



Manage MySQL like a devops sysadmin

Frédéric Descamps

Webinar – Oct 2012

Who am I ?

Frédéric Descamps

@lefred

<http://about.be/lefred>

Managing MySQL since 3.23 (as far as I remember)

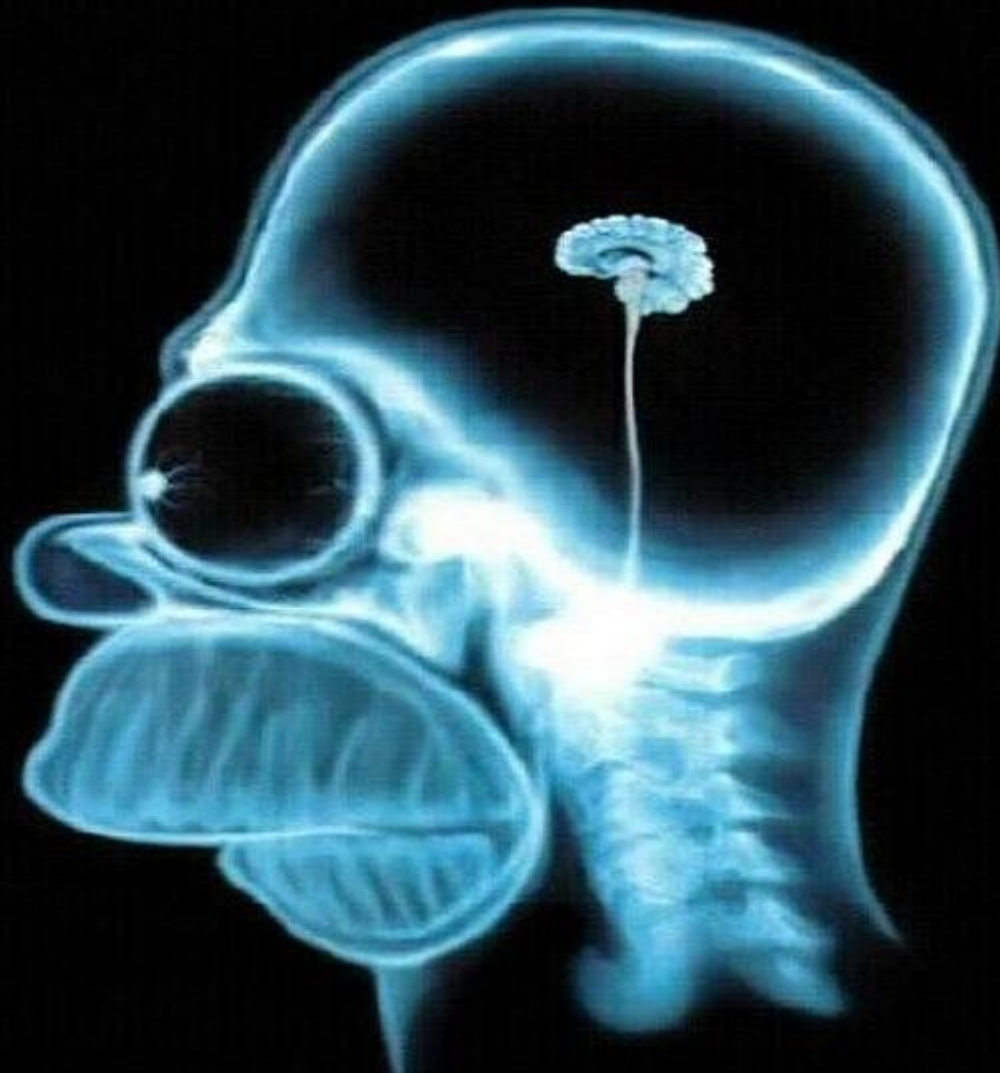
devops believer

To start with...

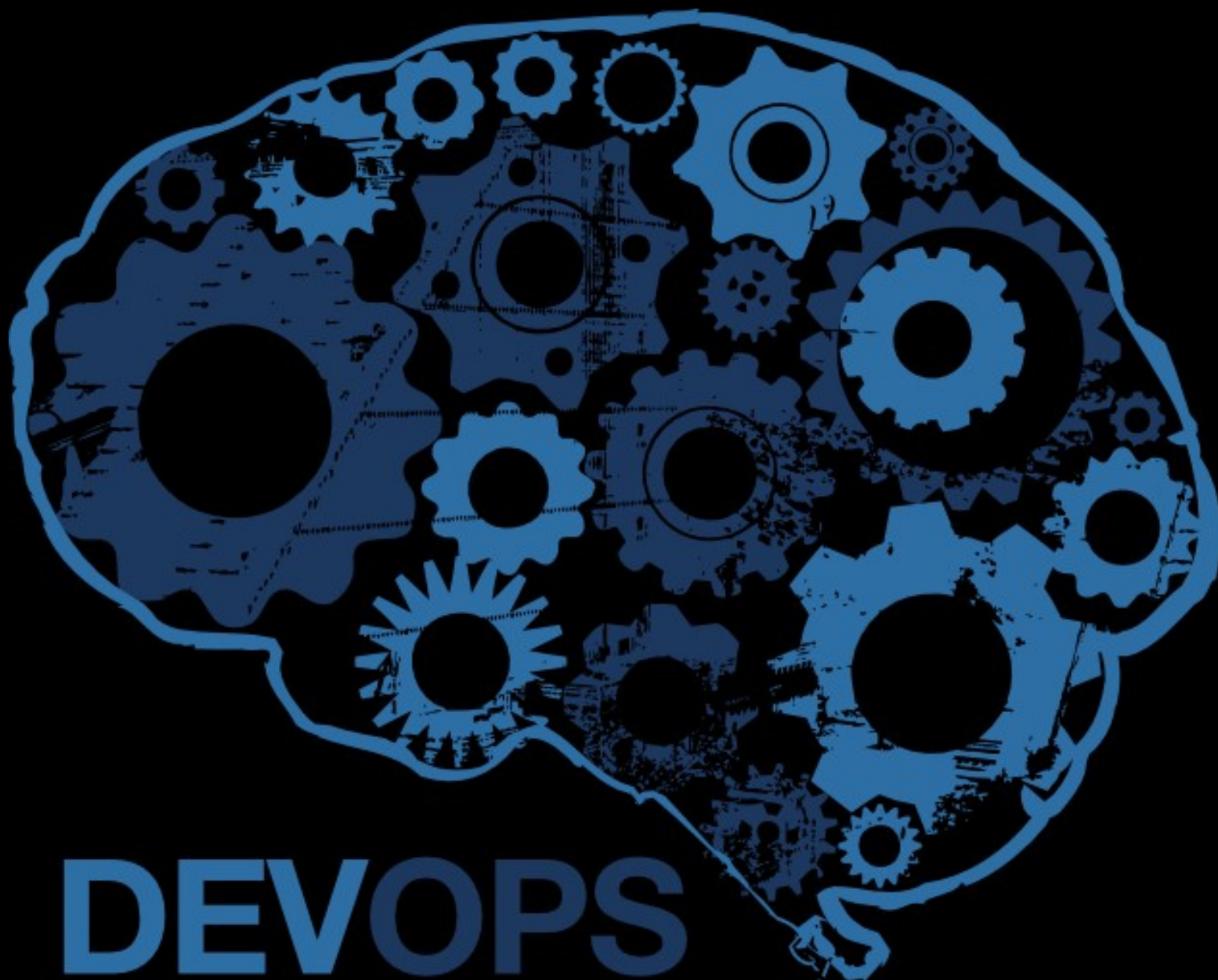
Let's use
devops like
slides...

Let's use devops
like slides...

from



to



devops

is a movement willing to help
aligning IT with company
needs

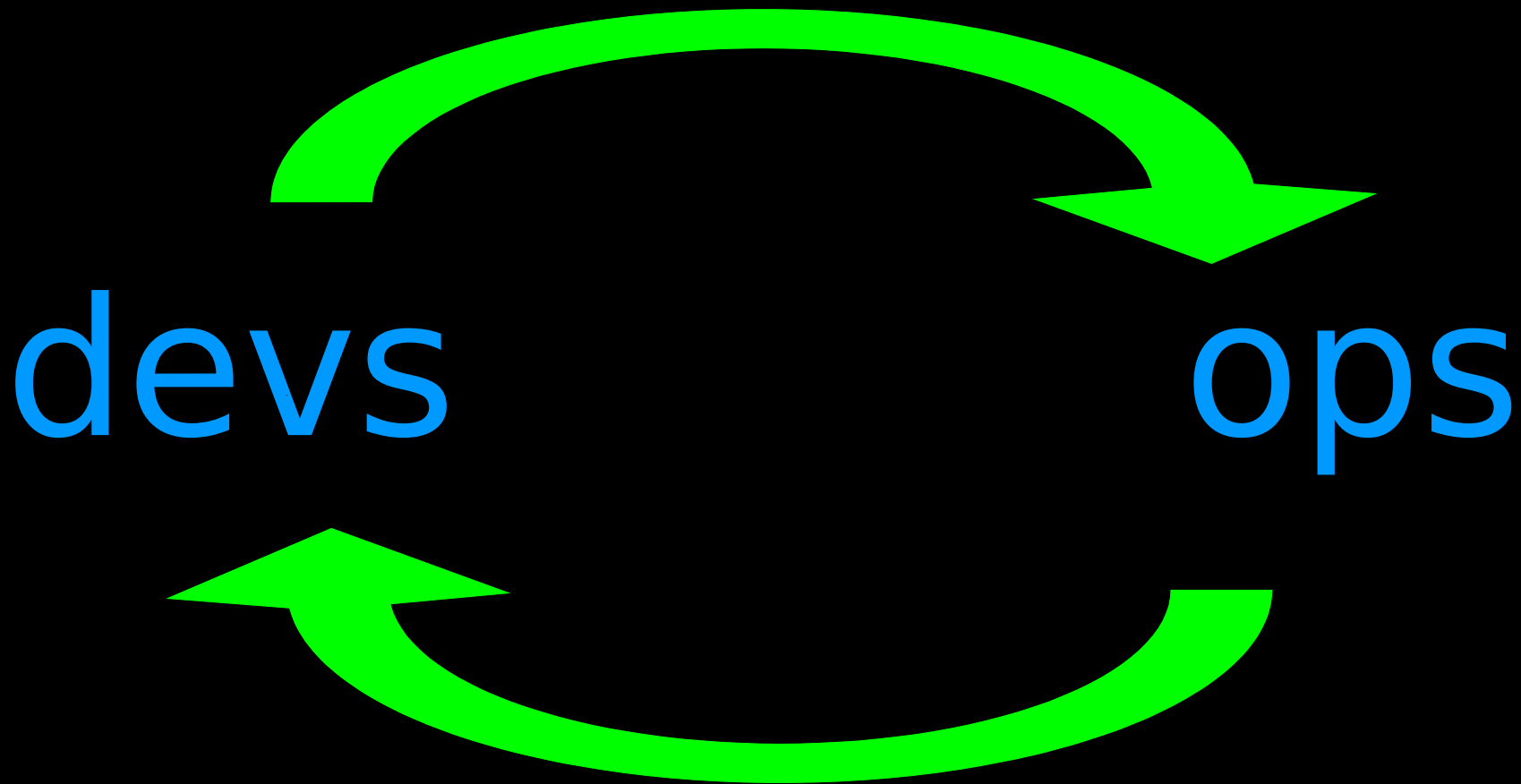
(starting by aligning devs and ops)

devops is about
communication !

devops is about
communication !

between
dev and ops

Small and frequent
successful iterations



This is one of the most fundamental ideas in devops:
the **feedback loop** !

We can say that
devops is bringing
together **people** and
practices from
development and
operations

I am a happy **op** !

I am a happy **dev** !



we can say that
development is bringing
together people and
product and
development and
operation

devops now

=

- being professional
- improving the collaboration by increasing trust and respect

/me likes !

HOW can I reach that
goal ?

devops
requires
visibility !

Everybody should have
a status **overview**

Is there any problem ?
How are the test going ?
How is performance ?

devops

is about

CAMS

CULTURE AUTOMATION MEASUREMENT SHARING

@botchagalupe @damonedwards

<http://www.opscode.com/blog/2010/07/16/what-devops-means-to-me/>

What
techniques
can I use ?

Monitoring Infrastructure as Code

Continuous
Improvement

What about **DBAs** ?



Join the team,
follow the
movement !

You are already
between devs
and ops

I am a happy **dba** !

You are already
better than
VS



But we are still
closer to the
dark side of the
force: sysadmins

But we're still
close to the
dark force
mins



How can we
apply **devops**
ideas to our **dba**
job ?

1. communication
2. configuration
management
3. testing
4. visibility

1. Communication

Usually **DEVS**
communicate
with managers,
quality managers,
products managers, ...



OPS

“communicate”
with support,
ISP's and... their
servers



DBAs don't communicate



The **interaction** between
developers, sysadmins,
dbas and all the actors
should begin from the **start**
of any projects

THINK

**Do it Right
The First
Time!**

Plan Ahead

DESIGN KIMBERLY BRADY.COM Y57017

1 800 800 800 800 800 800 800 800

**and do it
in team !**



Discuss everything **database**
related with **developers** and
sysadmins:

- *disks and raid design*
- *schema design & partitions*
- *additional columns & indexes*
- *tables repair, ...*

2. configuration management

Can we **automate** DBAs
job ?

Should we ?

Even if we have only **one**
database ?

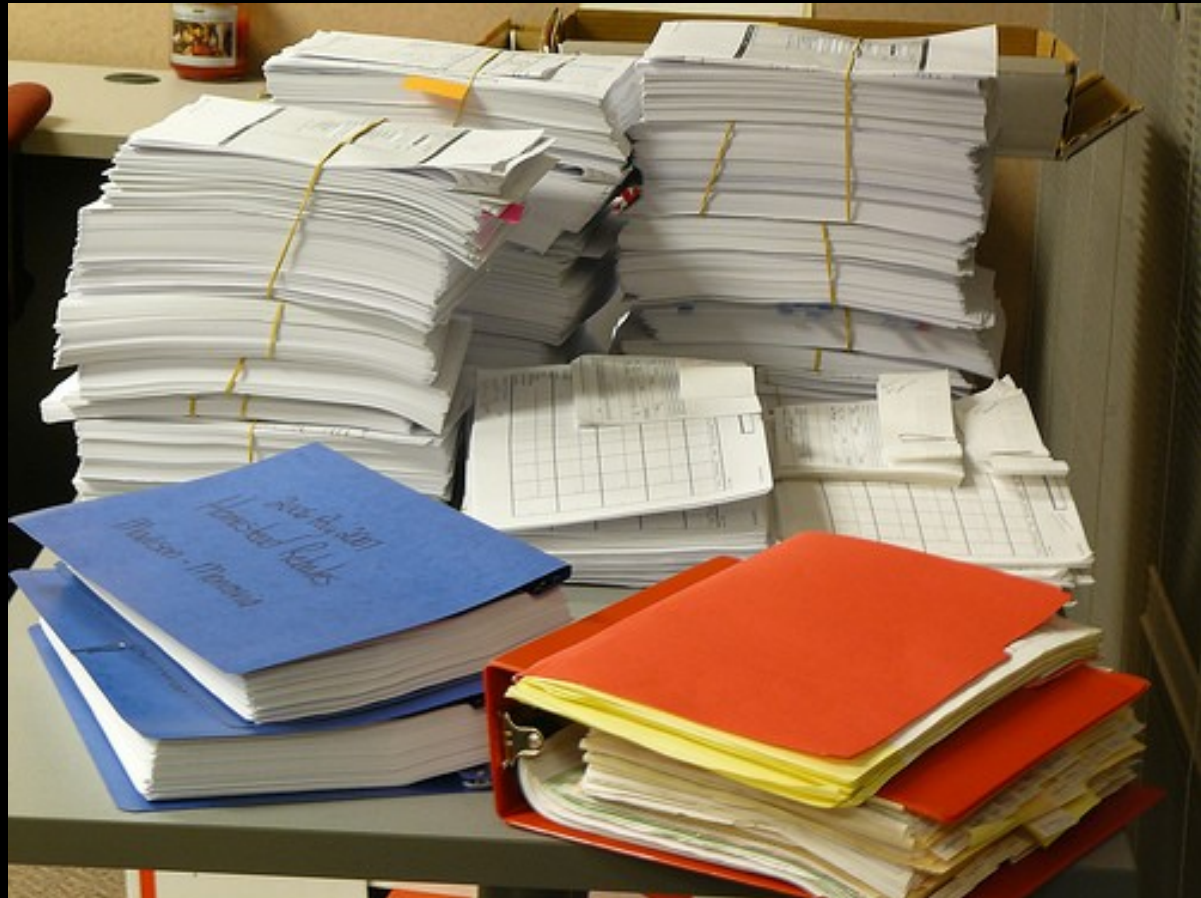




A lot of things to configure

- *binaries*
- *databases*
- *configs*
- *cron jobs*
- *monitoring*
- *grants*
- *replication*

**Documentation is
needed and must be up
to date**

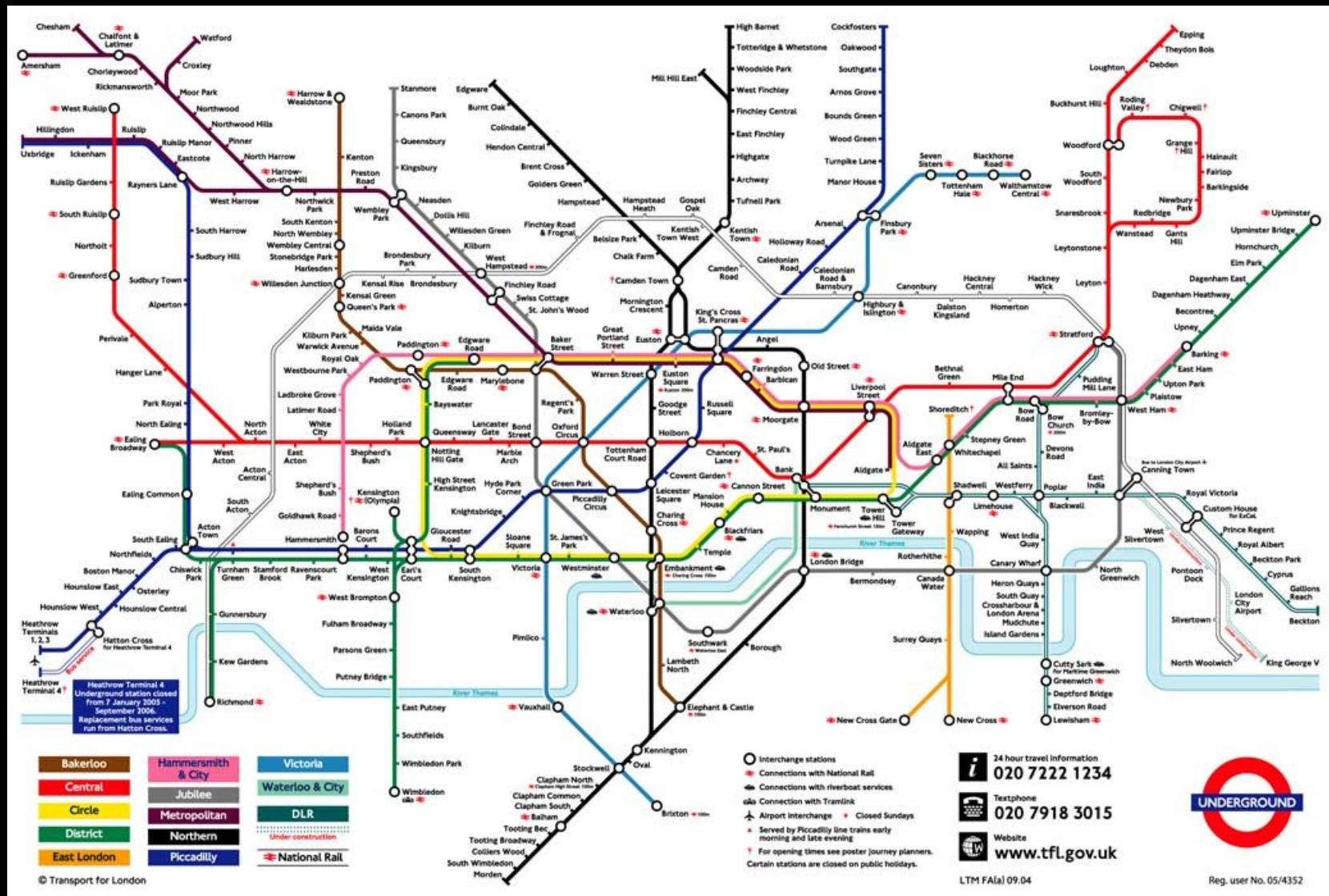


Needs to be **reproducible**



quickly **without** missing
anything !

Keep track of any changes



Mind the gap

3. testing

Test

every changes

every new package...

EVERYTHING

4. visibility

Expose monitoring, trending,
test results and project status
to **EVERYONE** in your
company!!

*Everybody should know why the
database is sloooooow !*

Use all these metrics to plan ahead all the changes !



NOT ALONE BUT WITH EVERYBODY

notions

practices

ideas

methodologies

We are **DBAs** we
work with **data**,
we need
something more
concrete !

We
we
data
some
o



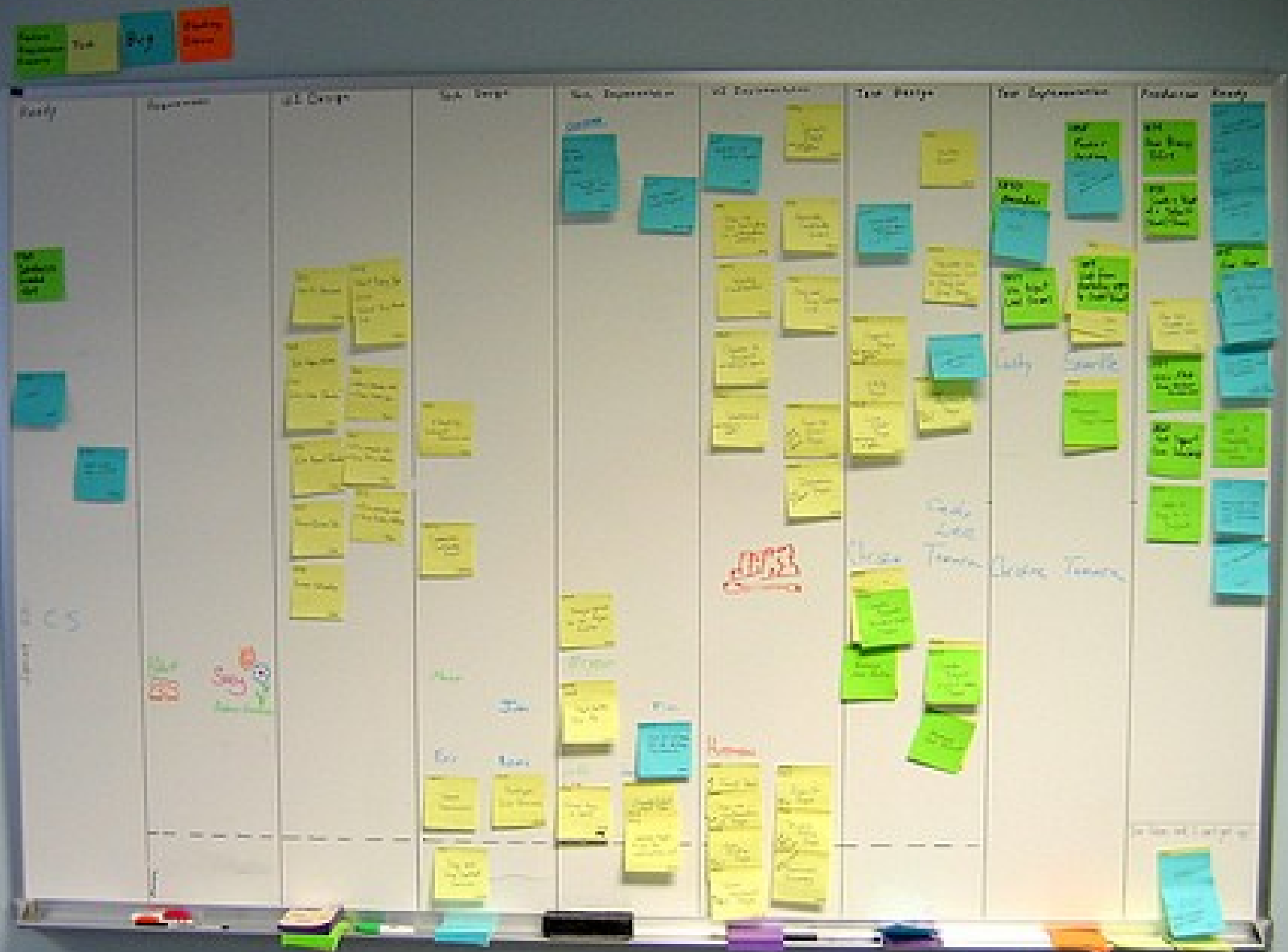
's
h
ed
ore







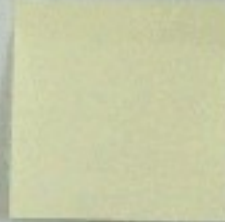

TRUE !

communication

Use an **agile**
method to
handle your
projects

For operations a
good method is
kanban



analyze 2	design 3	test 2	build 1	deploy 1
				
				
				

Use dashboards
and make them
available to
EVERYONE

share all your
dashboards with
everybody in
your company

**configuration
management**

consider your
infrastructure as
code

Use **puppet** (or any other
configuration management tool)
to **deploy**
everything related
to your **database**



Save your
puppet recipes
into a version
control system

but also save
your database
schema changes
in that system

test

TEST DRIVEN DESIGN?
I TRIED, BUT I DIDN'T FIND ANY TEST
THEY DID NOT FAIL.



What should you
test as a devops
dba ?

Test if you can
deploy
everything and if
you didn't forget
anything

Test also the
performance !

Is it faster with
this change ?

use

**Percona
Toolkit**

automate
your
tests

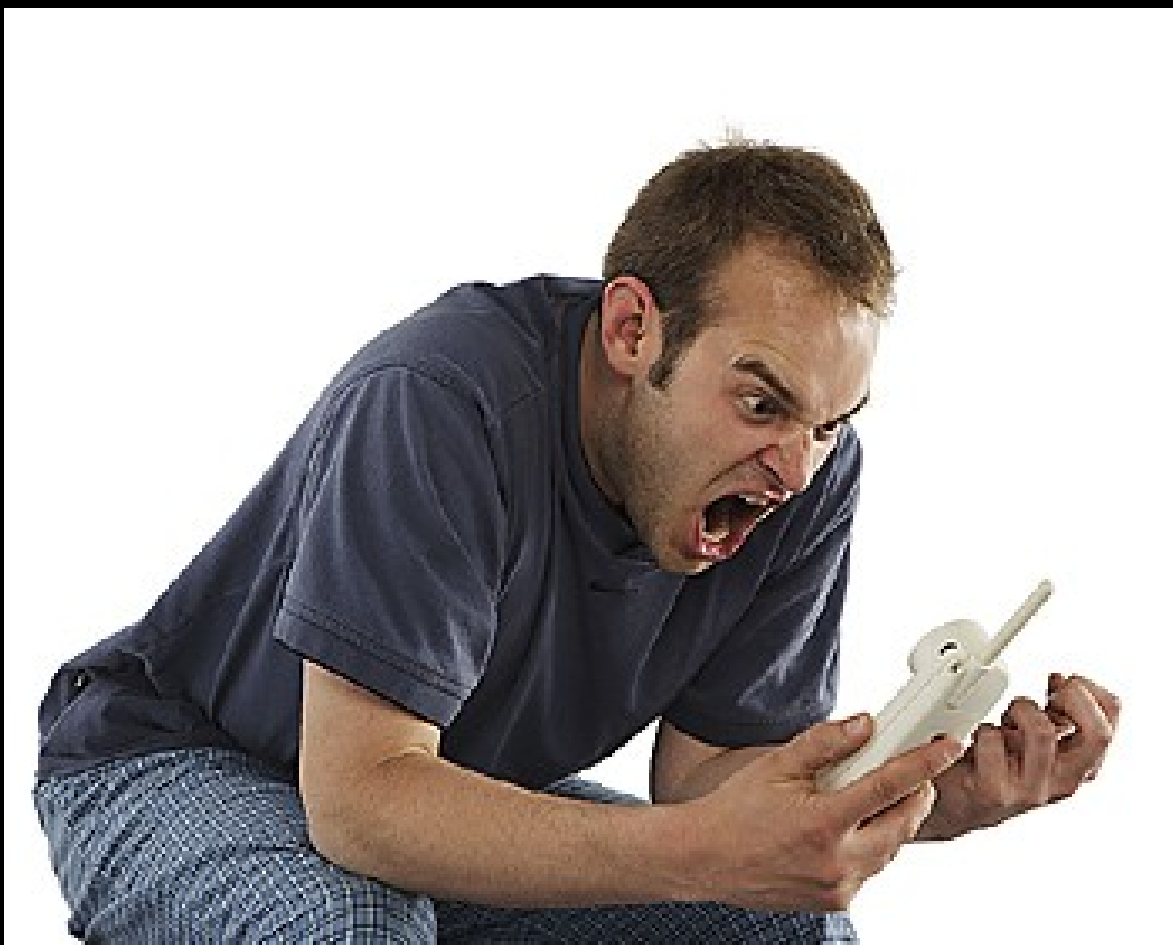
visibility

monitoring

Monitor your database
with an alerting system
(nagios).

Disk/IO, CPU load, Network
traffic, disk usage

But do you want to be
woken up at night
because replication of
the machine taking
backups is 60sec
behind ?



NO!

Then do you want to be
woken up in the middle
of the night because
they are 1000 more
selects than usual ?



NO!

Your **alerting** should answer those questions:

Is the database server running ?

Does it allow connections ?

Is replication lagging on production slaves ?

trending

The most **interesting** facts
for a **DBA** are the **trending**
results !

Use **cacti** ... with Percona
Monitoring Plugins
They provide very **good**
metrics

Correlate changes in
the trending results
with your configuration
changes...

**everything should be
tracked**

others ?

You can also
use other
“devops”
oriented **tools**
like :



puppet-dashboard



FOREMAN



REDMINE

with kanban plugin



If there is only
one thing you
remember from
this presentation

Yes, even a
schema change
MUST be
considered like a
migration !

review it

test it

plan it

do it

and now that we
have a
devdbaops
team

we can start
working on
really interesting
tasks for
everyone

the
smooth
migration

When this goal is achieved

DEVS have less pressure

OPS have less pressure

DBAs have less pressure

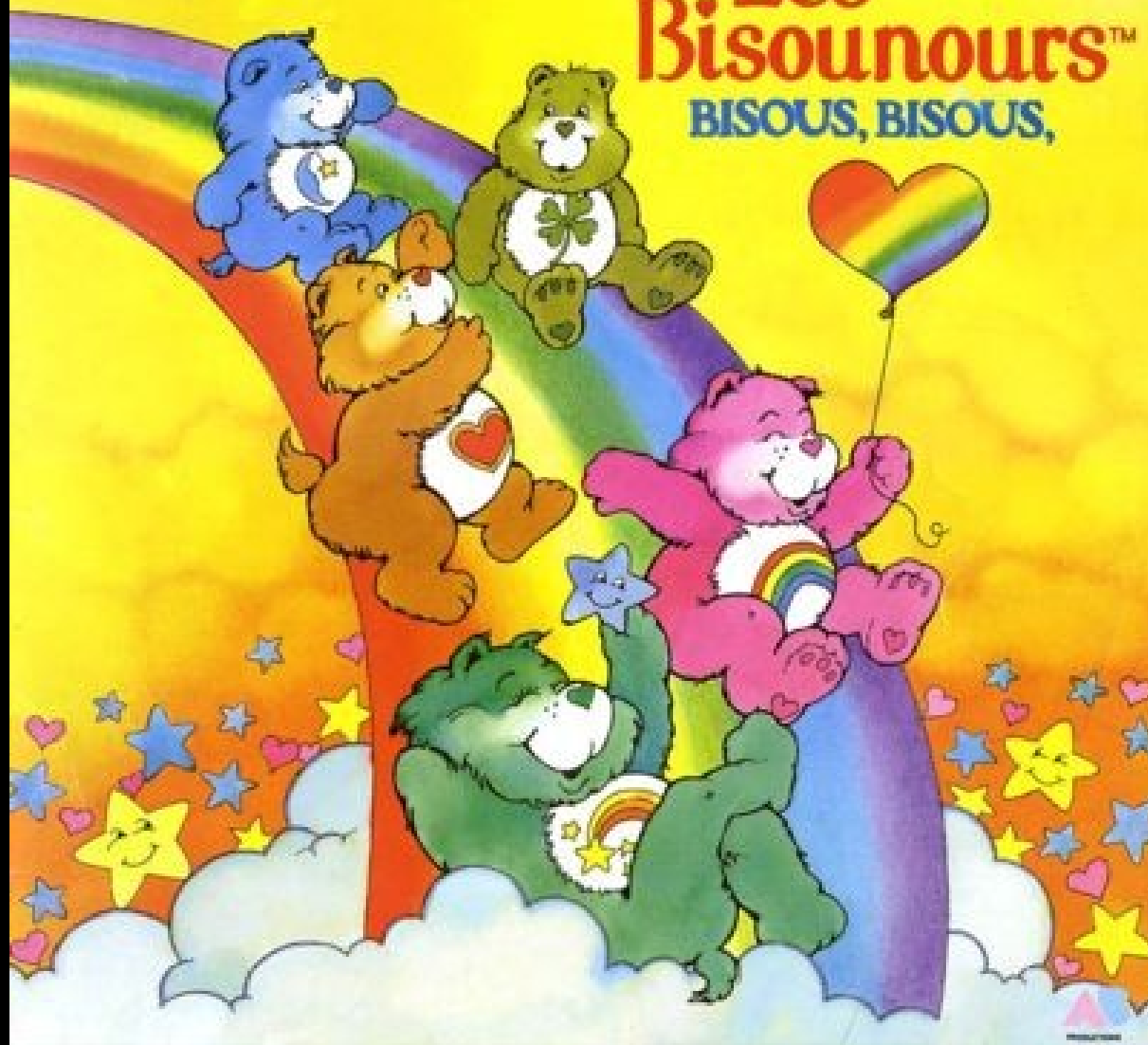
collaboration

=

less pressure

Les Bisounours™

BISOUS, BISOUS,



present

future

application

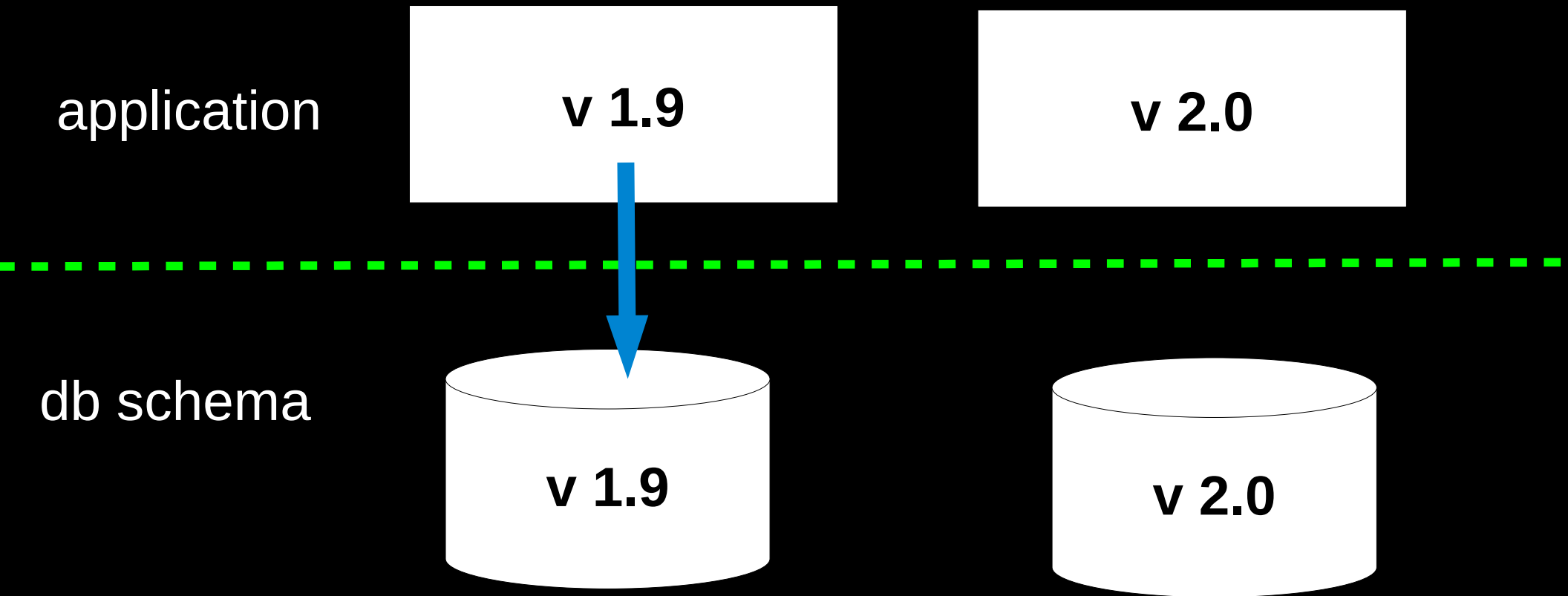
v 1.9

v 2.0

db schema

v 1.9

v 2.0



present

future

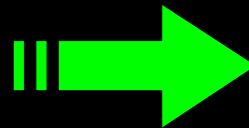
application

v 1.9

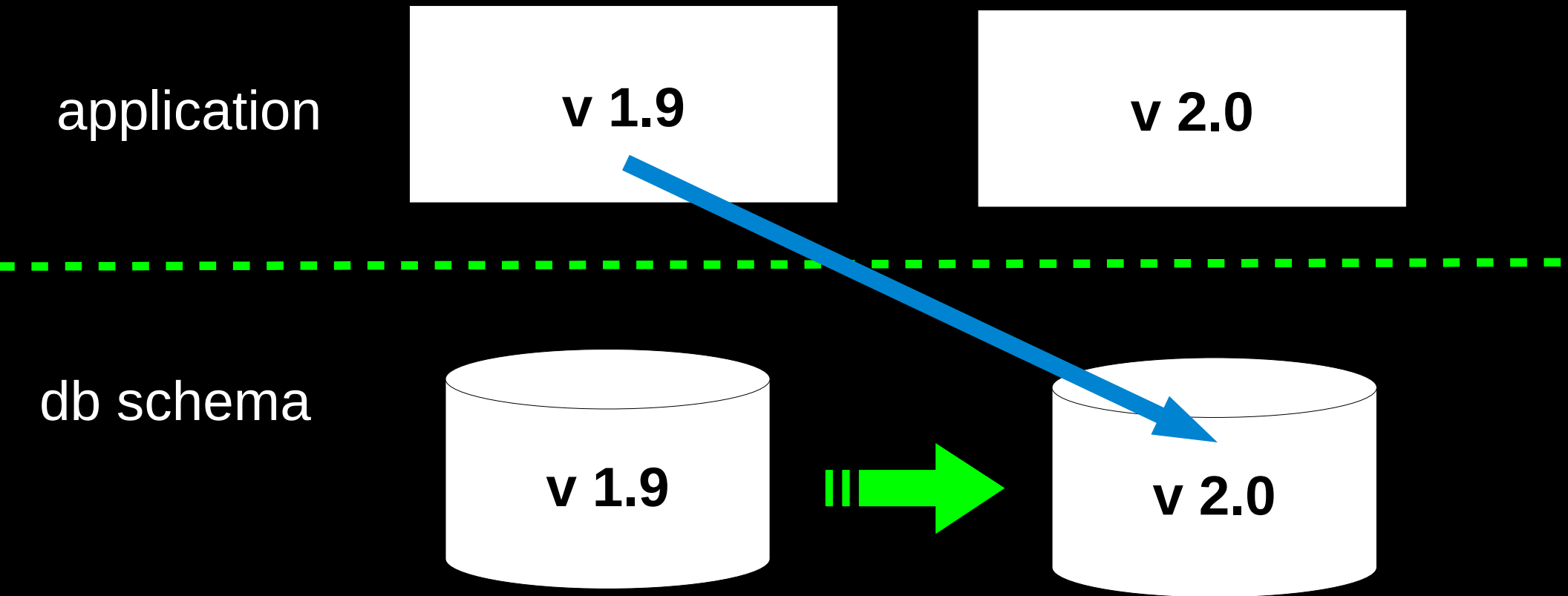
v 2.0

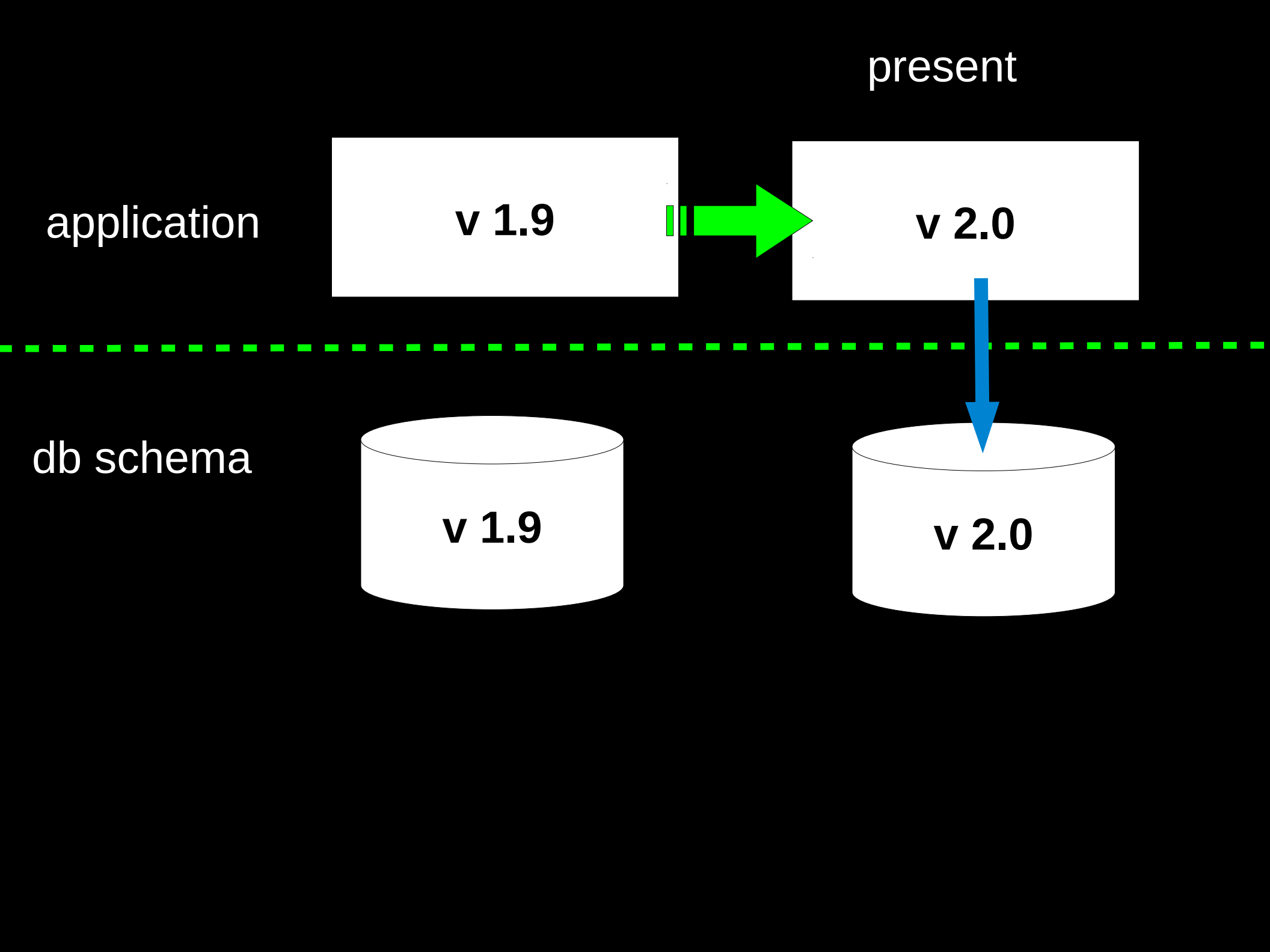
db schema

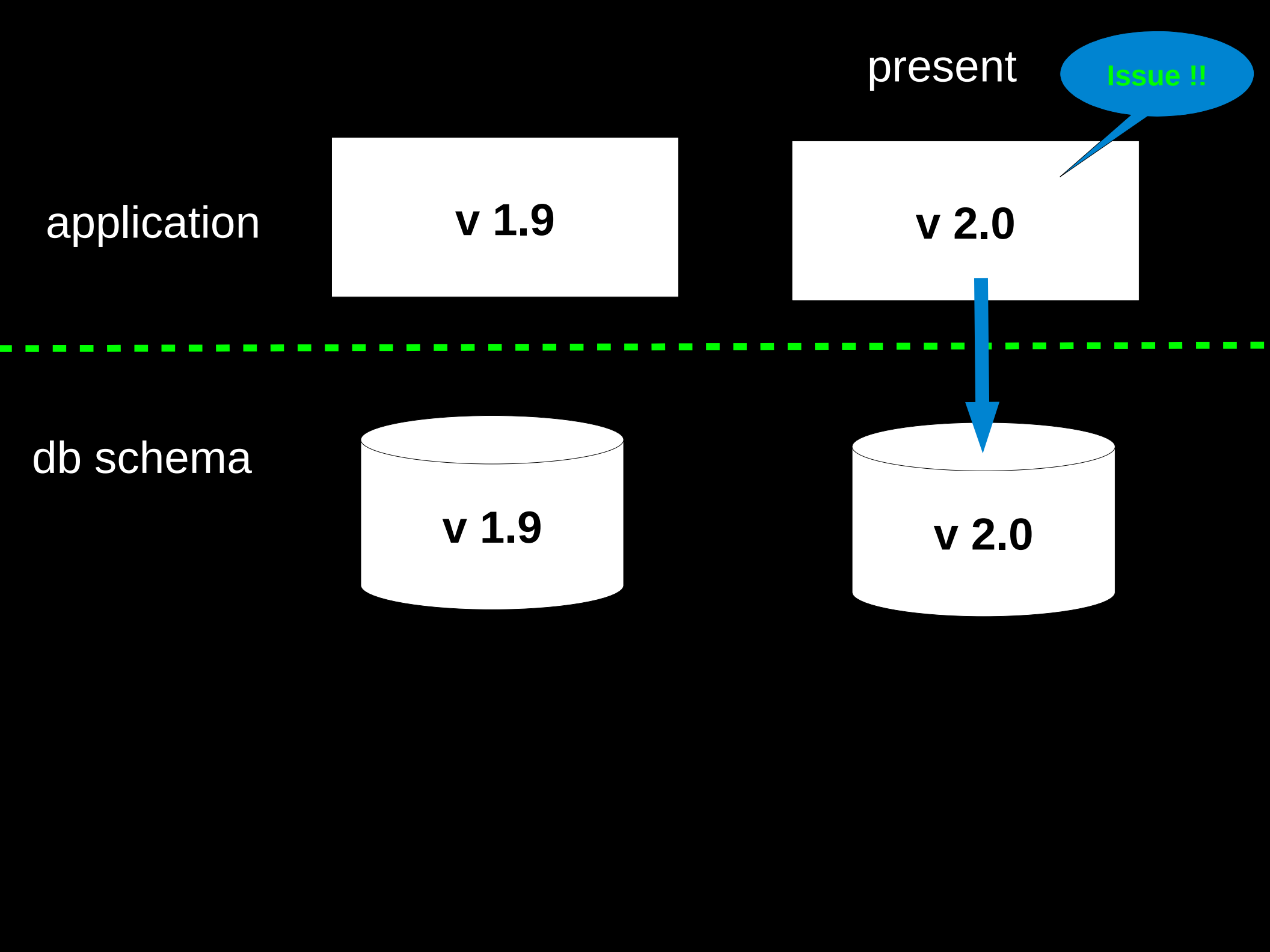
v 1.9



v 2.0



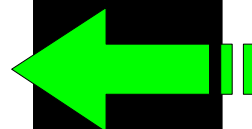




application

v 1.9

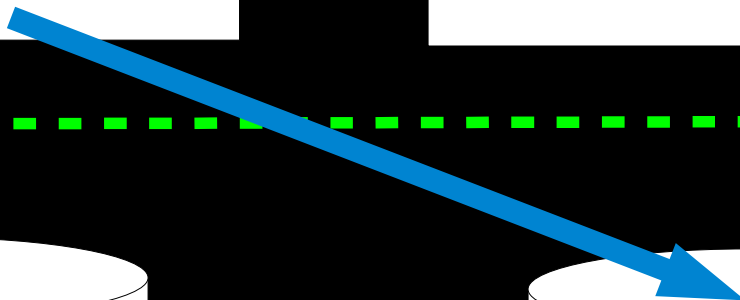
v 2.0



db schema

v 1.9

v 2.0



application

v 1.9

v 2.0

db schema

v 1.9

v 2.0

Issue !!

The diagram illustrates a version mismatch between an application and its database schema. A horizontal dashed green line separates the 'application' layer from the 'db schema' layer. In the application layer, there are two white rectangular boxes labeled 'v 1.9' and 'v 2.0'. In the db schema layer, there are two white cylindrical database icons labeled 'v 1.9' and 'v 2.0'. A blue arrow points from the 'v 2.0' application box down to the 'v 2.0' database cylinder. A blue speech bubble with the text 'Issue !!' points to the 'v 2.0' database cylinder, indicating a problem with the schema update.

application

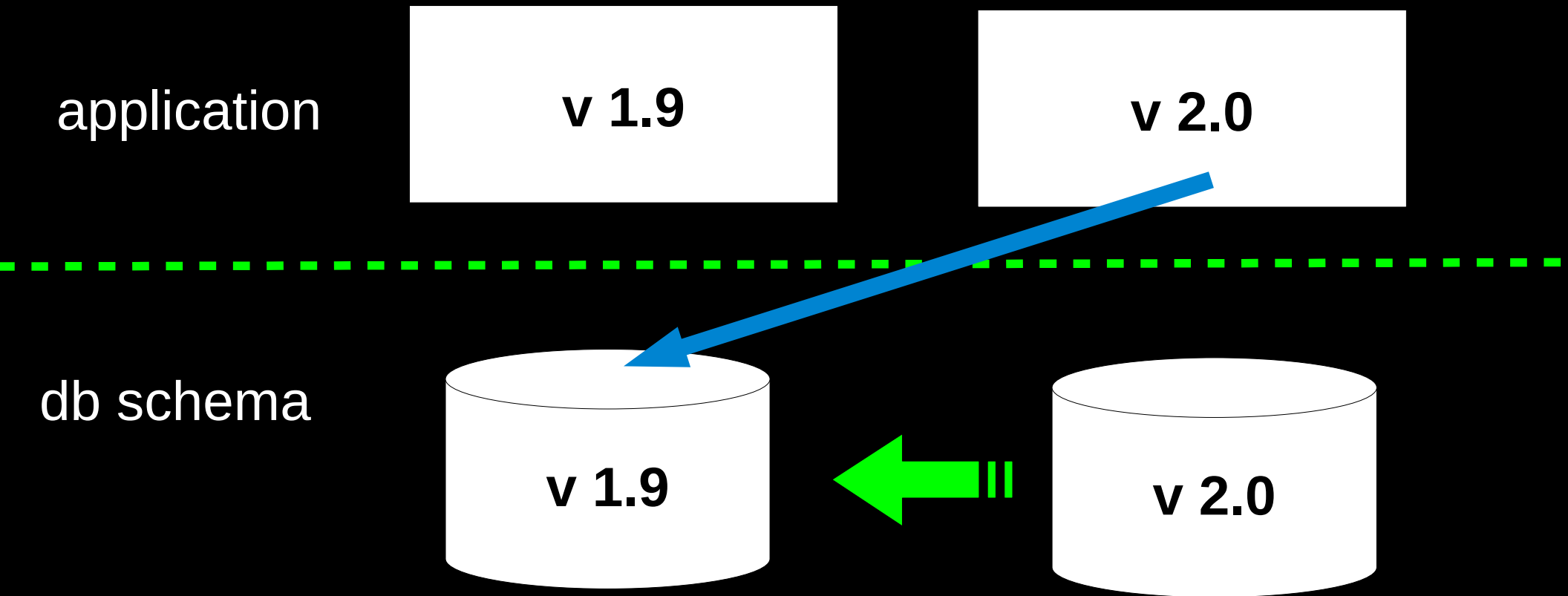
v 1.9

v 2.0

db schema

v 1.9

v 2.0



This can only be
the result of

collaboration
between **DEVs,**
DBAs & OPS

demo

git

puppet

Percona server

Jenkins

VirtualBox

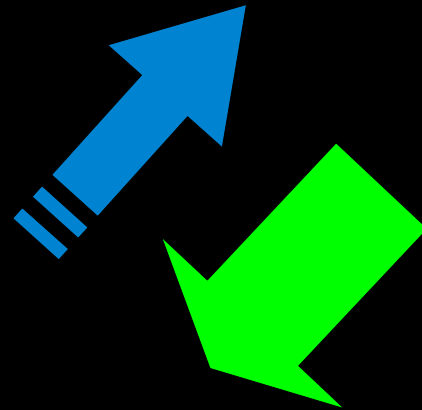
vagrant

mcollective

Change the schema
(add an index for example)

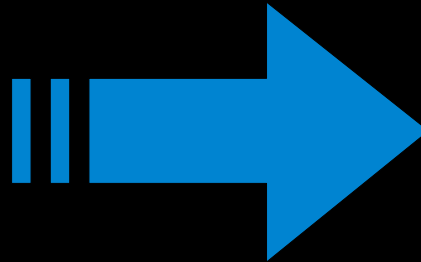
Commit !

+++ git





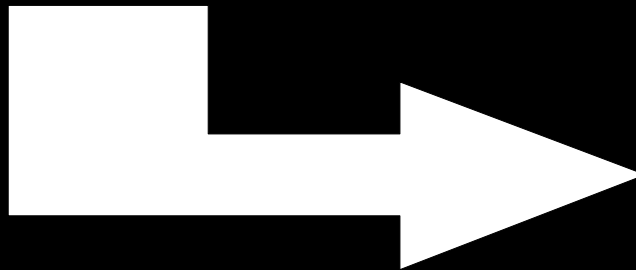
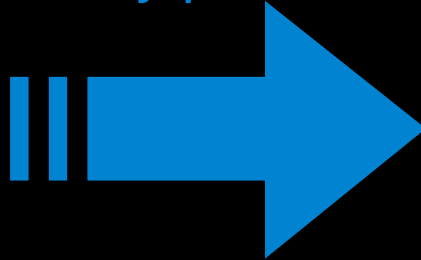
Launches a new
virtual machine from
scratch using vagrant



Install everything and
populate a test db



Start some basic
tests with cucumber-
puppet to see if all
puppet code was
correctly performed

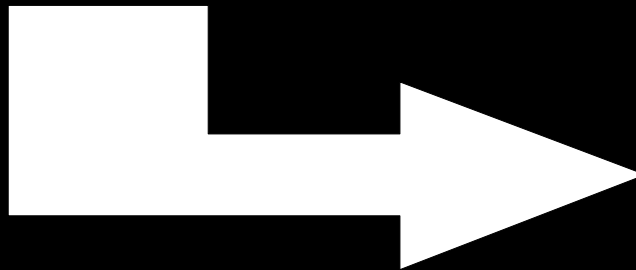
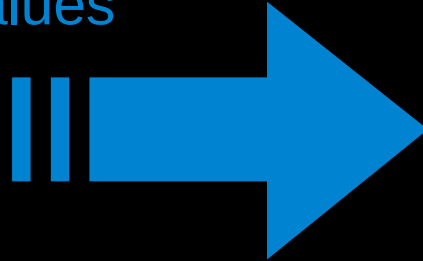


No error is a success

Otherwise is a fail



Then start some tests
with percona-
playback and/or
pt-query-digest and
compares to old
values



Faster is a success

Slower is a fail

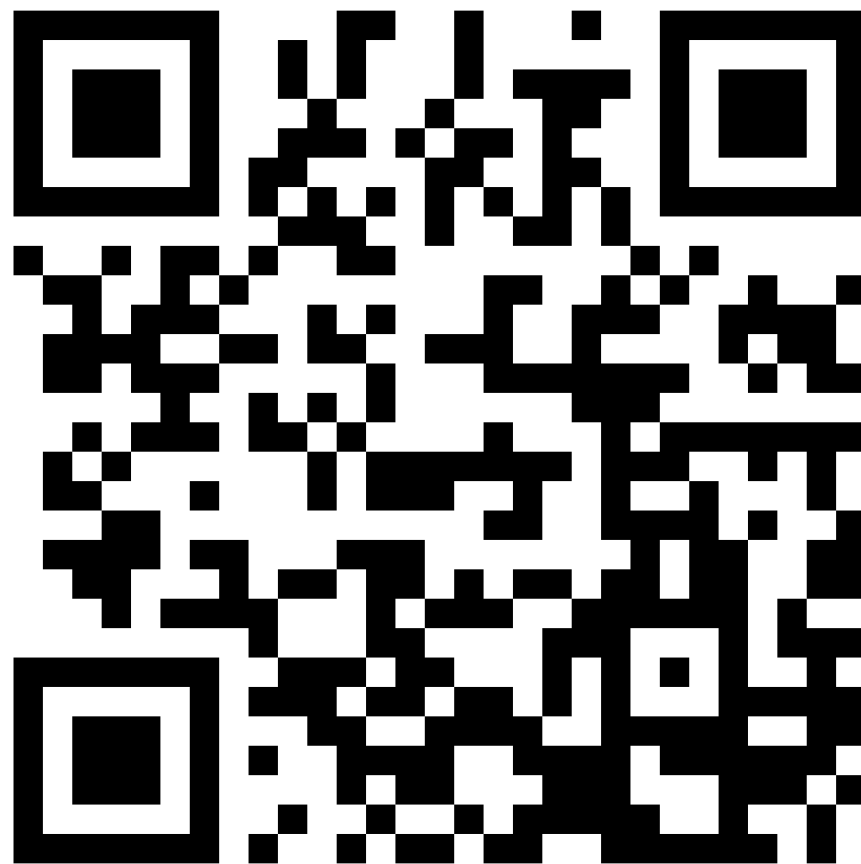
Projects links:

<http://puppetlabs.com/resources/overview-2/>
<http://puppetlabs.com/puppet/related-projects/dashboard/>
<http://theforeman.org/>
<http://git-scm.com/>
<http://www.redmine.org/>
<http://www.redmine.org/projects/redmine/wiki/PluginKanban>
<http://vagrantup.com>
<https://www.virtualbox.org/>
<http://jenkins-ci.org/>
<http://www.percona.com/software/percona-toolkit/>
<http://code.google.com/p/mysql-cacti-templates/>

Photo Credit:

<http://www.flickr.com/photos/kuma-kigurumi/>
<http://www.flickr.com/photos/airforceone/2125791018/>
<http://www.flickr.com/photos/akosvaradi/2617404213/>
<http://www.flickr.com/photos/doberdad/2264130166>
<http://www.flickr.com/photos/tomsaint/2714401733>
<http://www.flickr.com/photos/10335017@N07/4570943373/>
http://www.bxel.net/metro-de-bruxelles/metro_plan_bruxelles.jpg
<http://www.flickr.com/photos/youngkimstudio/6216328782>
<http://www.flickr.com/photos/26676383@N00/3690701110/>
<http://www.flickr.com/photos/blubbla/209864856>
<http://www.flickr.com/photos/chrishuffman/2336990347>
<http://geekandpoke.typepad.com/.a/6a00d8341d3df553ef0153922fa2ee970b-pi>

You can find all these recipes here !



Speaking at



In London, December 3rd & 4th

Speaking at

40% off passes
using the discount
code
“MySQL-UK”



3rd & 4th



lefred@percona.com
@lefred

