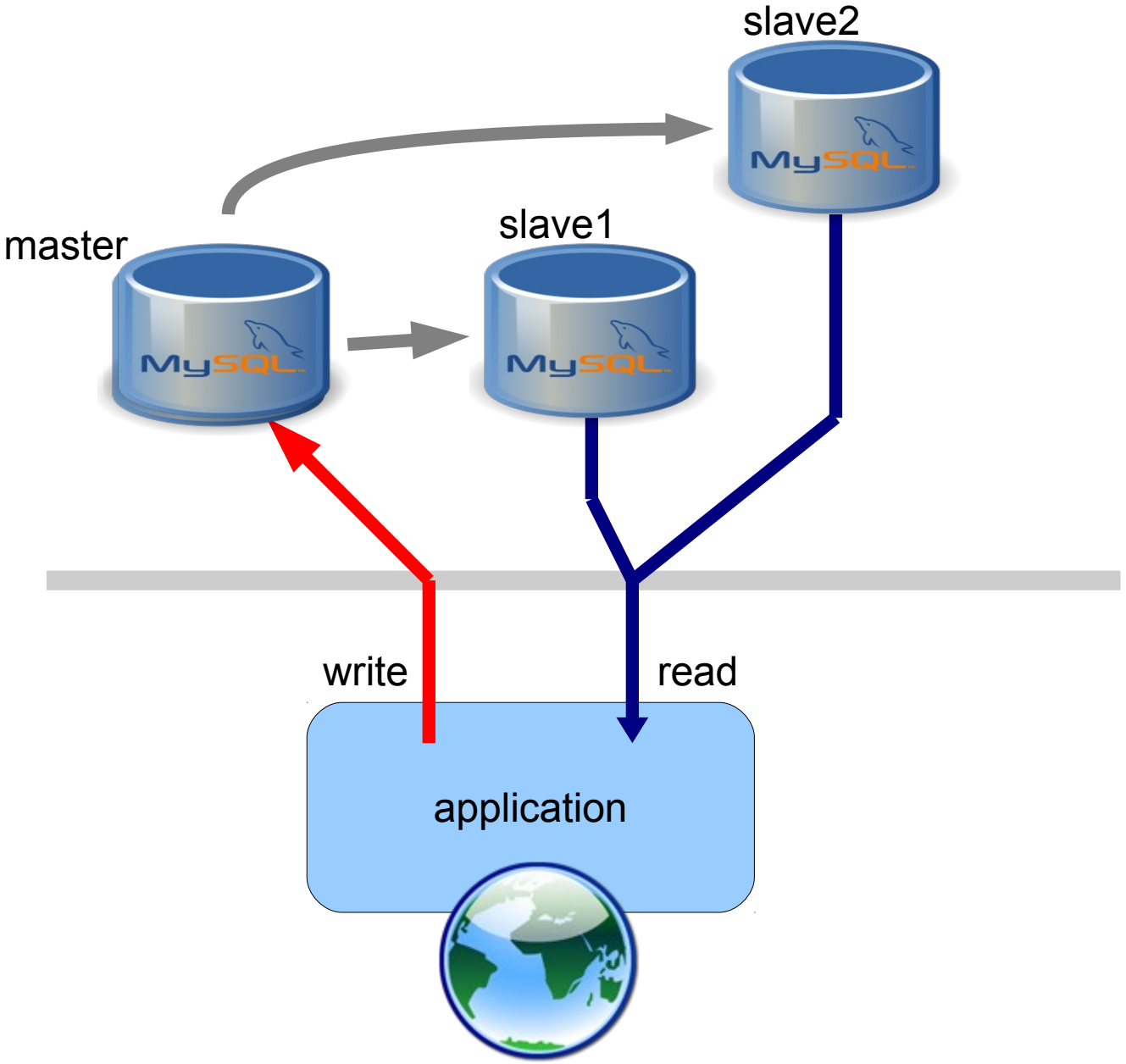
Mysql database replicatie

- waarom
  - hoge beschikbaarheid (high availability)
  - schaalbaarheid (scalability, load balancing)
  - niet voor backup (delete = delete!)
- focus voor nu: master -> slave replicatie

  (bv. voor een website (veel reads, minder writes))
  - master config
  - slave condig
  - demo
- andere vormen
  - master <-> master
  - clustering
  - combinaties

master - slave replication



Config

- master = rpi-1 (192.168.5.31), slave = rpi-2 (192.168.5.32)
- master config, in my.cnf:

```
server-id = 1
log-bin   = /var/log/mysql/mysql-bin.log
```

- create user for replication on master:

```
mysql> grant replication slave on *.* to 'repli_user'@'192.168.5.32' \
identified by '********';
```

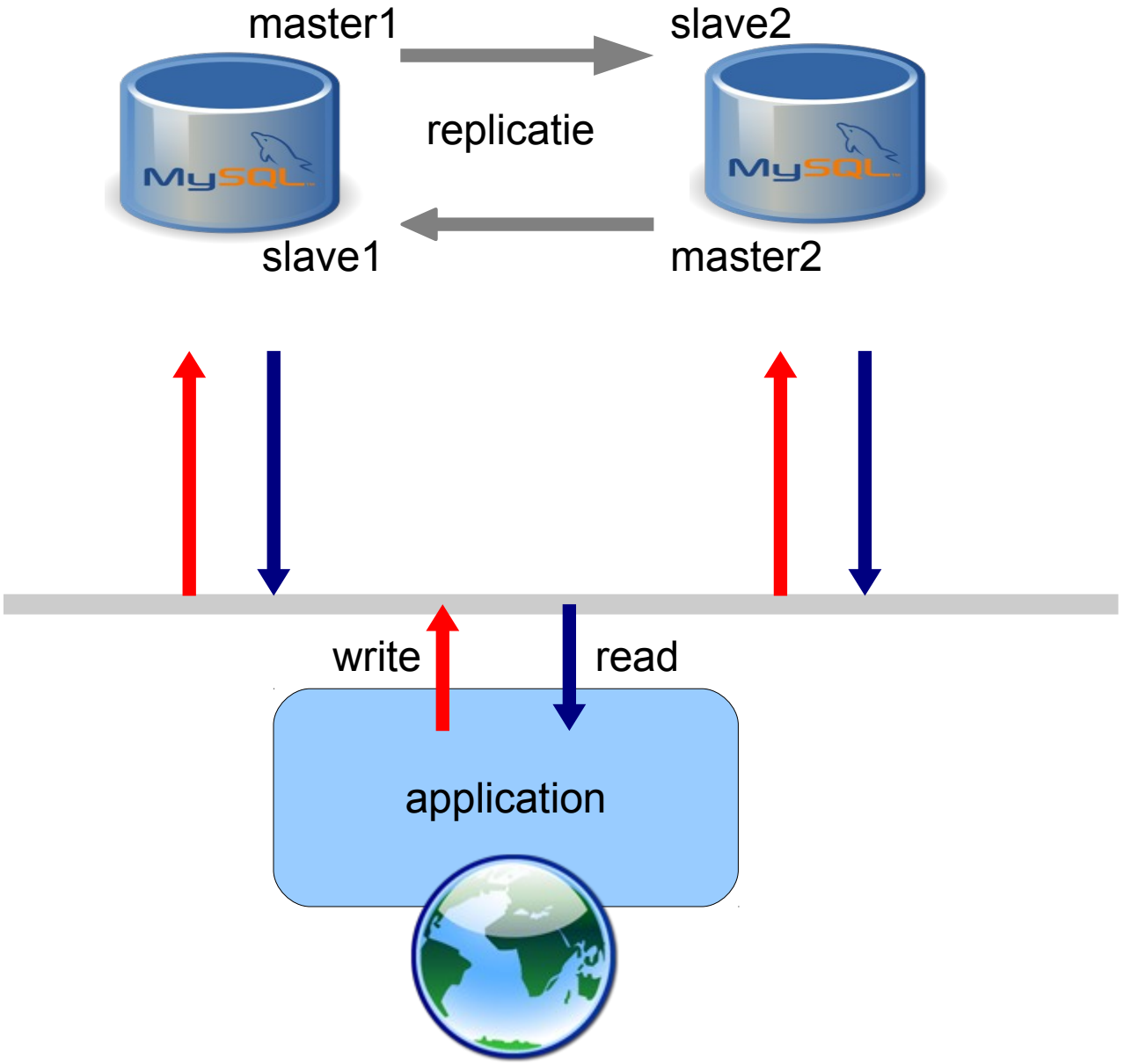
- slave config, in my.cnf:

```
skip-slave-start
server-id = 2
log-bin   = /var/log/mysql/mysql-bin.log
```

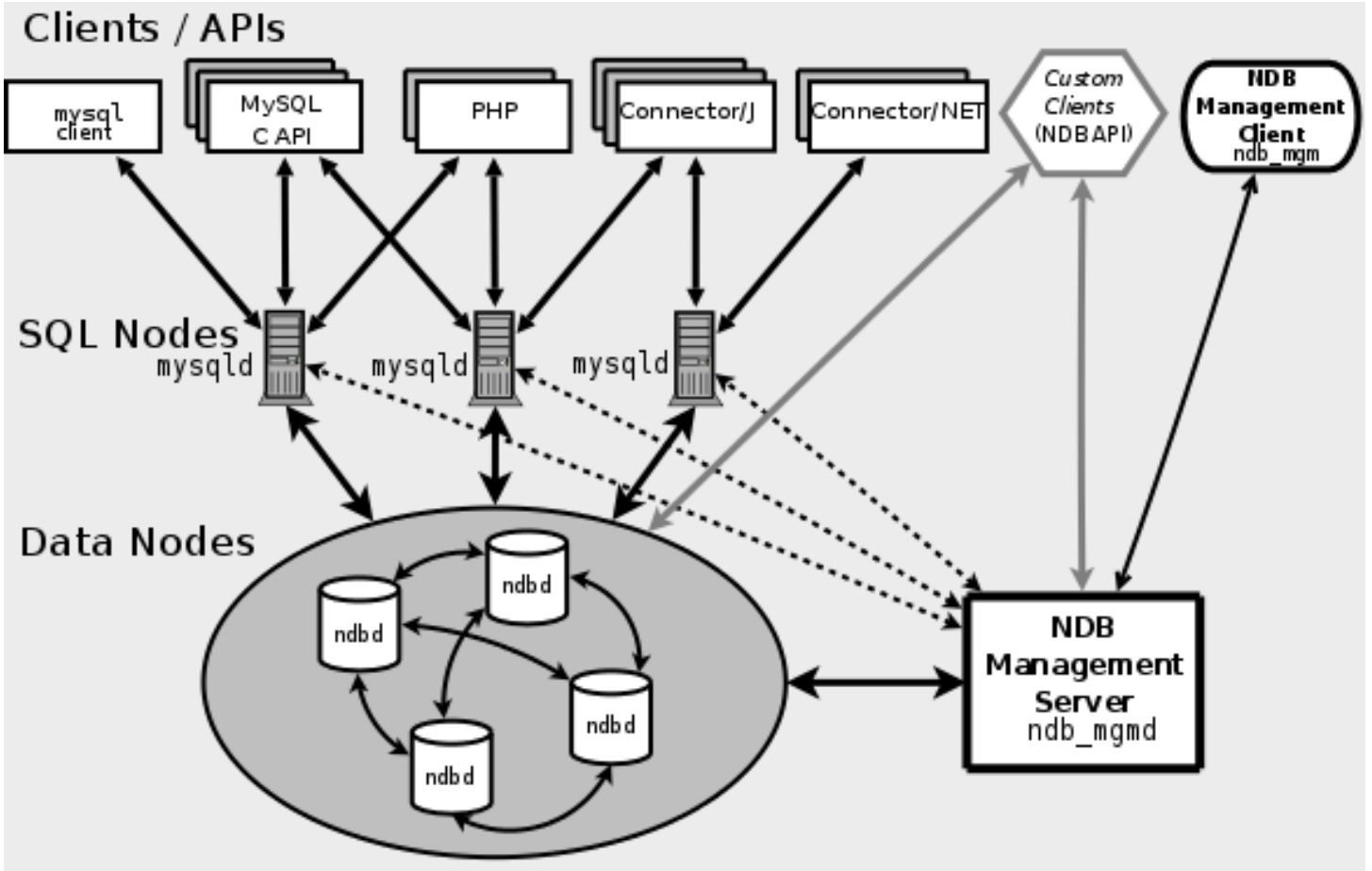
- enable slaving

```
mysql> CHANGE MASTER TO MASTER_HOST = 'masterhost.domain.com', \
       MASTER_USER = 'repli_user', MASTER_PASSWORD = '********';
mysql> start slave;
mysql> show slave status\G
```

master <-> master

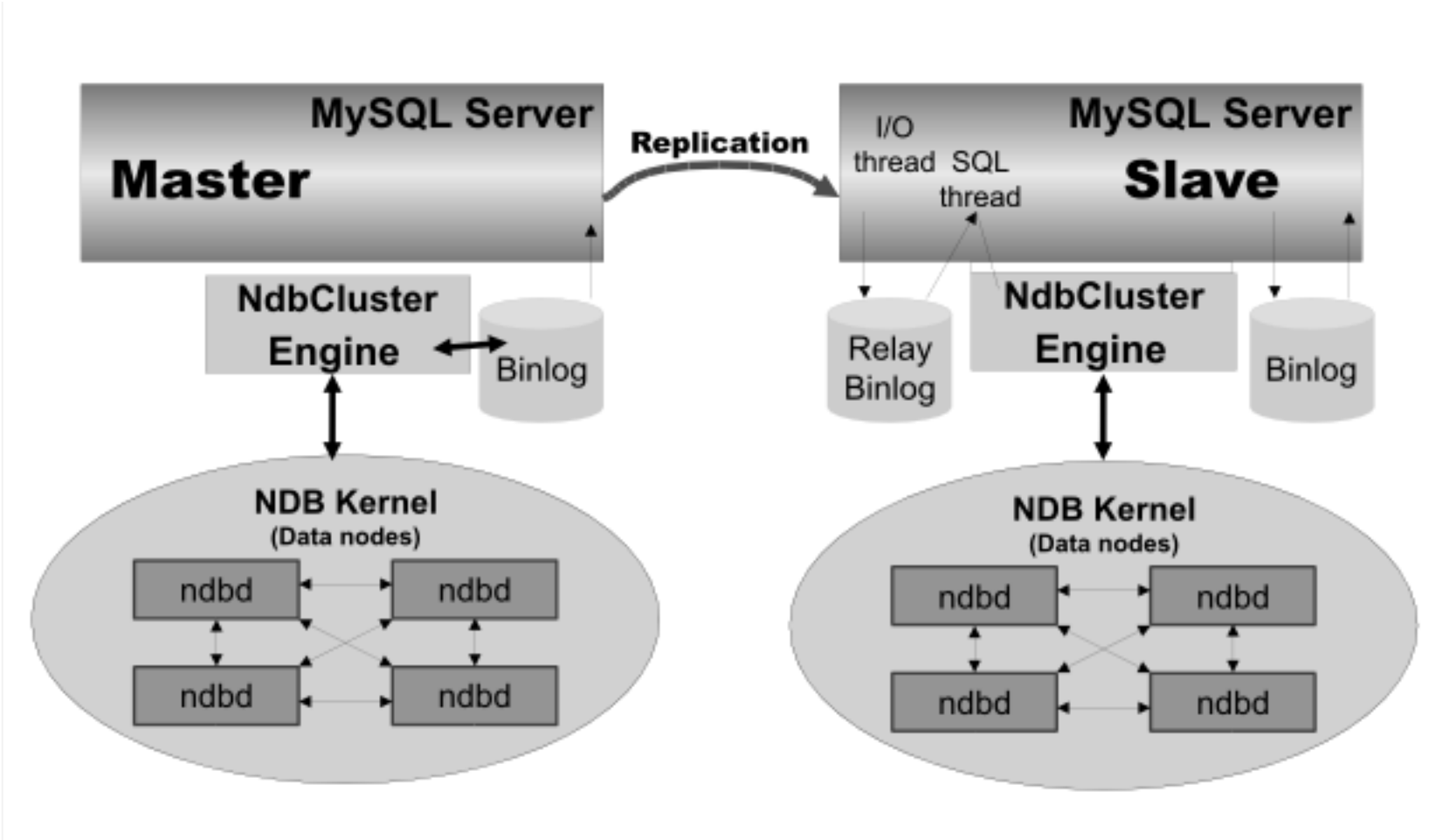


Mysql cluster



Bron: http://dev.mysql.com/doc/refman/5.1/en/mysql-cluster-overview.html

Combinaties



Bron: http://dev.mysql.com/doc/refman/5.1/en/mysql-cluster-replication.html