

MYSQL FAILOVER WITH CONSUL VOL. 2

MySQL Casual Talks vol.7

@ijin

12/12/2014

ABOUT ME (@IJIN)

- Michael H. Oshita
- Japanese American
- 10+ Years in Japan
- Software/Infra/Cloud Engineer
- <http://ijin.github.io>



ABOUT ME (@IJIN)



トラブル☆しゅーたーず

WALTI

The banner features a blurred background of a person's face and a code editor with JavaScript code. The Walti logo, which includes a small robot icon, is in the top left. Navigation links for '特徴' (Features), '料金' (Pricing), and 'ログイン' (Login) are in the top right. A large white circle on the left contains the text 'これまでの総スキャン回数' (Total number of scans so far) and the large number '456'. To the right of the circle, the main text reads 'サーバーサイドのセキュリティスキャンを身近に。' (Bring server-side security scanning closer to you.) followed by '初回スキャンは無料！その後は、スキャン1回5円〜' (First scan is free! After that, 5 yen per scan~). A red button at the bottom right says '招待を希望される方はこちら。' (Click here if you want to be invited.).

Walti

特徴 | 料金 | ログイン

これまでの総スキャン回数

456

サーバーサイドのセキュリティスキャンを身近に。

初回スキャンは無料！その後は、スキャン1回5円〜

招待を希望される方はこちら。

<https://walti.io>

BLOG

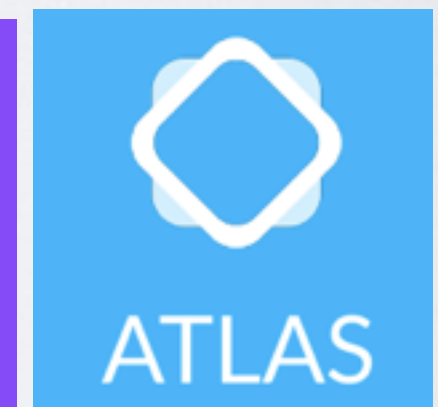


<https://ijin.github.io>

MYSQL FAILOVER

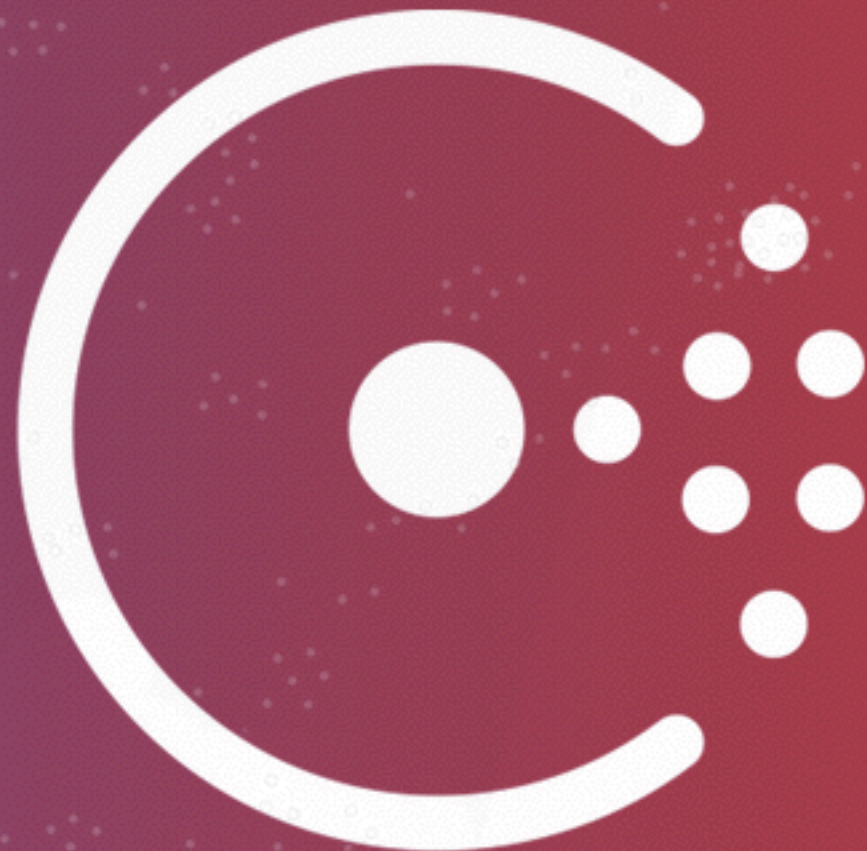
INGREDIENTES:

- MySQL
- MHA (MySQL Master HA)
- Consul



CONSUL

Service discovery
and configuration
made easy.
Distributed, highly
available, and
datacenter-aware.



CONSUL

HTTP APIとDNSで操作ができるKV内蔵な分散型クラスタ



CONSUL AGENT

HTTP API

key/value endpoint:

/v1/kv/<key>

RESTful:

GET, PUT, DELETE

CONSUL AGENT

HTTP API

key/value:

```
ijin@foo:~$ curl -X PUT -d 'bar' localhost:8500/v1/kv/foo
true
ijin@foo:~$ curl -s localhost:8500/v1/kv/foo | jq .
[
  {
    "CreateIndex": 515,
    "ModifyIndex": 519,
    "LockIndex": 0,
    "Key": "foo",
    "Flags": 0,
    "Value": "YmFy"
  }
]
ijin@foo:~$ curl -s localhost:8500/v1/kv/foo?raw
bar
ijin@foo:~$
```

base64

raw

CONSUL AGENT

DNS Interface

Node Lookup:

<node>.node.<datacenter>.<domain>

CONSUL AGENT

DNS Interface

```
ijin@foo:~$ dig @localhost -p8600 bar.node.consul

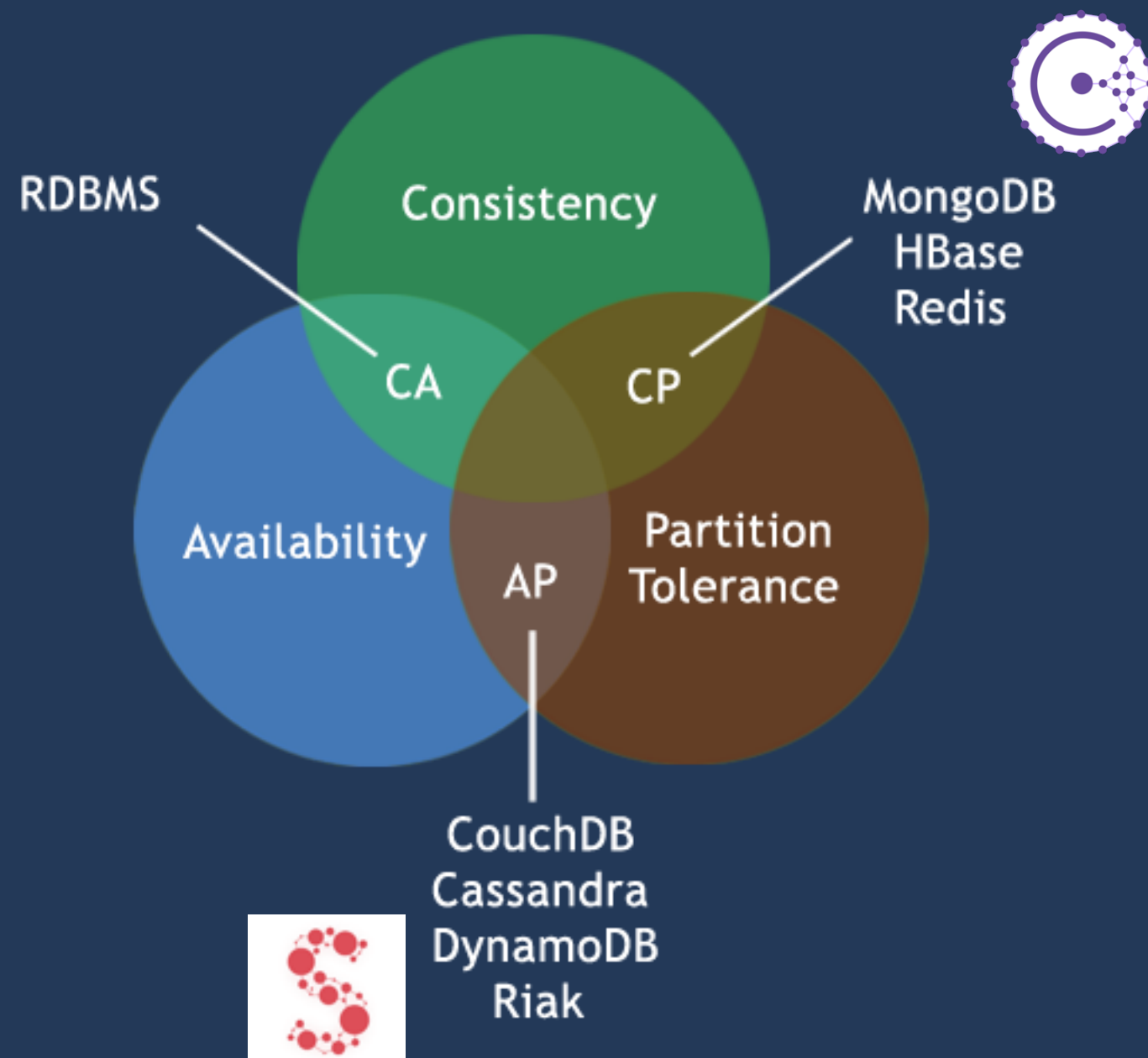
; <<>> DiG 9.8.1-P1 <<>> @localhost -p8600 bar.node.consul
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19212
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;bar.node.consul.                IN      A

;; ANSWER SECTION:
bar.node.consul.                10      IN      A      10.158.129.36
```

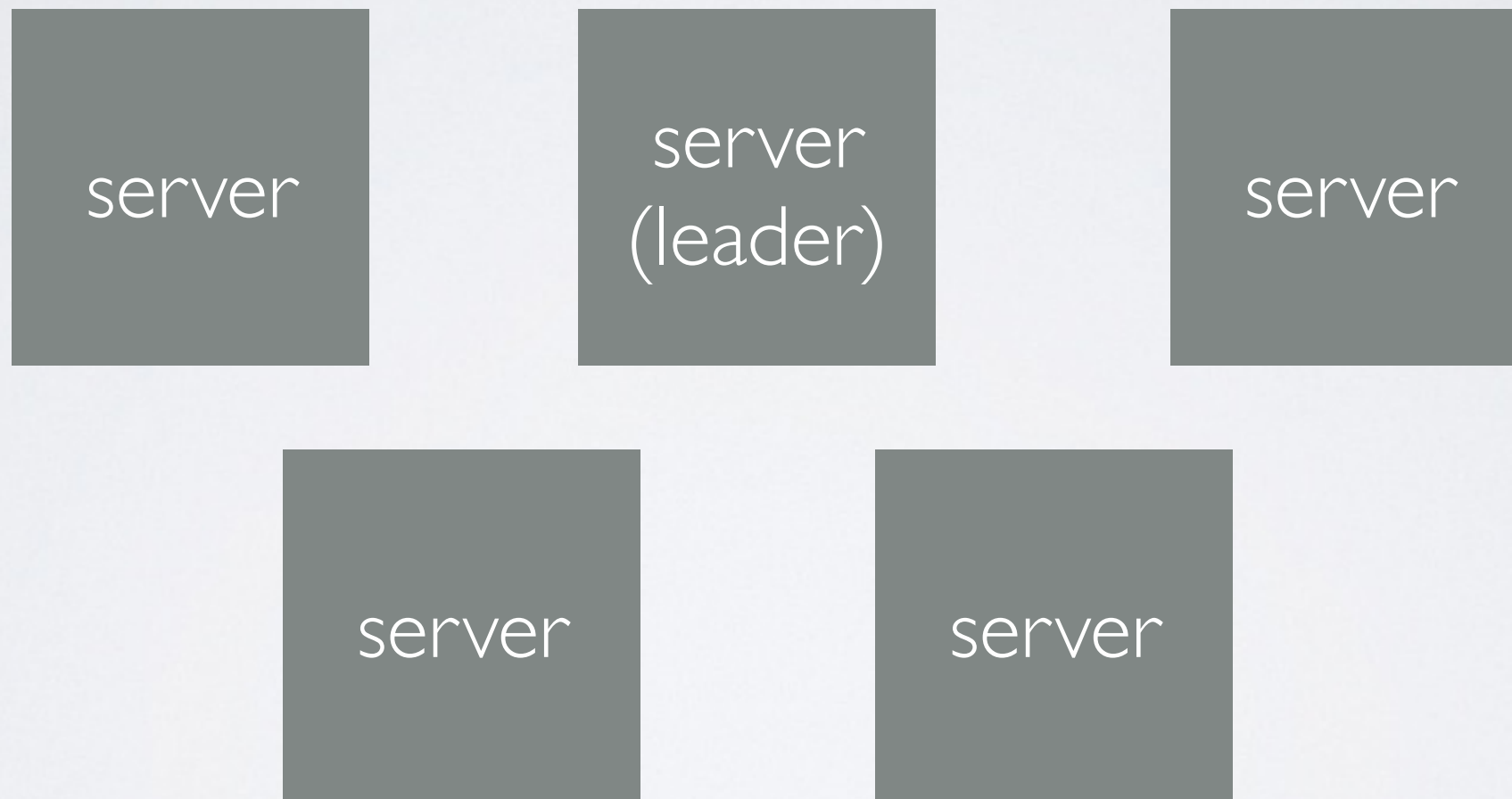
CONSISTENCY

CAP Theorem



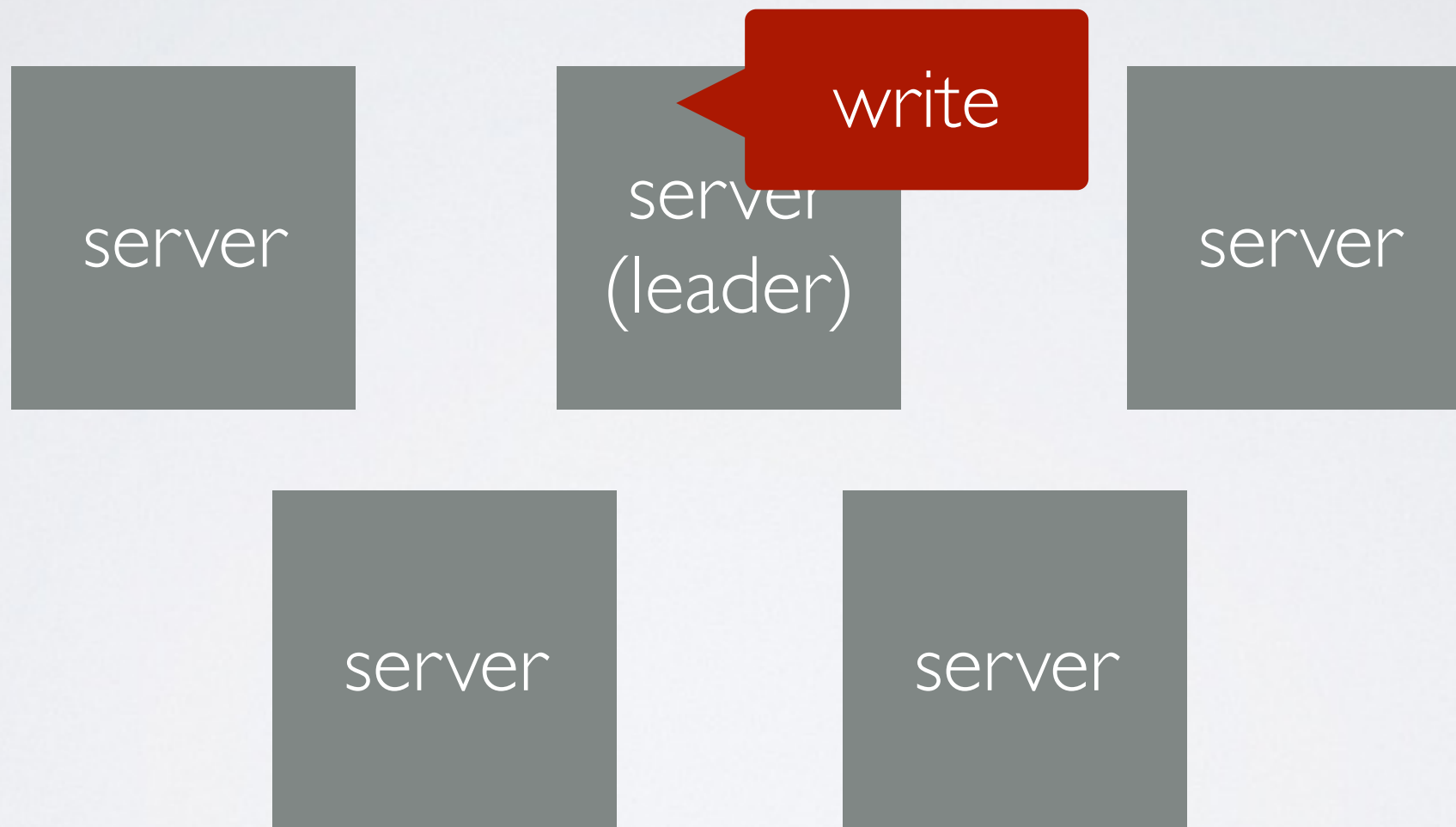
CONSISTENCY

consensus protocol using raft



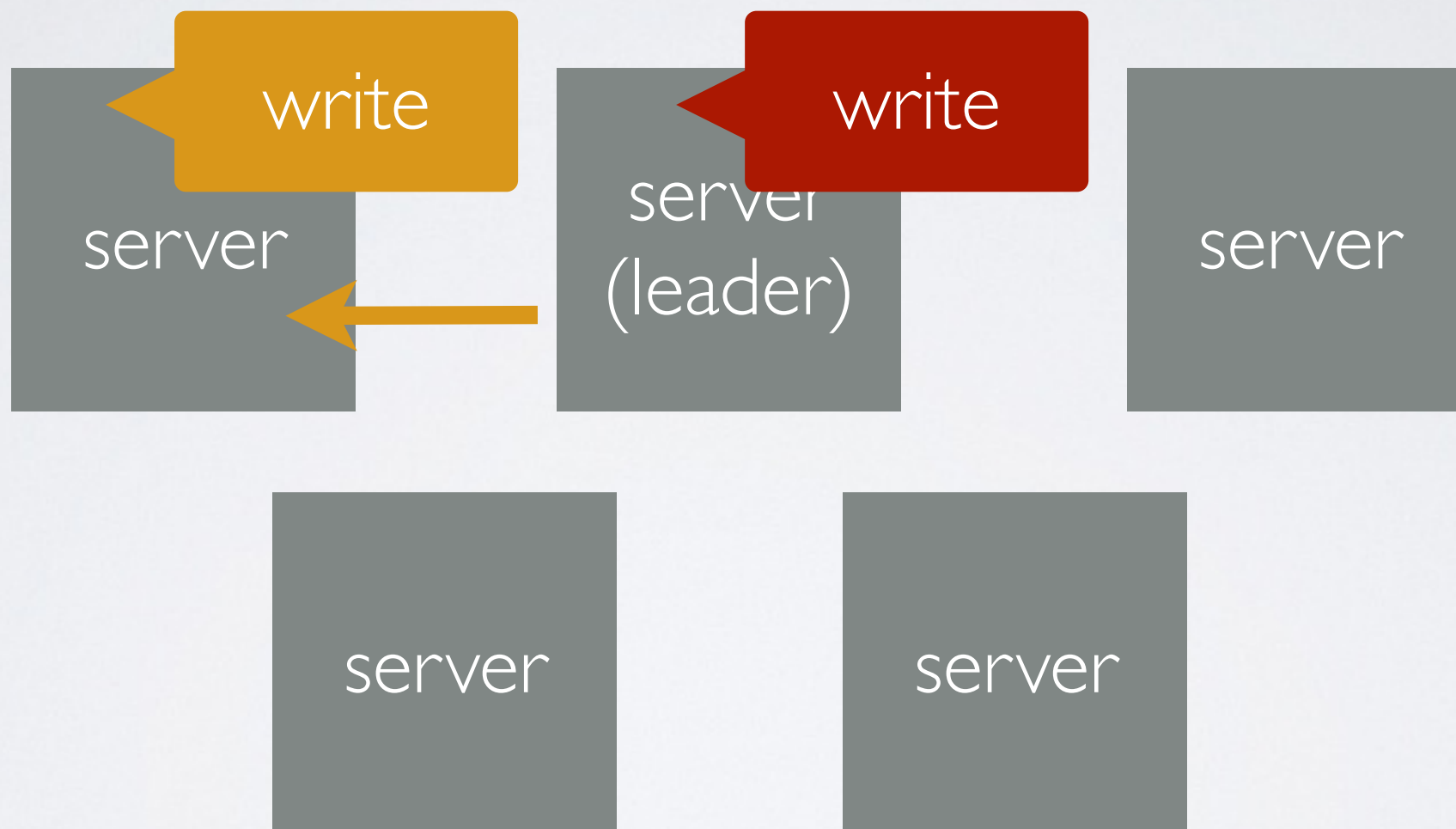
CONSISTENCY

consensus protocol using raft



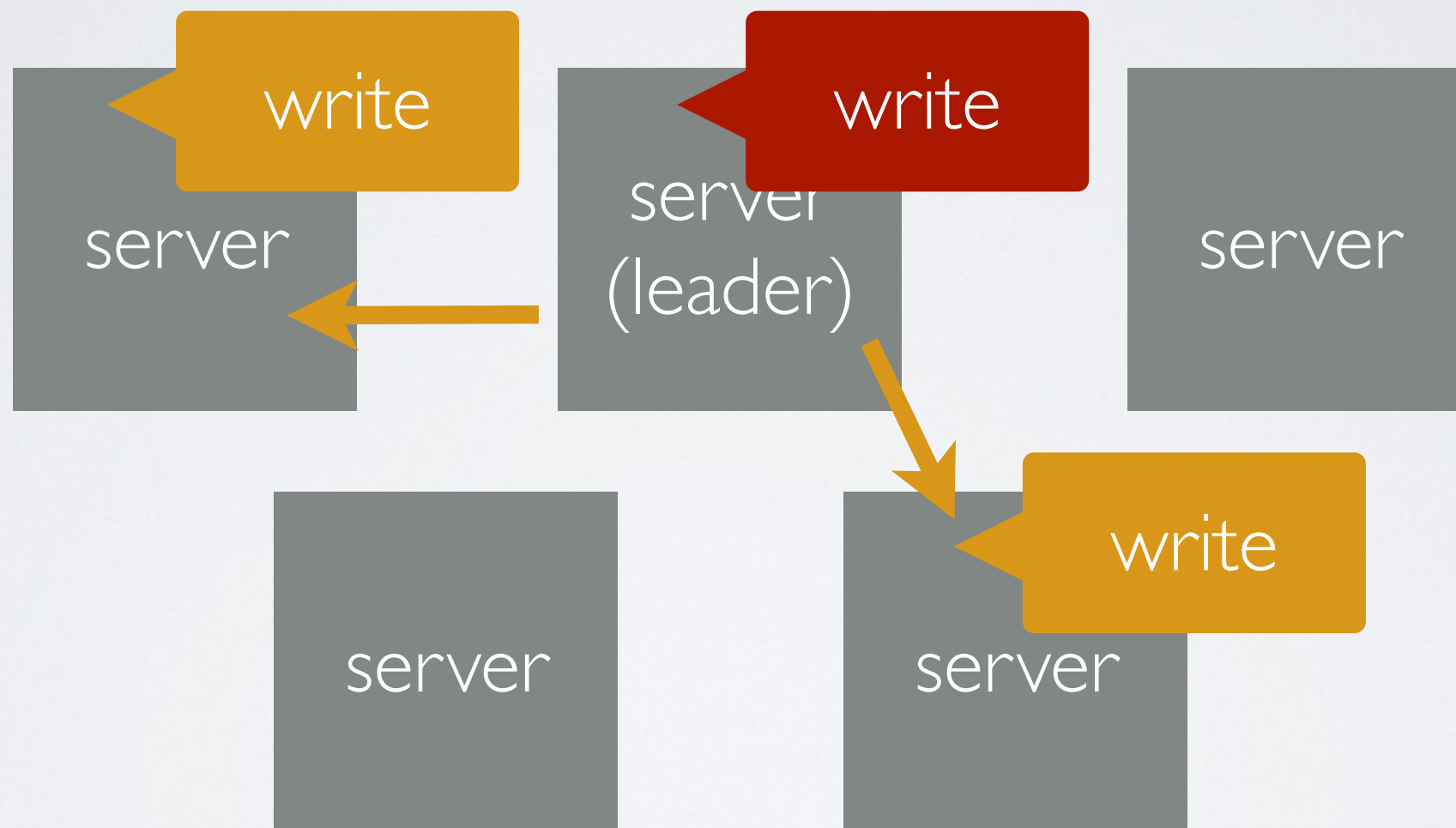
CONSISTENCY

consensus protocol using raft



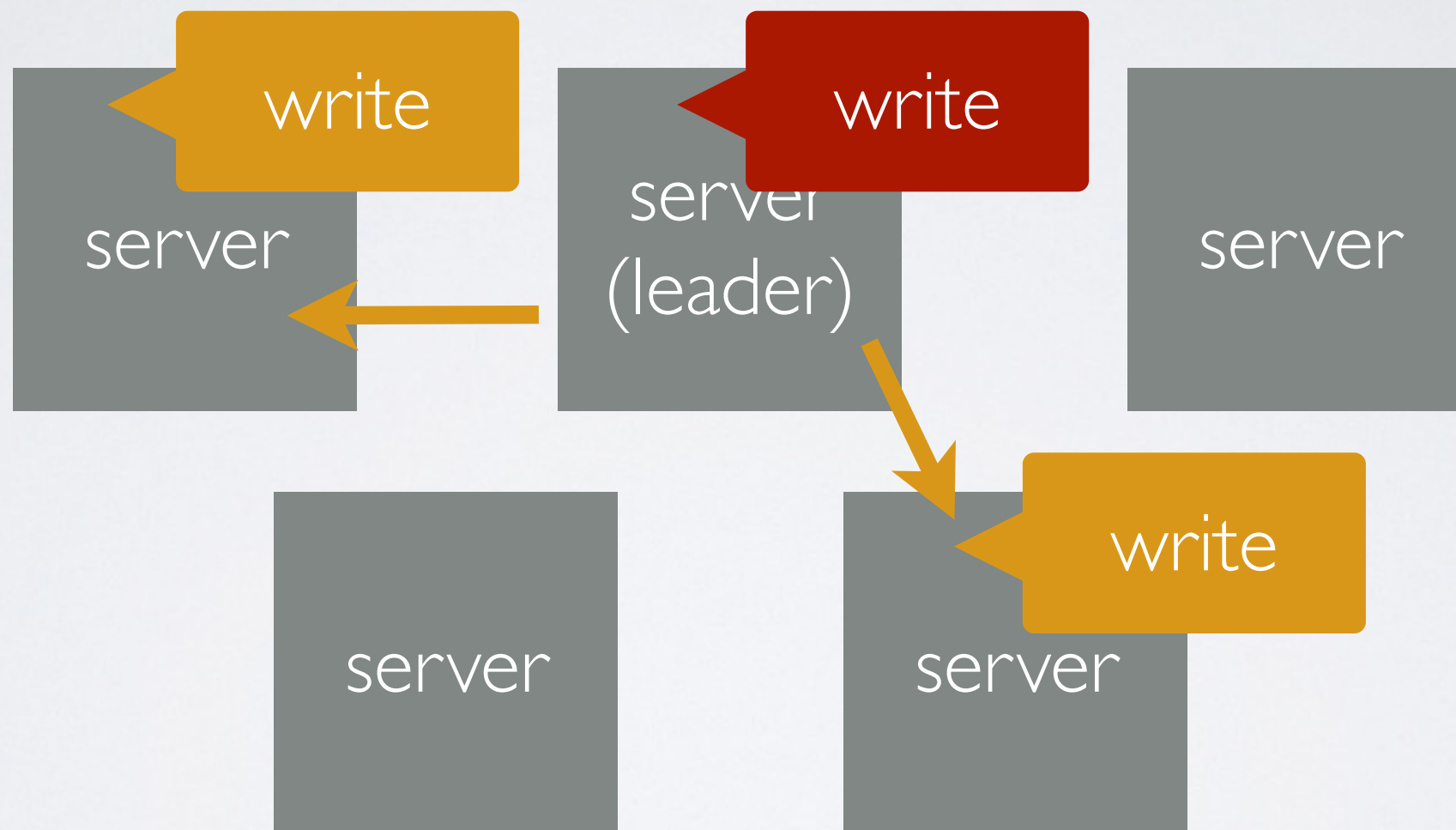
CONSISTENCY

consensus protocol using raft



CONSISTENCY

consensus protocol using raft



quorum $[(n/2)+1]$ reached, committed!

CONSISTENCY

consensus protocol using raft

writes:

Always Strongly Consistent

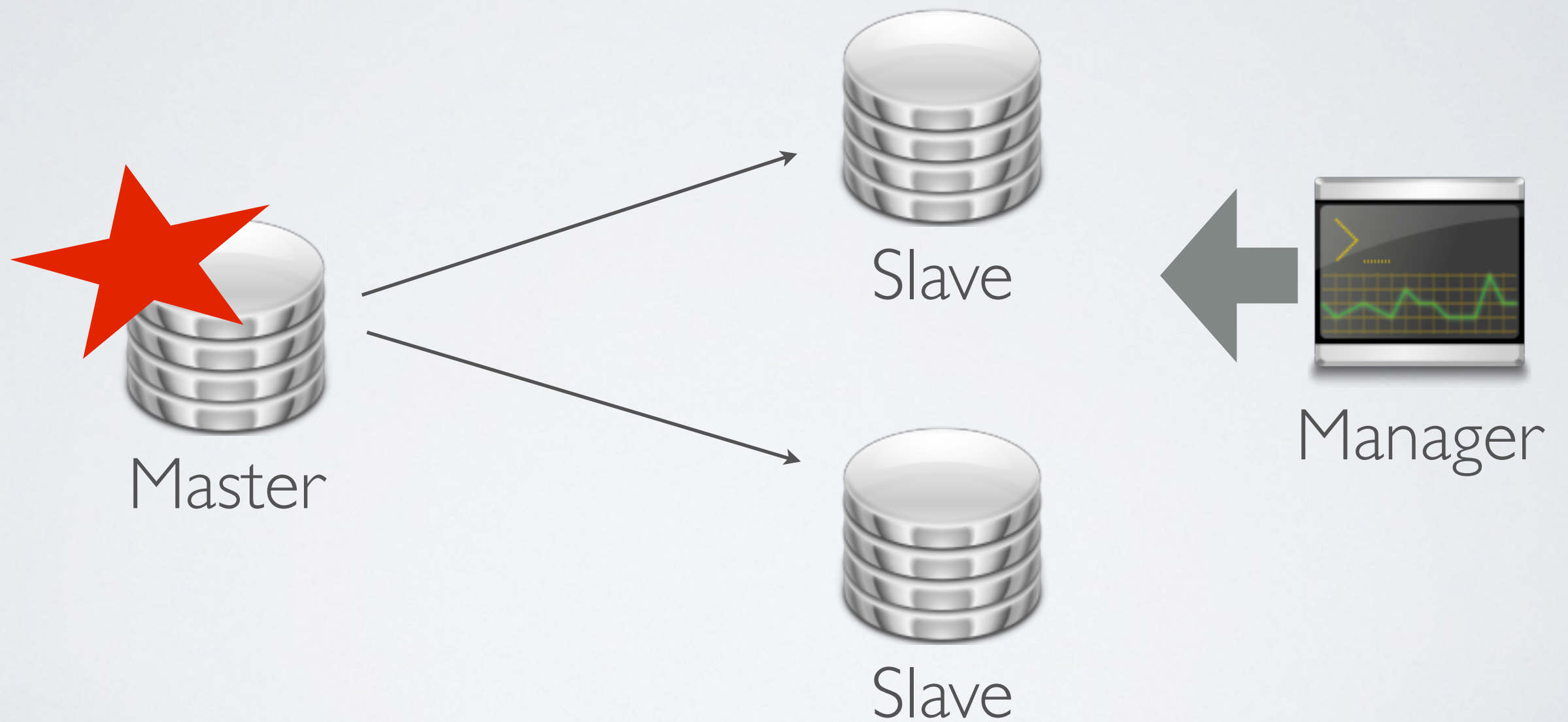
reads:

"?state" or "?consistent" parameter

- default
- strongly consistent
- stale

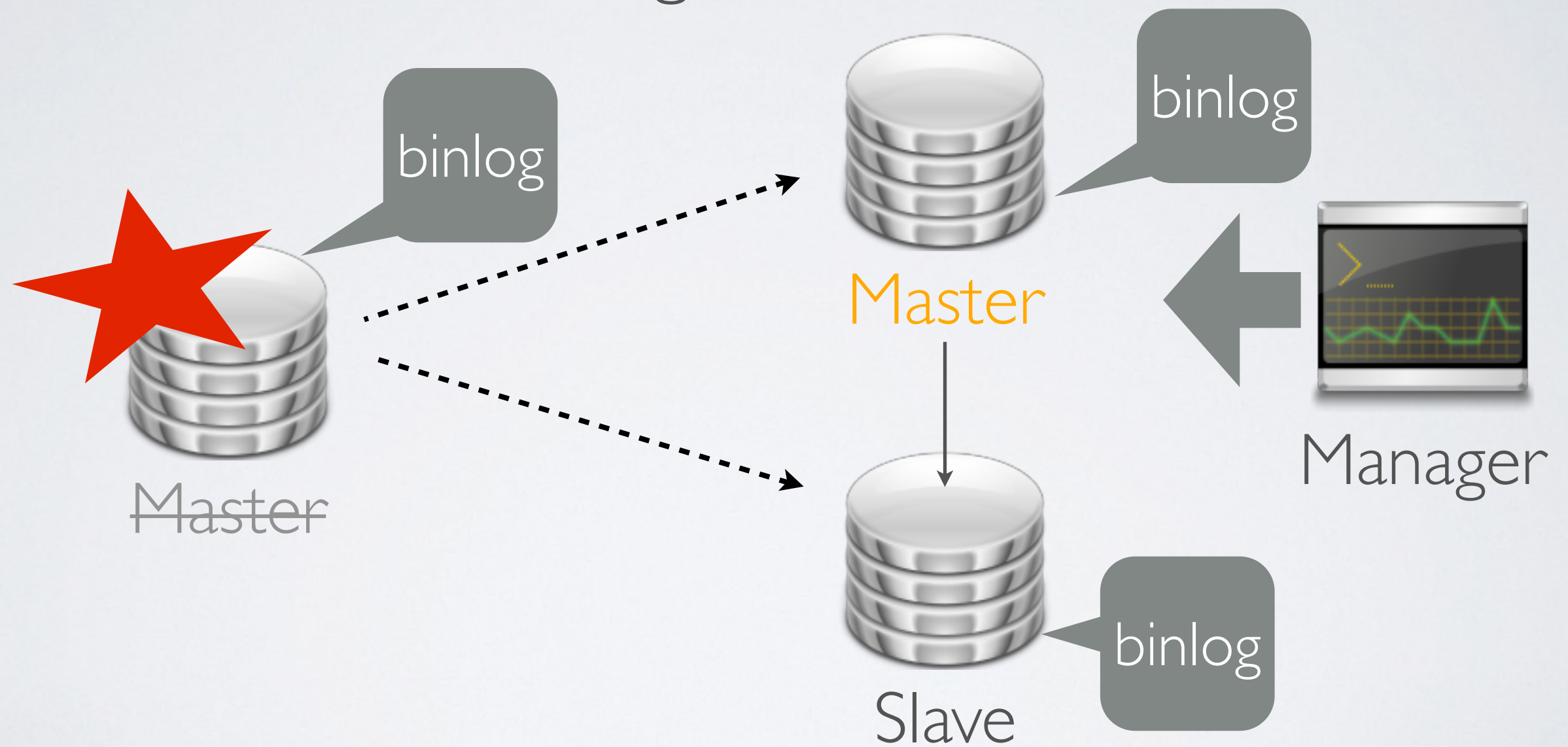
MHA

Masterで障害発生



MHA

各slaveに最新binlogを適用し、Master昇格



MHA

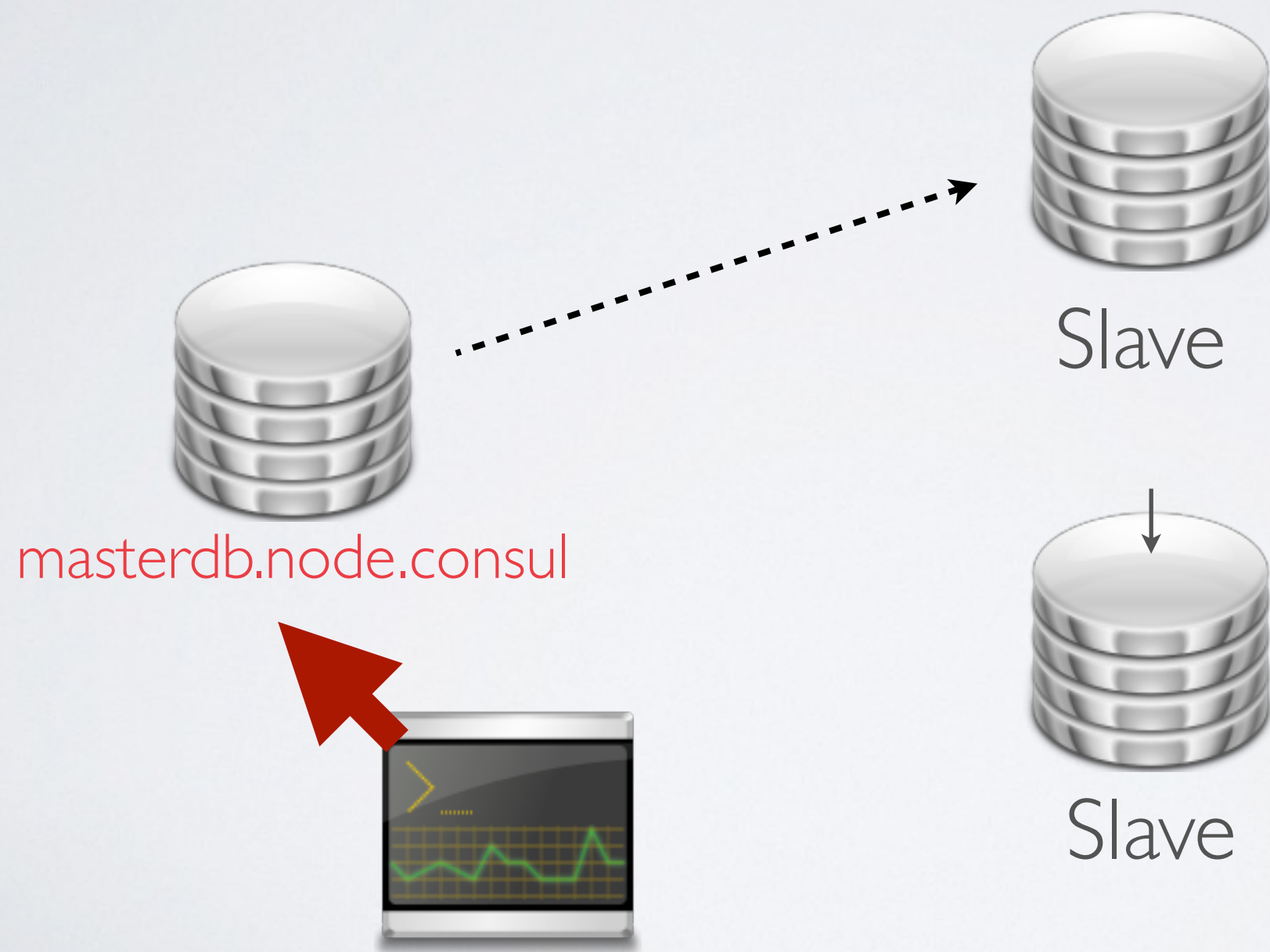
master_ip_failover

master_ip_failover_script

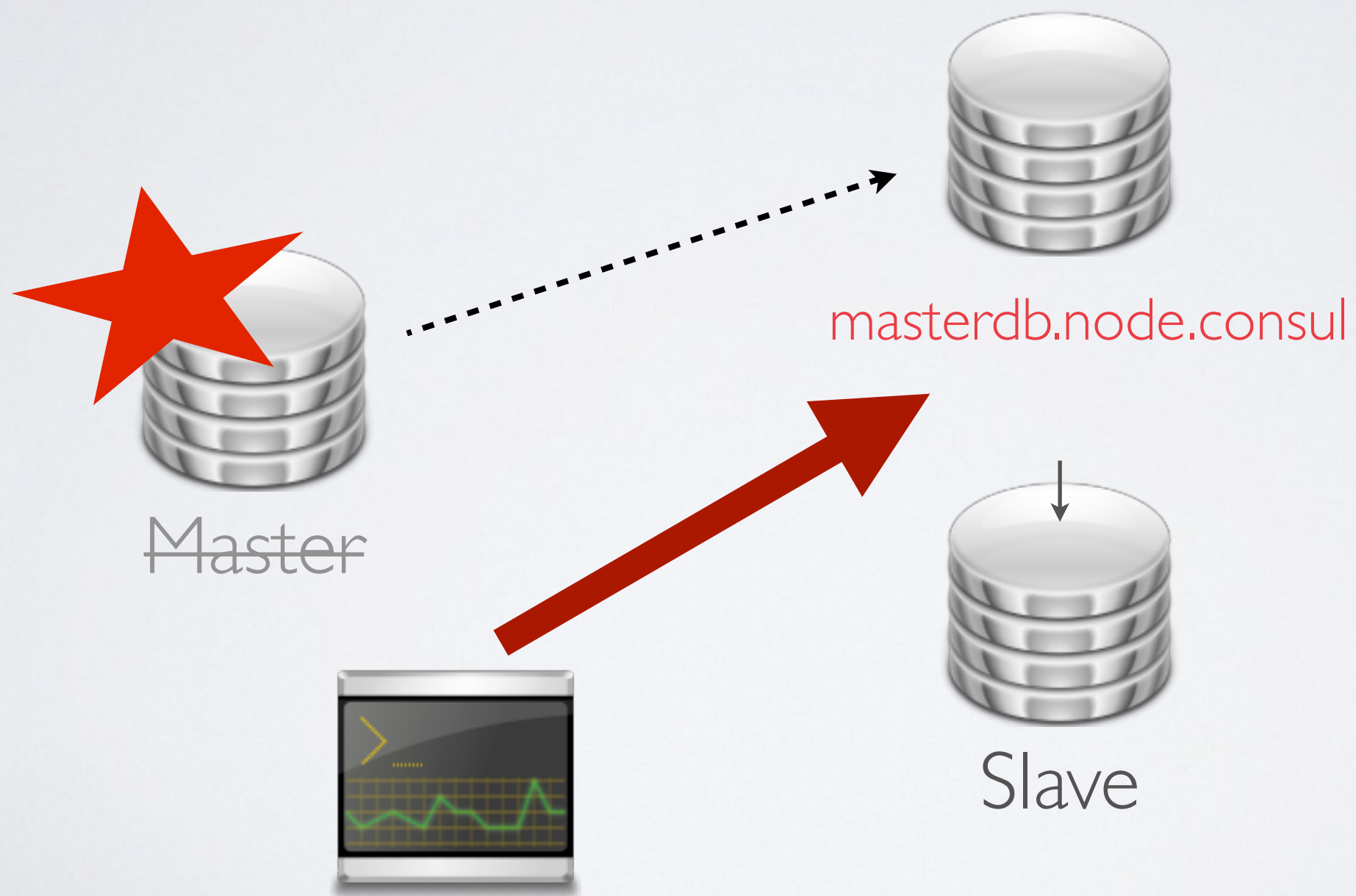
In common HA environments, many cases people allocate one virtual IP address on a master. Pacemaker takes over the virtual IP address to the standby server.

Another common approach is creating a global **catalog database** that has all mappings between addresses (i.e. {app1_master1, 192.168.0.1}, {app_master2, 192.168.0.2}, ...), instead of using need to update the **catalog database** when the current master dies.

MHA



MHA



MHA

FAILOVERシーケンス

"Starting master failover."

"* Phase 1: Configuration Check Phase...\n"

"* Phase 2: Dead Master Shutdown Phase...\n"
==> ここで旧Master情報削除 (master_ip_failover)

"* Phase 3: Master Recovery Phase...\n"

"* Phase 3.1: Getting Latest Slaves Phase...\n"

"* Phase 3.2: Saving Dead Master's Binlog Phase...\n"

"* Phase 3.3: Determining New Master Phase...\n"
==> ここで新Master情報登録 (master_ip_failover)

"* Phase 4: Slaves Recovery Phase...\n"

MHA & CONSUL

master_ip_failover

旧master IPを無効化

```
system("curl -X PUT -d '{\"Node\": \"master-testdb\"}'  
localhost:8500/v1/catalog/deregister");
```

新master IPを登録

```
system("curl -X PUT -d '{\"Node\": \"master-testdb\",  
\"Address\": \"$new_master_host\"}'  
localhost:8500/v1/catalog/register");
```

IN ACTION

The screenshot shows a terminal window with a macOS-style title bar. The terminal output is as follows:

```
affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show master status \G
***** 1. row *****
      File: mysql-bin.000018
      Position: 107
      Binlog_Do_DB:
      Binlog_Ignore_DB:
1 row in set (0.00 sec)

mysql>
      Master_User: repl
      Master_Port: 3306
      Connect_Retry: 60
      Master_Log_File: mysql-bin.000018
      Read_Master_Log_Pos: 107
      Relay_Log_File: mysql-relay-bin.000002
      Relay_Log_Pos: 253
      Relay_Master_Log_File: mysql-bin.000018
      Slave_IO_Running: Yes
      Slave_SQL_Running: Yes
      Replicate_Do_DB:
      Replicate_Ignore_DB:
      Replicate_Do_Table:
      Replicate_Ignore_Table:
      Replicate_Wild_Do_Table:
```

On the right side of the terminal, there is a vertical dashed line. To the right of this line, the following text is visible:

```
|111.109.27(10.111.109.27:3306)..
|Sun Jun 22 12:34:40 2014 - [info] Ping(SELECT) succeeded, waiting u
ntil MySQL doesn't respond..
```

At the bottom of the terminal, there is a status bar with the following information:

```
u 12.04.4 1: bash- [26/1081] root@km-app1:/etc/consul.d# while true; do mysql -umha -pmasterha20
13 -h master-testdb.node.consul -e 'select now()'; sleep 1; done

root@ubuntu-chef:~# while true; do dig @localhost -p8600 master-tes
tdb.node.consul +short; sleep 1; done
```

The status bar also includes system information: 42!! 0.07 2.4GHz 3.9G15% 2014-06-22 12:34:53.

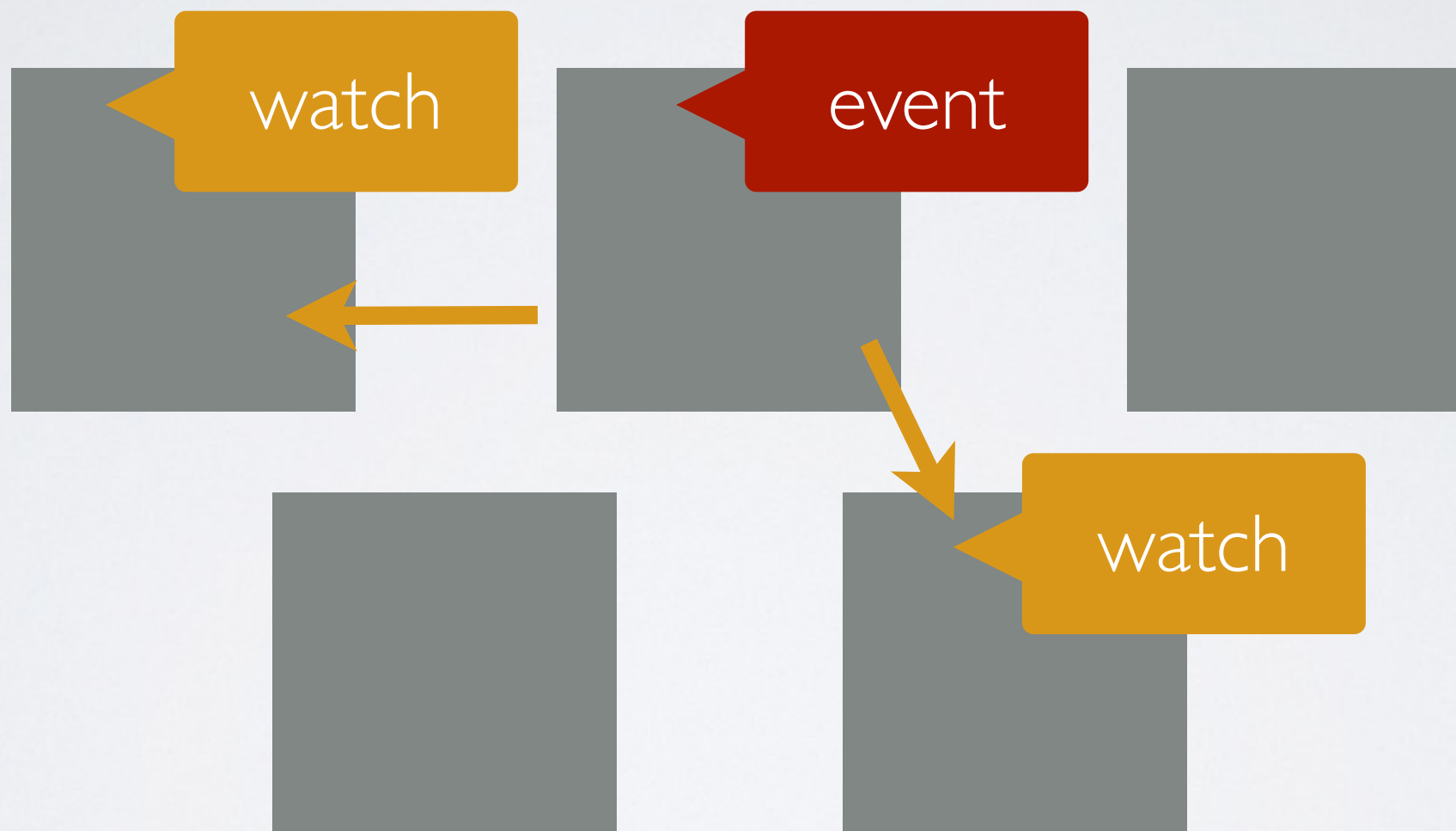
<https://www.youtube.com/watch?v=rA4hyJ-pccU>

ALTERNATIVES

- consul event & consul watch
- consul-template

CONSUL EVENT

events propagated over gossip protocol



```
$ consul watch -type key -key foo/bar/baz /usr/bin/my-key-handler.sh
```

CONSUL EVENT

Under the hood, events are propagated using the [gossip protocol](#). While the details are not important for using events, an understanding of the semantics is useful. The gossip layer will make a best-effort to deliver the event, but there is **no guarantee** delivery. Unlike most Consul data, which is replicated using [consensus](#), event data is purely peer-to-peer over gossip. This means it is not persisted and does not have a total ordering. In practice, this means you cannot rely on the order of message delivery. An advantage however is that events can still be used even in the absence of server nodes or during an outage.

CONSUL-TEMPLATE

haproxy.cfg

```
defaults
    log      global
    mode     tcp
    #option  httplog
    option   dontlognull
    timeout  5000
    clitimeout 50000
    srvtimeout 50000

listen mysql-master
    bind 127.0.0.1:3306
    balance roundrobin
    {{ range service "master-testdb" }}
    server {{.Node}} {{.Address }} {{end}}:3306 check port 3306
```

CONSUL-TEMPLATE

```
consul-template -consul 127.0.0.1:8500  
-template "/tmp/haproxy.cfg:/etc/haproxy/haproxy.cfg: \  
service haproxy reload"
```

IN CLOSING

- Use consul DNS catalog for MHA failover
- Consul-Template over consul event