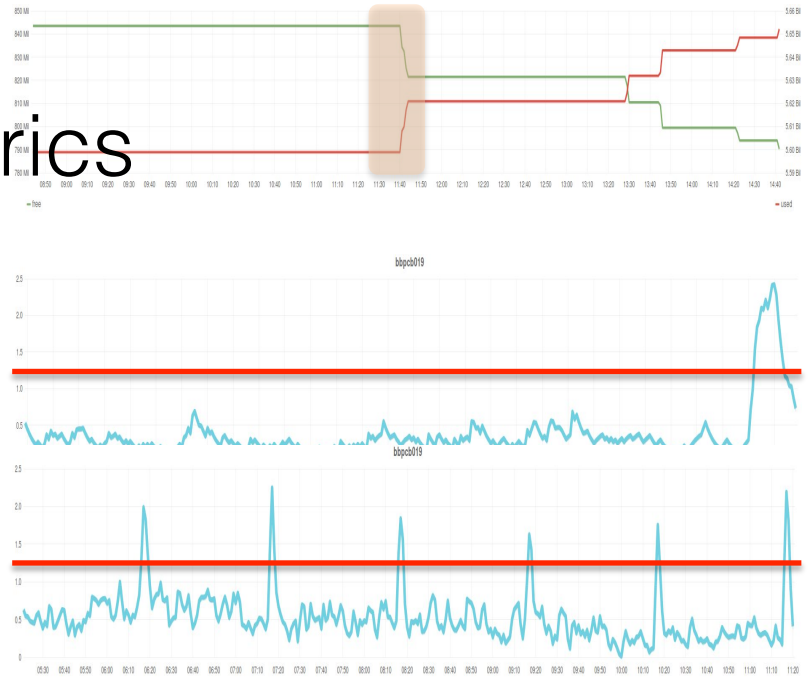
- Impossible to analyze all metrics
  - Automatic filtering required
- Requires heavy tuning
  - Highly correlated metrics
  - Recurrent patterns



Anomaly detection (skyline)



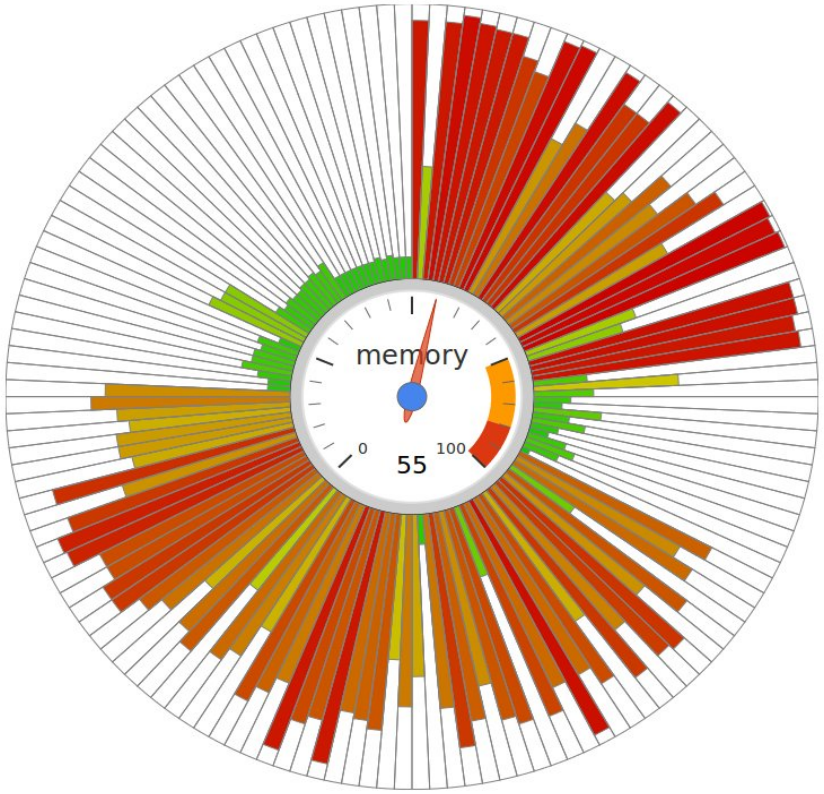
- Impossible to analyze all metrics
  - Automatic filtering required
- Requires heavy tuning
  - Highly correlated metrics
  - Recurrent patterns
- *How to deal with logs?*
  - *Machine learning*



Interesting work

- Complex Event Processing engine:
  - Paradigm shift: streams replace datasets
  - Limitless capability by correlating ALL data sources
  - Technologies: Riemann, ESPER
- Anomaly / Correlation detection:
  - Requires tuning (but promising)
  - Skyline + Oculus (etsy) for Time series
  - Machine learning (Mahout) for logs

Advanced visualization

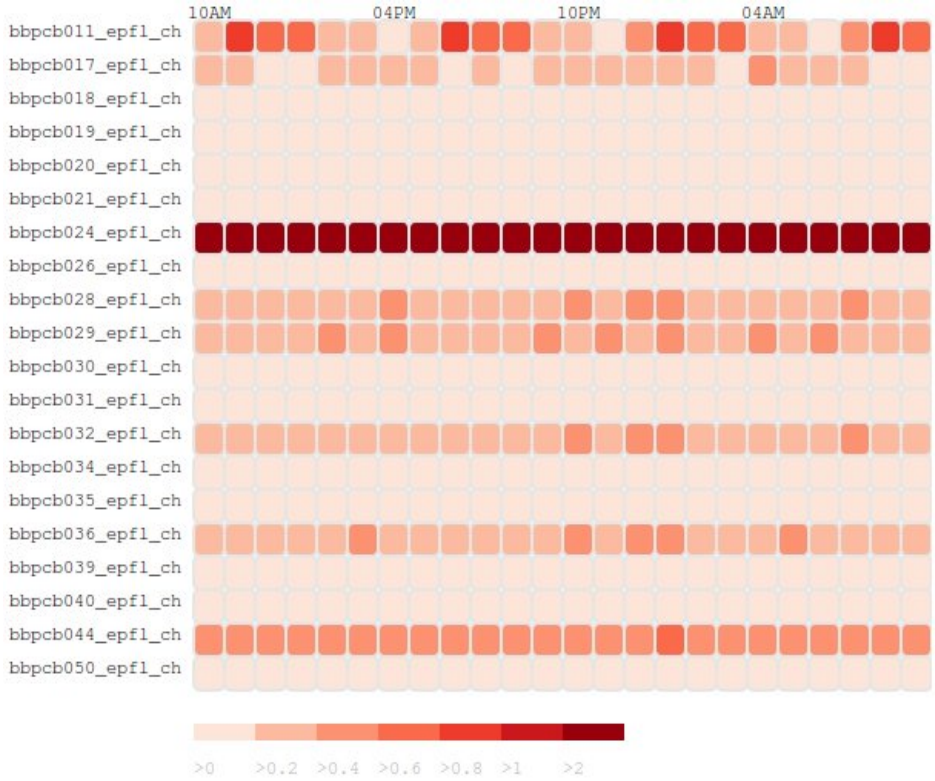


Sunburst chart:

- "Cluster" aware
- Metric agnostic (100% normalizable)
- Snapshot in time
- Scalable representation

Heat map:

- "Cluster" aware
- Metric agnostic (100% normalizable)
- Evolution over time



Summary

- Project still in “ramp-up” phase
  - Infra needs to be ready for future requirements (scalable / elastic)
- Very diverse infra
  - Impose strict deployment rules and reproducibility
  - Automating as much as possible
- Infra should stay “under control”
  - Monitor as much as we can:
    - To ease operations / developments

Thanks for your attention

Alexandre.beche@epfl.ch