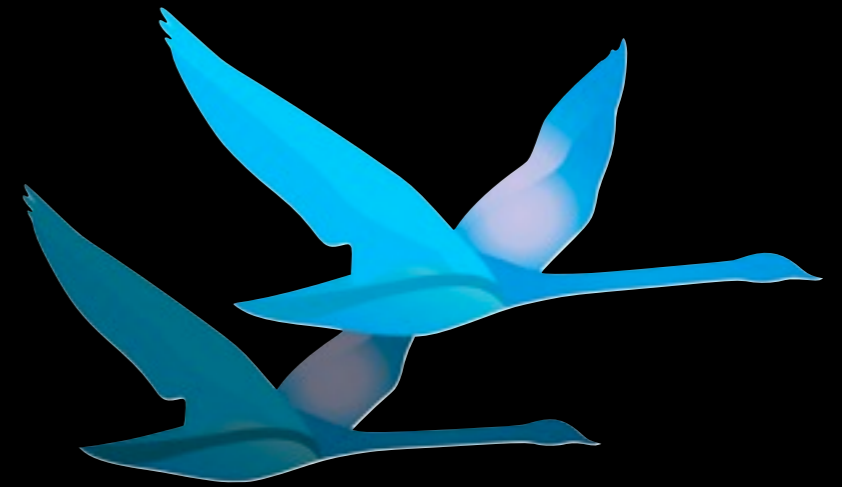


# The art of sandboxing

Reducing Complex Systems to  
Manageable Boxes



Giuseppe Maxia, QA Director, Continuent, Inc

**Facts. And Demos. Possibly fun**

# about me - Giuseppe Maxia

- a.k.a. The Data Charmer
- QA Director at Continuent, Inc
- Long time hacking with MySQL features
- Formerly, community manager, db consultant, designer, coder.
- A passion for QA and open source
- Blogger
- <http://datacharmer.blogspot.com>



# What is this all about

- Sandboxes vs. VMs
- MySQL Sandbox: a toolkit for laziness
- Genesis of a sandbox
  - Understand the system
  - Find the boundaries
  - Tame the system
- Make it usable
- Make it extendable
- Tricks of the trade

AGENDA

# Sandboxes and VMs

# public Sandbox definition

- *A sandbox is a testing environment that isolates untested code changes and outright experimentation from the production environment or repository.*
- *Sandbox is a security mechanism for separating running programs. It is often used to execute untested code, or untrusted programs from unverified third-parties, suppliers and untrusted users.*

*(from Wikipedia)*

# my Sandbox definition

- Sandbox is a testing environment that:
  - isolates a given running program
  - avoids conflicts with similar programs
  - makes the program easy to install and use

# Why not a VM?

- VMs are great.
- But expensive. They use RAM and storage.
- You can use a dozen sandboxes in your laptop. Not VMs.

# Again, what's a sandbox?

- Installation tamer:
  - Take a program, which is designed to run in a separate host
  - Make it behave, so it runs with more of its kind in the same host
- Usage simplifier:
  - provide shortcuts to make that program easy to use

A popular example

# MySQL Sandbox, A toolkit for productive laziness



# Laziness

Laziness is a disinclination to activity or exertion despite having the ability to do so.



<http://en.wikipedia.org/wiki/Laziness>

# I am an experienced DBA

- I have **the ability** of installing multiple MySQL servers.
- In the same host.
- Without conflicting.
- Manually.
- Do I feel **inclined to do so**?
- Several times a day?
- I DON'T THINK SO.

# I am a command line wizard

- **After installing multiple servers**
- **I can use them**
- **with various long options.**
- **Manually.**
- **Do I feel inclined to do so?**
- **Many dozen times a day?**
- **I DEFINITELY DON'T THINK SO.**

# I can set up replication

- **Almost without errors.**
- **And then I can connect to masters and slaves with long options on the command line.**
- **Do I feel **inclined to do so?****
- **I DON'T THINK SO.**

# I can set up circular replication

- Almost always with errors.
- And cursing.
- Do I feel **inclined to do so?**
- I DON'T THINK SO.

# I can install plugins

- **After reading the manual.**
- **And translating from intentions to reality.**
- **And copying-and-pasting.**
- **Do I feel inclined to do so?**
- **I DON'T THINK SO.**

# I have the ability

- but I don't feel inclined to do repetitive work
- **I ADMIT IT: I AM LAZY**



**YES: I AM A  
LAZY  
DEVELOPER**



# **A lazy developer ®**

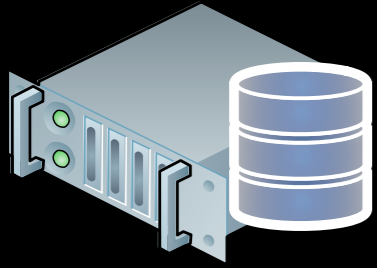
**Someone who writes 12,000 lines of code to spare himself (\*) the trouble of typing 15 lines on a terminal.**

**(\*) And another 1,000,000 people**

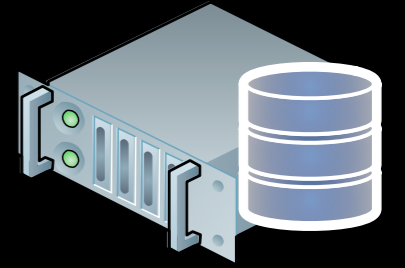
# That's me



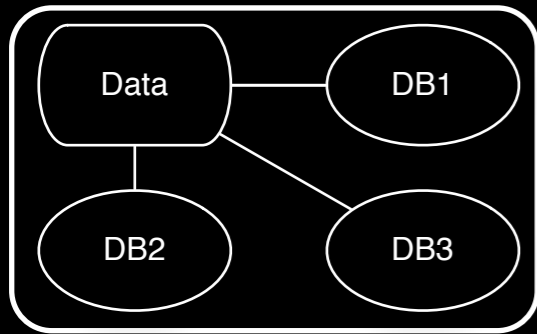
# overview



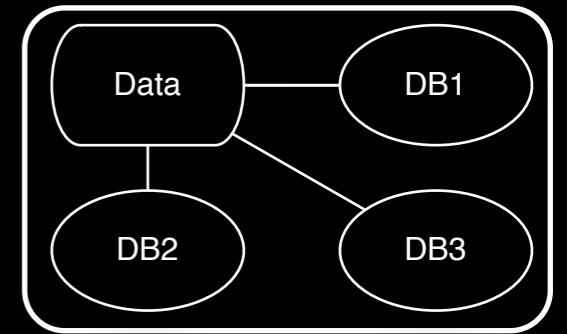
MySQL  
server



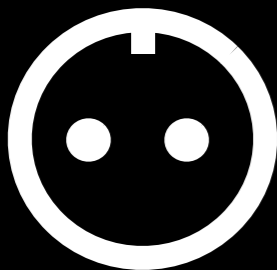
MySQL  
server



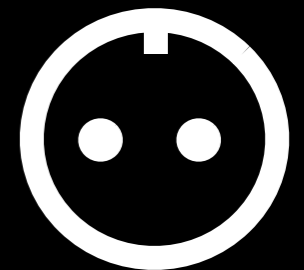
DATA DIRECTORY



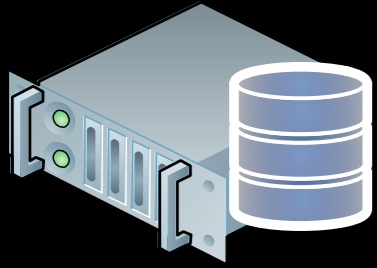
PORT



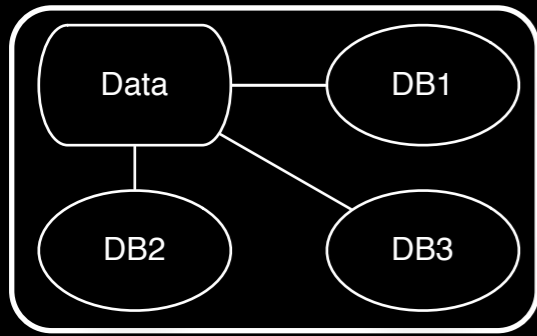
SOCKET



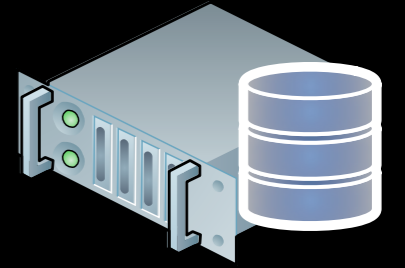
# overview



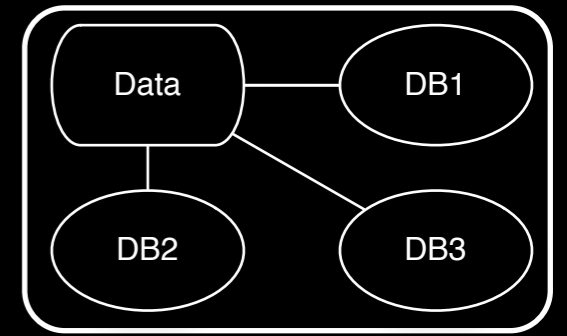
MySQL  
server



`/var/lib/mysql`



MySQL  
server

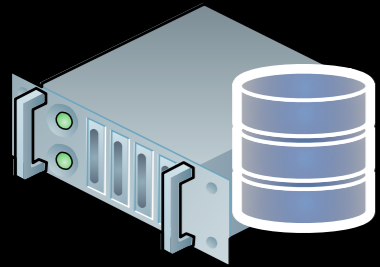


`/var/lib/mysql`

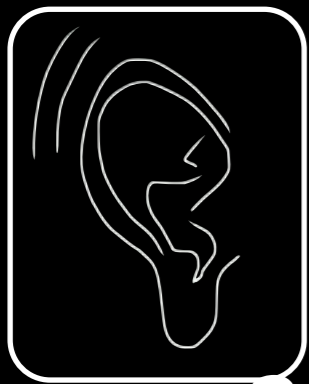


**DATA CORRUPTION**

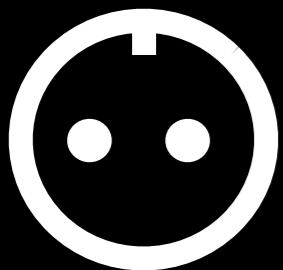
# overview



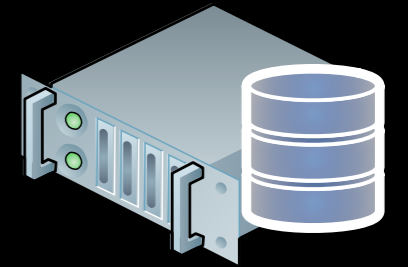
MySQL  
server



3306



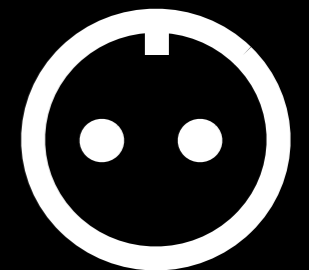
/tmp/mysql.sock



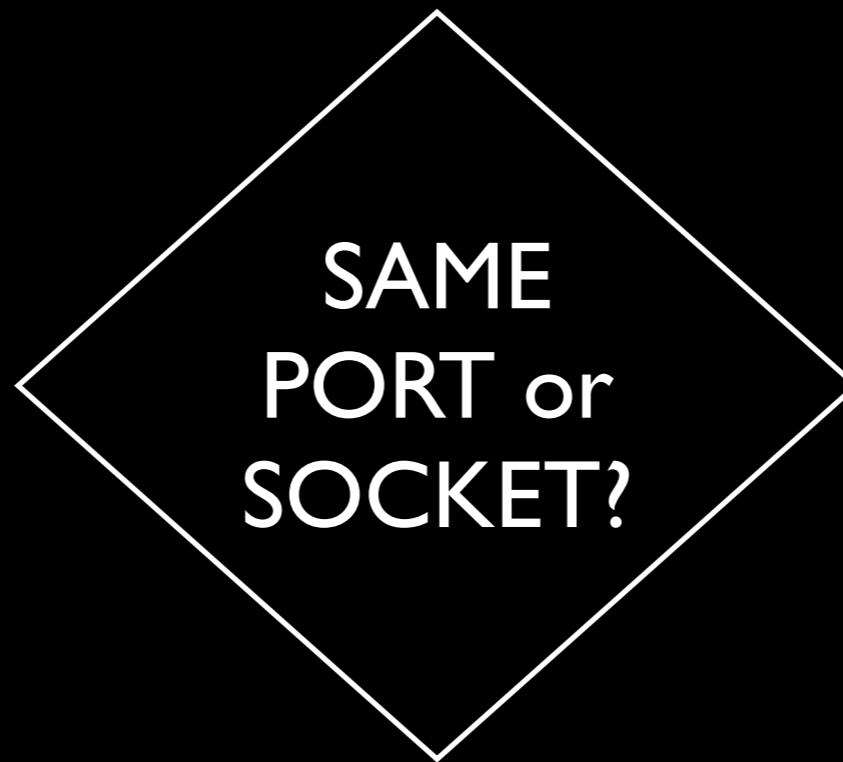
MySQL  
server



3306



/tmp/mysql.sock



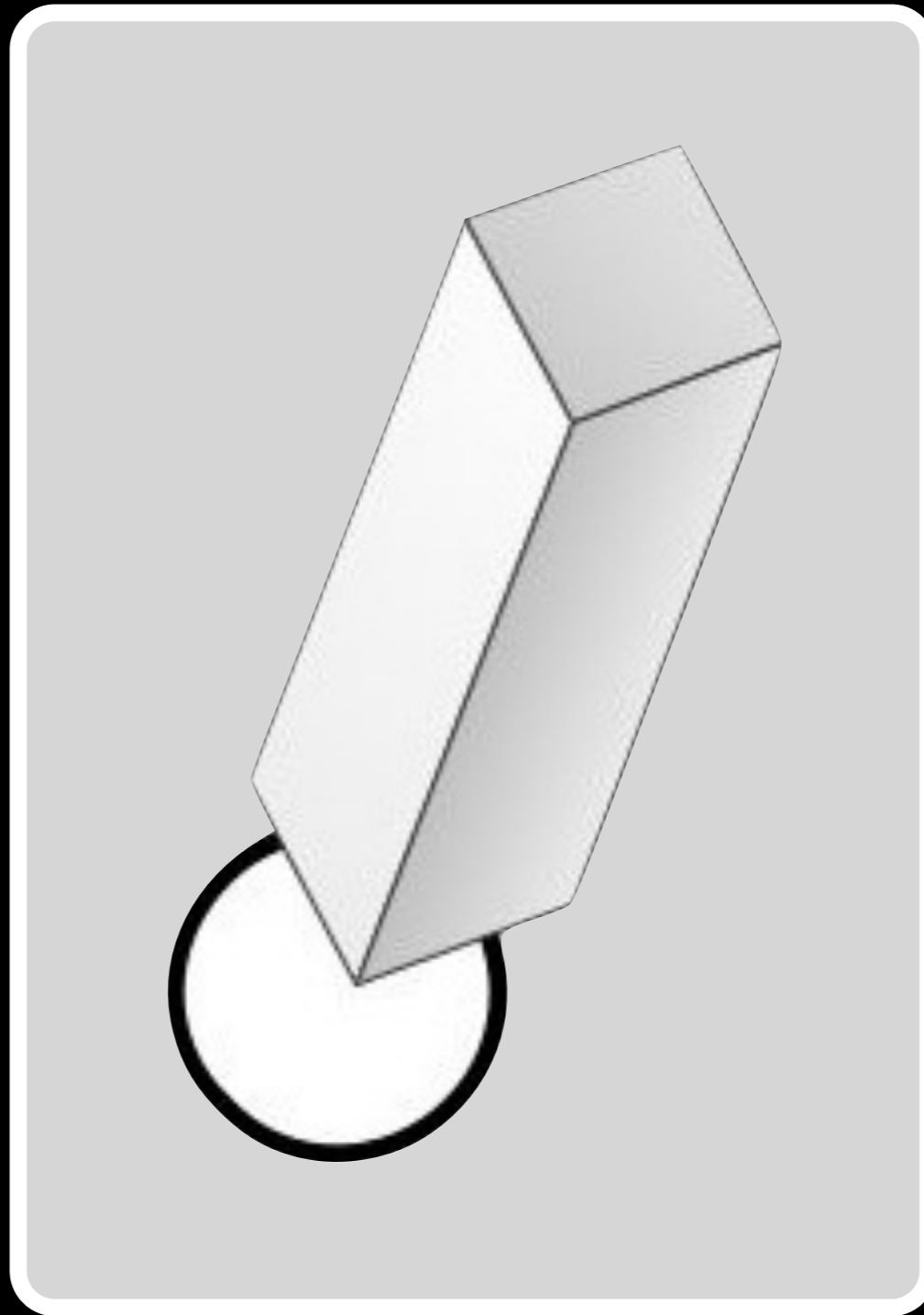
SAME  
PORT or  
SOCKET?

**DOES NOT START**

# The hard way (1)



# the hard way (2)



# The easy way

```
$ make_sandbox \  
    /path/to/mysql-5.1.54_linux.tar.gz
```

```
# it should work always
```

# The easier way

```
$ make_sandbox 5.1.54
```

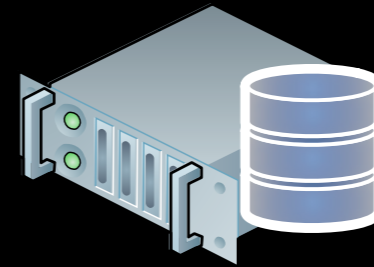
```
# Needs some preliminary work
```

# The easiest way

\$ sb 5.1.54

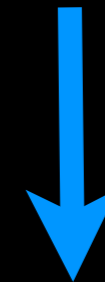
# Needs the same preliminary work

# MySQL Sandbox

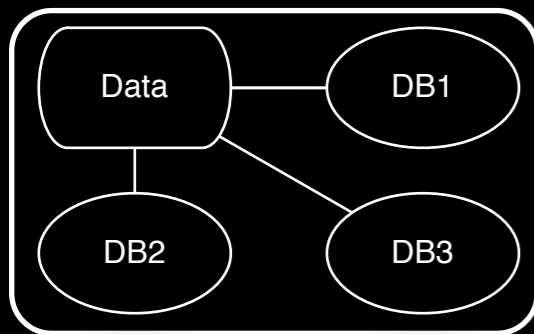


MySQL  
server

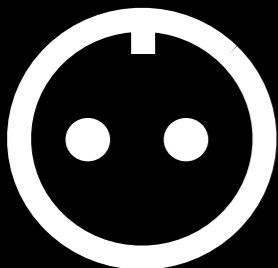
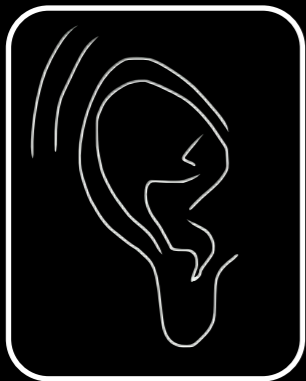
**VERSION**



`$SANDBOX_HOME/msb_`**VERSION**`/data`

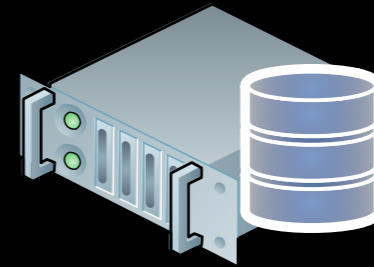


**VERSION**



`/tmp/mysql_`**VERSION**`.sock`

# MySQL Sandbox

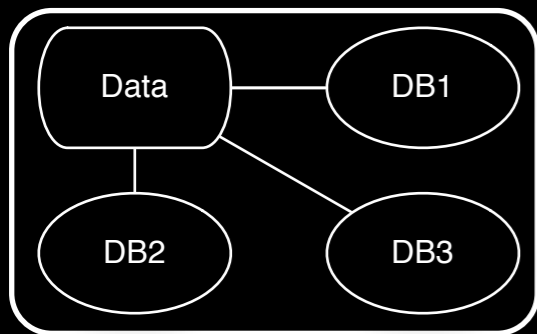


MySQL  
server

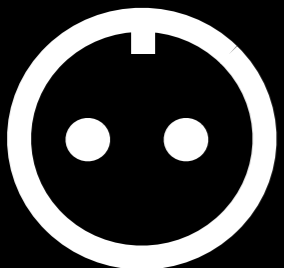
5.1.54



`$SANDBOX_HOME/msb_5_1_54/data`

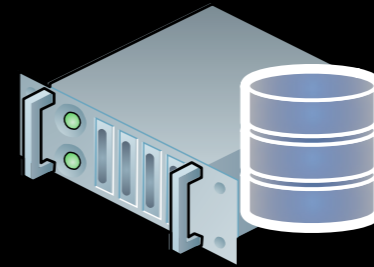


5154



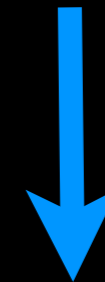
`/tmp/mysql_5154.sock`

# MySQL Sandbox

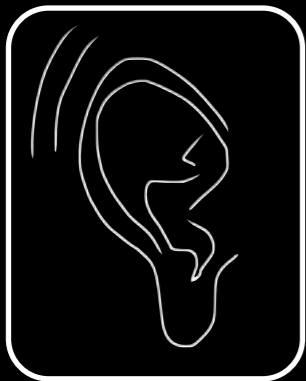
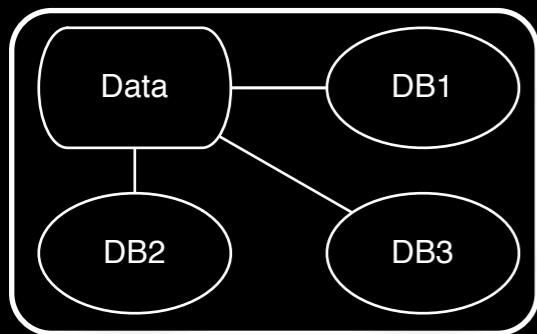


MySQL  
server

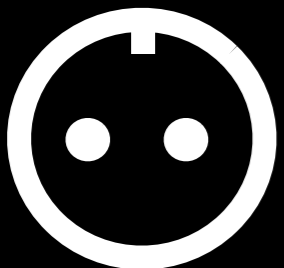
5.5.9



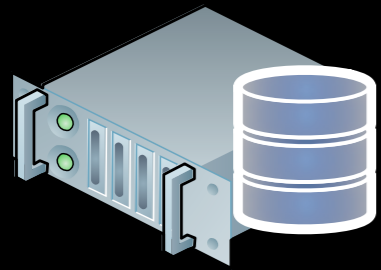
```
$SANDBOX_HOME/msb_5_5_09/data
```



5509



```
/tmp/mysql_5509.sock
```

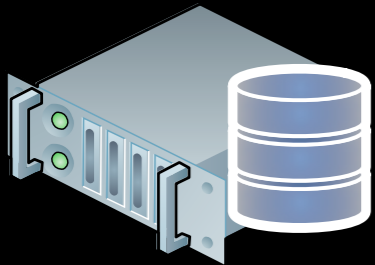


MySQL  
server

# Single Sandbox

customized scripts

```
start
stop
restart
status
clear
send_kill
_
use
```



MySQL  
server

# Multiple Sandbox

customized scripts

```
start_all
stop_all
restart_all
status_all
clear_all
send_kill_a
ll
use_all
```

m

s1

s2

n1

n2

n3



# Where do you get it

- from CPAN

```
sudo cpan MySQL::Sandbox
```

- from launchpad

```
http://launchpad.net/mysql-sandbox
```

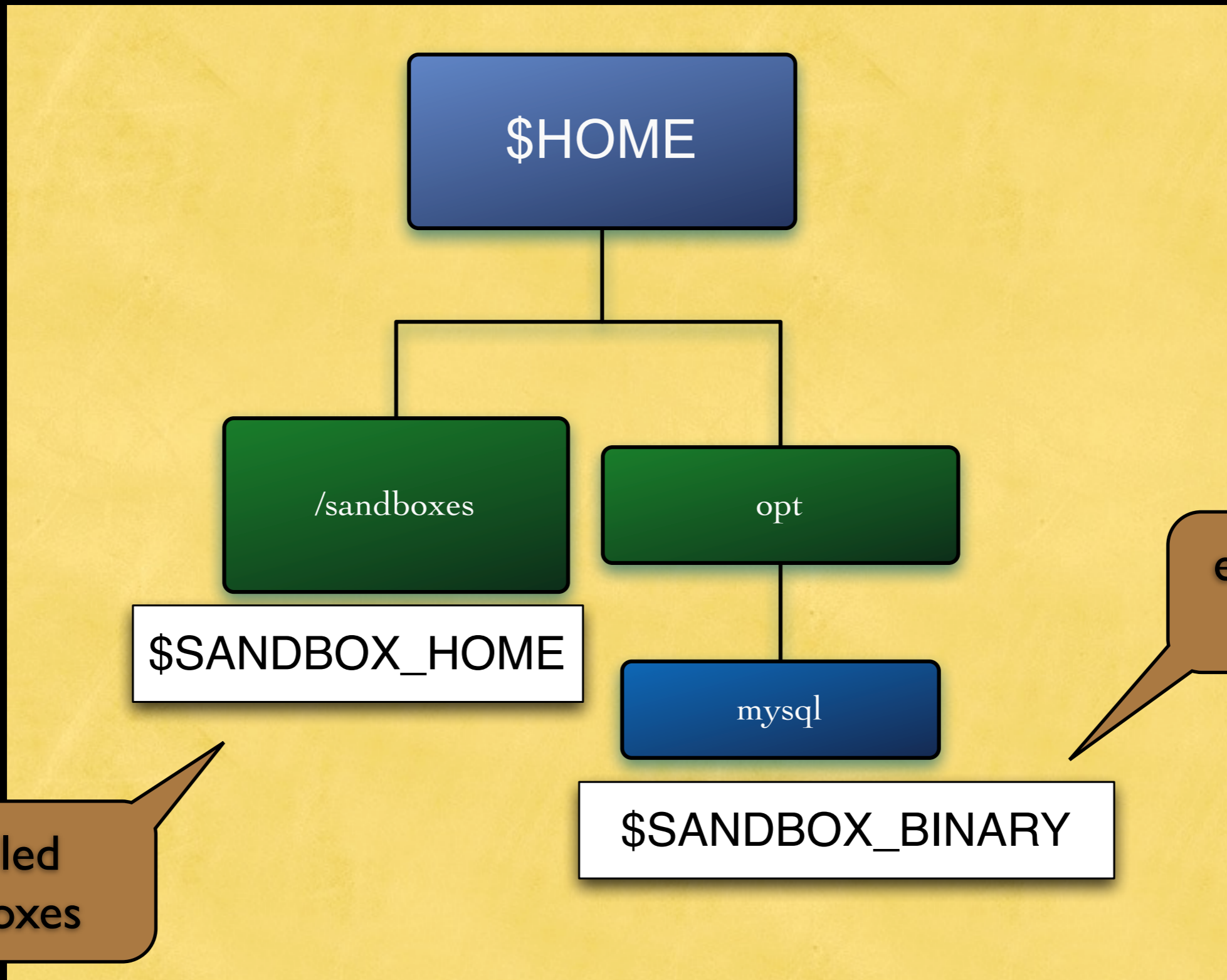
# The easy replication way

```
$ make_replication_sandbox \  
    /path/to/mysql-5.1.54_linux.tar.gz
```

```
# or, after some preparation
```

```
$ make_replication_sandbox 5.1.54
```

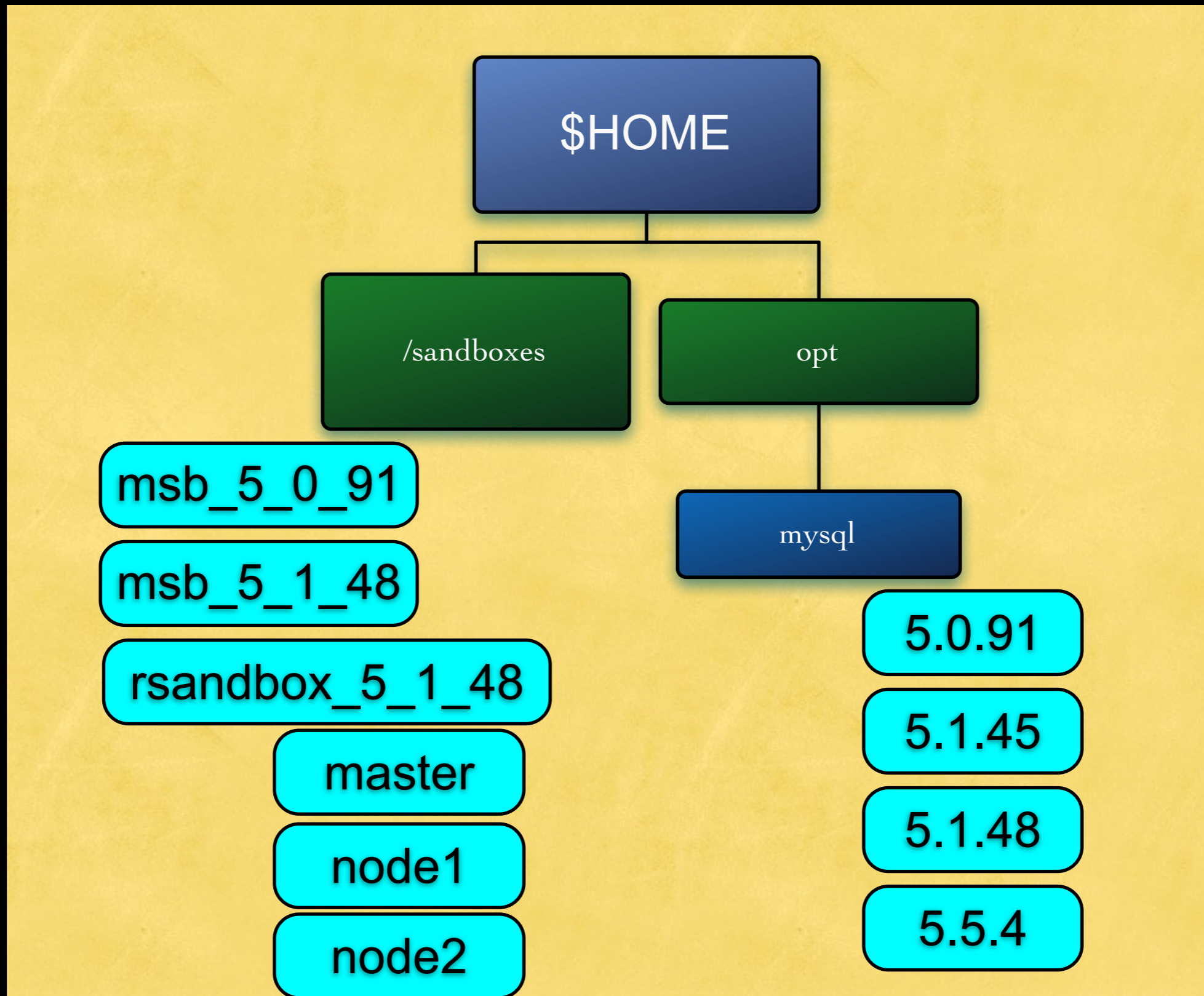
# default architecture



installed sandboxes

expanded tarballs

# default architecture

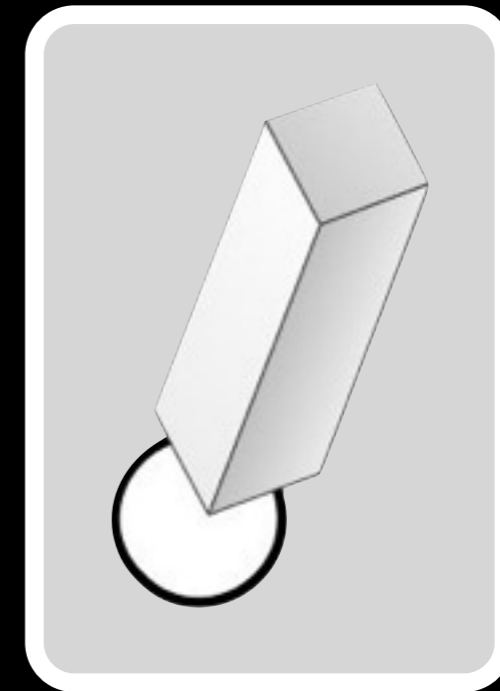


# A Sandbox, step by step

# Step 1: make an installer



+



# installer

- A program that knows the basics of how to install a complex application
- Hiding the details and the ugly output
- Returning clear success or error codes

# step 1.1: make it work!

- make sure the installer works
- even if it is ugly
- even if it requires long commands
- you can refine it later

# Step 2: design a framework

- define:
  - where you want to install
  - where you get the binaries from
  - how to shape the sandbox directories

# step 3: define an interface

- put the installer inside an easy to use shell
- make the installer create scripts that help the user
- Ask yourself: *would I use this stuff?*

# Techniques: ports (1)

- Easy:
  - the program has a --port option
  - tell the installer to use that option

# Techniques: ports (2)

- Difficult:
- the program does not have a --port option
  - (a) the program has a configuration file:
    - make the installer modify that file
  - (b) no configuration file:
    - use a tunnel

# Techniques: hostnames

- Make fake hostnames in /etc/hosts

# techniques:conf files

- method 1: use a template
  - PRO: easy to parse
  - CON: stiff and hard to evolve
- method 2: parse and replace
  - PRO: resilient to changes
  - CON: difficult to parse

# template example

```
repl_boot_script=_BOOT_SCRIPT_  
repl_dataserver_host=_THIS_HOST_  
repl_dbport=_REPL_PORT_  
repl_logs_dir=_TUNGSTEN_BASE_/logs  
repl_backup_storage_dir=_TUNGSTEN_BASE_/backups
```

# techniques: hardcoded calls to programs

- if your program calls a given program
  - change the \$PATH
  - put a fake program earlier in \$PATH
  - make that program do what you want

# techniques:make it extensible

- A demo

# demos

# THANKS