



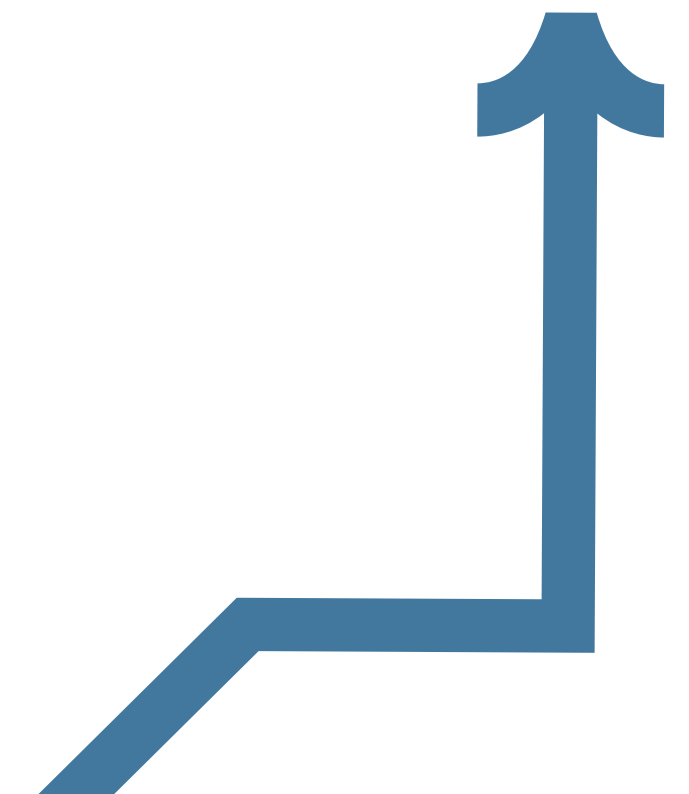
La Base de Données Clé-Valeur Distribuée

PRÉSENTÉ PAR :

Hafid GARHOUM

ENCADRÉ PAR :

Pr. Lamia KARIM



PLAN

- 01 Introduction à etcd
- 02 Pourquoi maîtriser etcd ?
- 03 Étude de Cas : etcd dans Kubernetes
- 04 Fonctionnement d'etcd
- 05 Travaux pratiques
- 06 Projet : Système de vote distribué en temps réel
- 07 Références



1. Introduction à etcd

etcd ?

Base de données clé-valeur

distribuée

fortement consistante

Ecrit en Golang



**etcd est un projet CNCF
open source**



Utilisé par :



kubernetes



CoreDNS



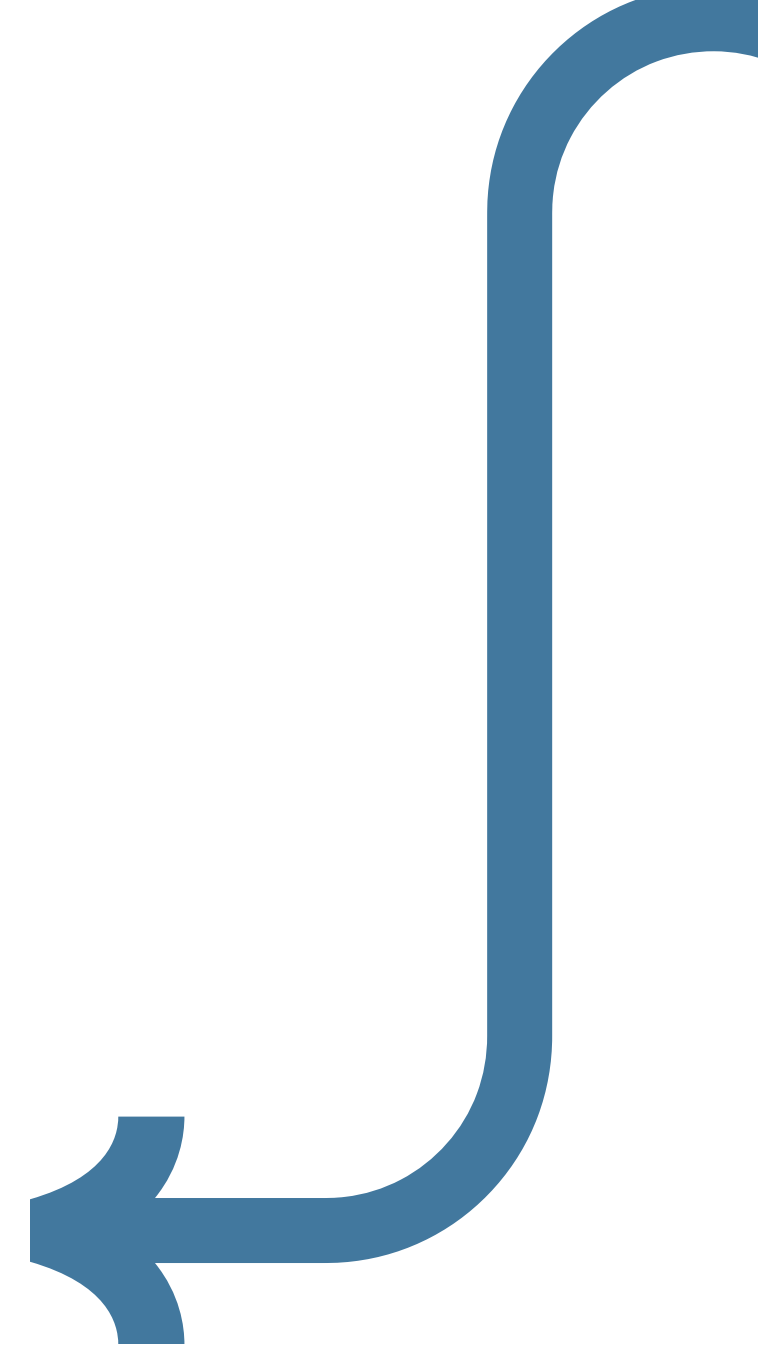
ROOK

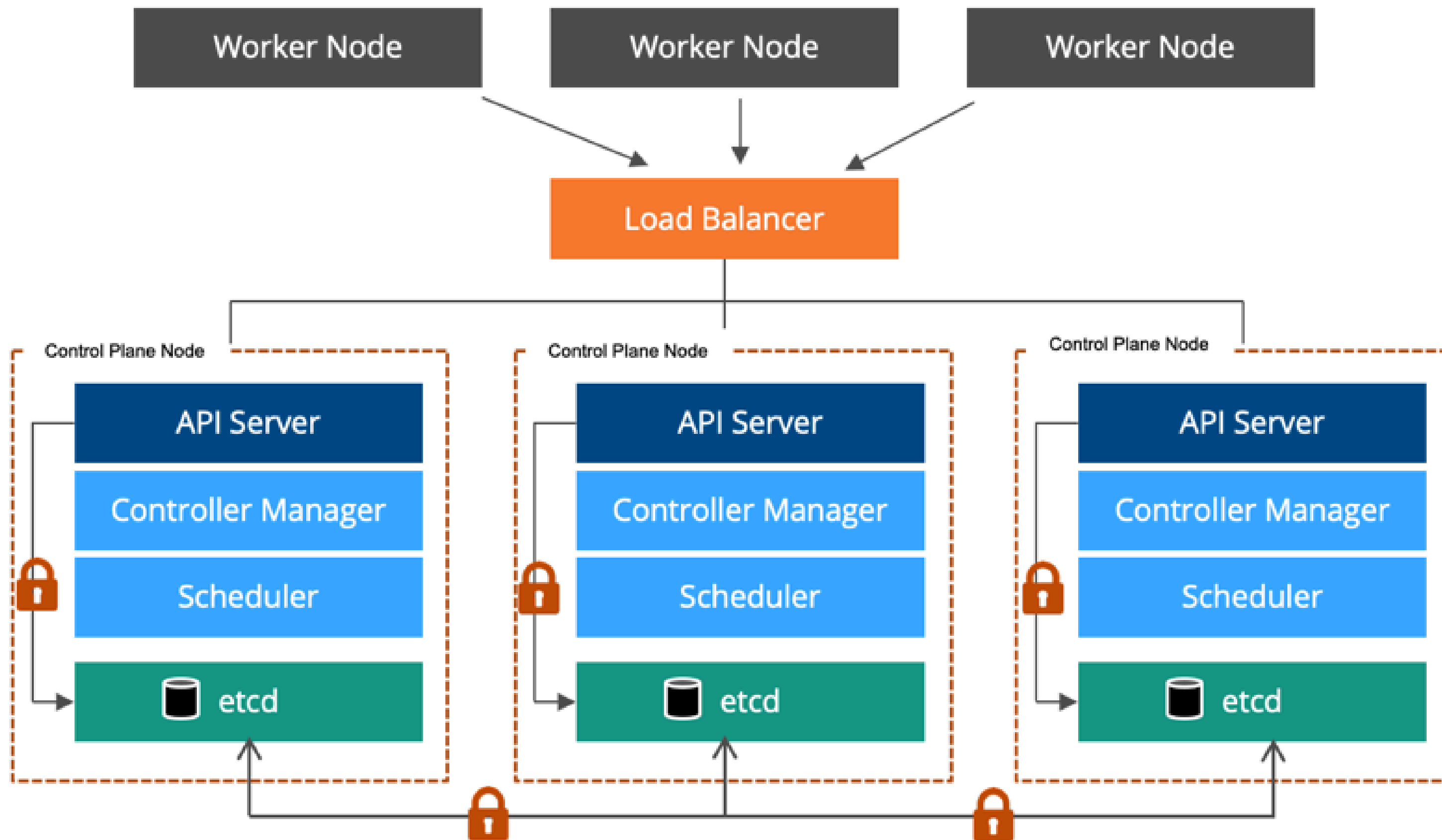


2. Pourquoi maîtriser etcd ?

- **Similaire à ZooKeeper**, utilisé dans les clusters Hadoop pour la coordination.
- **Cœur de Kubernetes** : etcd stocke tout l'état du cluster (pods, services, configurations...).
- **Fiable et cohérent grâce à l'algorithme Raft** : idéal pour les systèmes critiques.
- **Parfait pour les systèmes distribués** : assure synchronisation et coordination automatique entre nœuds.
- **API simple et rapide** : basée sur gRPC/JSON, facile à intégrer dans les infrastructures modernes.

3. Étude de Cas : etcd dans Kubernetes





1. etcd stocke tout l'état du cluster :

- Déploiements, pods, services, secrets, configmaps...
- Informations sur les nœuds, quotas, RBAC, etc.

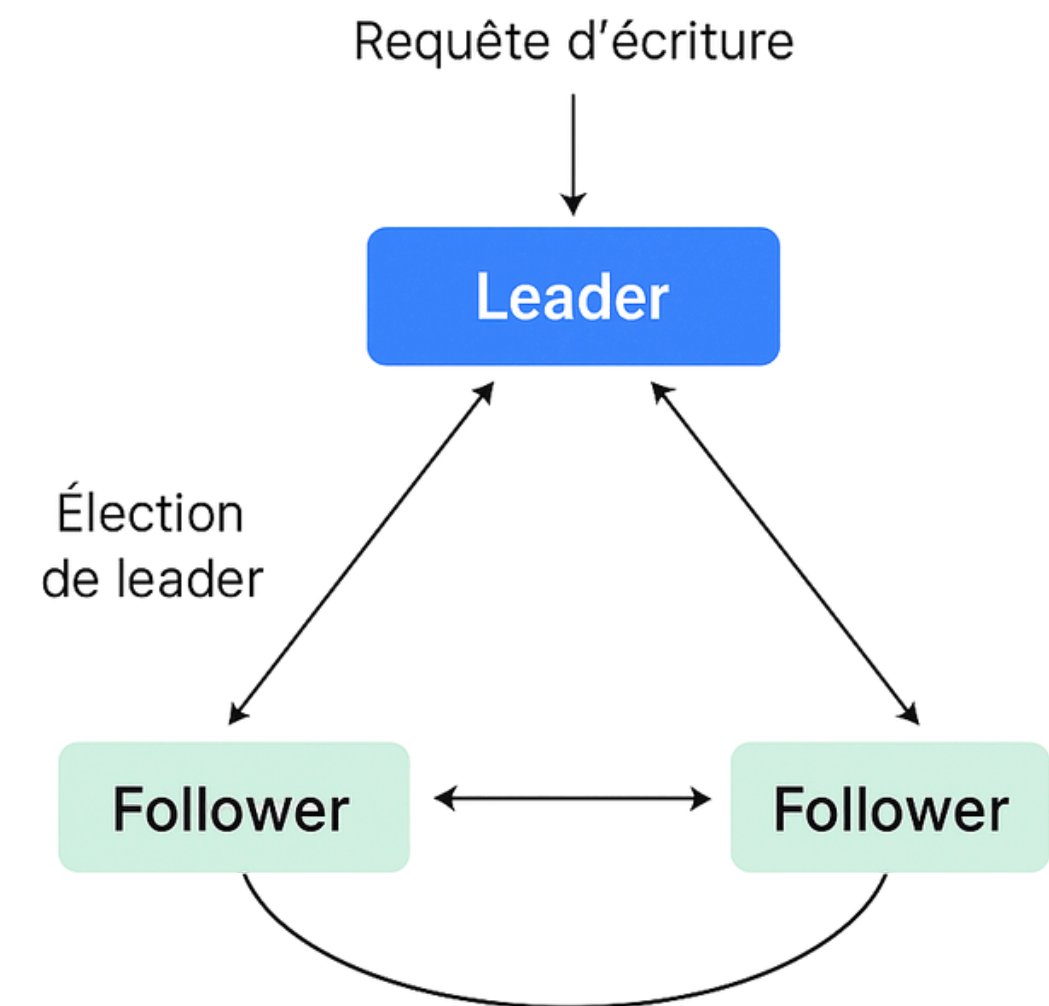
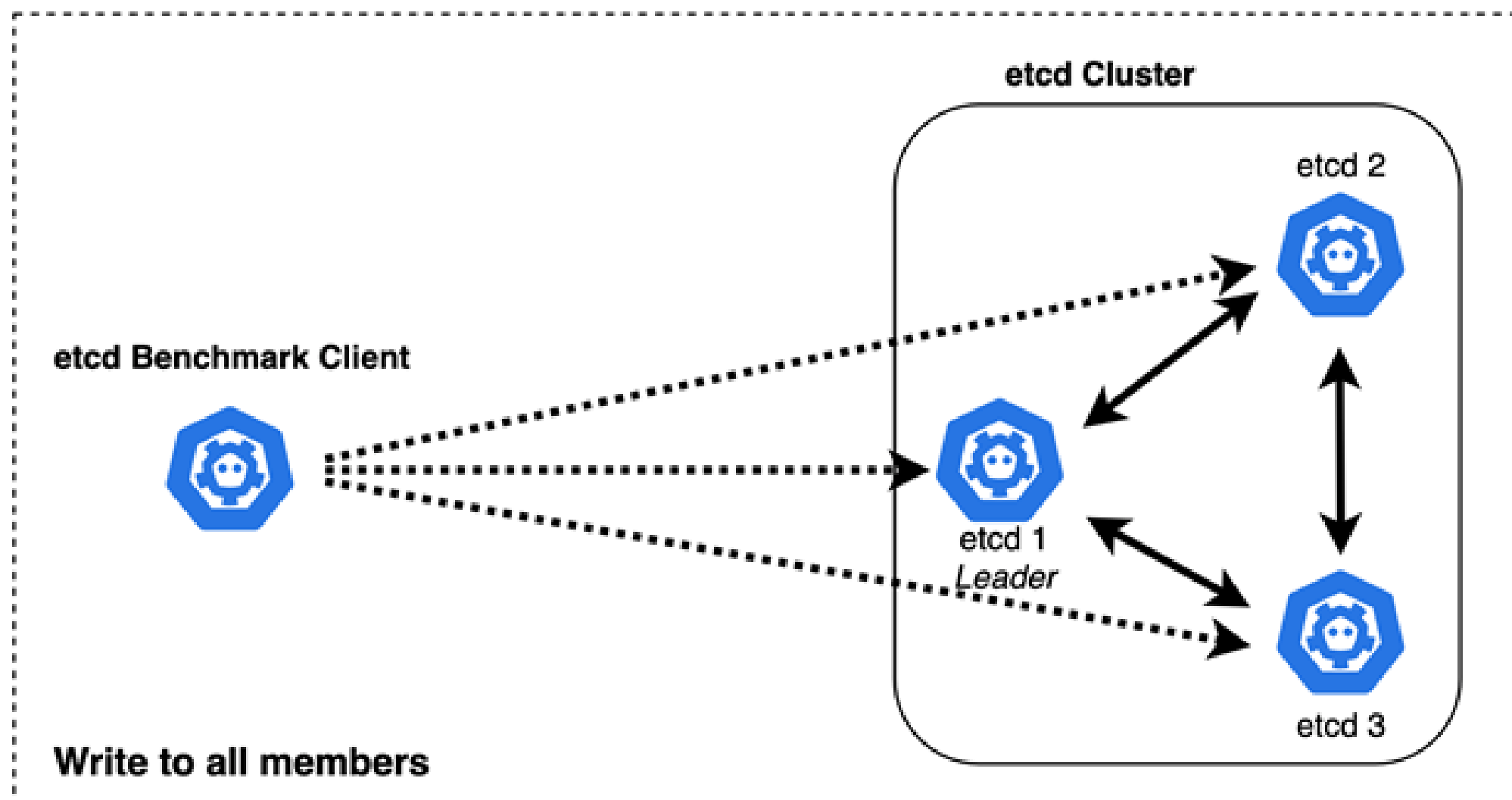
2. À chaque changement (ex : création d'un pod), Kubernetes met à jour etcd.

3. Les composants maîtres de Kubernetes (API server, scheduler, controller manager) lisent et écrivent dans etcd en permanence.



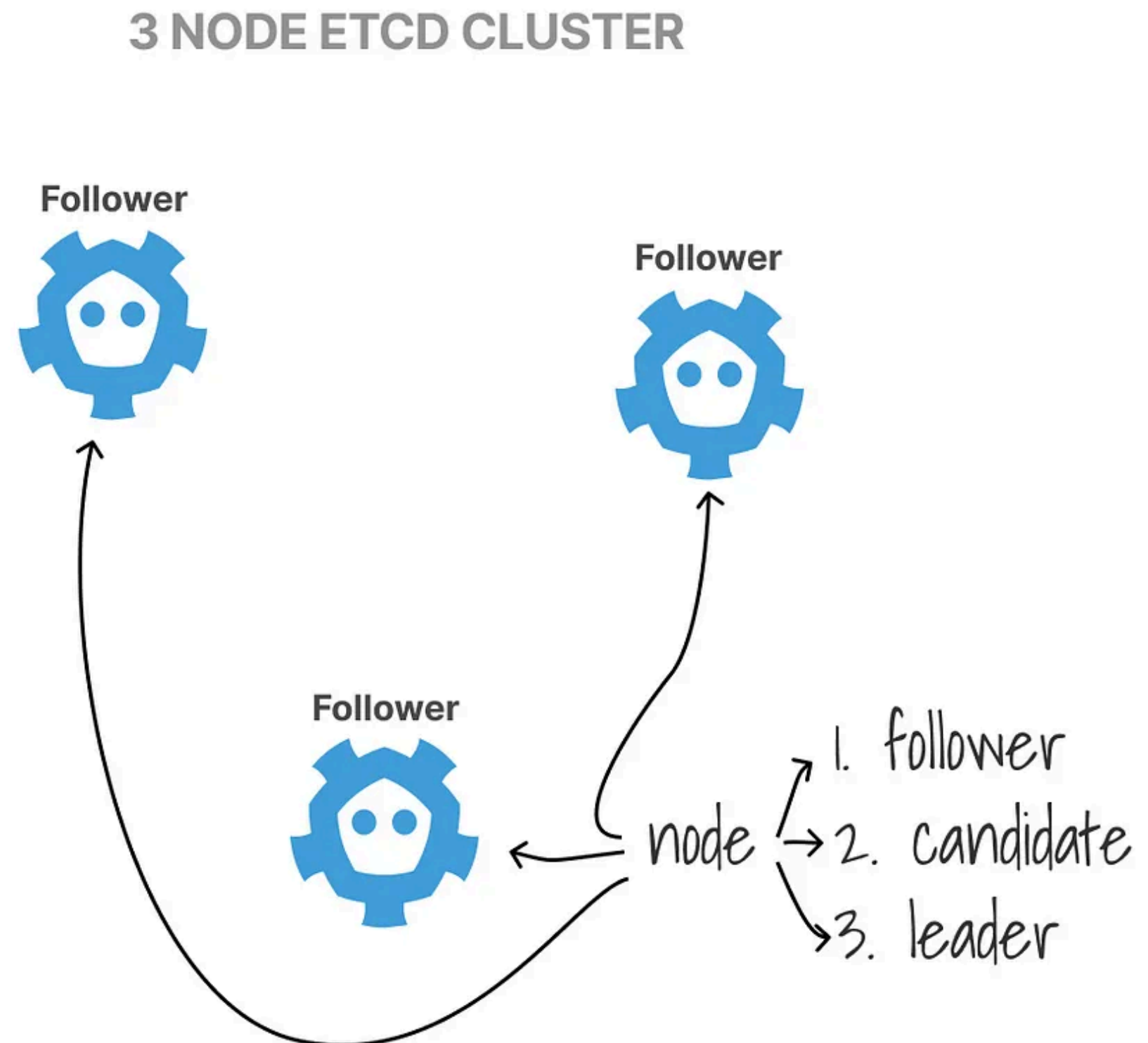
4. Fonctionnement d'etcd

4.1 etcd cluster



4.2 Algorithme Raft : Cohérence forte

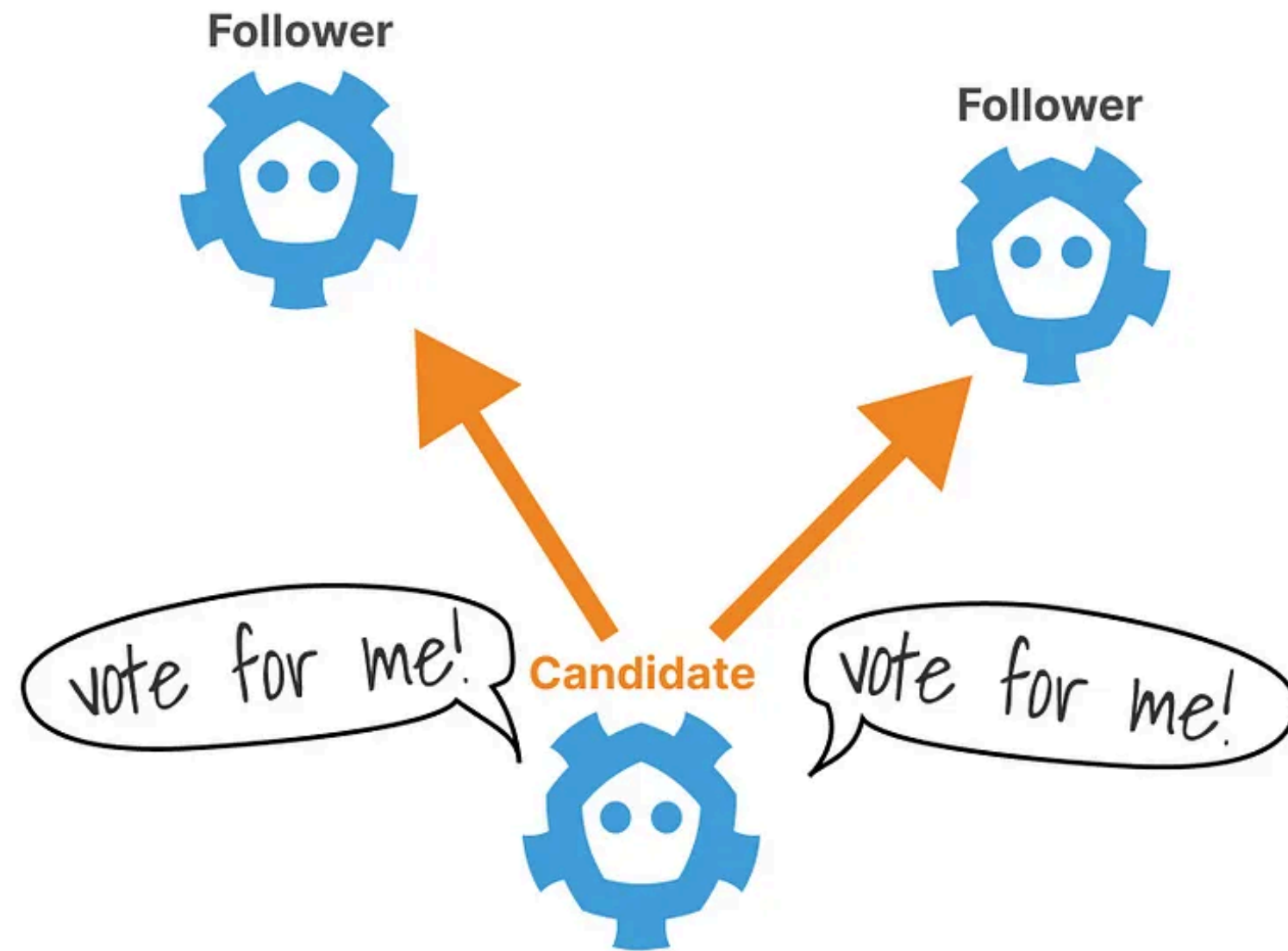
Tout d'abord, tous les nœuds démarrent dans l'état Follower.



4.2 Algorithme Raft : Cohérence forte

Les nœuds répondent avec leur vote.

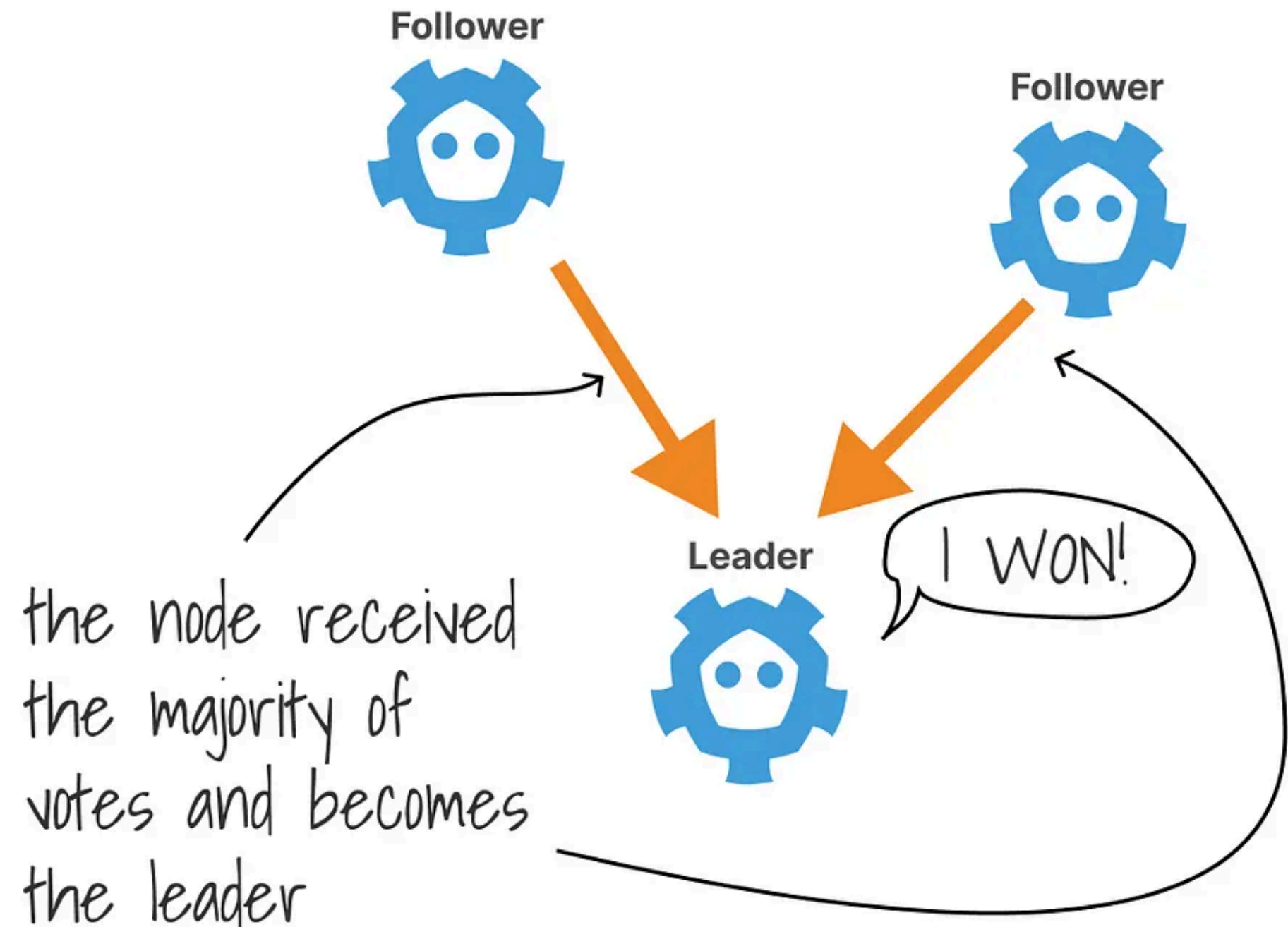
3 NODE ETCD CLUSTER (ELECTION)



4.2 Algorithme Raft: Cohérence forte

Le candidat qui obtient la majorité des voix devient le Leader.

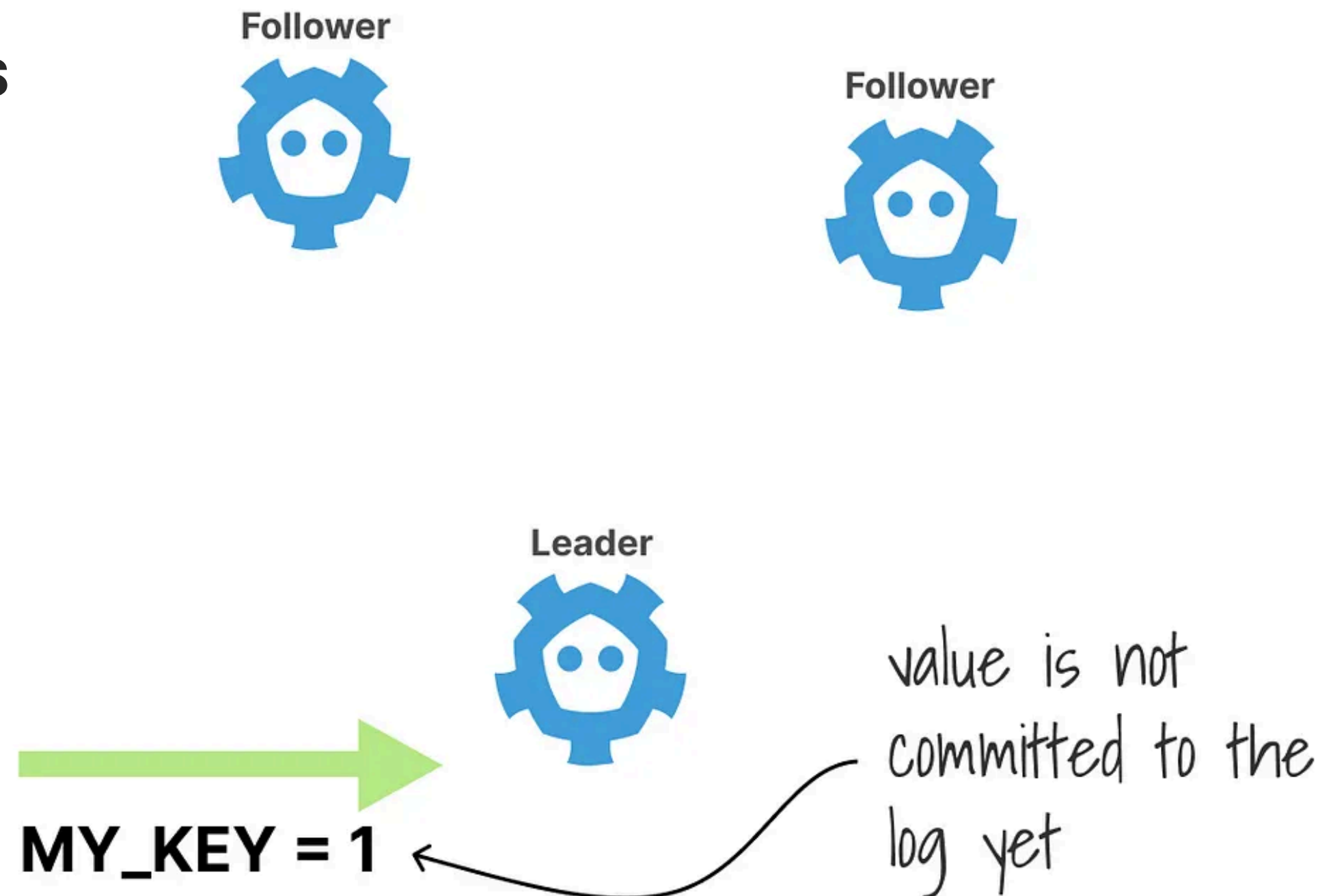
3 NODE ETCD CLUSTER (LEADER ELECTED)



4.2 Algorithme Raft : Cohérence forte

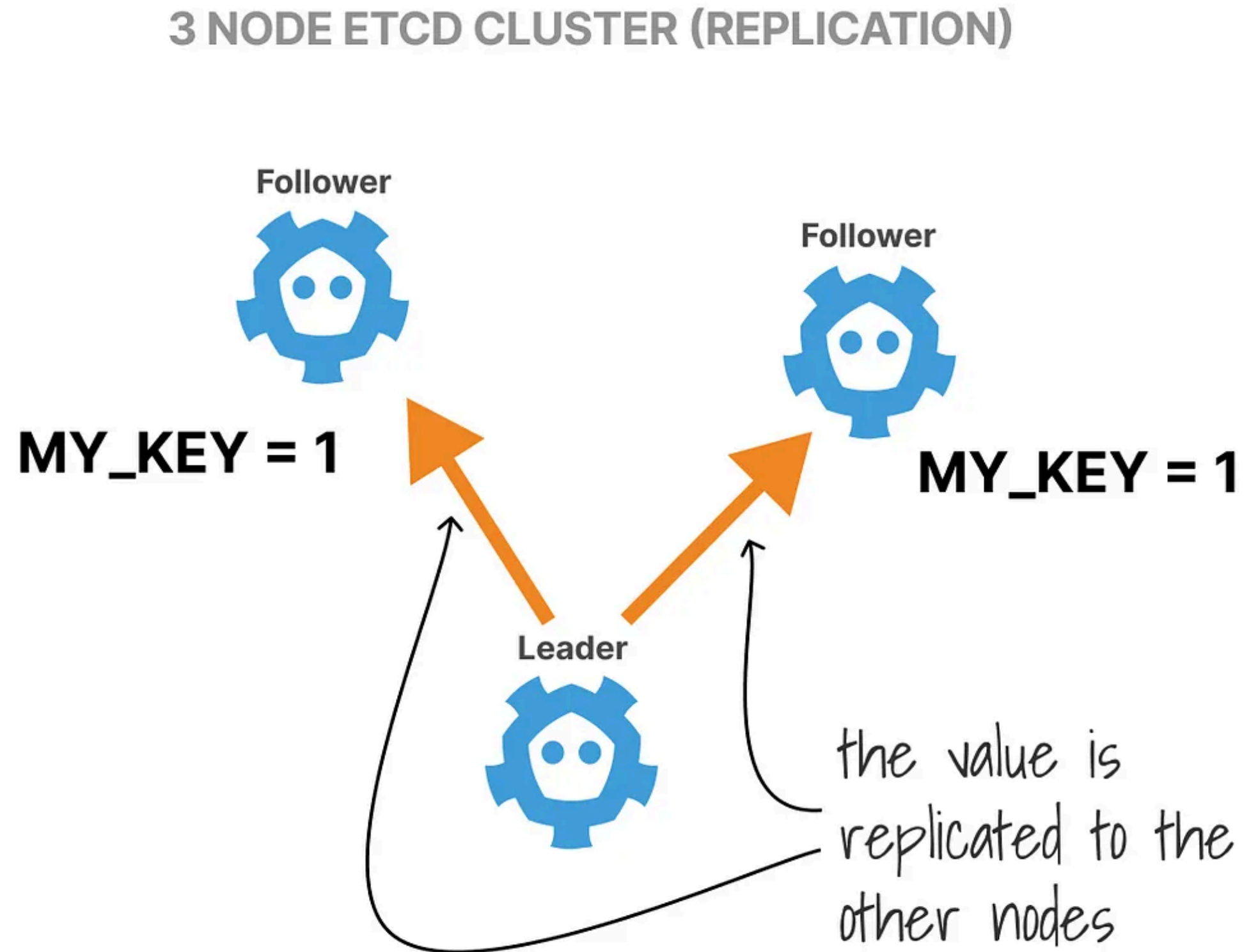
Toutes les demandes d'écriture sont redirigées vers le leader.

3 NODE ETCD CLUSTER (REQUEST TO WRITE)



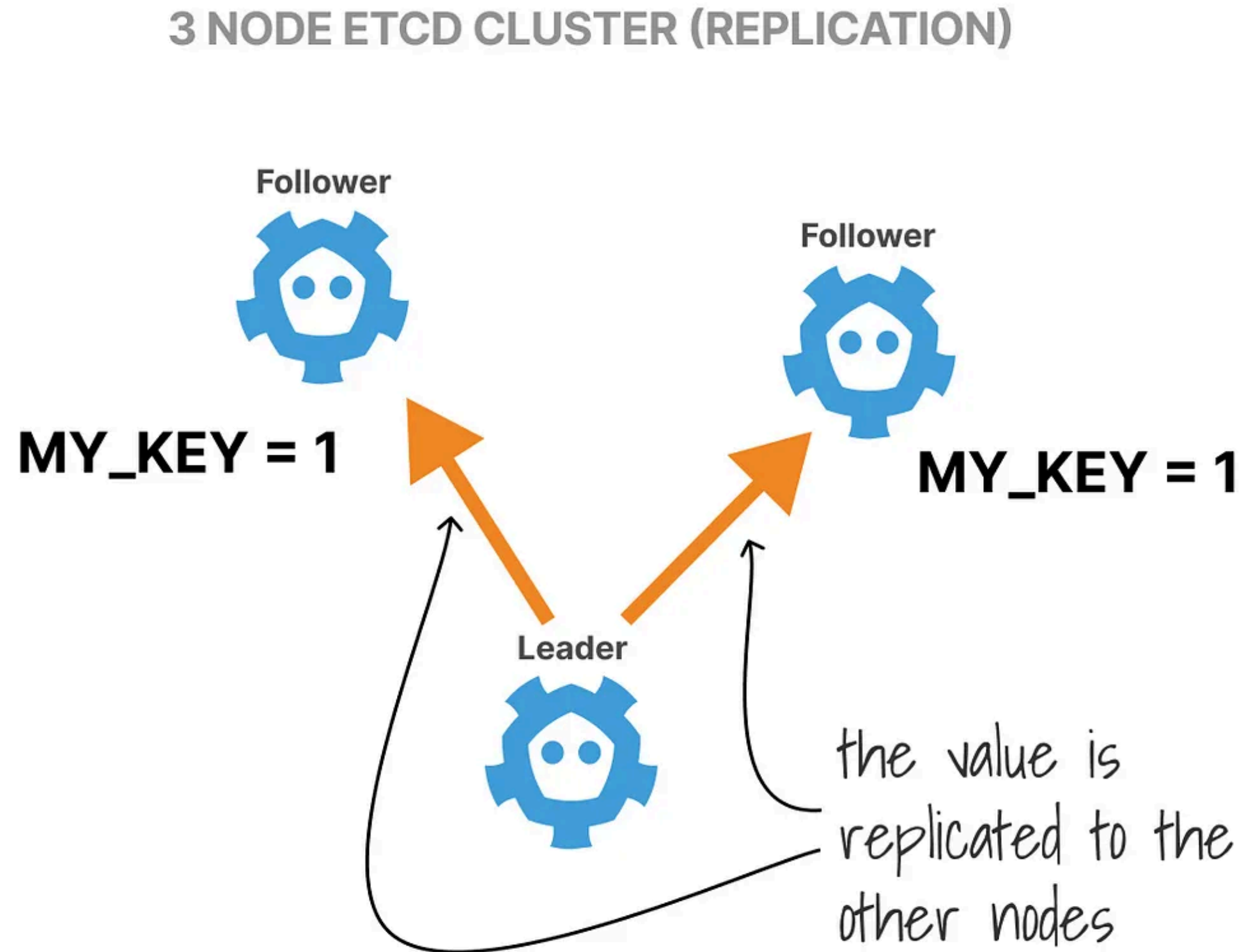
4.2 Algorithme Raft : Cohérence forte

the Leader replicates the value to the rest of the (followers) nodes.



4.2 Algorithme Raft : Cohérence forte

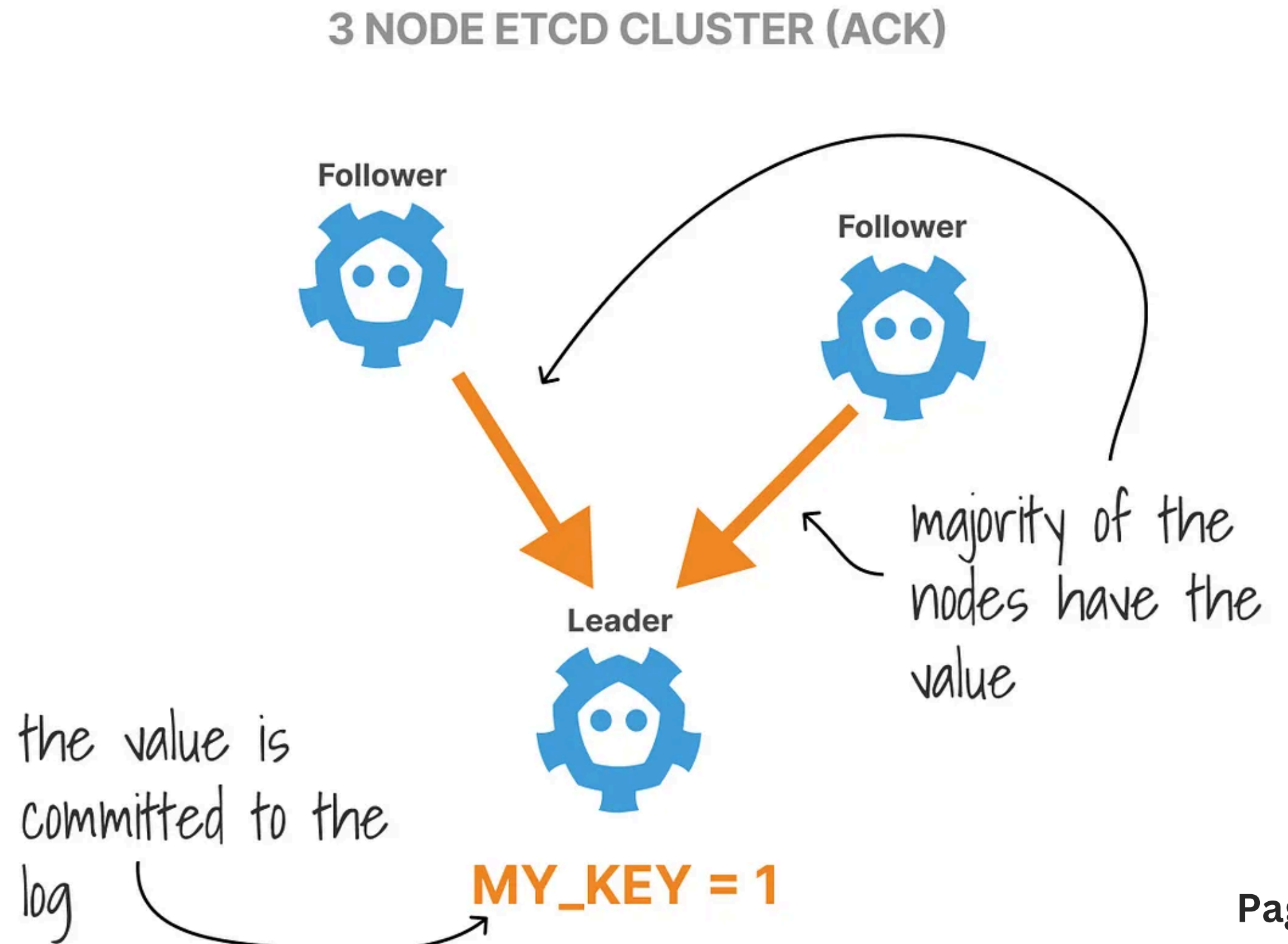
le Leader réplique la valeur sur le reste des nœuds (suiveurs).



4.2 Algorithme Raft : Cohérence forte

Enfin, le leader attend qu'une majorité de nœuds Follower ait écrit l'entrée et commit (valide) la valeur.

Une nouvelle élection est organisée si le Leader du cluster tombe pour une raison quelconque.



4.3 Algorithme Raft: Tolérance aux pannes

most common setup has 3 nodes

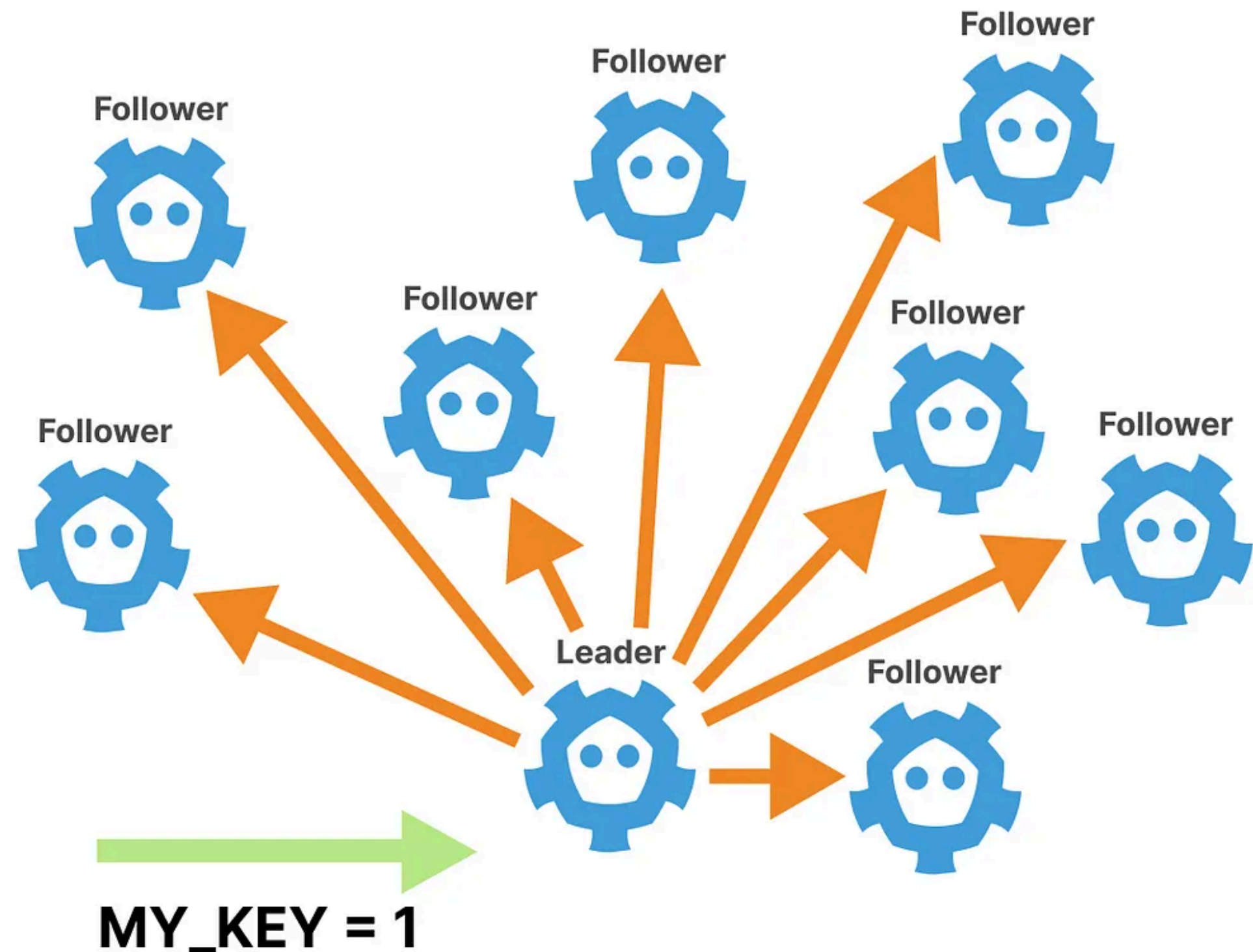


CLUSTER SIZE N	FAILURE TOLERANCE $T = (N-1) / 2$	MAJORITY $M = (N/2) + 1$
1	0	1
2	0	2
3	1	2
4	1	3
5	2	3
6	2	4
7	3	4
8	3	5
9	4	5
10	4	6

Raft continue à fonctionner même si $N/2$ nœuds tombent (dans un cluster de N nœuds impairs).

4.3 Algorithme Raft: Tolérance aux pannes

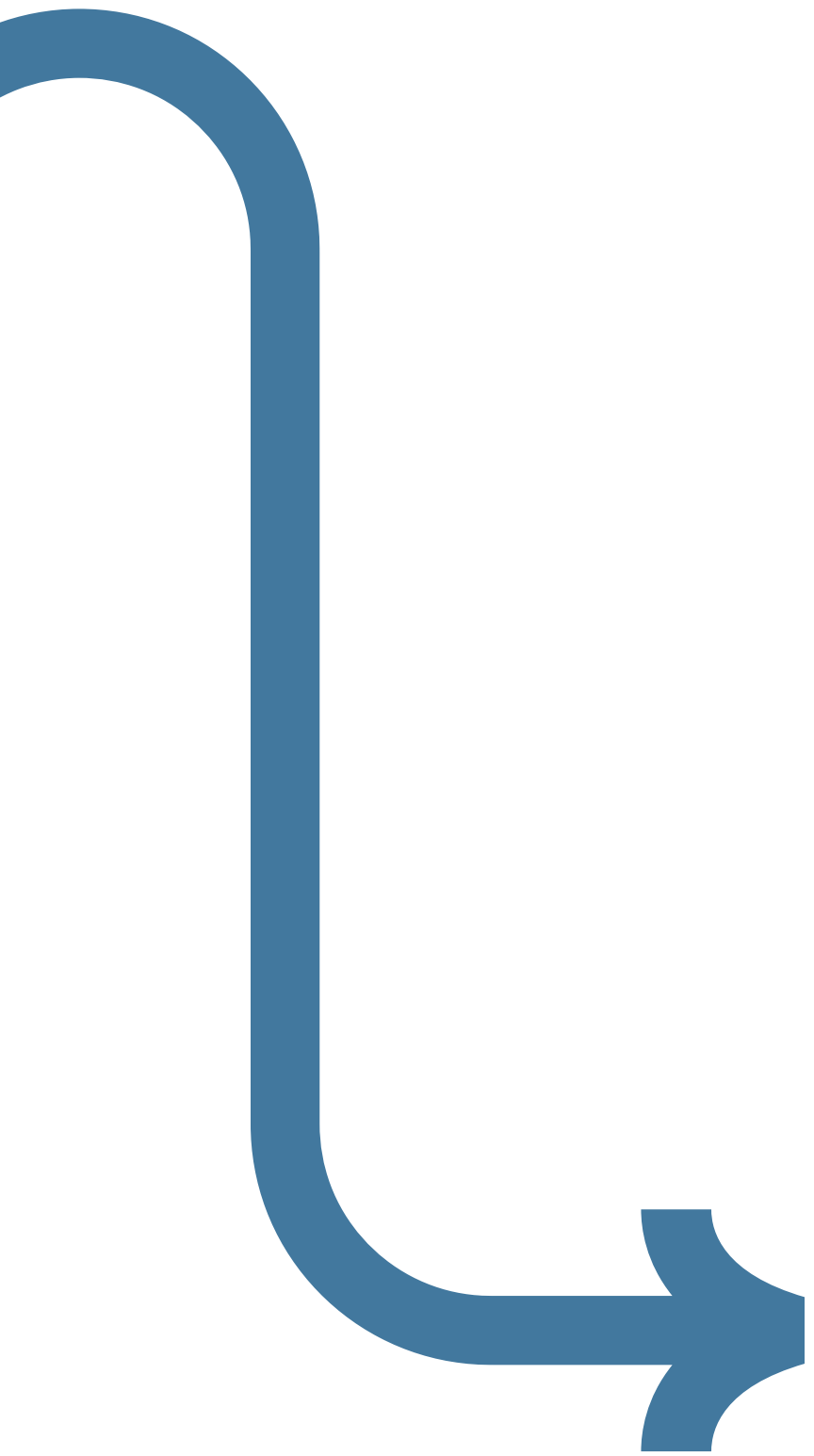
Plus il y a d'abonnés dans le cluster, plus il faut de temps pour parvenir à un consensus.



4.4 Comparaison entre etcd et redis



Caractéristique	etcd / consul	Redis
Modèle	Clé-valeur simple	Clé-valeur, en mémoire
Cohérence	✓ Forte (Raft)	✗ Faible
Tolérance aux pannes	Tolérance aux pannes	✓ Oui (cluster Raft)
Cas d'usage principal	Système distribué	Caching, sessions
Persistance	Oui	Optionnelle



5. Travaux Pratiques

- **TP1 : Installation locale et opérations CRUD avec etcdctl**
 - **Objectif** : prise en main rapide et manipulation des clés
 - **Environnement** : Ubuntu/Debian, etcd en local
- **TP2 : Déploiement d'un cluster etcd à 3 nœuds**
 - **Objectif** : comprendre l'architecture distribuée
 - **Contient** : configuration complète, test de réplication, simulation de panne, reprise automatique



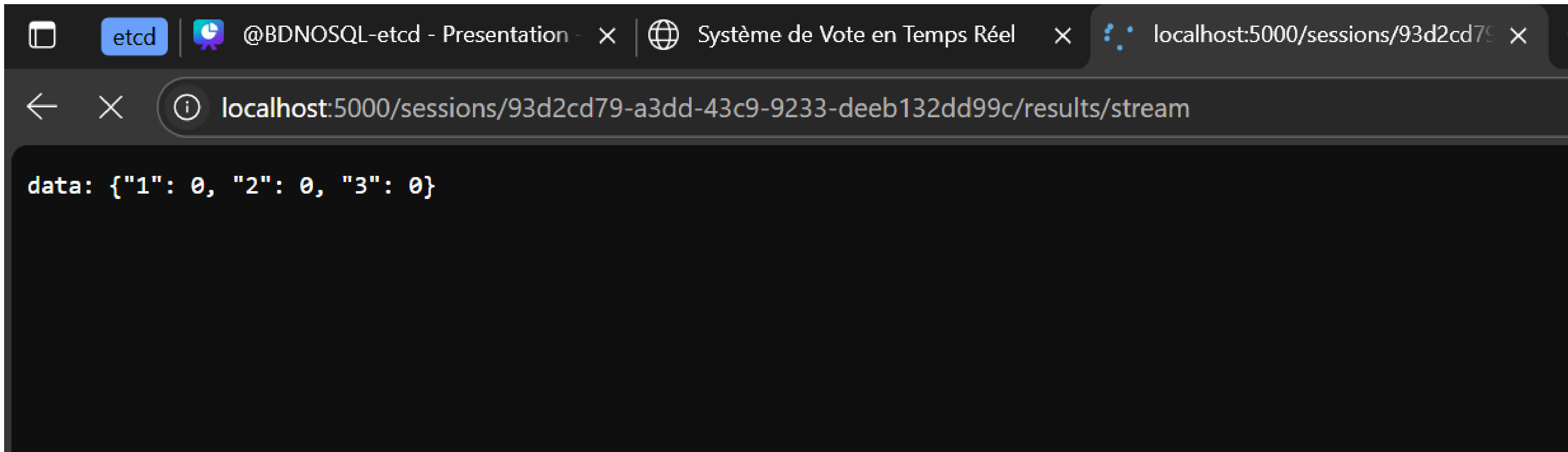
6. Projet : Système de vote distribué en temps réel

- On crée une nouvelle session de vote

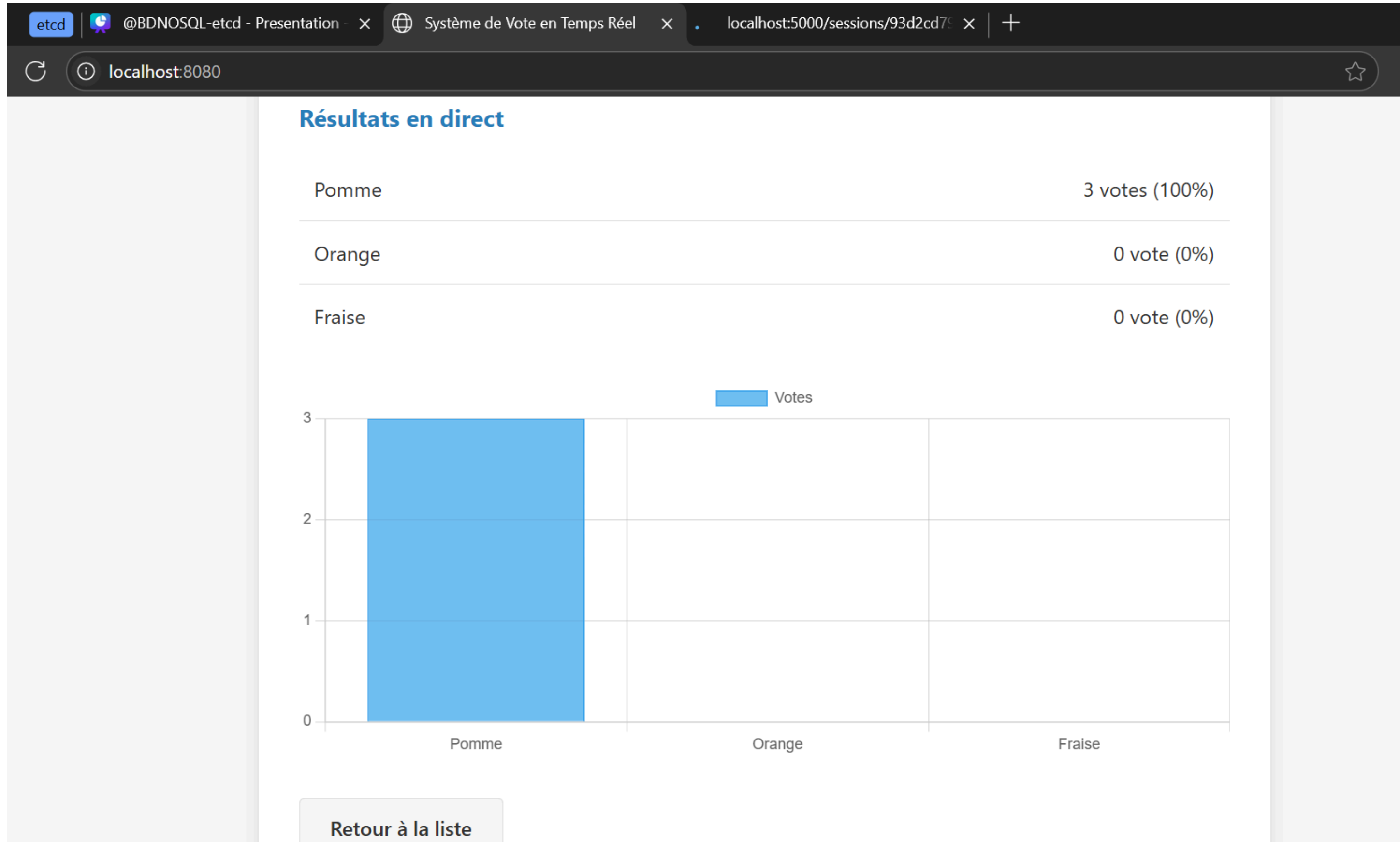
```
80 -> 2379-2380/tcp -> projet_systeme_vote-ecdc-1
hafid@DESKTOP-VCL7S4H:/mnt/c/Users/hp/Downloads/Projet_BD_NoSQL/Projet_systeme_vote$ curl -X POST http://localhost:5000/sessions \
-H "Content-Type: application/json" \
-d '{
  "title": "Meilleur fruit",
  "options": ["Pomme", "Orange", "Fraise"]
}'
{
  "info": {
    "active": true,
    "created_at": 1747588087.138111,
    "description": "",
    "title": "Meilleur fruit"
  },
  "options": [
    "Pomme",
    "Orange",
    "Fraise"
  ],
  "session_id": "93d2cd79-a3dd-43c9-9233-deeb132dd99c"
}
```

- **Suivre le streaming en temps réel :**

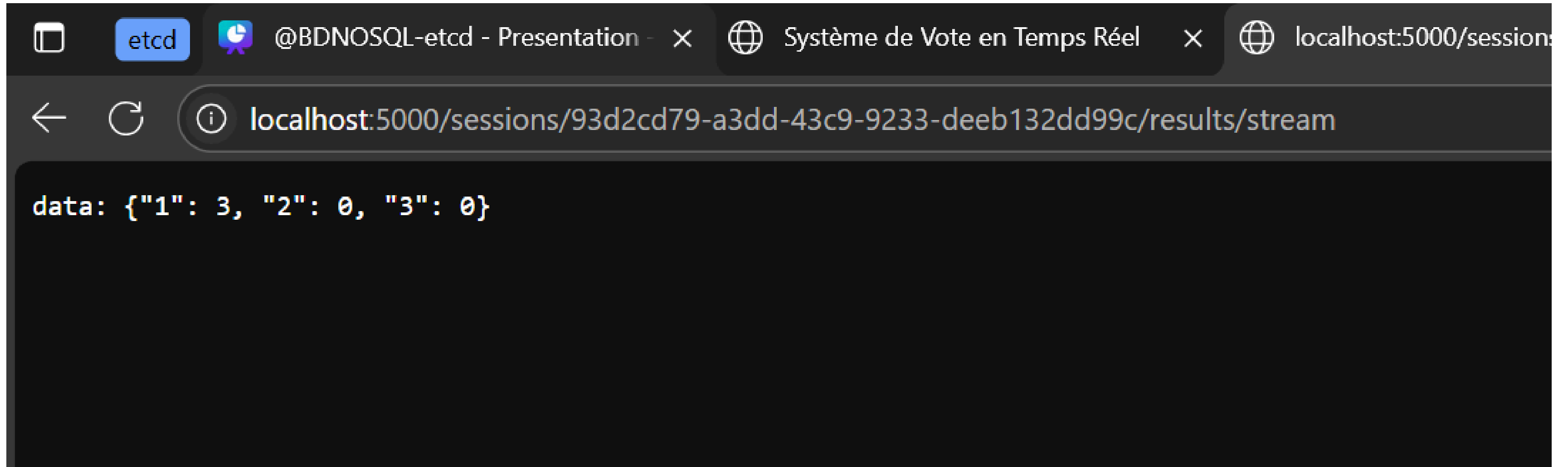
`http://localhost:5000/sessions/<ton_session_id>/results/stream`



- On réalise des votes via l'interface web



- Résultats de vote actualisé :





projet_systeme_vote-etcd-1

[bitnami/etcd:latest](#)

0d02c5237c19

[2379:2379](#) [2380:2380](#)

Logs

Inspect

Bind mounts

Exec

Files

Stats

```
$ ^C
$ etcdctl --endpoints=http://localhost:2379 get / --prefix --keys-only
/votes/sessions/584ddc09-c3ea-4e2c-9109-d208a6c51773/info

/votes/sessions/584ddc09-c3ea-4e2c-9109-d208a6c51773/options/1

/votes/sessions/584ddc09-c3ea-4e2c-9109-d208a6c51773/options/2

/votes/sessions/584ddc09-c3ea-4e2c-9109-d208a6c51773/options/3

/votes/sessions/584ddc09-c3ea-4e2c-9109-d208a6c51773/results/1

/votes/sessions/584ddc09-c3ea-4e2c-9109-d208a6c51773/results/2

/votes/sessions/584ddc09-c3ea-4e2c-9109-d208a6c51773/results/3

/votes/sessions/93d2cd79-a3dd-43c9-9233-deeb132dd99c/info

/votes/sessions/93d2cd79-a3dd-43c9-9233-deeb132dd99c/options/1
```

- **Afficher les données stockés dans etcd:**

```
$ etcdctl --endpoints=http://localhost:2379 get / --prefix
```



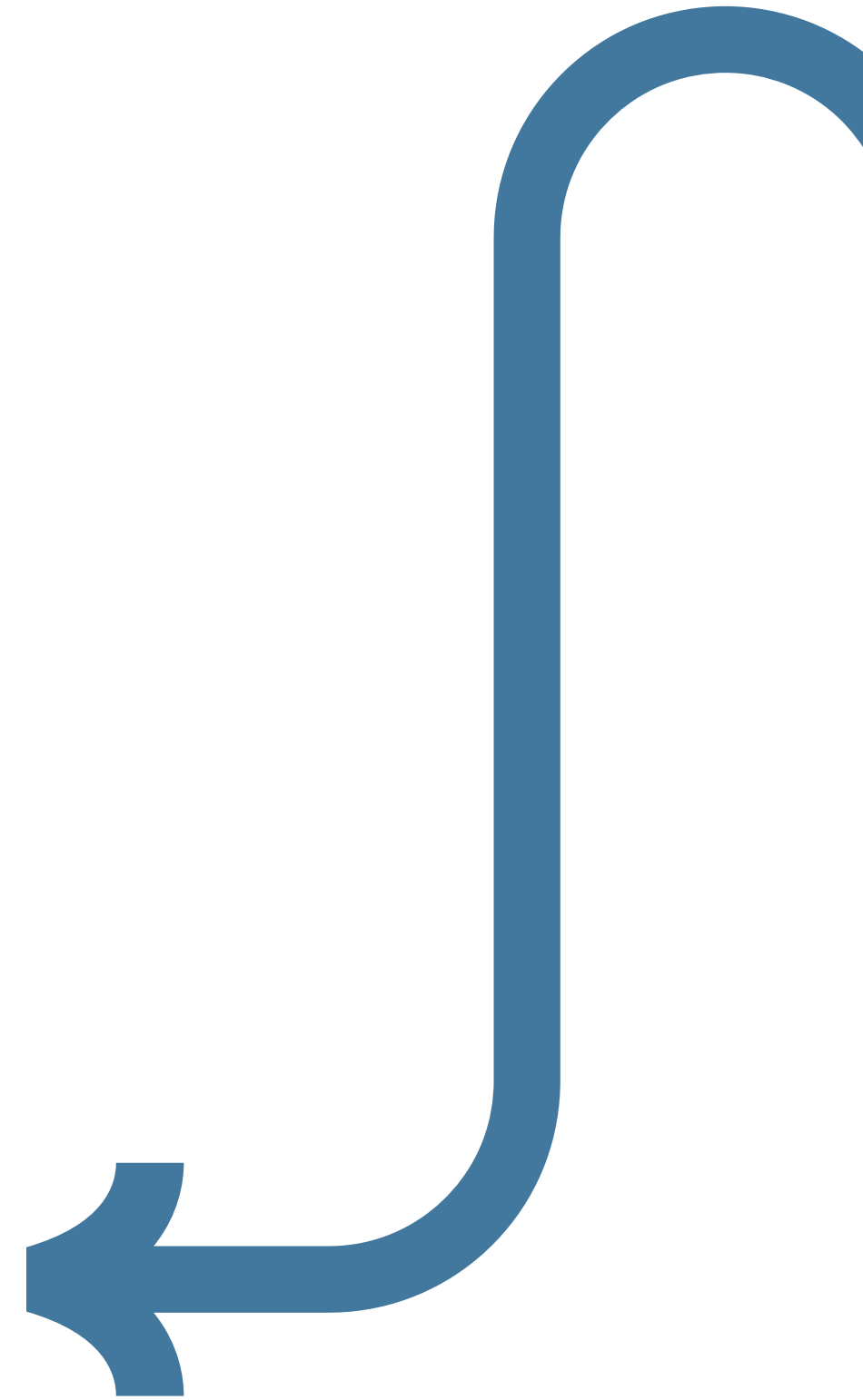
projet_systeme_vote-etcd-1

 [bitnami/etcd:latest](#) **STATUS**
0d02c5237c19  Running (6 min)
[2379:2379](#)  [2380:2380](#) 

Logs Inspect Bind mounts **Exec** Files Stats

```
5
/votes/sessions/584ddc09-c3ea-4e2c-9109-d208a6c51773/results/2
2
/votes/sessions/584ddc09-c3ea-4e2c-9109-d208a6c51773/results/3
2
/votes/sessions/93d2cd79-a3dd-43c9-9233-deeb132dd99c/info
{"title": "Meilleur fruit", "description": "", "created_at": 1747588087.138111, "active": true}
/votes/sessions/93d2cd79-a3dd-43c9-9233-deeb132dd99c/options/1
{"text": "Pomme", "id": "1"}
/votes/sessions/93d2cd79-a3dd-43c9-9233-deeb132dd99c/options/2
{"text": "Orange", "id": "2"}
/votes/sessions/93d2cd79-a3dd-43c9-9233-deeb132dd99c/options/3
{"text": "Fraise", "id": "3"}
/votes/sessions/93d2cd79-a3dd-43c9-9233-deeb132dd99c/results/1
3
/votes/sessions/93d2cd79-a3dd-43c9-9233-deeb132dd99c/results/2
0
/votes/sessions/93d2cd79-a3dd-43c9-9233-deeb132dd99c/results/3
0
$
```

7. Références



- etcd adopters.md - <https://github.com/etcd-io/etcd/blob/main/ADOPTERS.md>
- etcd in kubernetes : <https://itnext.io/how-etcd-works-in-kubernetes-e18adf97d24d>
- etcd cluster sur AWS : <https://proventa.de/en/building-a-high-availability-etcd-cluster-on-aws/>
- etcd, the easy way : <https://medium.com/nerd-for-tech/etcd-the-easy-way-4c01e243f285>