

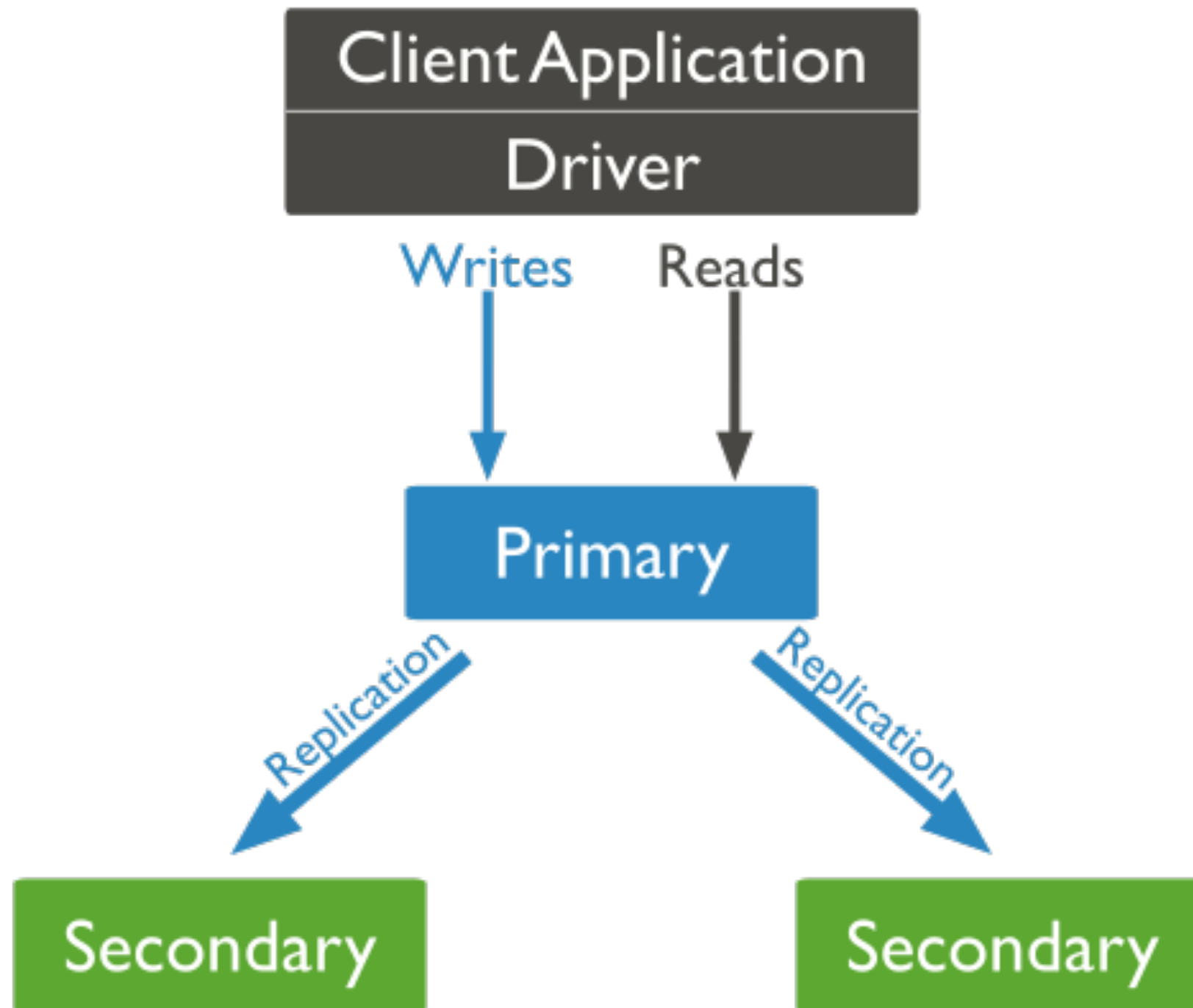
mongoDB

Banco de dados NoSQL - Replicaset

Prof. Gustavo Leitão

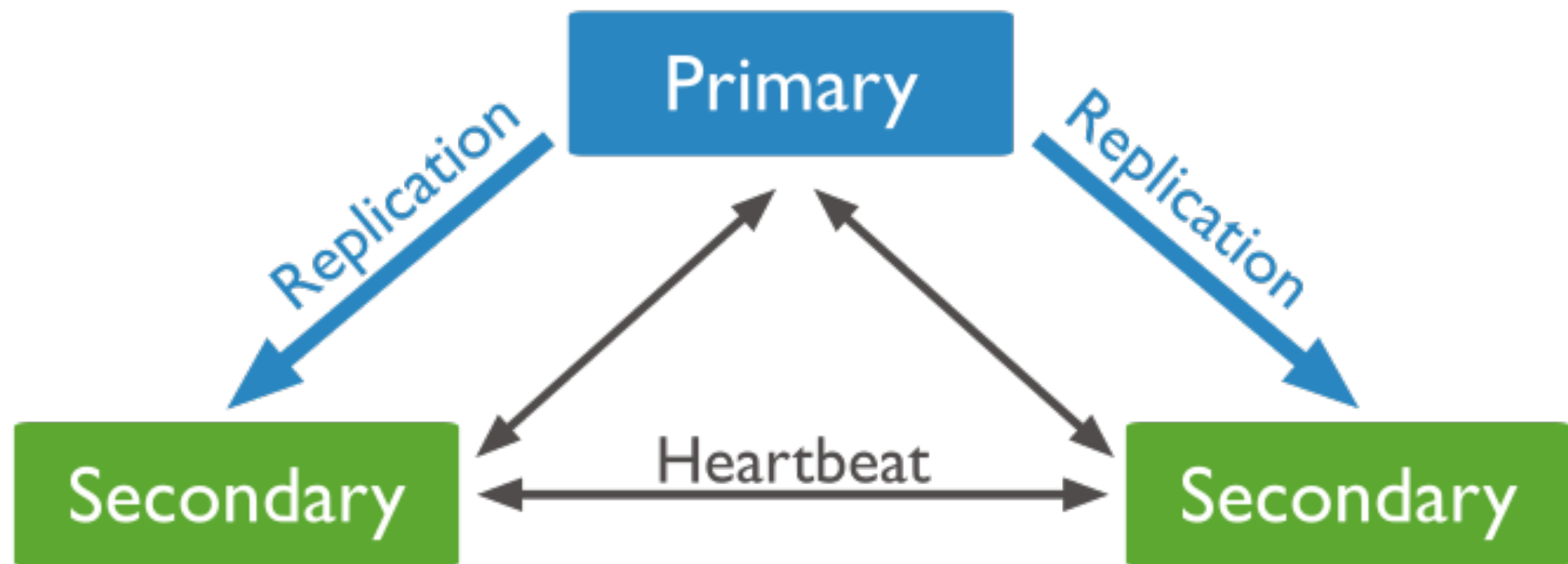


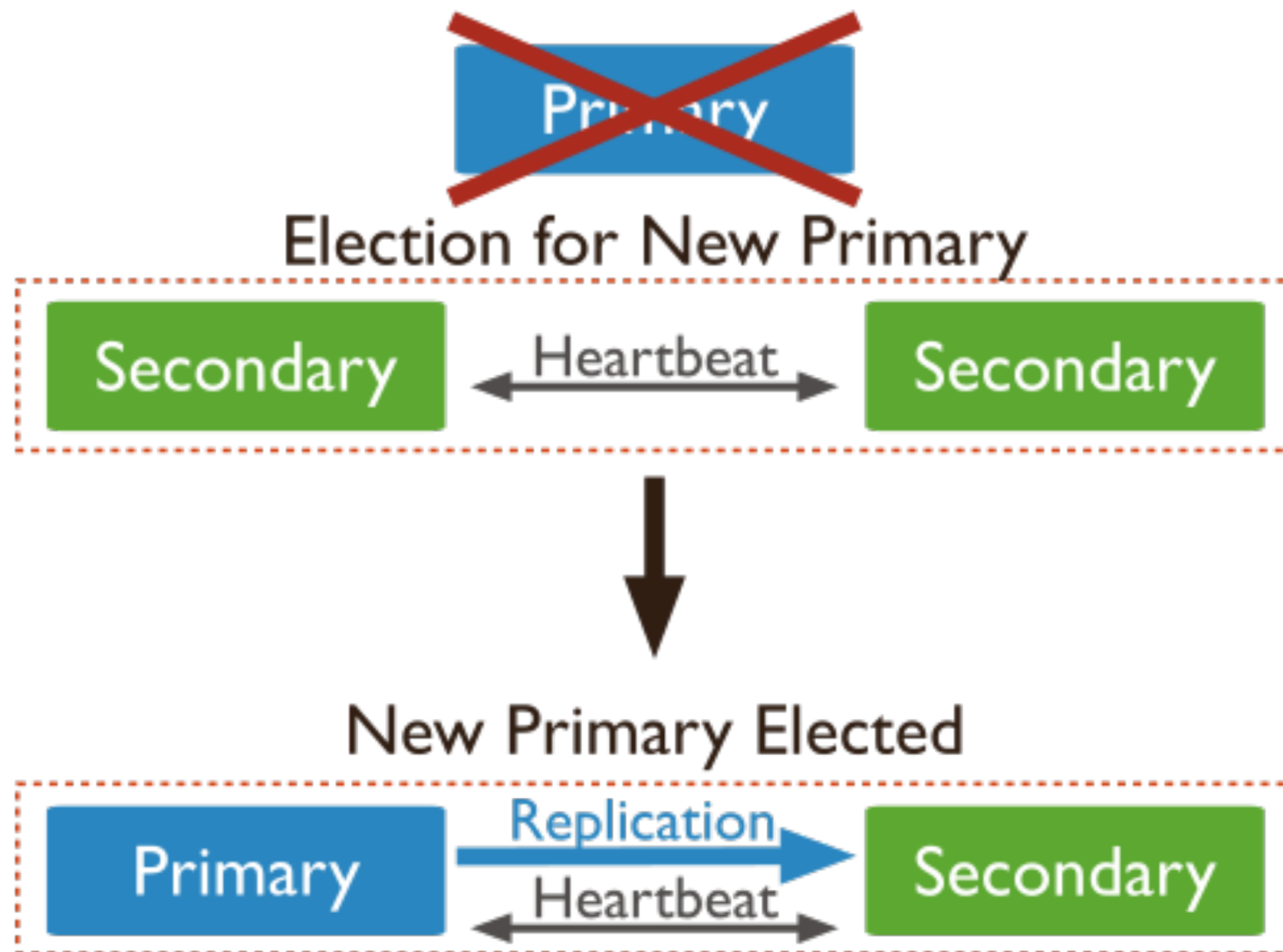
- O MongoDB permite criar instâncias replicantes
- O grupo de instancias que replicam os dados é chamado de Replica Set
- A replicação aumenta a disponibilidade. Com múltiplas cópias dos dados o sistema deixa de ter um único ponto de falha
- A replica set é um grupo de instâncias de **mongod** que armazena o mesmo conjunto de dados.

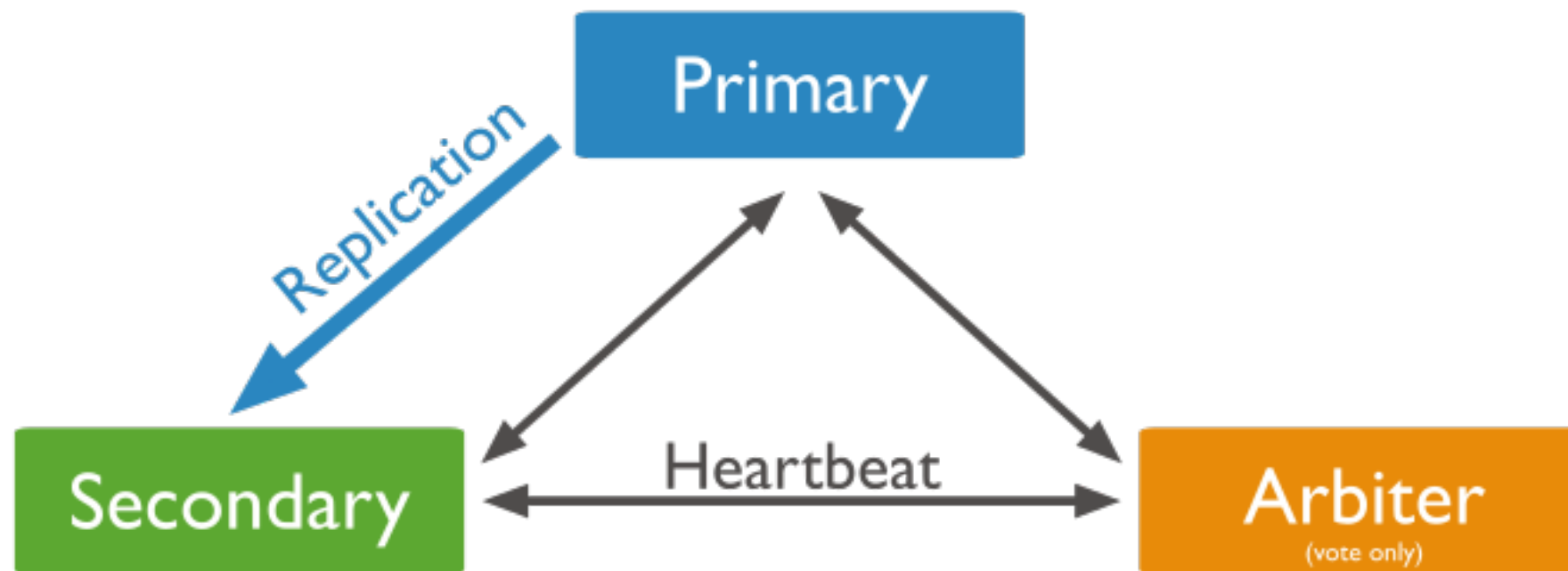




- O nó primário recebe todos o comandos de escrita
- Só pode existir um único nó primário
- Os dados são replicados nos nós secundários assincronamente pelo nó primário
- Caso o nó primário fique indisponível, haverá uma eleição entre os nós para eleger um novo nó primário
- O mínimo recomendado de nós é de três: um primário e dois secundários.



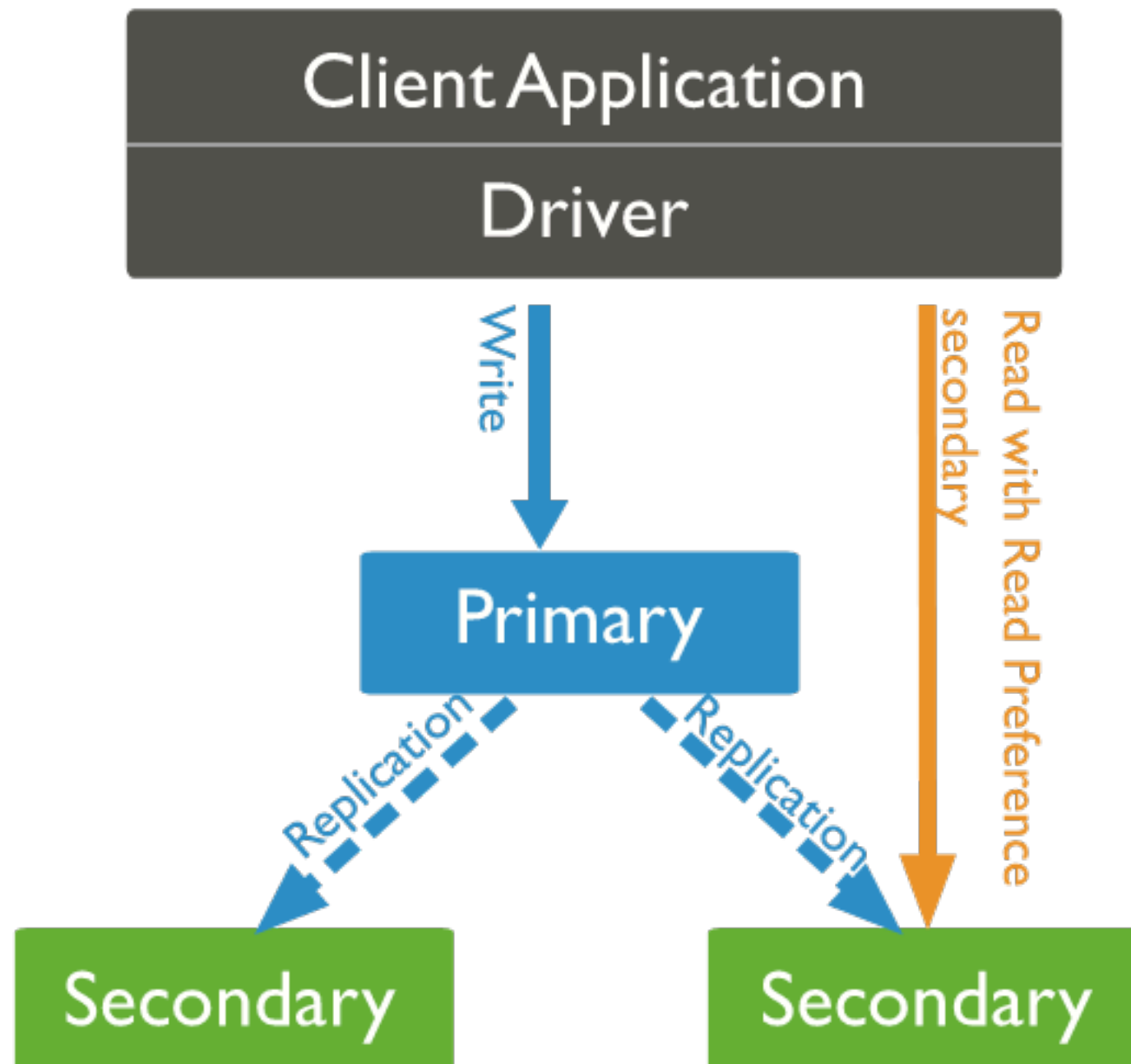


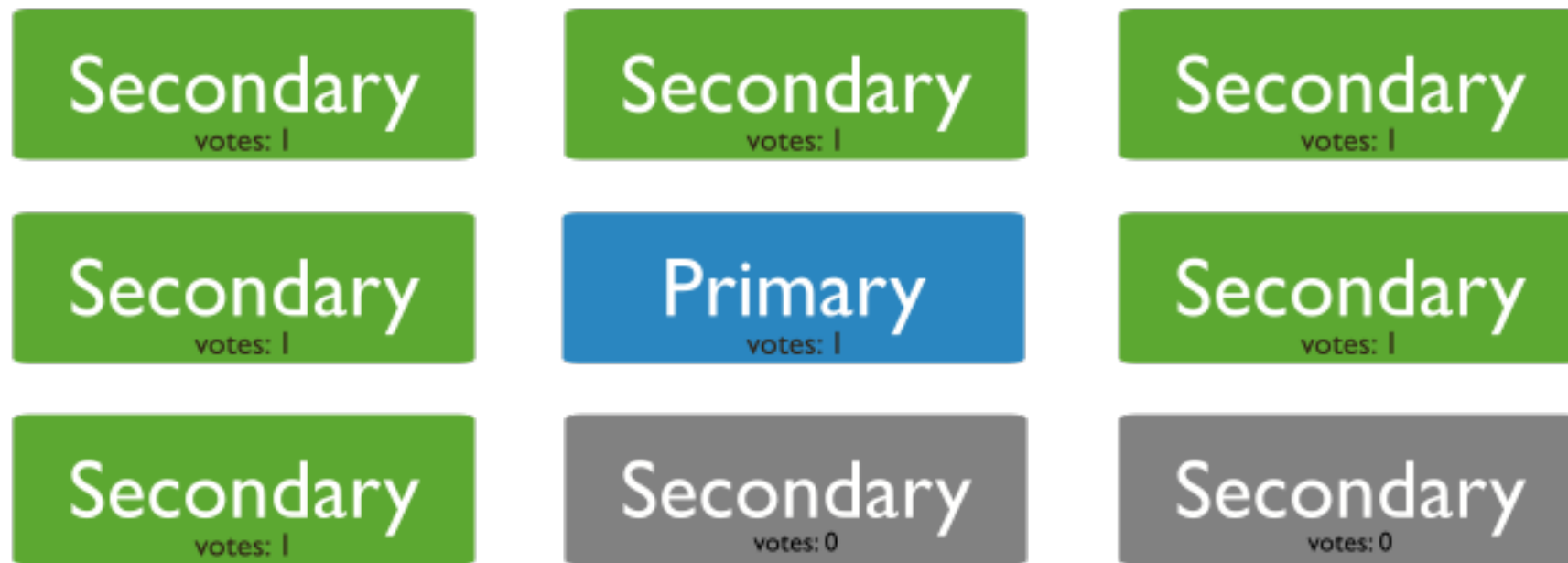


- É possível adicionar uma instancia especial chamada de árbitro
- O árbitro não armazena dados e é utilizado apenas para proporcionar quorum nas eleições
- Adicionado em cluster com número par de nós
- O árbitro não necessita de hardware dedicado



- As operações de leitura e escrita devem ser realizadas através do nó primário
- No entanto é possível realizar leitura a partir de nós secundários. Nesse caso, como a replicação funciona assincronamente, é possível que os dados não estejam plenamente atualizados

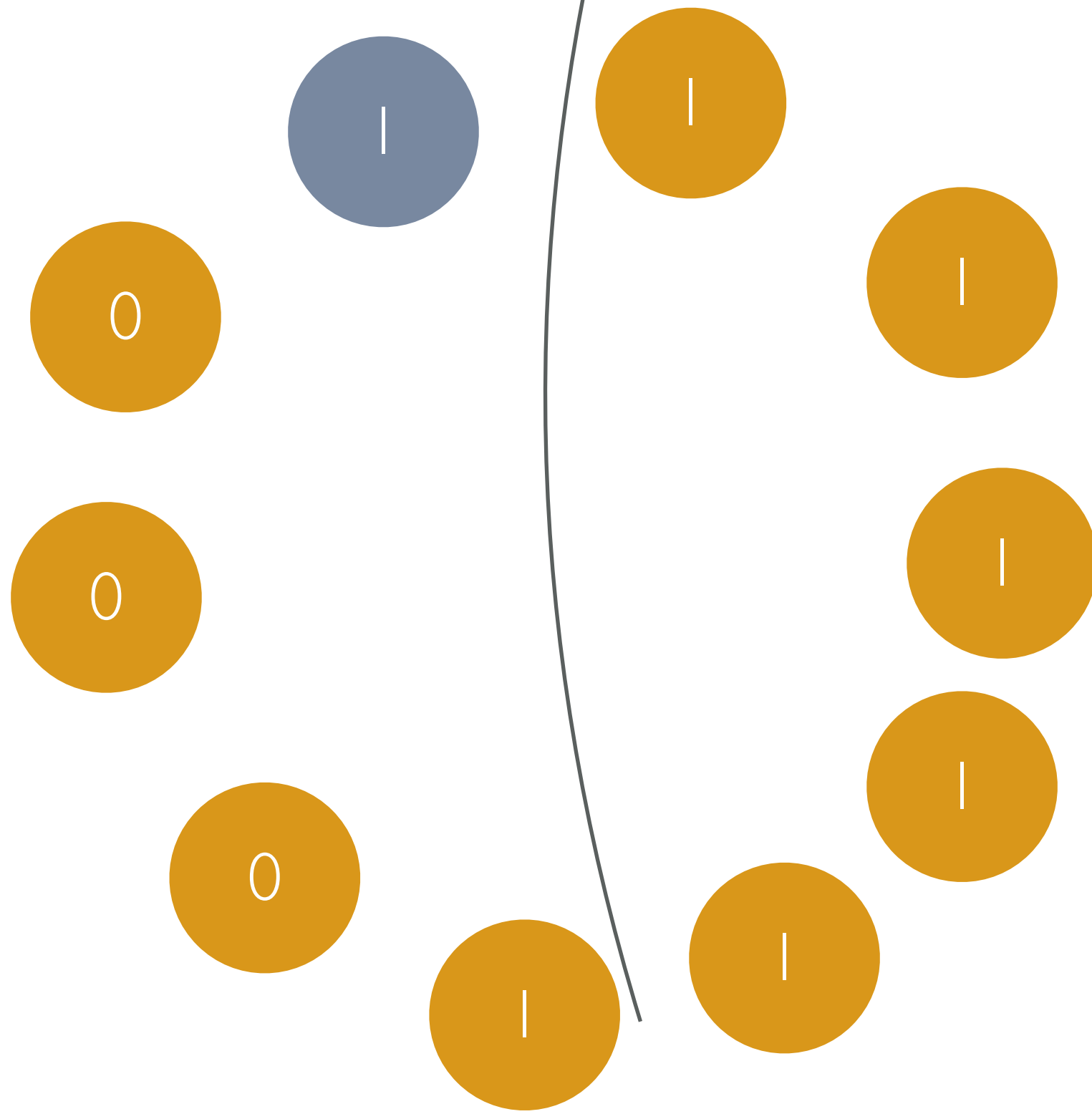




Mongo suporta até 50 nós replicantes. No entanto apenas 7 deles devem ter poder de voto.



mongoDB®

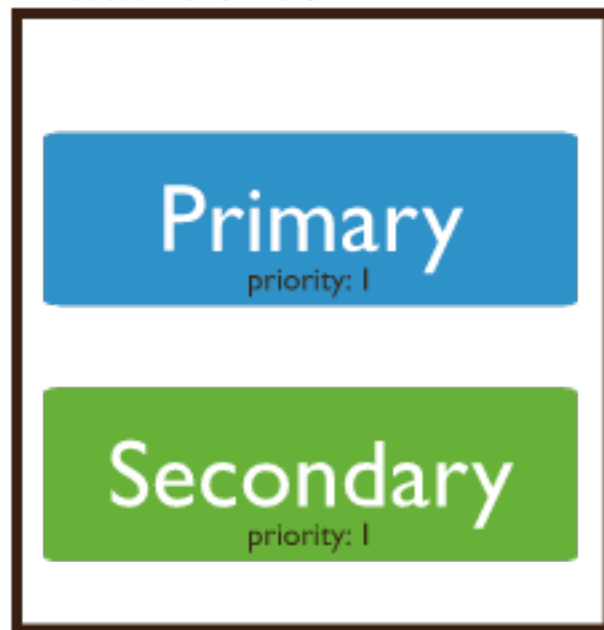




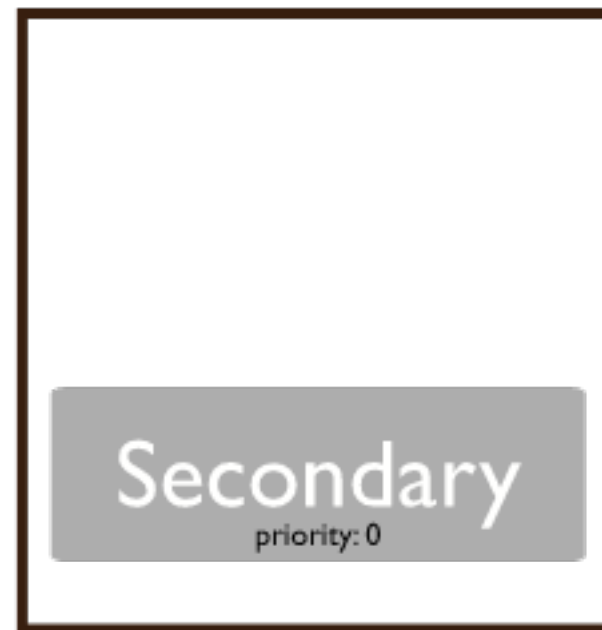
- A prioridade é número que controla a prioridade de um nó se tornar primário em caso de indisponibilidade.
- É um número entre 0 e 1000 para os nós primários e secundários.
- Especifique um número maior se o nó é mais elegível a se tornar o primário e menor se for menos elegível
- Padrão: 1.0 para primary/secondary; 0 para arbiters.



Data Center 1



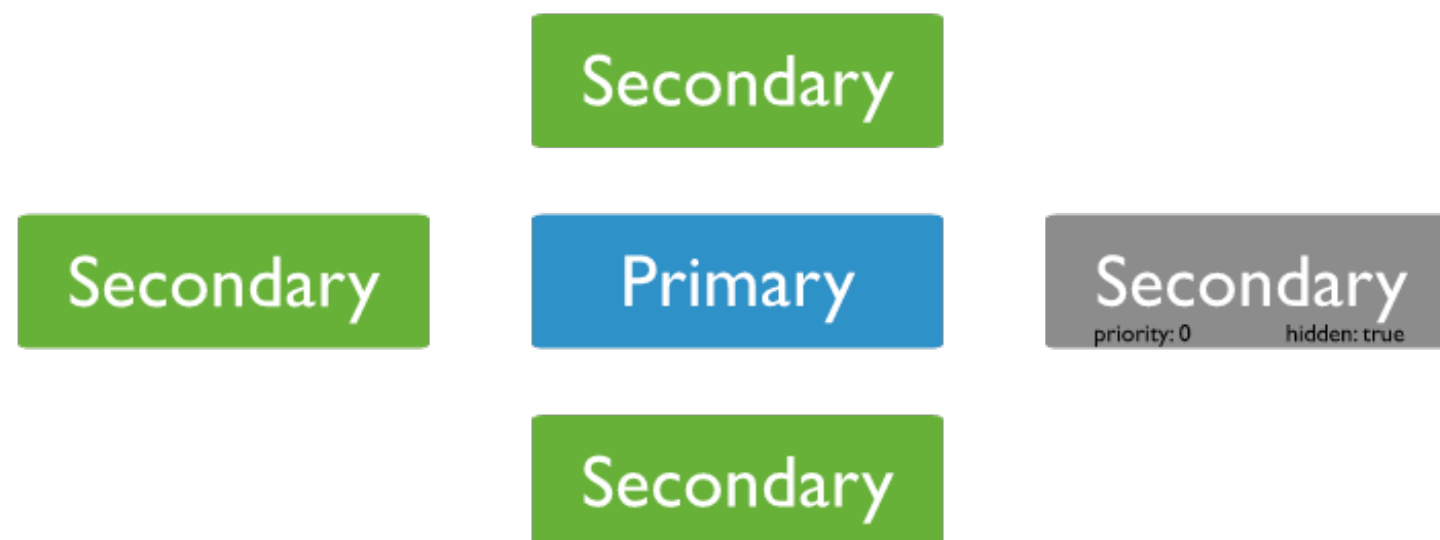
Data Center 2



Prioridade zero indica que o nó não poderá se tornar primário.
No entanto, ele permanece agindo normalmente como secundário.

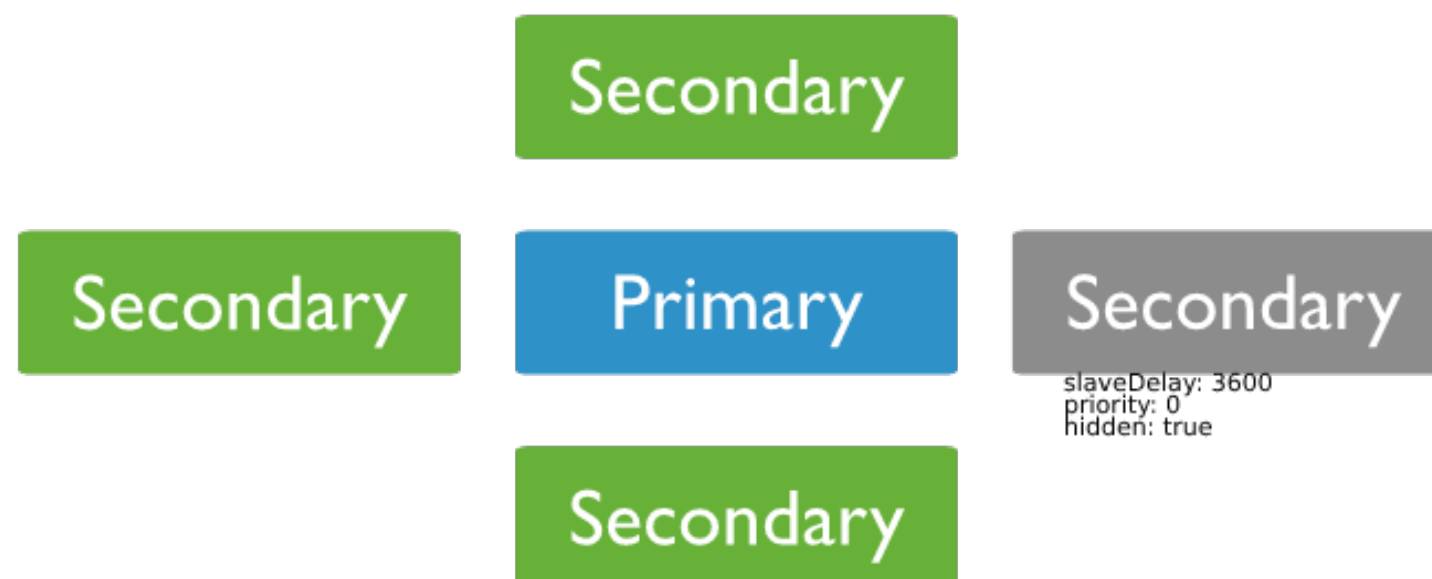
Nós ocultos (Hidden Replica Set Members)

- Um membro oculto mantém uma cópia do conjunto de dados do primário, mas é invisível para aplicativos clientes.
- Membros ocultos sempre devem ser membros de prioridade 0 e, portanto, não podem se tornar primários.
- Os clientes não distribuirão leituras para membros ocultos. Como resultado, esses membros não recebem tráfego além da replicação básica.
- Use membros ocultos para tarefas dedicadas, como relatórios e backups.



Membros atrasados (Delayed Replica Set Members)

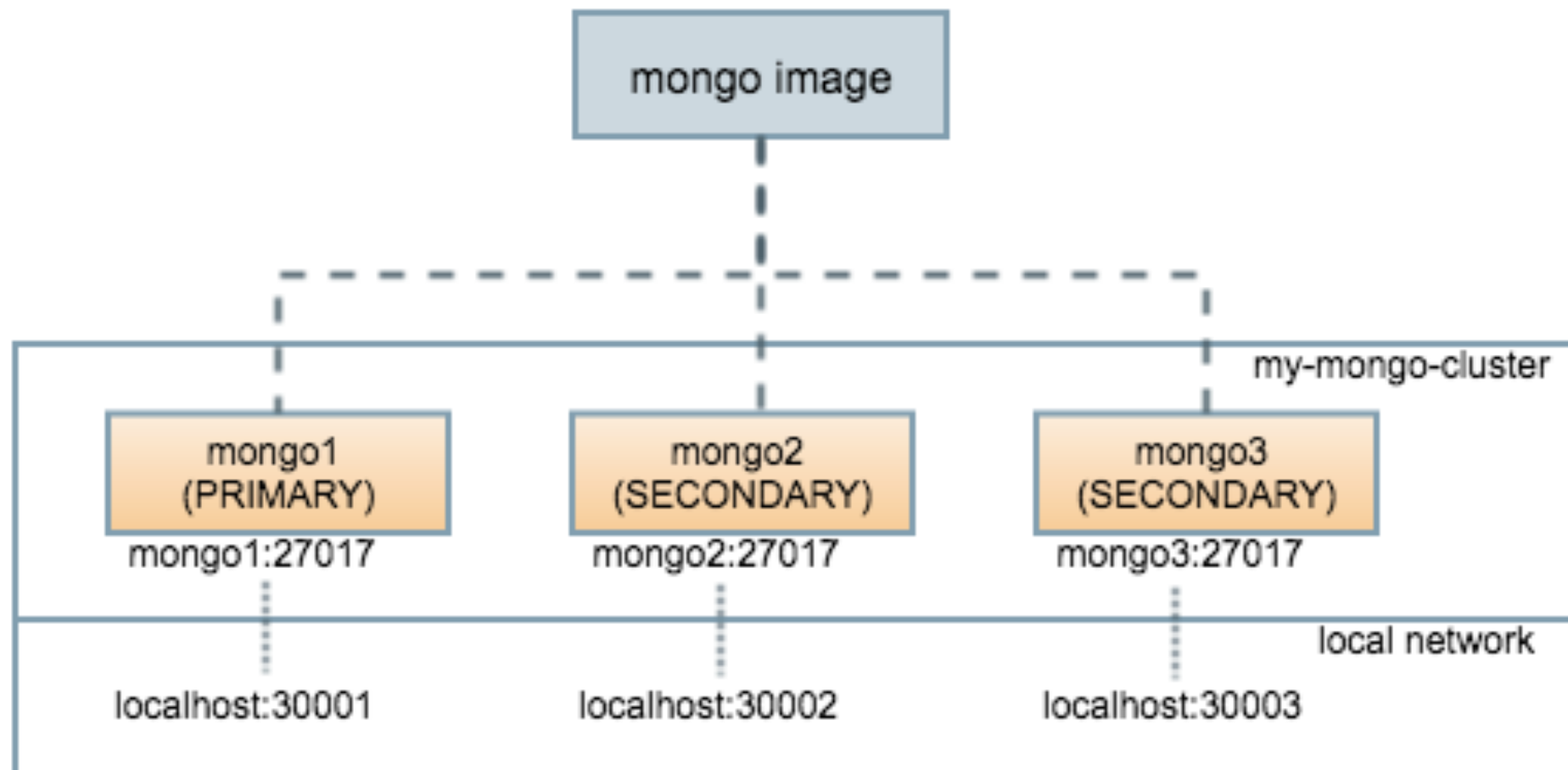
- Membros atrasados contêm cópias do conjunto de dados de um conjunto de réplicas. No entanto, o conjunto de dados de um membro atrasado reflete um estado anterior ou atrasado do conjunto.
- Por exemplo, se o horário atual for 09:52 e um membro tiver um atraso de uma hora, o membro atrasado não terá nenhuma operação mais recente que 08:52.
- Como os membros atrasados são um "backup contínuo" ou um instantâneo "histórico" em execução do conjunto de dados, eles podem ajudá-lo a se recuperar de vários tipos de erros humanos.
- Por exemplo, um membro atrasado pode possibilitar a recuperação de atualizações malsucedidas de aplicativos e erros do operador, incluindo *drop* de bancos de dados ou coleções.



Tolerância a falhas

Number of Members	Majority Required to Elect a New Primary	Fault Tolerance
3	2	1
4	3	1
5	3	2
6	4	2

Na prática...





Checando as interfaces

```
$ docker network ls
```

NETWORK ID	NAME	DRIVER	SCOPE
2a4e341c6039	bridge	bridge	local
4fbef5286425	host	host	local
8062e4e7cdca	none	null	local



Criando Interface de rede para o cluster

```
docker network create my-mongo-cluster
```



Checando novamente as interfaces

```
$ docker network ls
```

NETWORK ID	NAME	DRIVER	SCOPE
2a4e341c6039	bridge	bridge	local
4fbef5286425	host	host	local
f65e93c94e42	mongo-cluster	bridge	local
8062e4e7cdca	none	null	local



Inicializando o mongo l

```
docker run -p 30001:27017 --name mongol --net my-mongo-  
cluster -d mongo mongod --replSet mongo-rs
```



Inicializando o mongo2

```
docker run -p 30002:27017 --name mongo2 --net my-mongo-cluster -d mongo mongod --replSet mongo-rs
```



Inicializando o mongo3

```
docker run -p 30003:27017 --name mongo3 --net my-mongo-cluster -d mongo mongod --replSet mongo-rs
```




Configurando replicação

```
docker exec -it mongol mongo
```

```
> rs.initiate()  
> cfg = rs.conf()  
> cfg.members[0].host = "mongo1:27017"  
> rs.reconfig(cfg)  
> rs.add("mongo2:27017")  
> rs.add("mongo3:27017")  
> rs.status();
```

Configurando nós ocultos

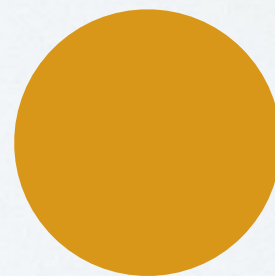
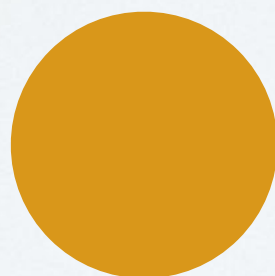
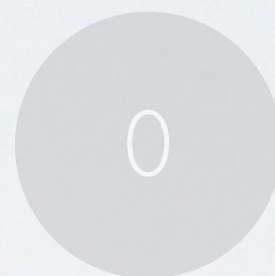
```
cfg = rs.conf()
cfg.members[0].priority = 0
cfg.members[0].hidden = true
rs.reconfig(cfg)
```

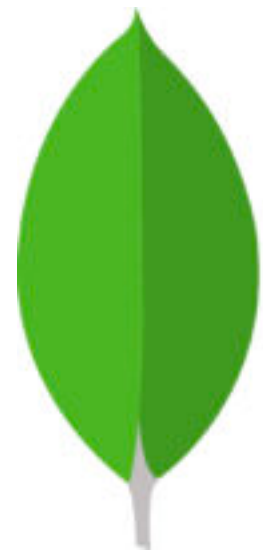
[copy](#)

Configurando replicas atrasadas

```
cfg = rs.conf()
cfg.members[0].priority = 0
cfg.members[0].hidden = true
cfg.members[0].slaveDelay = 3600
rs.reconfig(cfg)
```

[copy](#)





mongoDB

Banco de dados NoSQL - Replicaset

Prof. Gustavo Leitão