Introduction à Cassandra

Nicolas Romanetti
@nromanetti

www.jaxio.com

mercredi 18 avril 2012

University Talk

- Tout public
- 3h pour vraiment comprendre Cassandra
- Pédagogique

University Talk

- Démarre doucement
- Continue doucement
- Mais en 3h nous allons couvrir beaucoup de sujets
- Devrait vous chatouiller un peu le cerveau
- Conseil: prenez des notes

Cassandra LAN Party

(Hands On Cassandra)

Cet après midi
de 13h30 à 16h30

A Propos

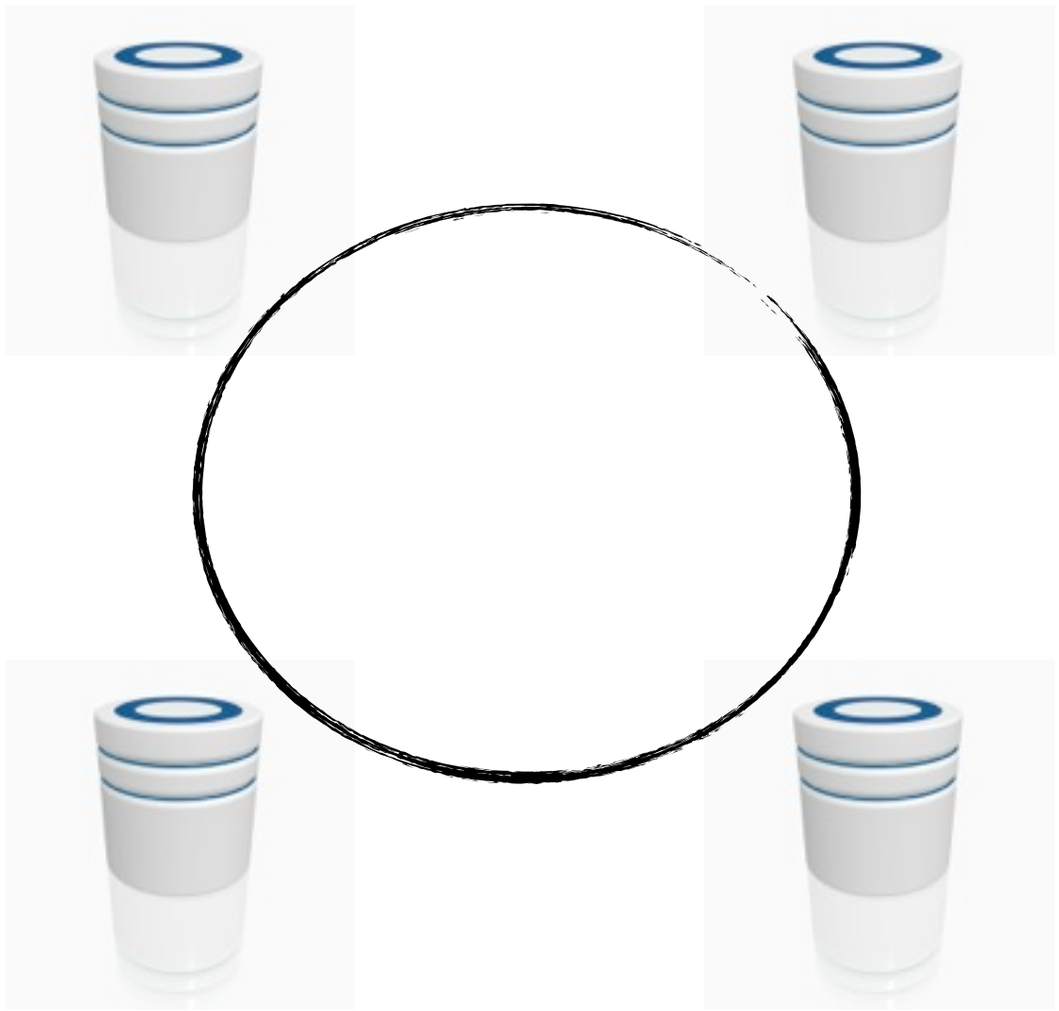
- Nicolas Romanetti
- @nromanetti
- www.jaxio.com
- Légitimité?

Cassandre

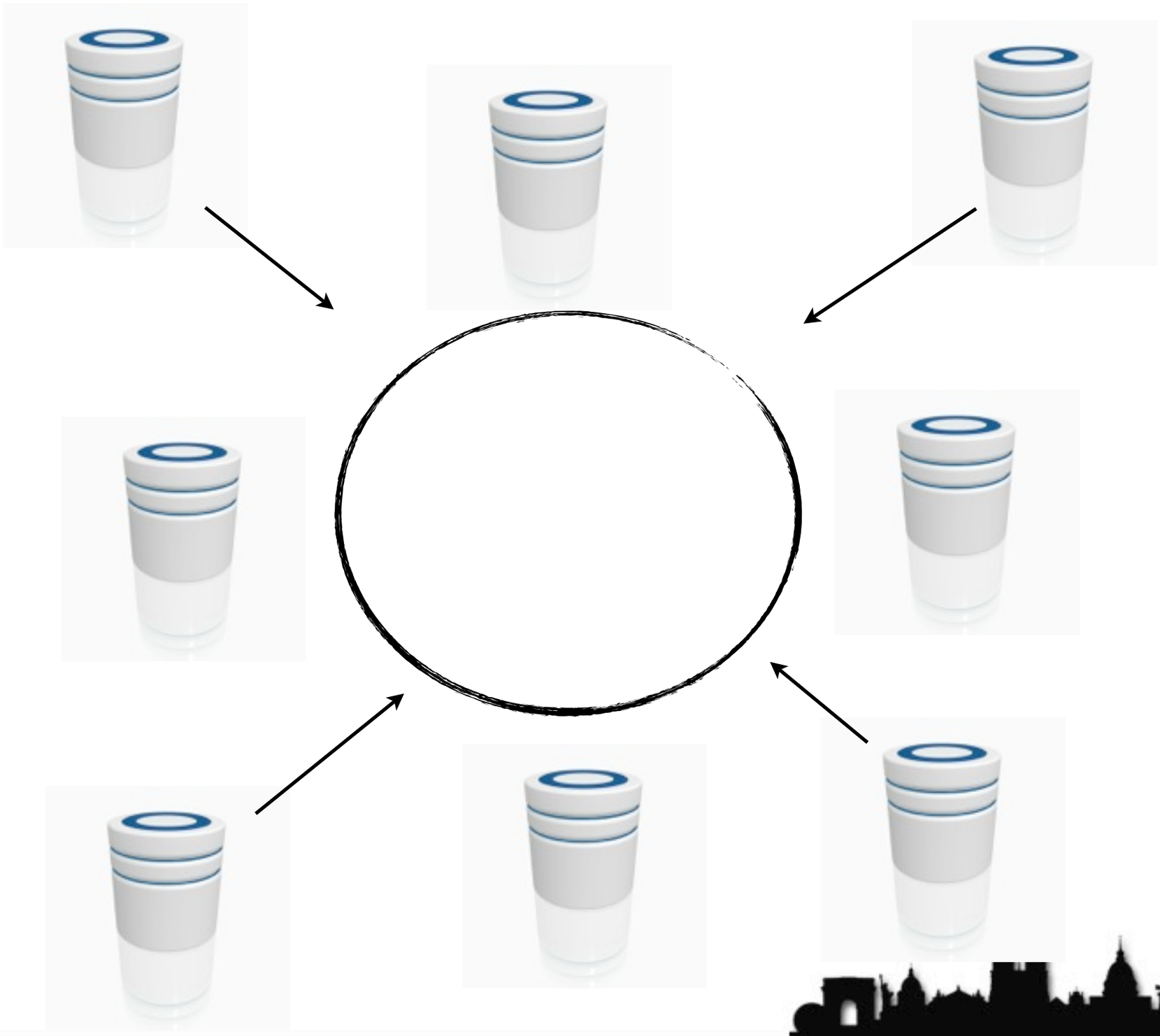
- Très belle
- Libre
- Avait raison

Cassandra c'est...

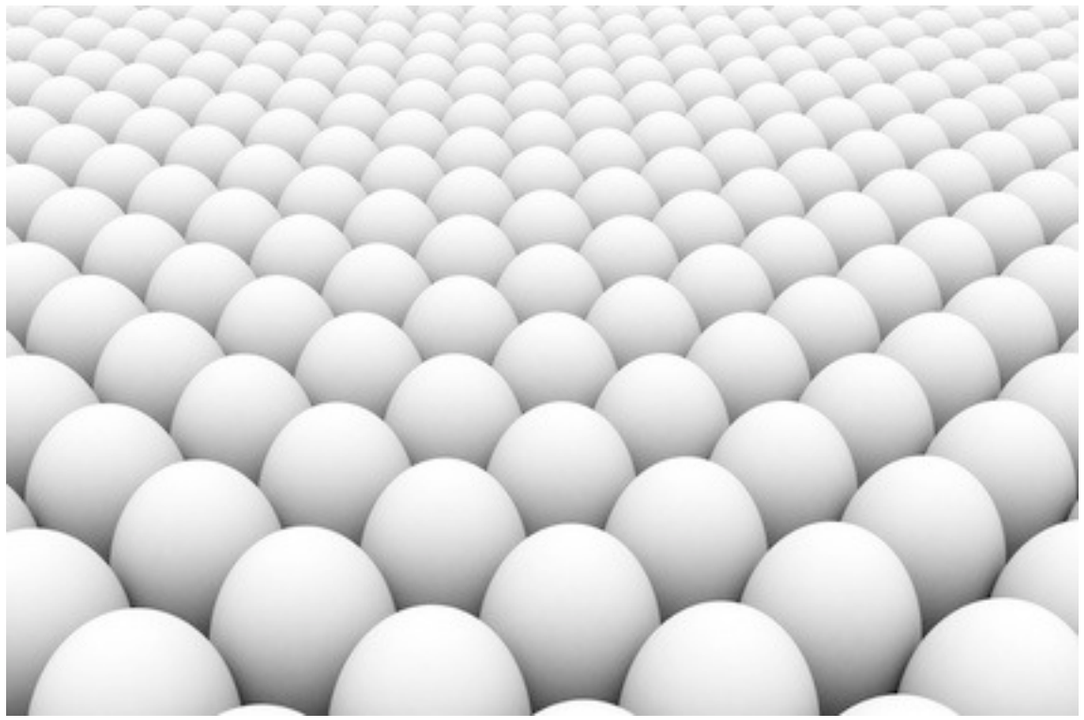
Cassandra c'est...



Cassandra c'est...



Symétrique



Conçue pour les pannes

RDBM

scalabilité

réplication

dénormalisation

indexes

cache

transaction

join

performance

Très important

Installation de Cassandra

Historique

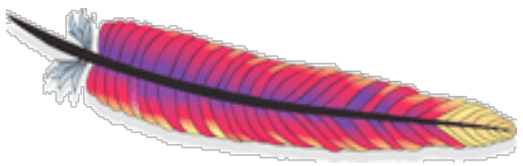
amazon

Dynamo

Google

Big Table

facebook



DATASTAX

NETFLIX

twitter

Data Model Cassandra

- Attention au vocabulaire

Version courte

- Une Map distribuée de Map

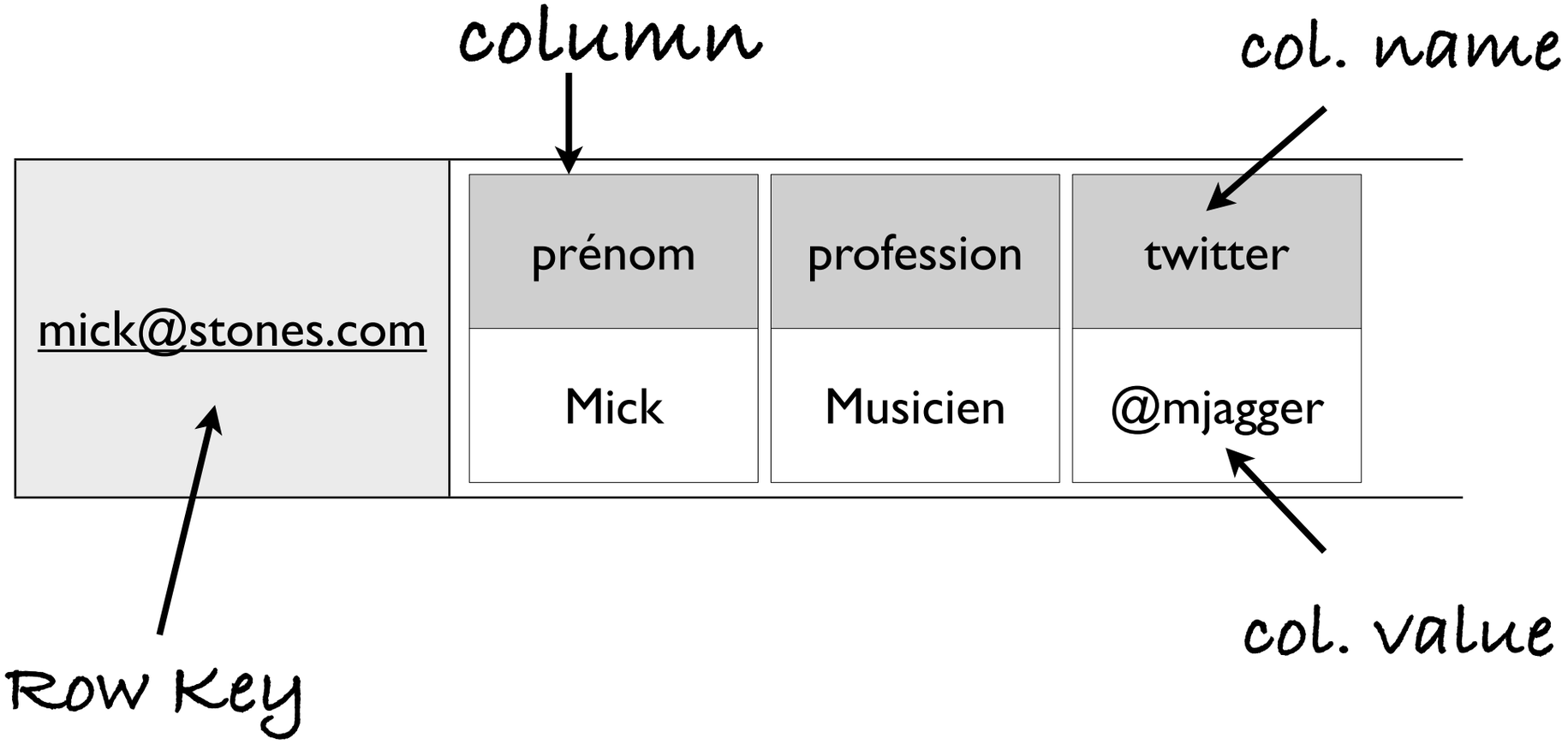
Column

name
value
timestamp

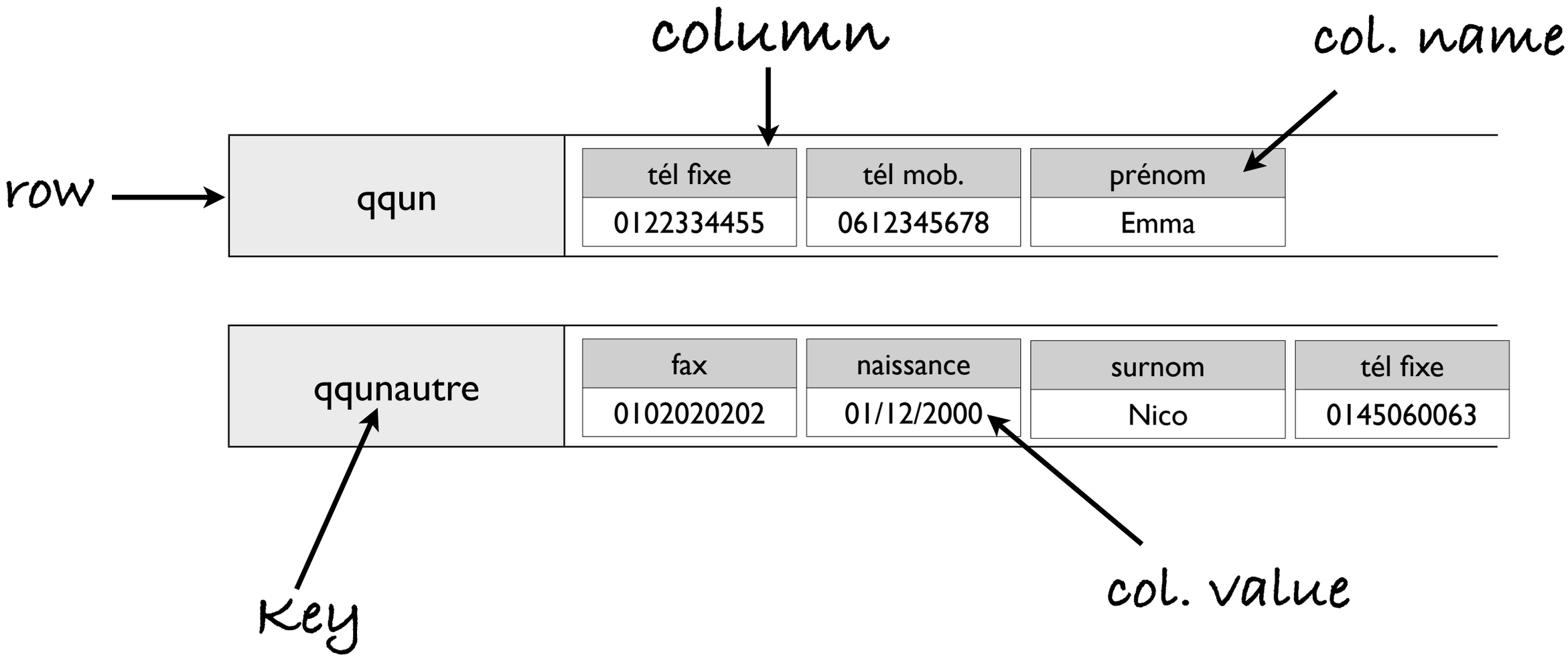
Row

Key	col. name1	col. name2	col. name3	...	col. nameN
	col. value1	col. value2	col. value3		col. valueN

Exemple de Row



Libre...



comparator/validator

Column Family

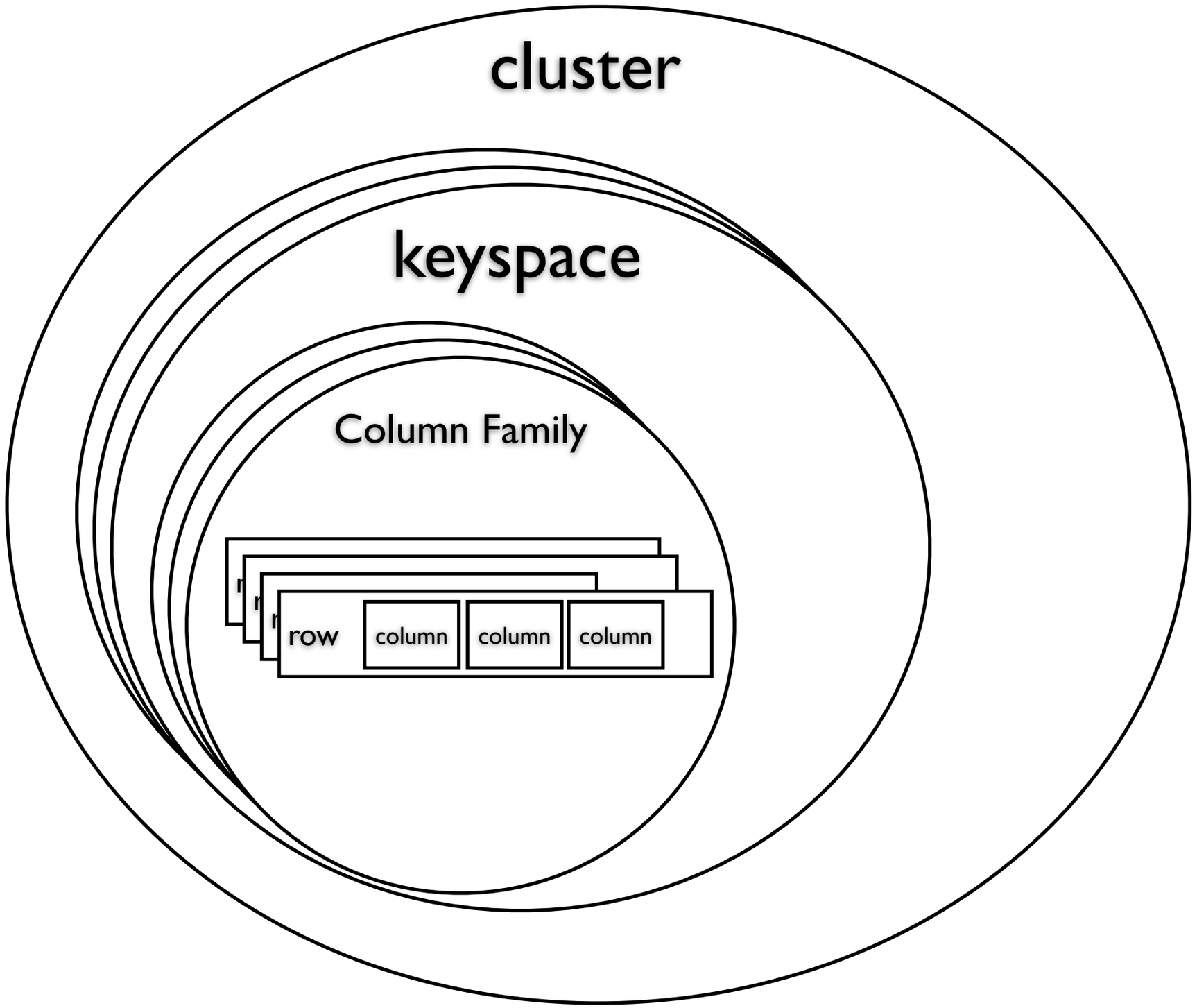
user

qqun	prénom	tél fixe	tél mob.
	Aurélien	0122334455	0612345678

qqunautre	fax	naissance	tél fixe
	0102020202	01/12/2000	0122334455

```
user['qqun']['tél mob.'] = 0612345678
```

col. Family | row key | col. name | col. value


cluster
keyspace
Column Family
row
column
column
column

Le nom est une valeur!

‘vol’

XK_4501_2012_01_06	6:55	7:00	7:20
	décollage face à la mer	survol Capo di Feno	survol Cannes

etc...

Liens

‘BlogEntries’

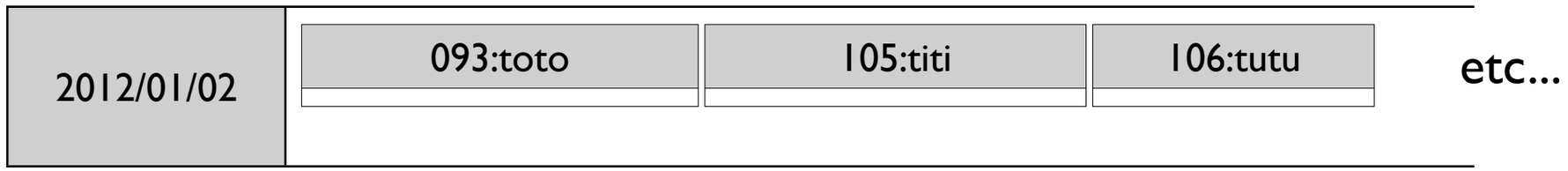
jaxio	2012/04/18 10:52	2012/04/12 10:12	2012/04/01 10:11	etc...
	rowkeyA	rowkeyB	rowkeyC	

‘BlogEntry’

rowkeyA	Content	date	Title	etc...
	Plusieurs talks se ...	2012/04/18 10:52	DevoxxFR rox	

Composite name, valueless

‘TopScores’



Super column

nromanetti@jaxio.com	
login	nroma
twitter	@nromanetti
ville	Paris


```
users['france']['nromanetti@jaxio.com']['ville'] = 'Paris'
```

Super col. Family

row key

super col. name

sub col. name

sub col. value

Fourre tout

contactInfo
tel:04..., email: nr@..., city: Ajaccio, etc...

Gros contenu

Une_Video	0	1	2	etc...
	chunk_0	chunk_1	chunk_2	

Column expirante

name
value

Compteur Distribué

nb de like
57832

Tombstone

nom

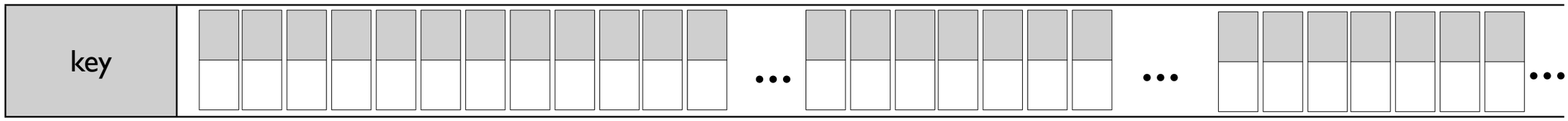
Questions?

Query

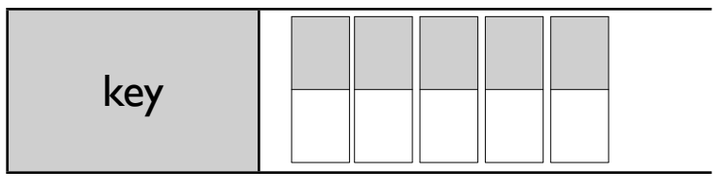
- Attention au vocabulaire

2 types de row

Wide row



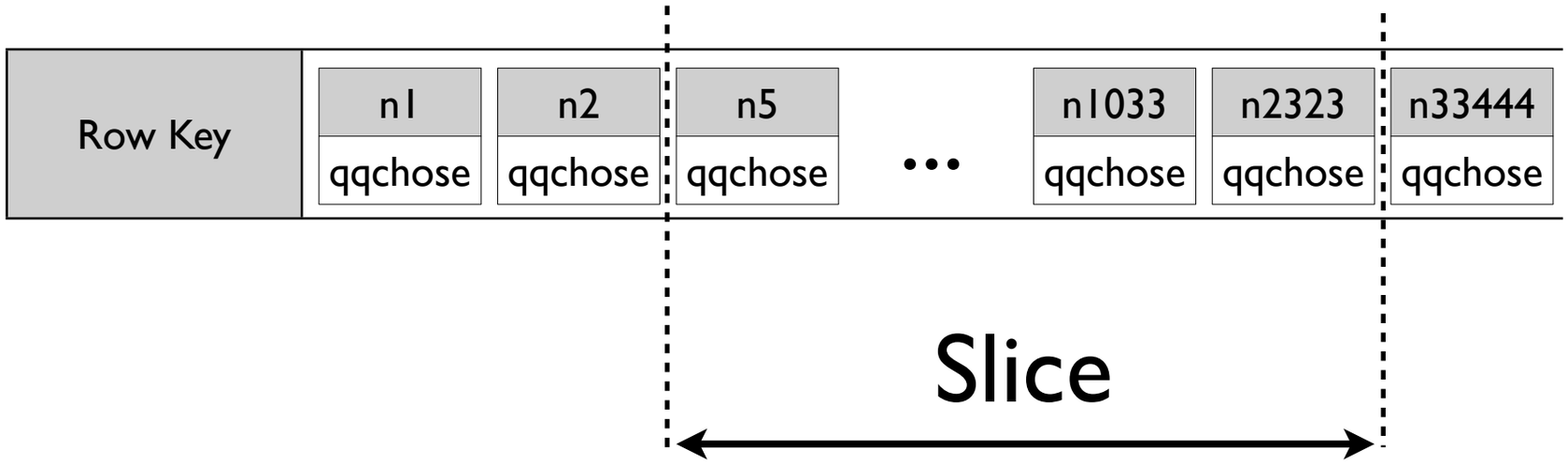
Skinny row



get

- `get user['toto']['firstname'];`
- `get user['toto']['state'];`

Column Slice



ordre..

Cassandra Query Language

- `select user where country = 'FR';`

Secondary Index

«Modélisation»

- Avoir un don de prophétie

Disponibilité ☞ Réplication

Où stocker une row?

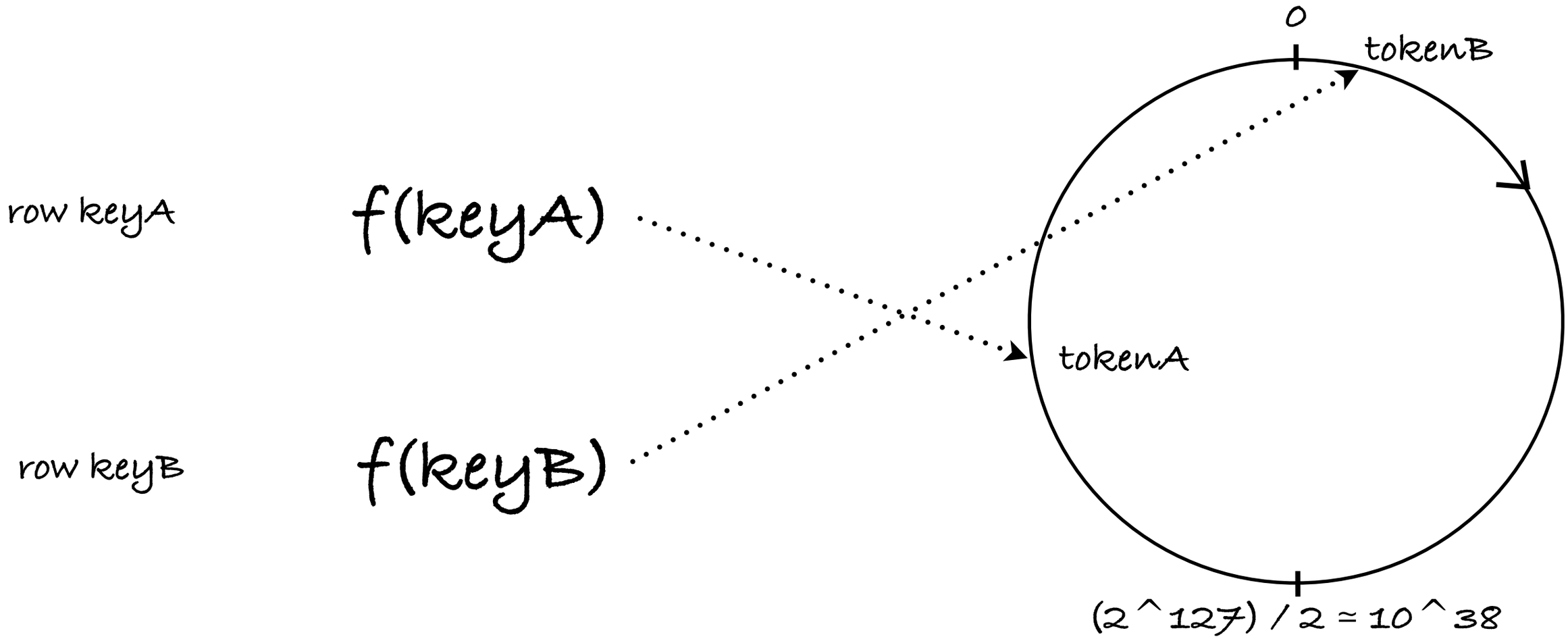
- Tous les serveurs sont identiques (symétrie)
- Postulats
  - 1 row doit tenir sur 1 seul serveur
  - On veut répartir les rows uniformément

Consistent Hashing

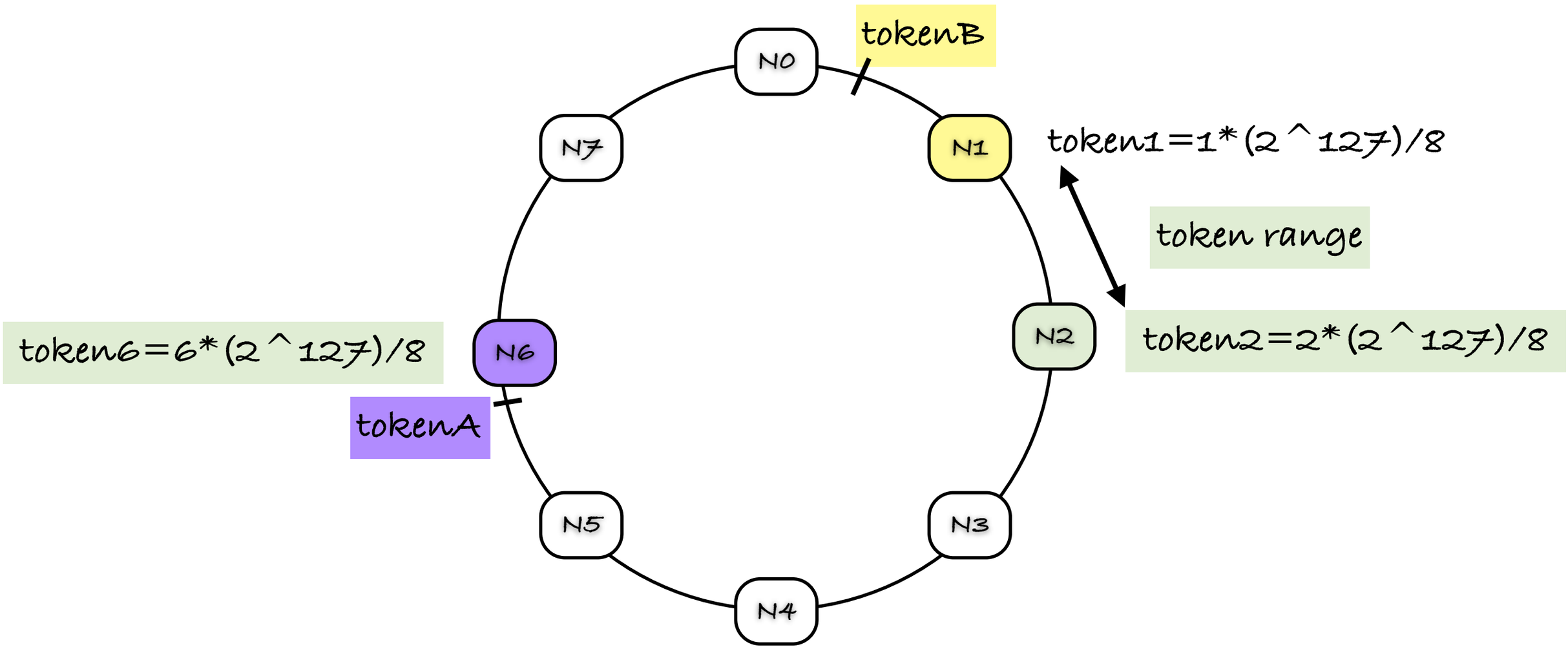
bytes
e.g. String

Consistent
Hash

Un ‘token’ entre
0 et 2^127



Consistent Hashing



node 2 = contient 1ère replica des rows ayant leur token compris entre token1 et token2

nodetool ring

```
MacBook-Pro-de-Nicolas-Romanetti:dsc-cassandra-1.0.8 nromanetti$ bin/nodetool -h 10.1.1.1 ring
Address         DC          Rack        Status State   Load            Owns    Token
                                                                               164480732853838332588838221182322480353
10.1.1.1        1           1           Up     Normal  34,34 KB        3,33%   0
10.2.1.1        2           1           Up     Normal  34,34 KB        0,00%   1
10.3.1.1        3           1           Up     Normal  34,34 KB        0,00%   2
10.1.1.2        1           1           Up     Normal  43,33 KB        16,67%  28356863910078205288614550619314017621
10.2.1.2        2           1           Up     Normal  43,32 KB        0,00%   28356863910078205288614550619314017622
10.3.1.2        3           1           Up     Normal  43,33 KB        0,00%   28356863910078205288614550619314017623
10.1.1.3        1           1           Up     Normal  43,33 KB        16,67%  56713727820156410577229101238628035242
10.2.1.3        2           1           Up     Normal  38,72 KB        0,00%   56713727820156410577229101238628035243
10.3.1.3        3           1           Up     Normal  43,33 KB        0,00%   56713727820156410577229101238628035244
10.3.1.4        3           1           Up     Normal  43,33 KB        16,67%  85070591730234615865843651857942052866
10.1.1.5        1           1           Up     Normal  43,33 KB        16,67%  113427455640312821154458202477256070485
10.3.1.5        3           1           Up     Normal  43,33 KB        0,00%   113427455640312821154458202477256070487
10.2.1.5        2           1           Up     Normal  98,79 KB        3,25%   118959467553234022797339489649815746085
10.1.1.6        1           1           Up     Normal  145,25 KB       13,06%  141181442192599215569169119445432872465
10.2.1.6        2           1           Up     Normal  43,33 KB        0,35%   141784319550391026443072753096570088107
10.3.1.6        3           1           Up     Normal  43,33 KB        0,00%   141784319550391026443072753096570088108
10.1.1.4        1           1           Up     Normal  47,71 KB        3,03%   146946219305601782975974550082043979626
10.2.1.4        2           1           Up     Normal  47,71 KB        10,31%  164480732853838332588838221182322480353
MacBook-Pro-de-Nicolas-Romanetti:dsc-cassandra-1.0.8 nromanetti$
```

Vocabulaire

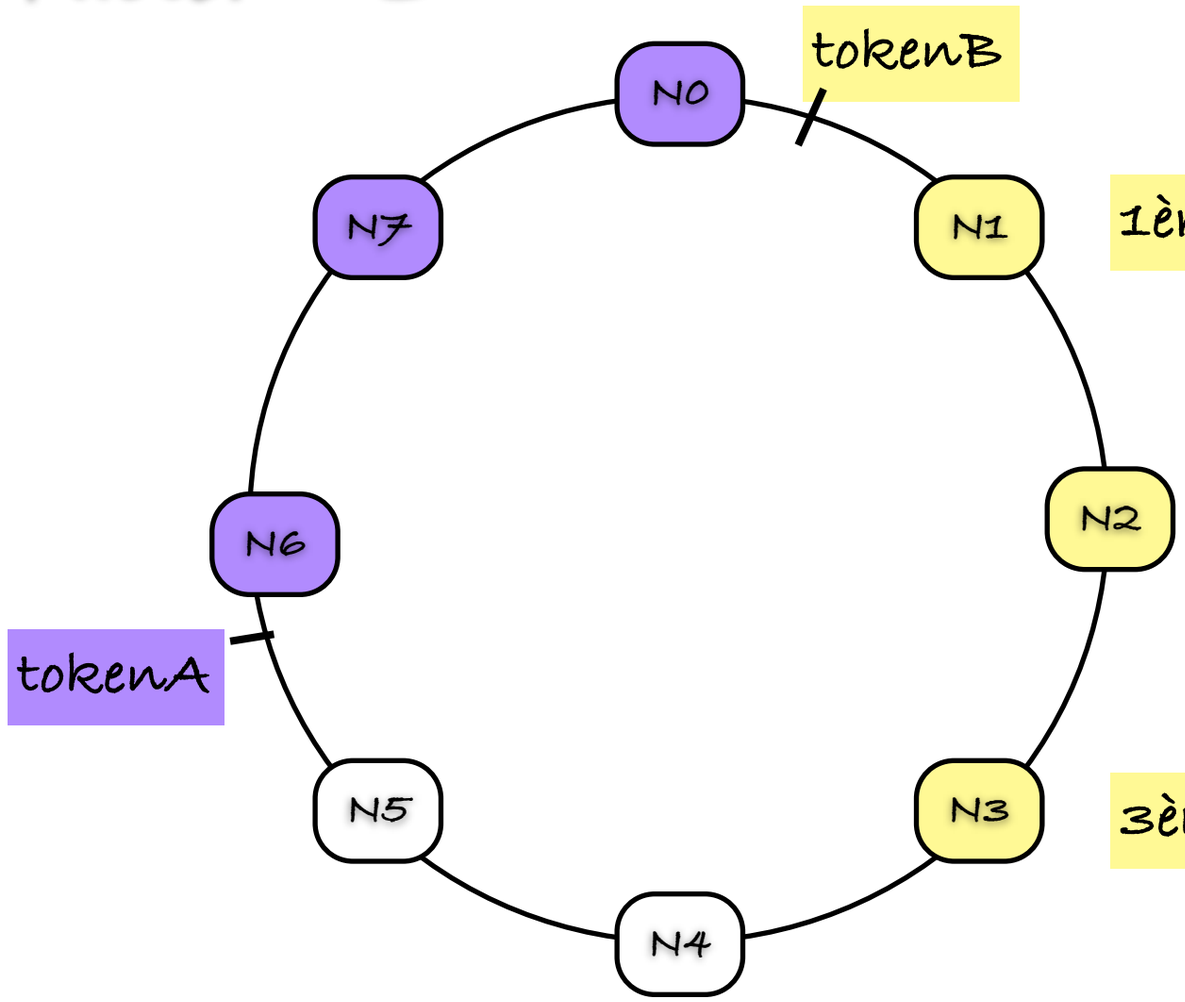
Node

Replication Factor

Replica

Stratégie de placement

Replication Factor = 3

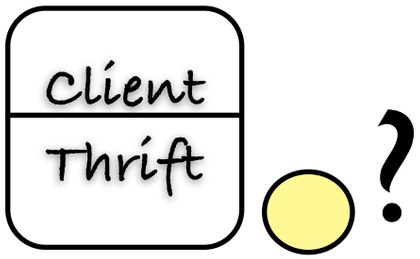


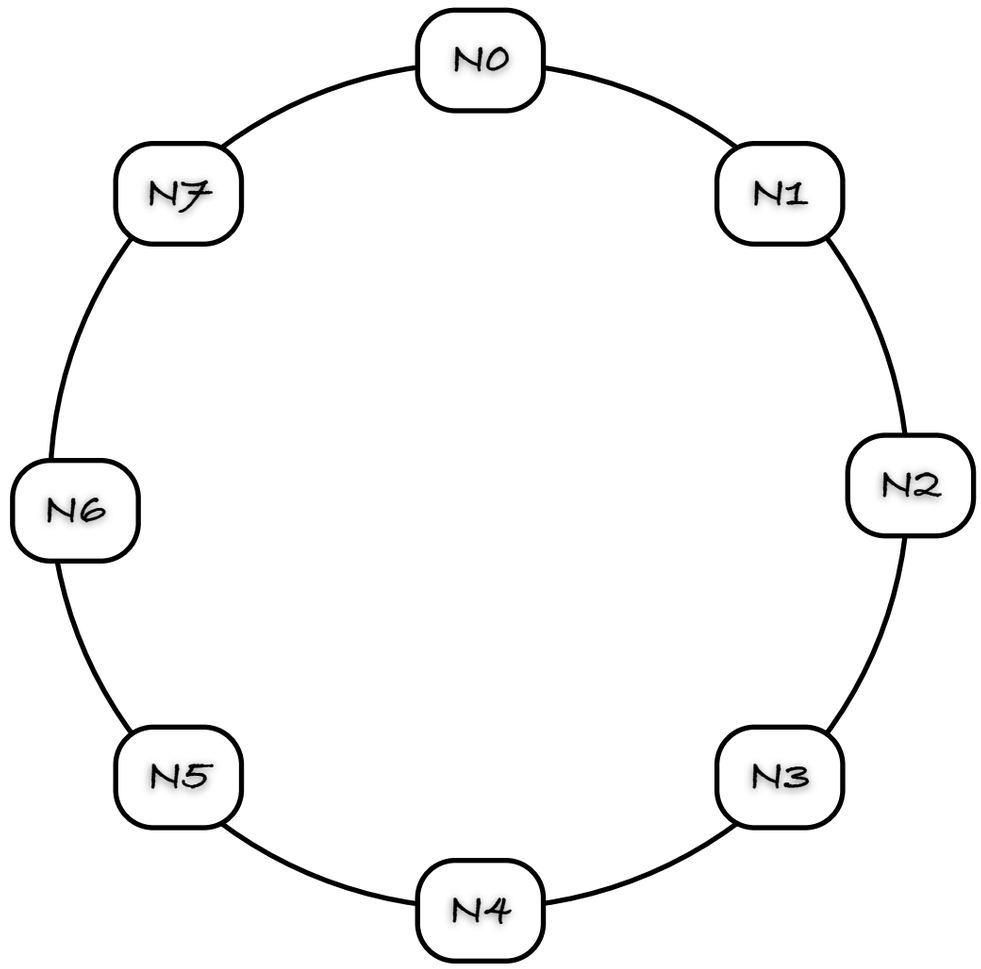
API client

- Hector... Frère de Cassandre
- Astyanax (créé par Netflix)... Fils d'Hector
- Driver CQL...

Thrift

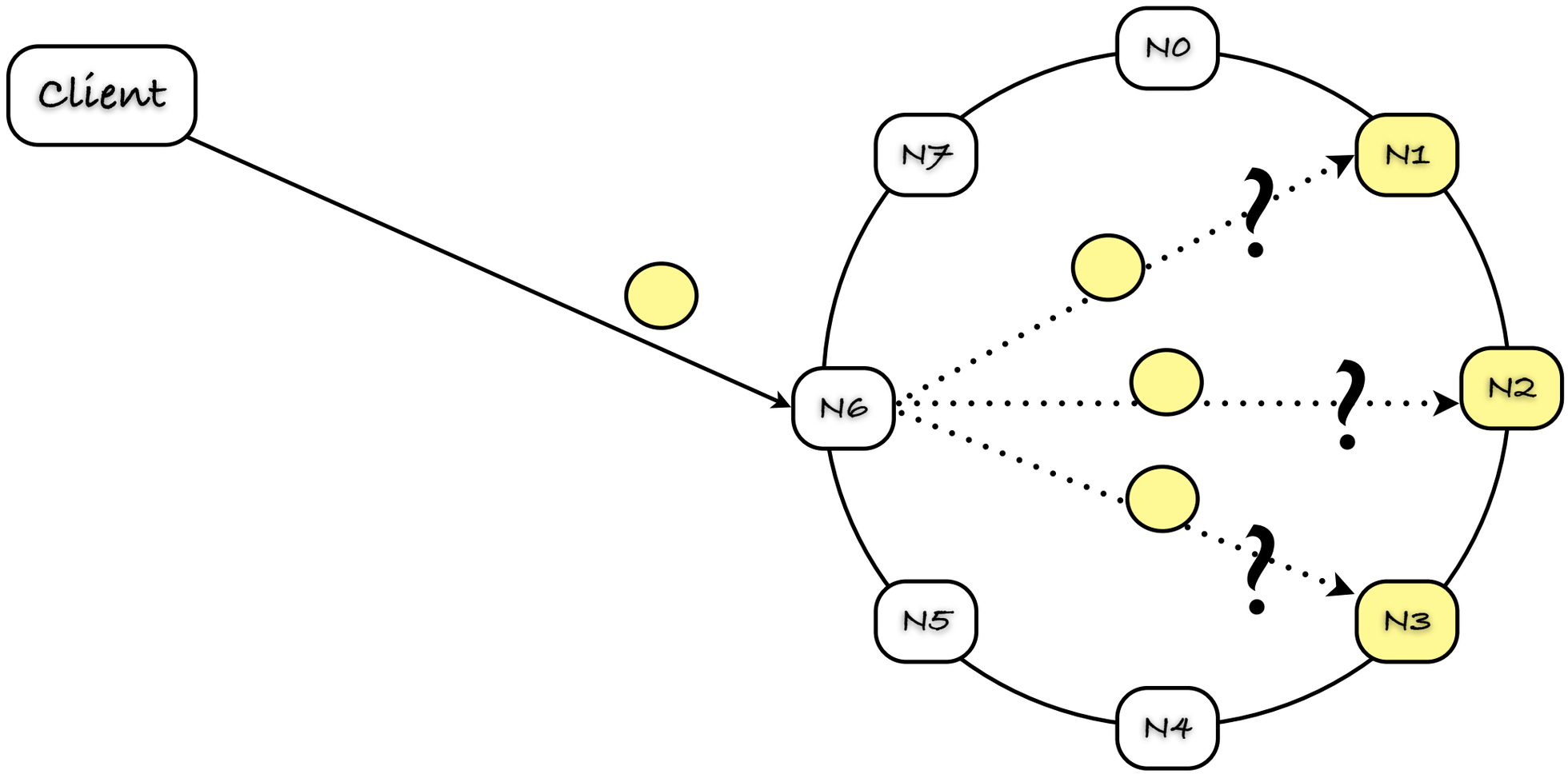
Ecriture: qui contacter?





Une replica?

Attendre les réponses?



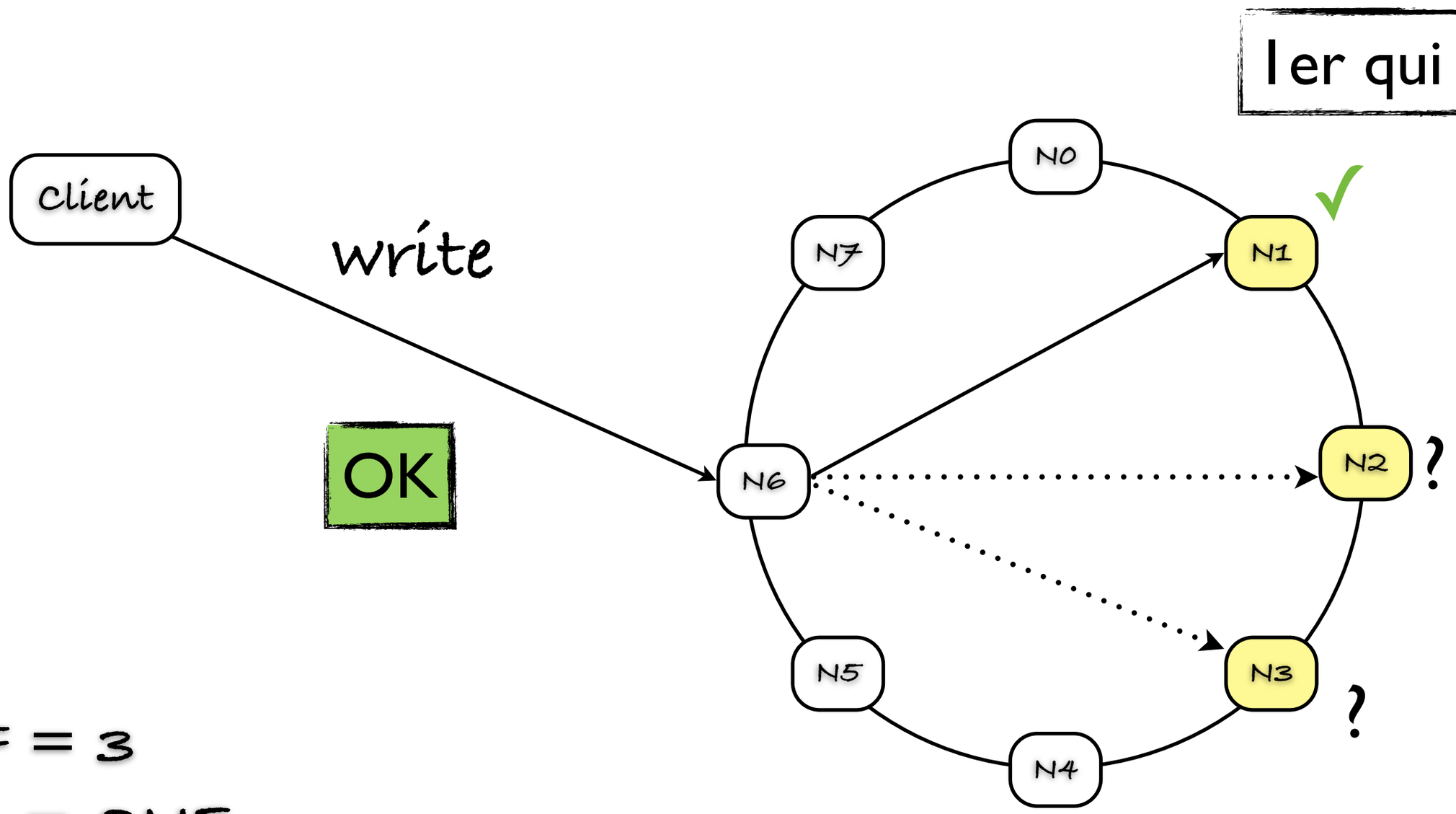
solution: Tuneable Consistency

CONSISTENCY LEVEL

CL - ONE

écriture

Exemple 1

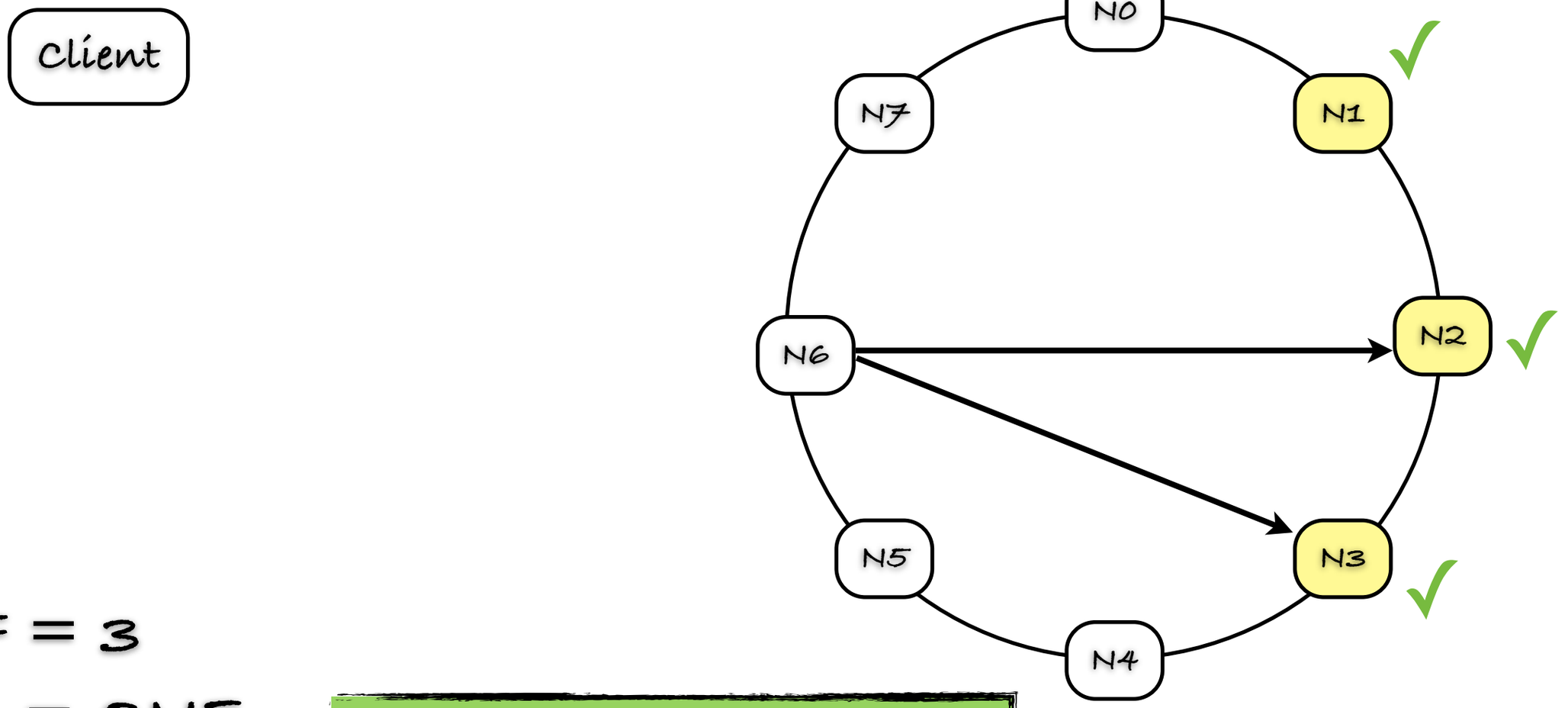


RF = 3

CL = ONE

·····➤ asynchrone

Exemple 1 (suite)

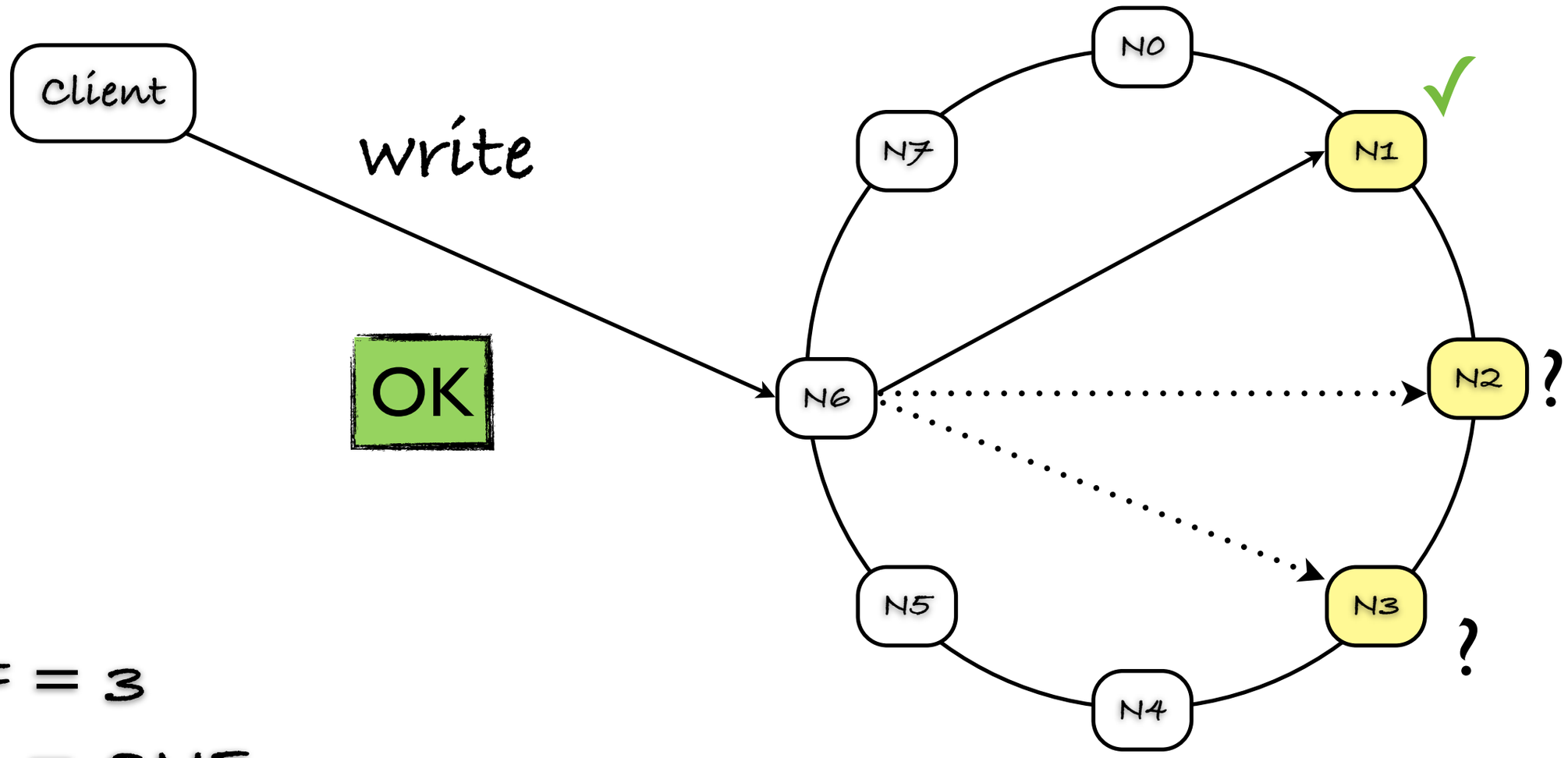


RF = 3

CL = ONE

OK en background...

Exemple 2

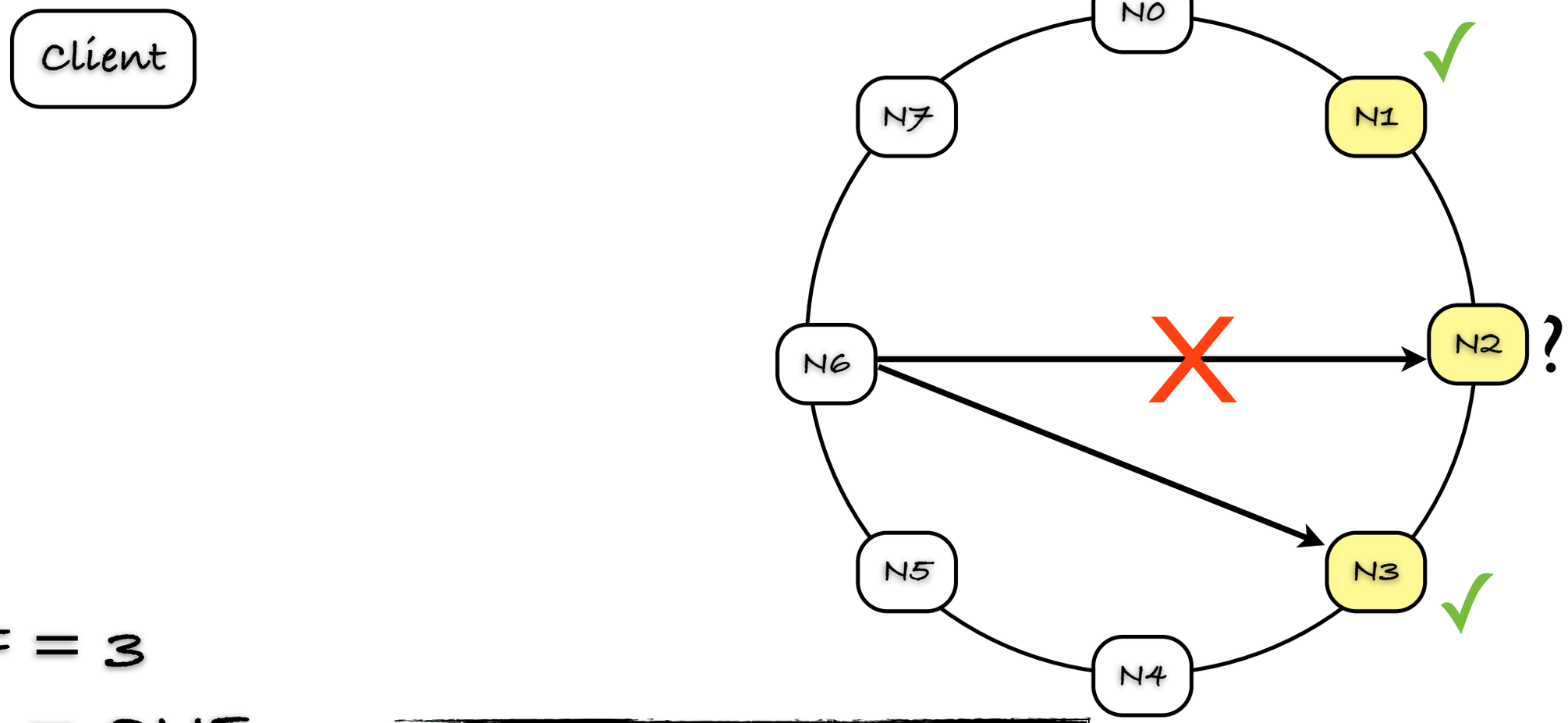


RF = 3

CL = ONE

·····➤ asynchrone

Exemple 2 (suite)

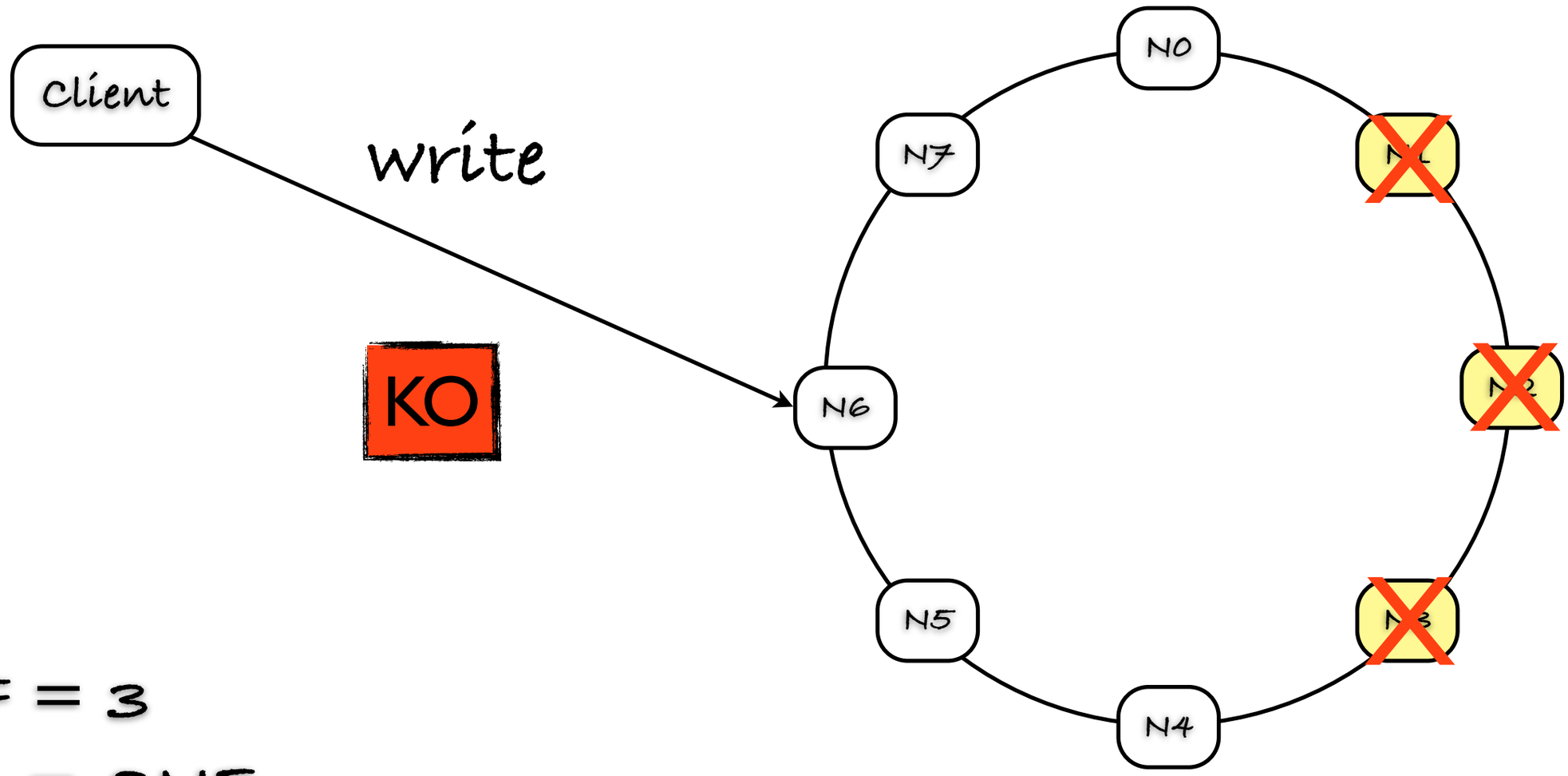


RF = 3

CL = ONE

KO en background...

Exemple 3



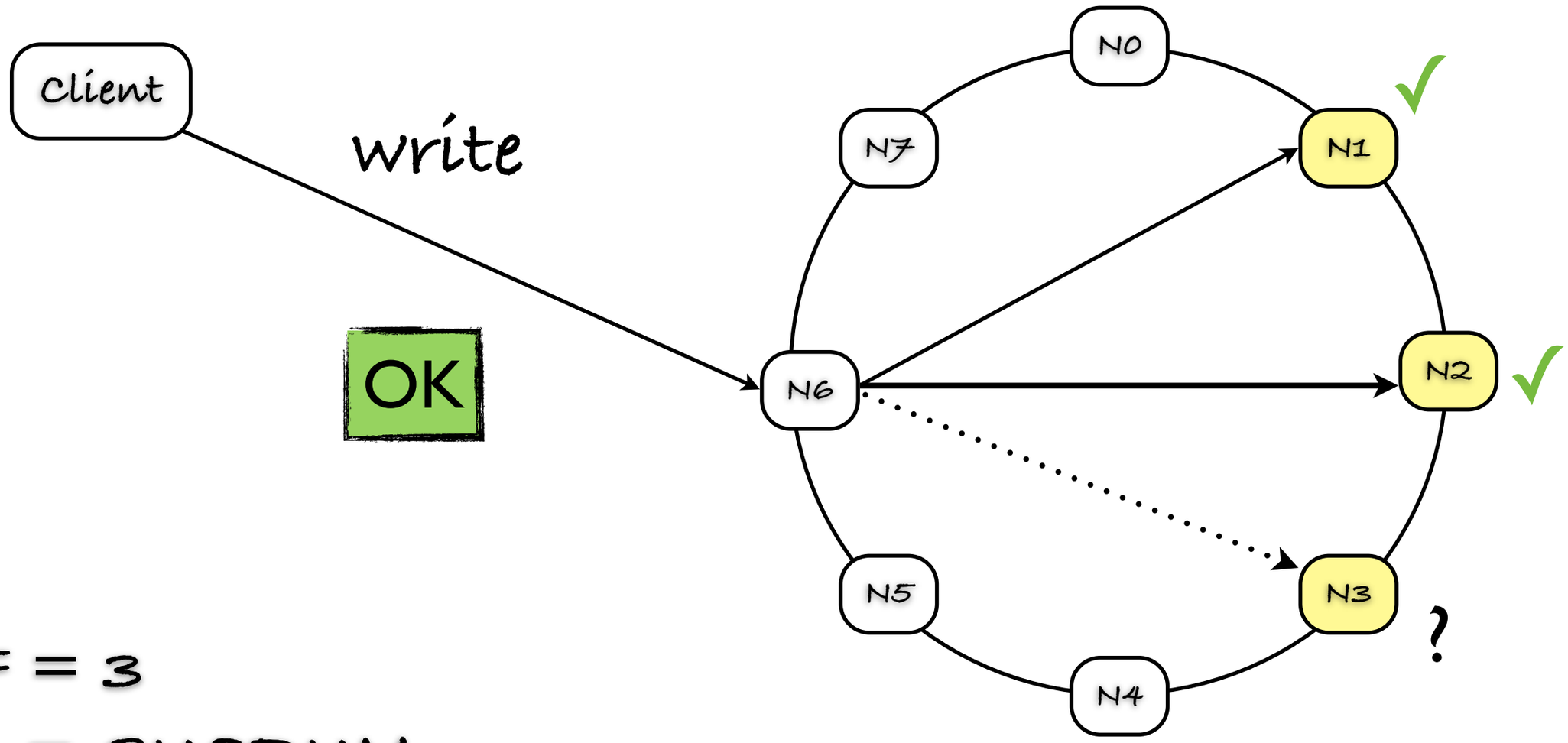
RF = 3

CL = ONE

CL - QUORUM

écriture

Exemple 1

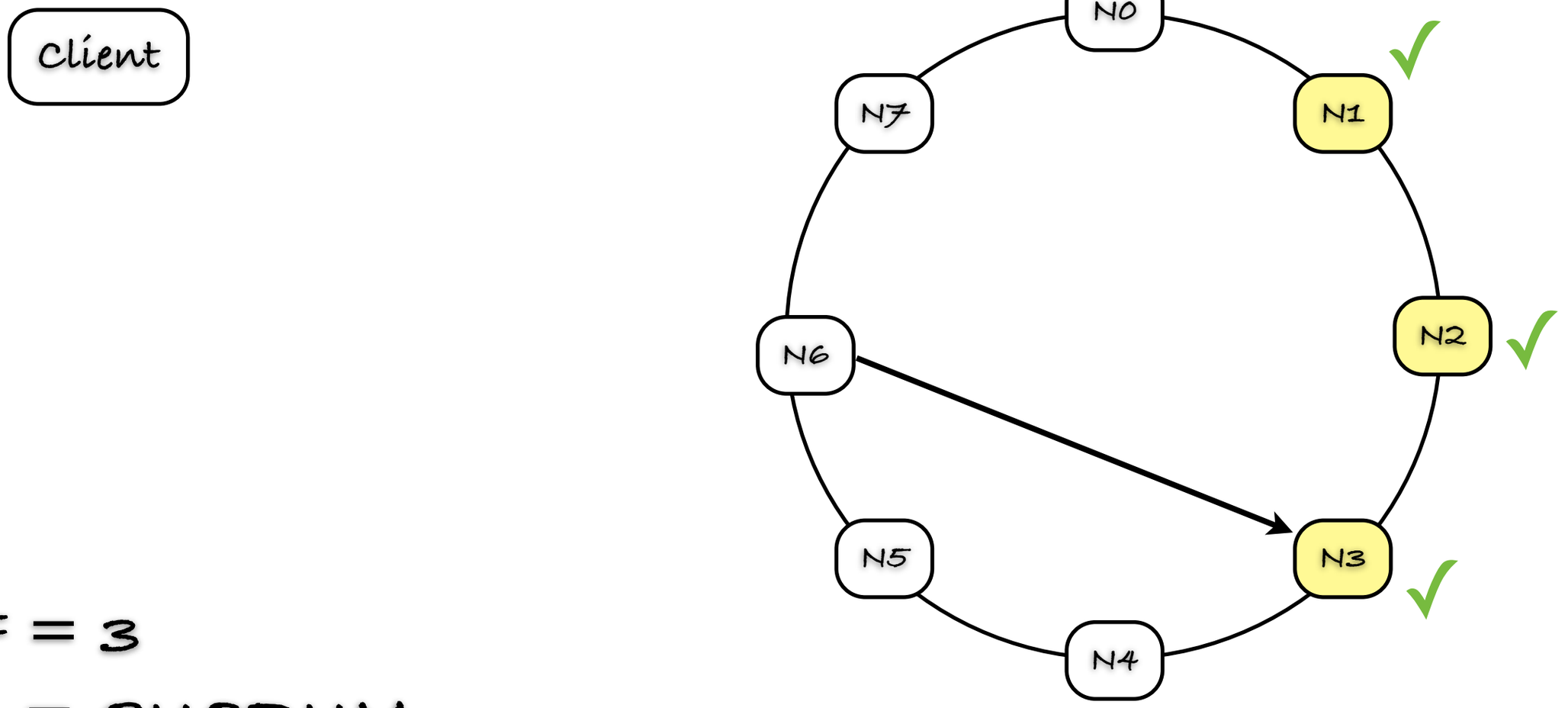


RF = 3

CL = QUORUM

·····➤ asynchrone

Exemple 1 (suite)

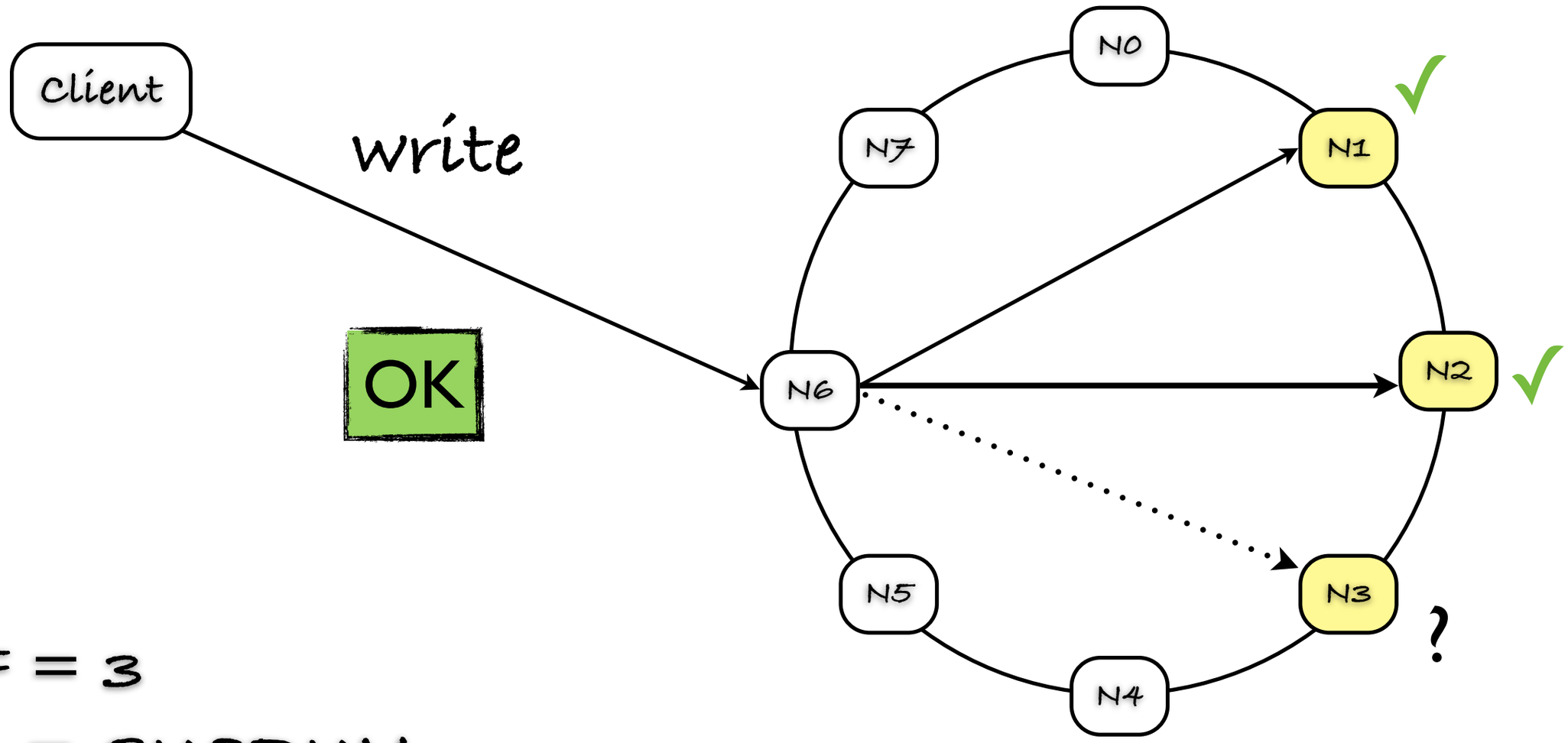


RF = 3

CL = QUORUM

OK en background...

Exemple 2

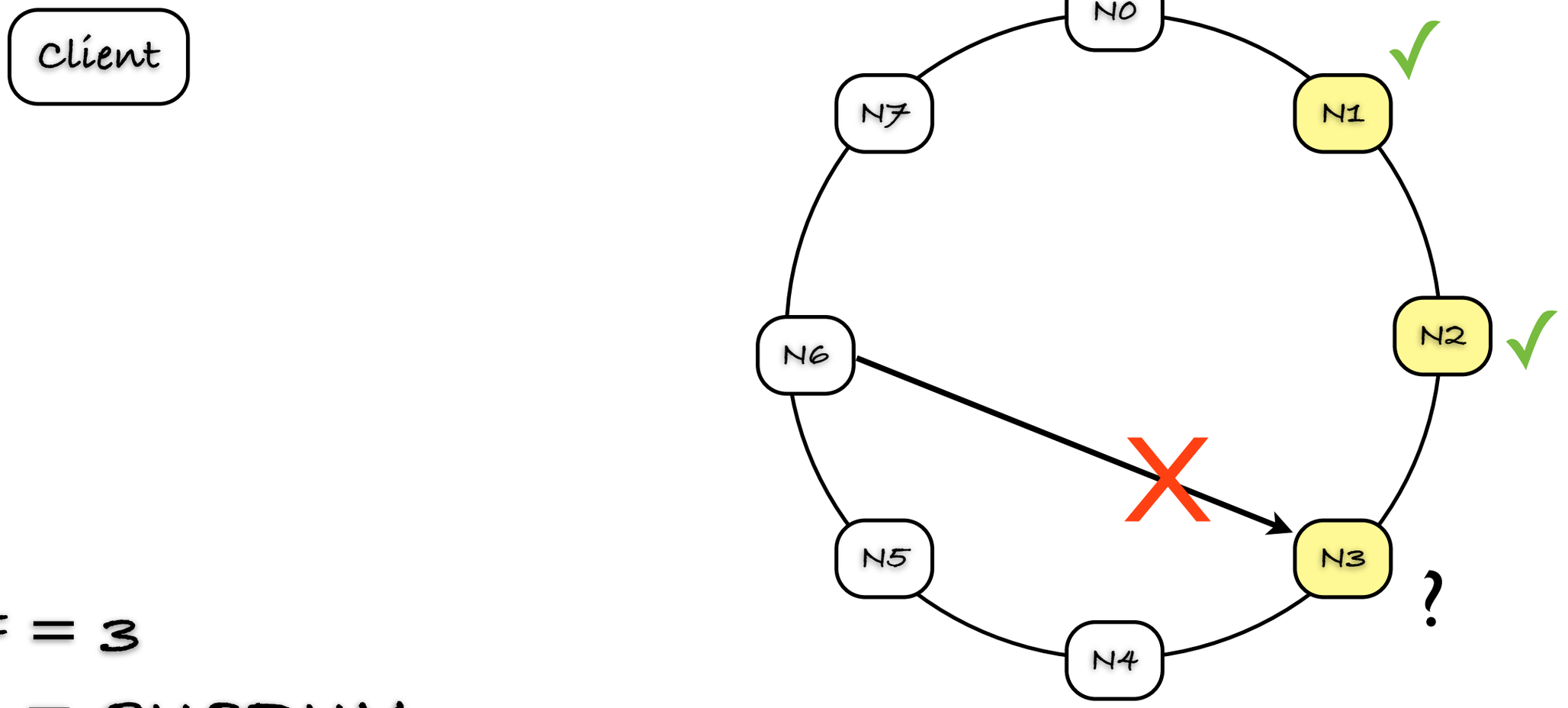


RF = 3

CL = QUORUM

·····➤ asynchrone

Exemple 2 (suite)

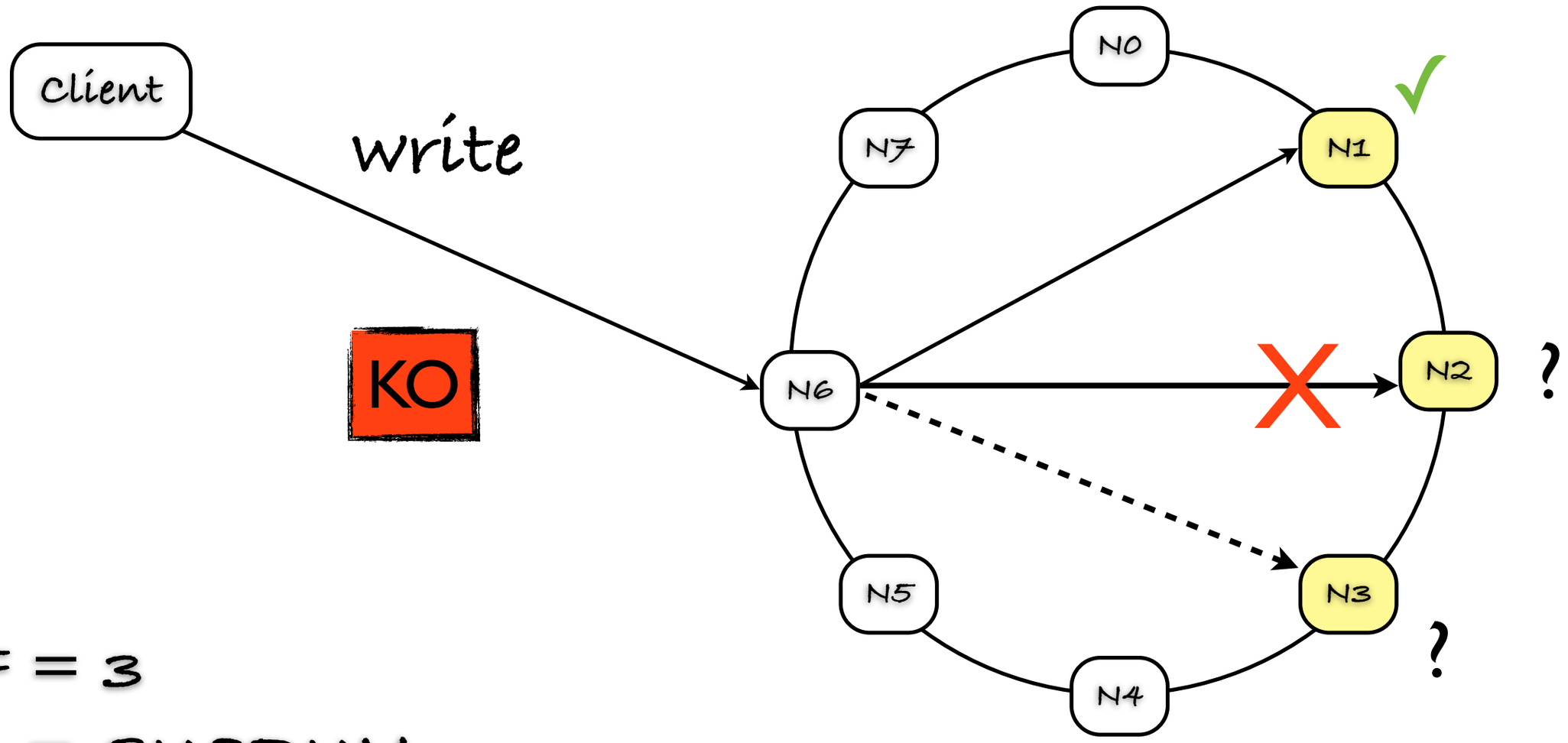


RF = 3

CL = QUORUM

KO en background...

Exemple 3

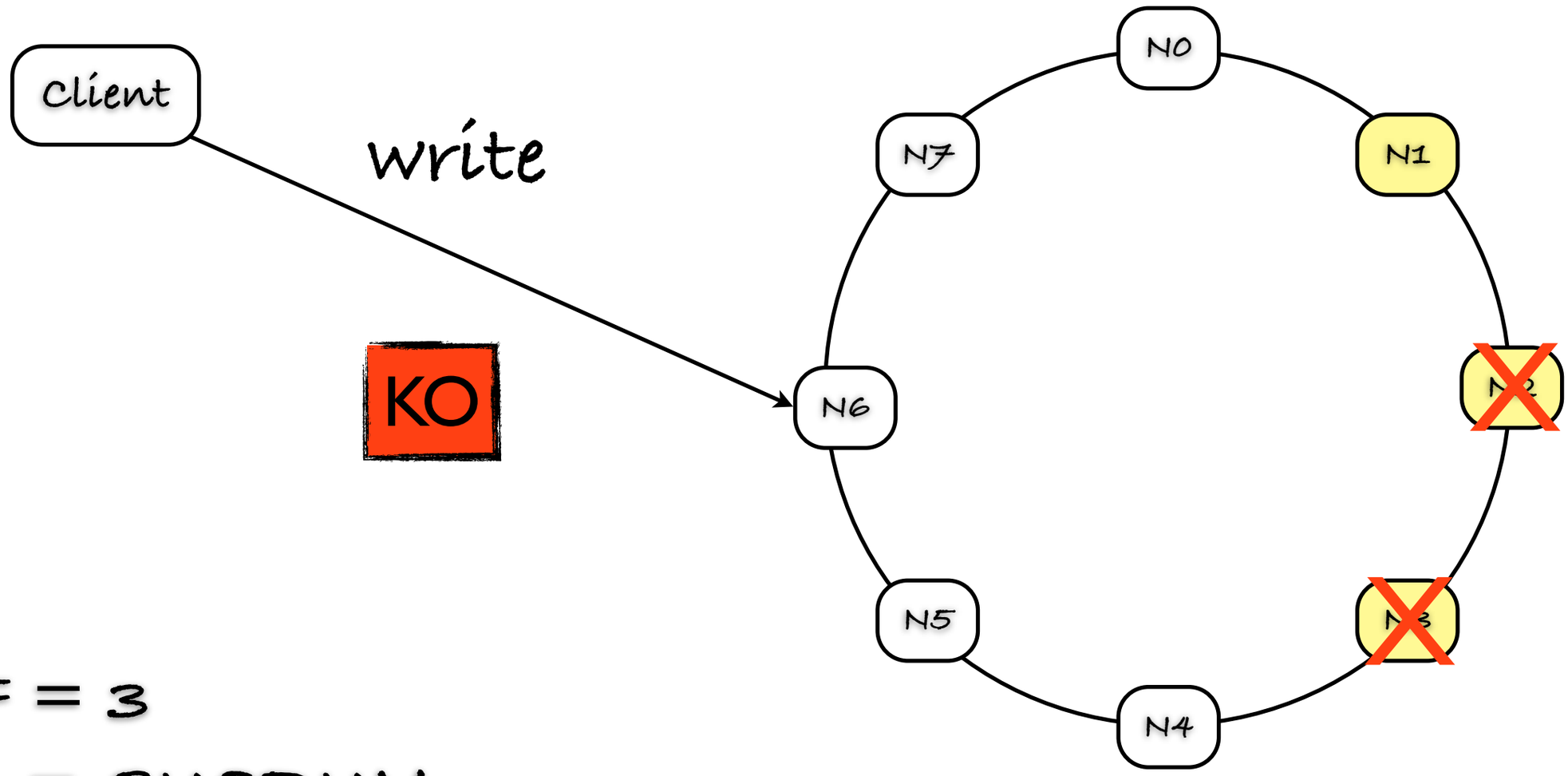


RF = 3

CL = QUORUM

·····➤ asynchrone

Exemple 4



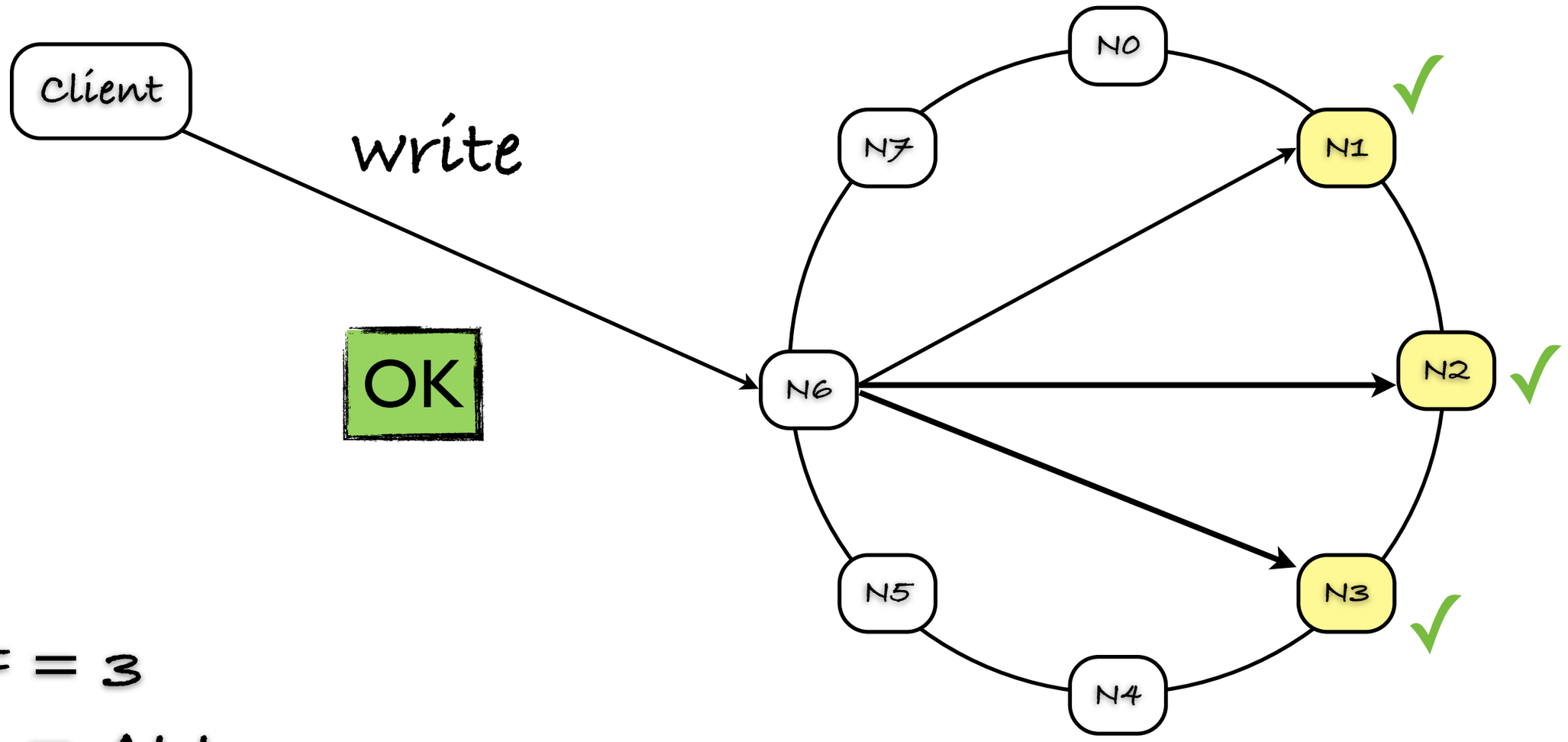
RF = 3

CL = QUORUM

CL - ALL

écriture

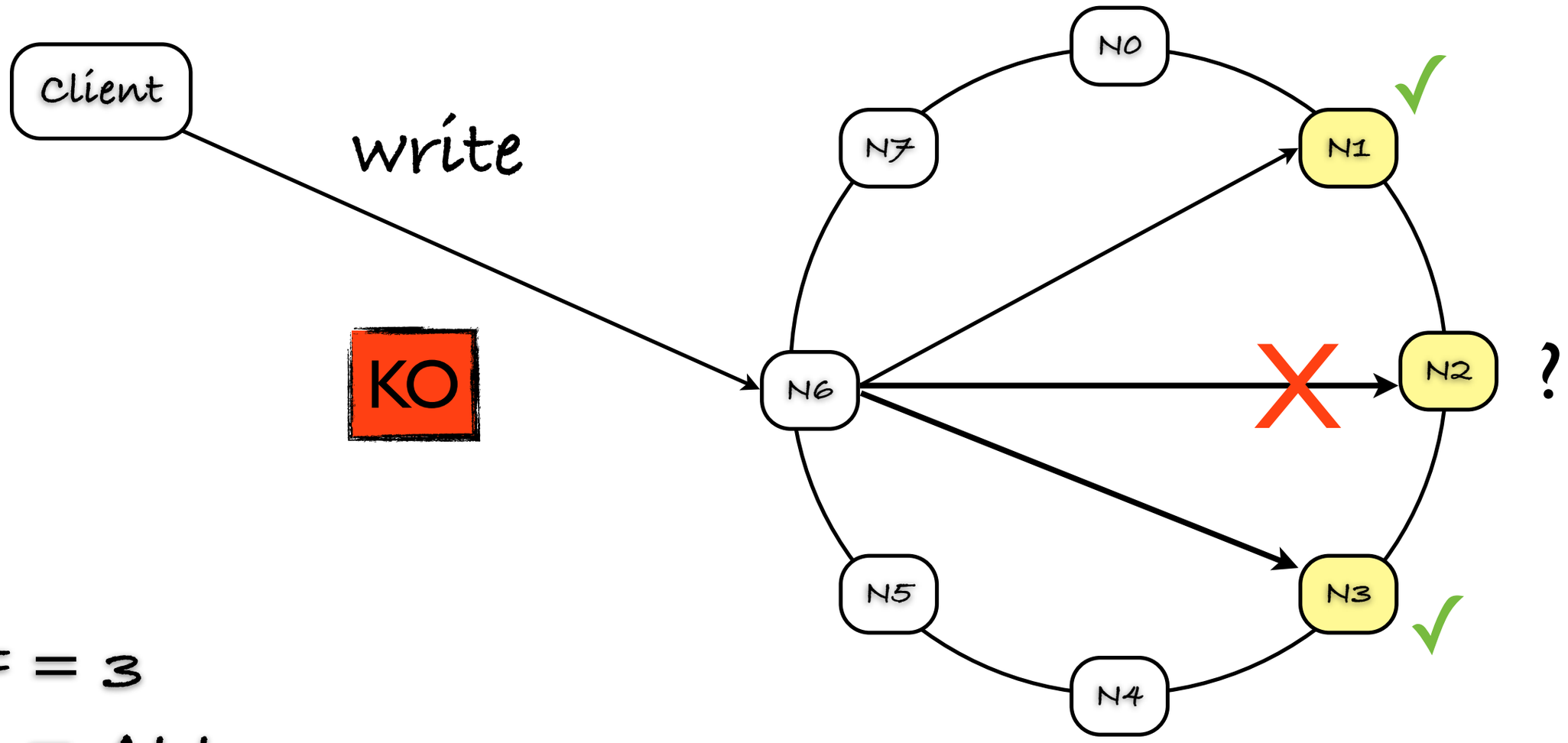
Exemple 1



RF = 3

CL = ALL

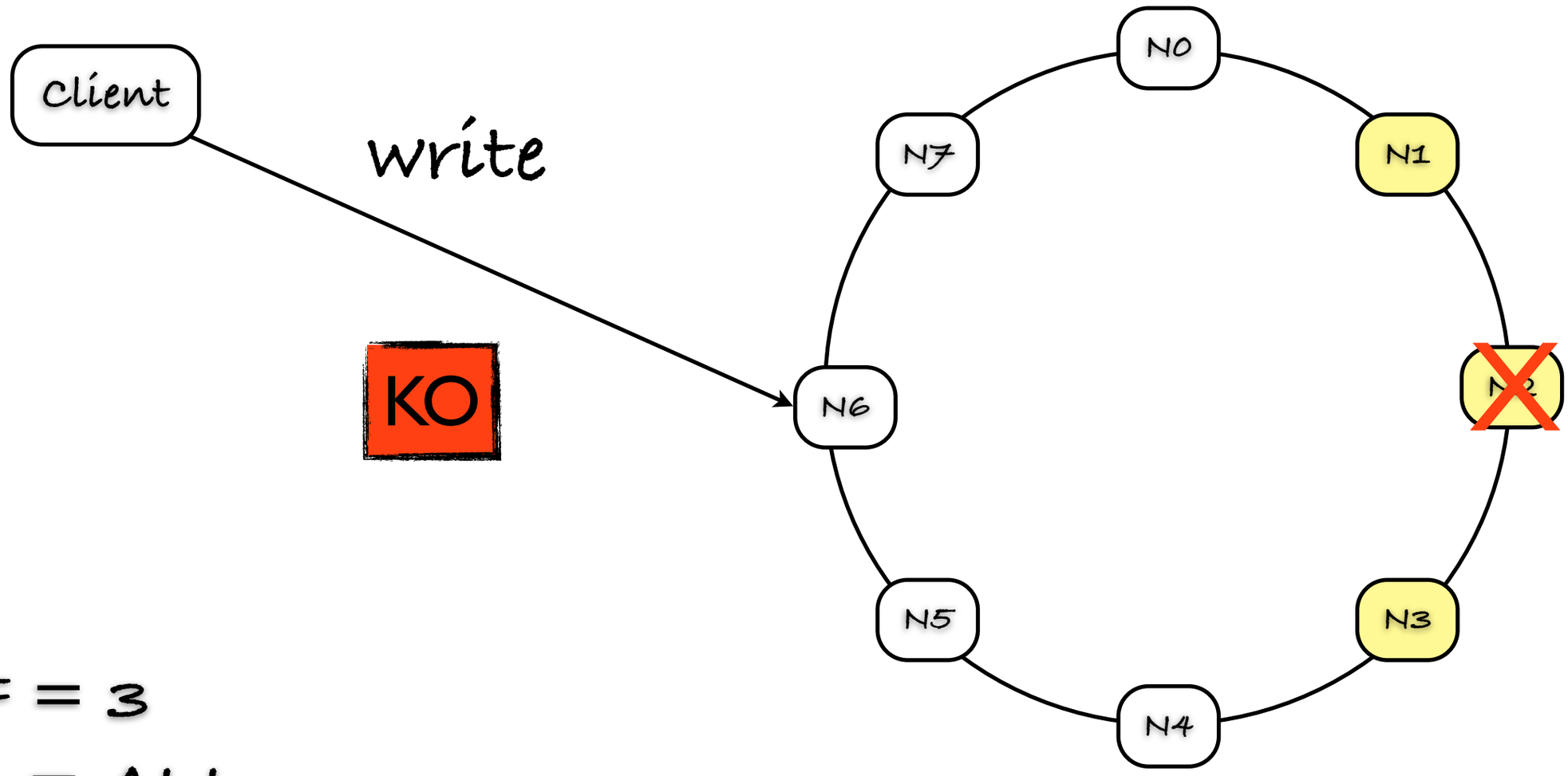
Exemple 2



RF = 3

CL = ALL

Exemple 3



RF = 3

CL = ALL

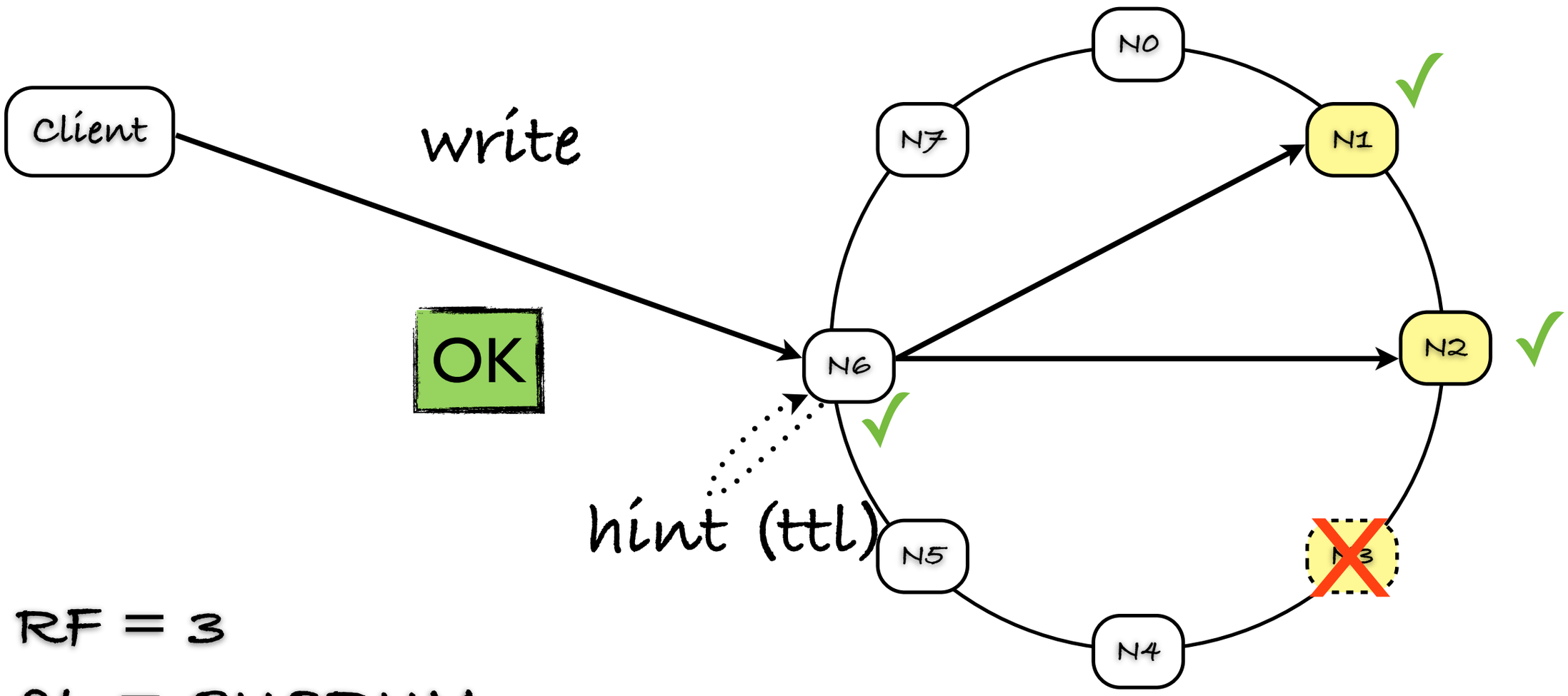
Sondage

Rollback ou pas?

Pas de Rollback...

- La donnée va être propagée
- Mais comment?

Hinted Handoff

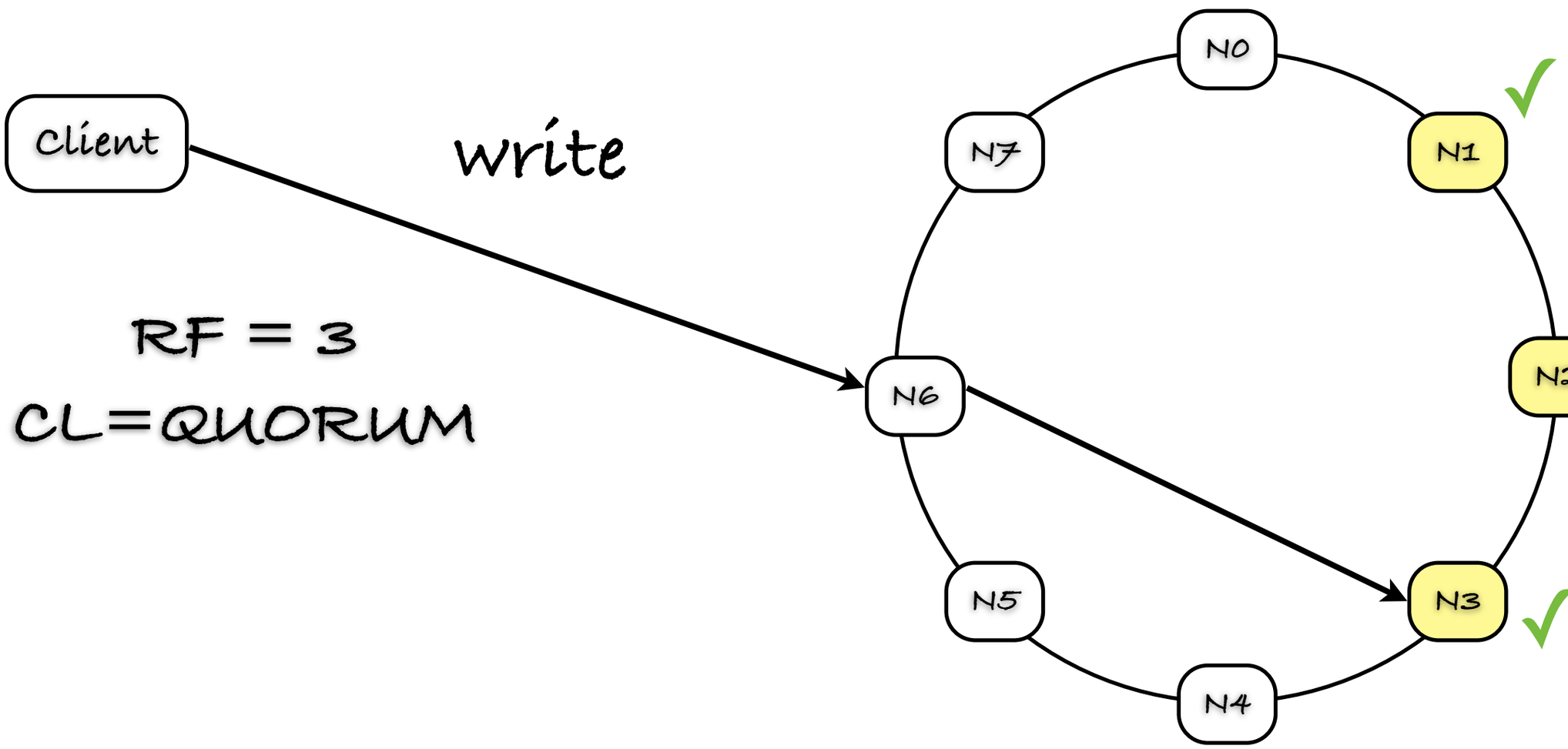


RF = 3

CL = QUORUM

·····➤ asynchrone

Hinted Handoff



N3 revient dans le ring

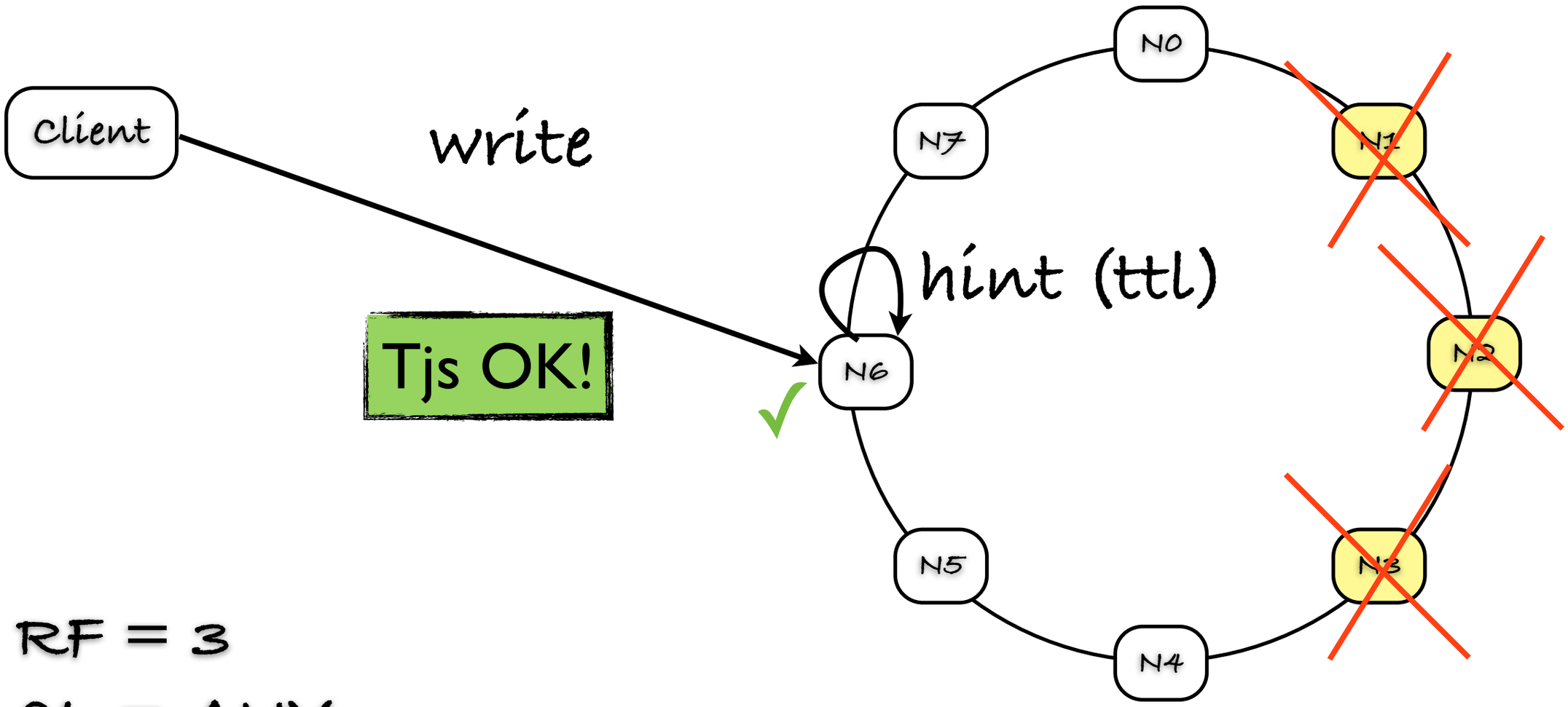
Hinted Handoff

- 1er mécanisme de reprise sur erreur
- TTL du hint

CL - ANY

écriture

Exemple CL Any

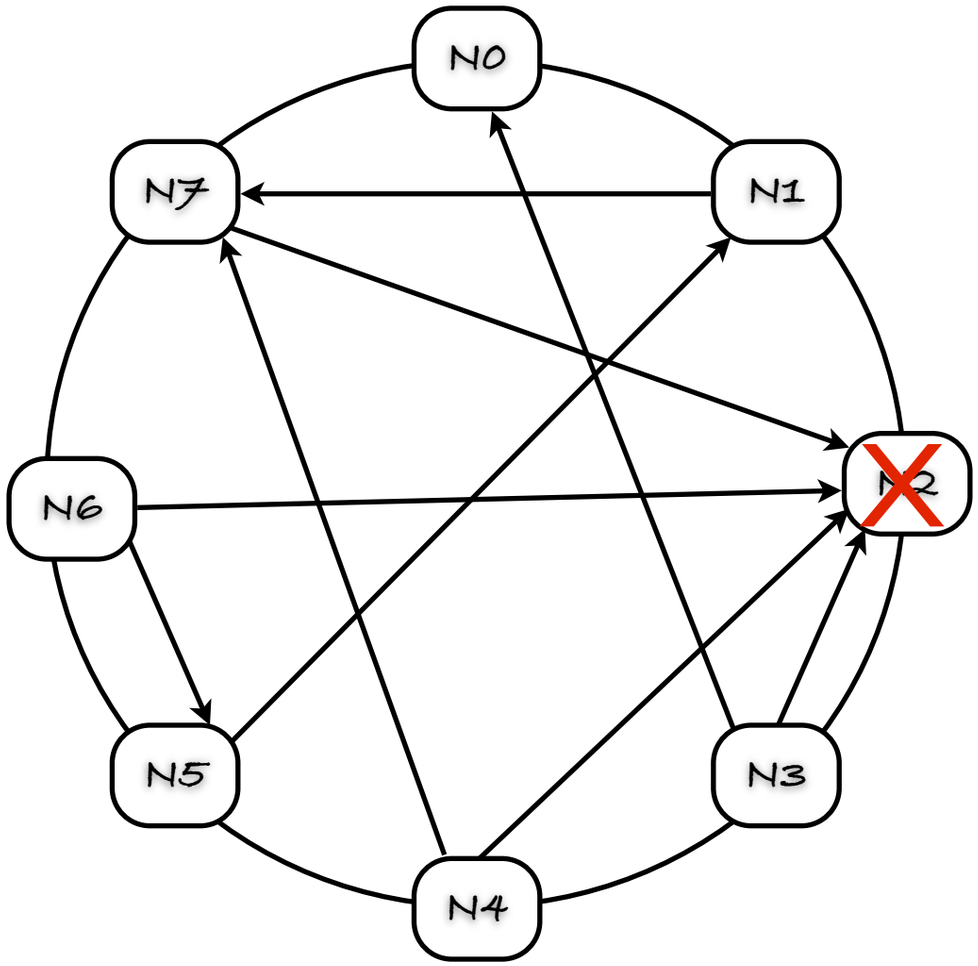


RF = 3

CL = ANY

Etat du ring?

Gossip Protocol



- Pas d'ilots
- Découverte du ring
- Echange d'info sur le système

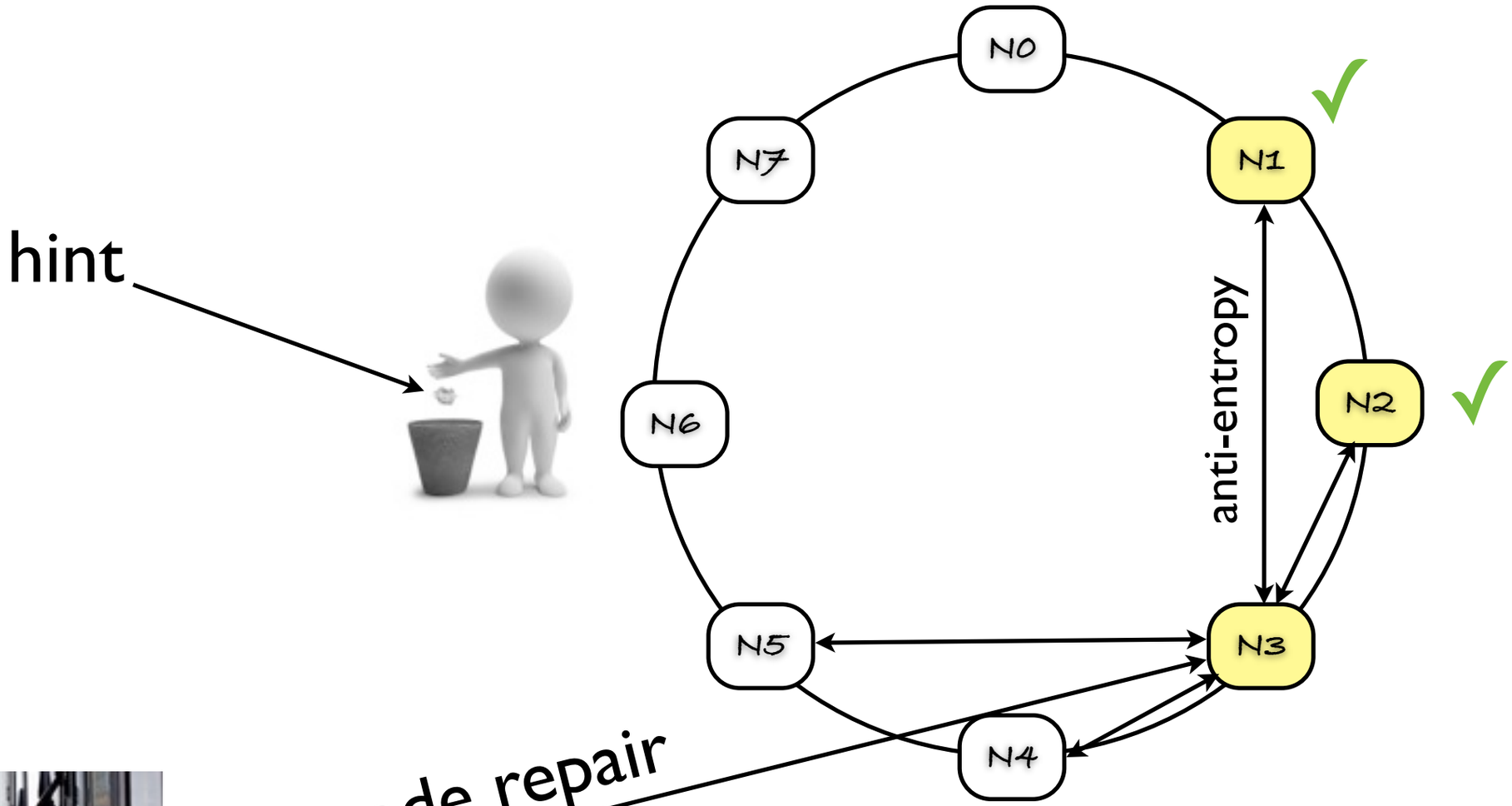
∿φ Accrual Failure Detector

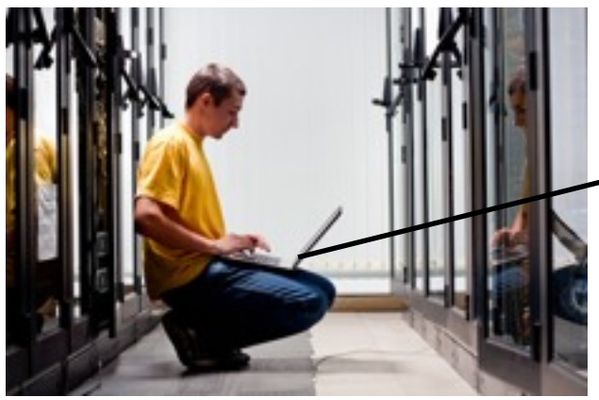
- Ni oui ni non
- Probabiliste

Expiration du hint

- Comment synchroniser les données lorsque le noeud revient dans le ring après expiration des hints?

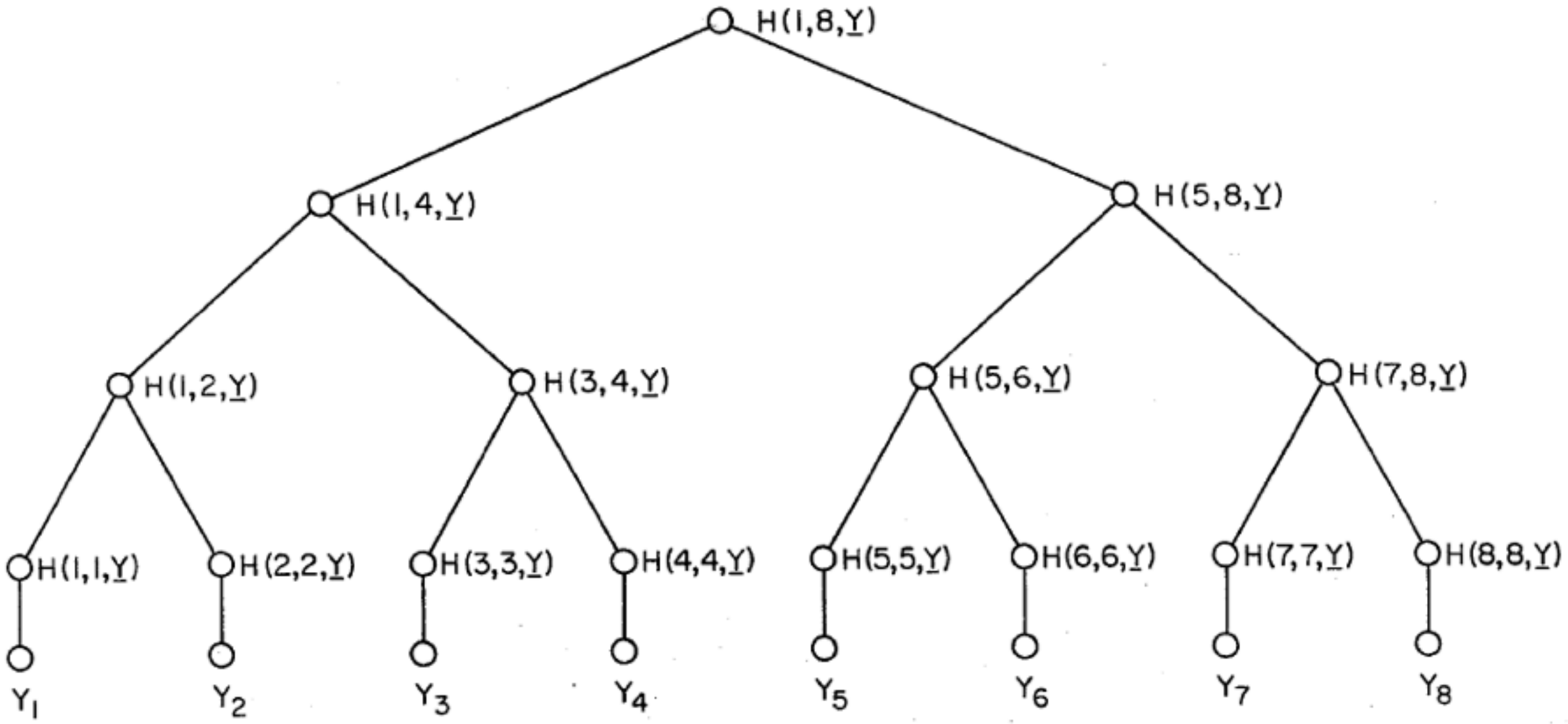
Anti-Entropy





2ème mécanisme de reprise sur erreur

Merkle Tree (1982)



United States Patent [19]

Merkle

Best A

[54] **METHOD OF PROVIDING DIGITAL SIGNATURES**

[75] Inventor: **Ralph C. Merkle**, Mountain View, Calif.

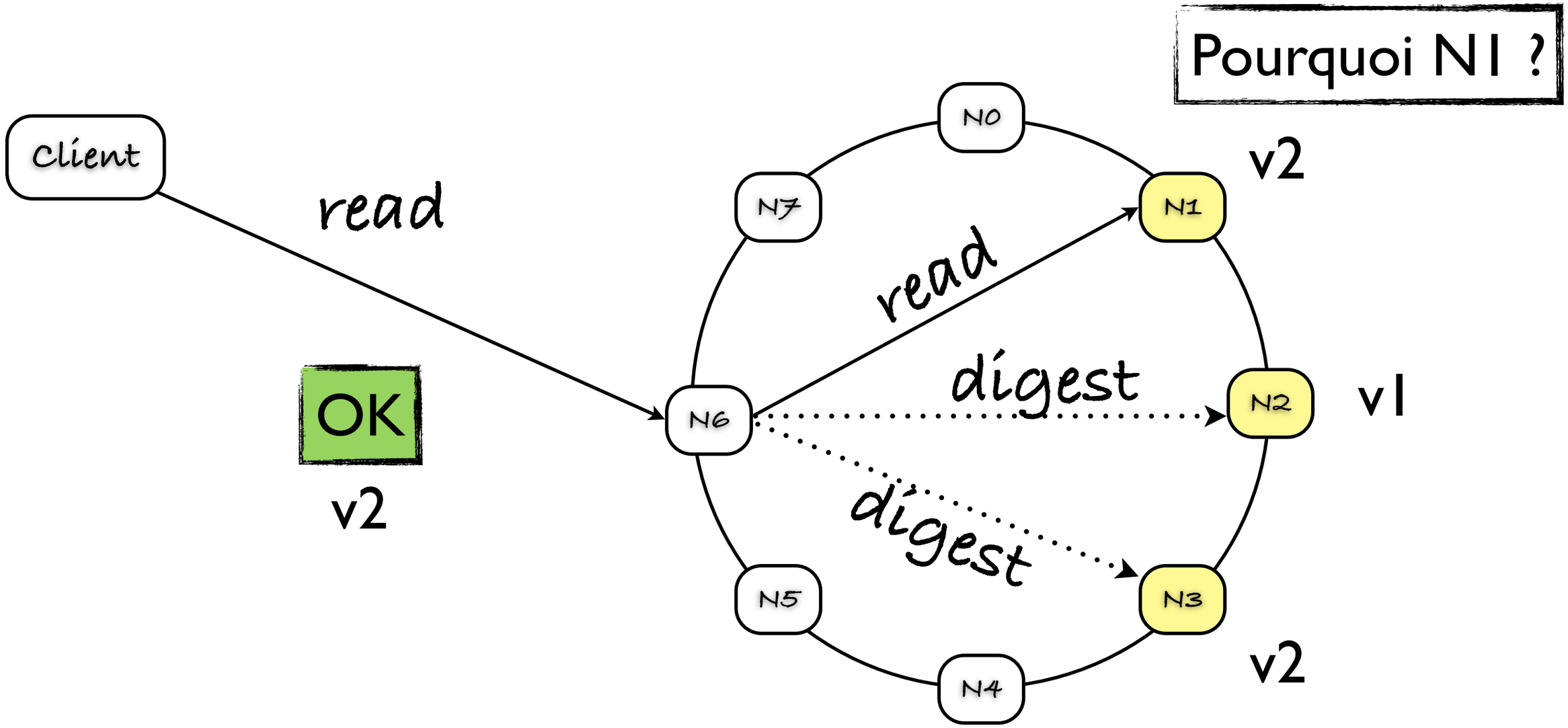
En lecture

- Tuneable consistency

CL - ONE

lecture

Exemple 1

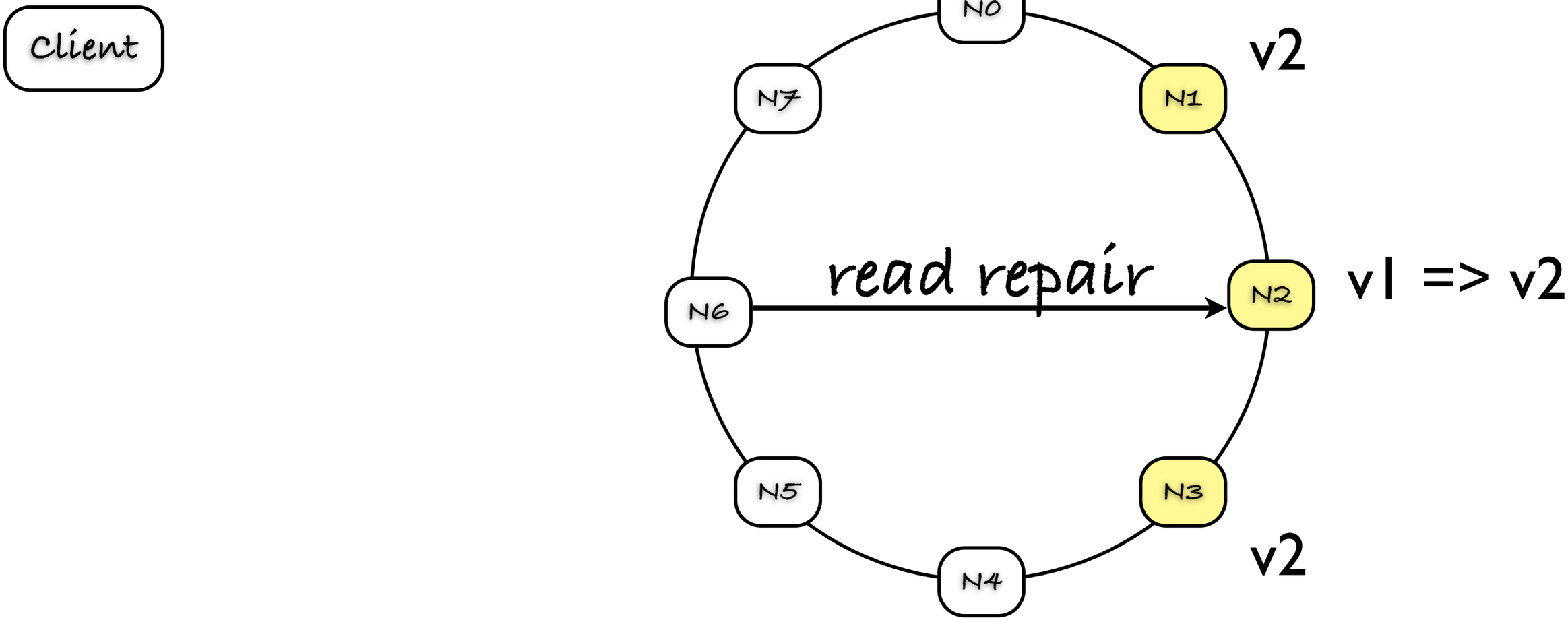


RF = 3

CL = ONE

Donnée la plus récente parmi ONE replica

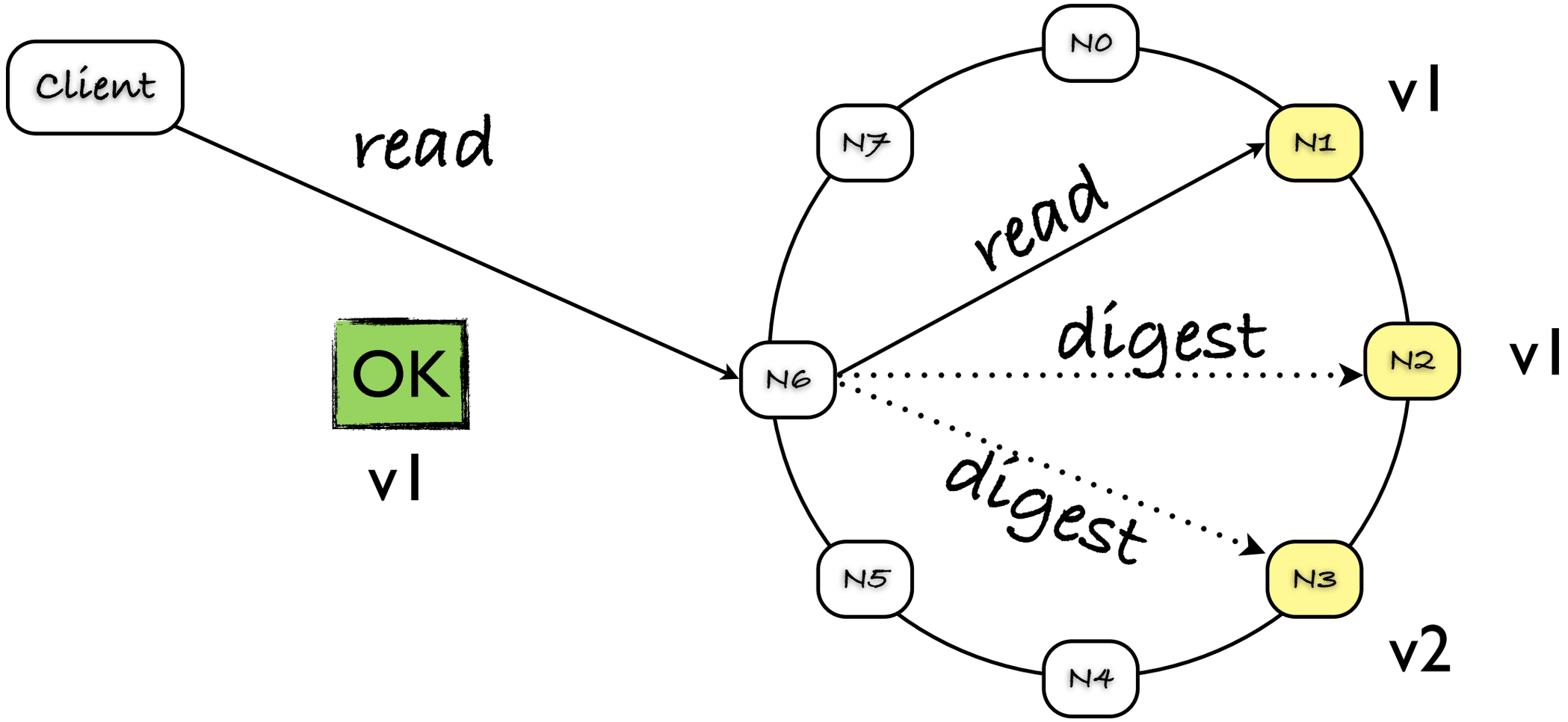
Exemple 1 (suite)



RF = 3

CL = ONE

Exemple 2



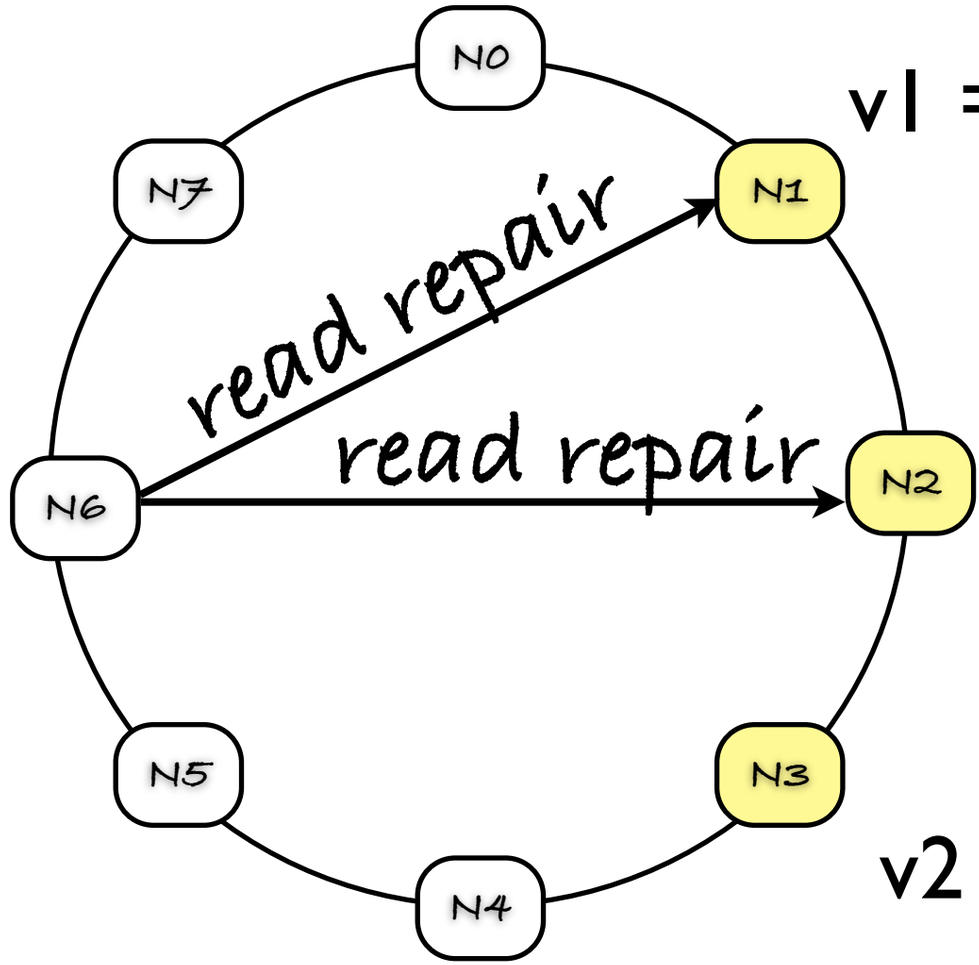
RF = 3

CL = ONE

Donnée la plus récente parmi ONE replica

Exemple 2 (suite)

Client



RF = 3

CL = ONE

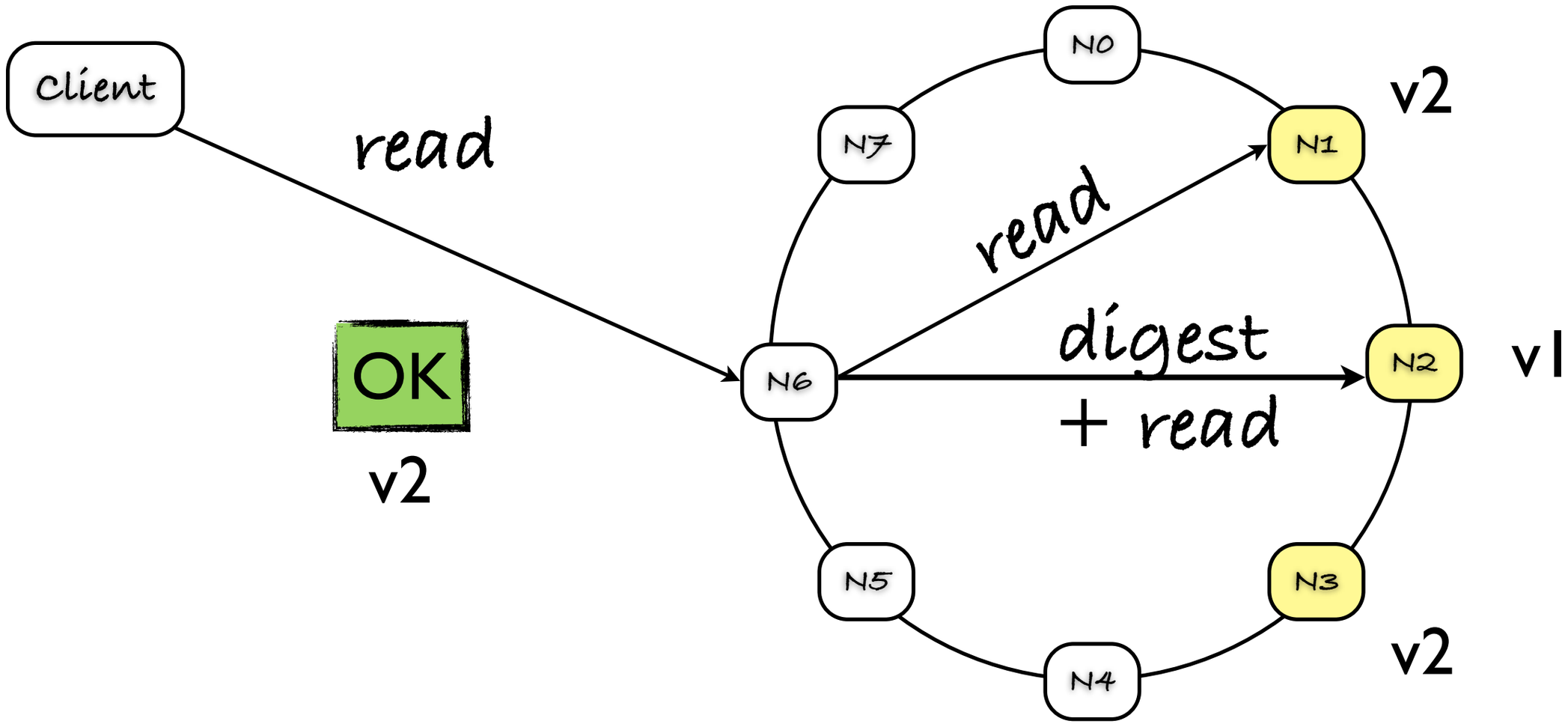
Read Repair

- 3ème mécanisme de reprise sur erreur
- Configurable (% de chance)

CL - QUORUM

lecture

Exemple 1



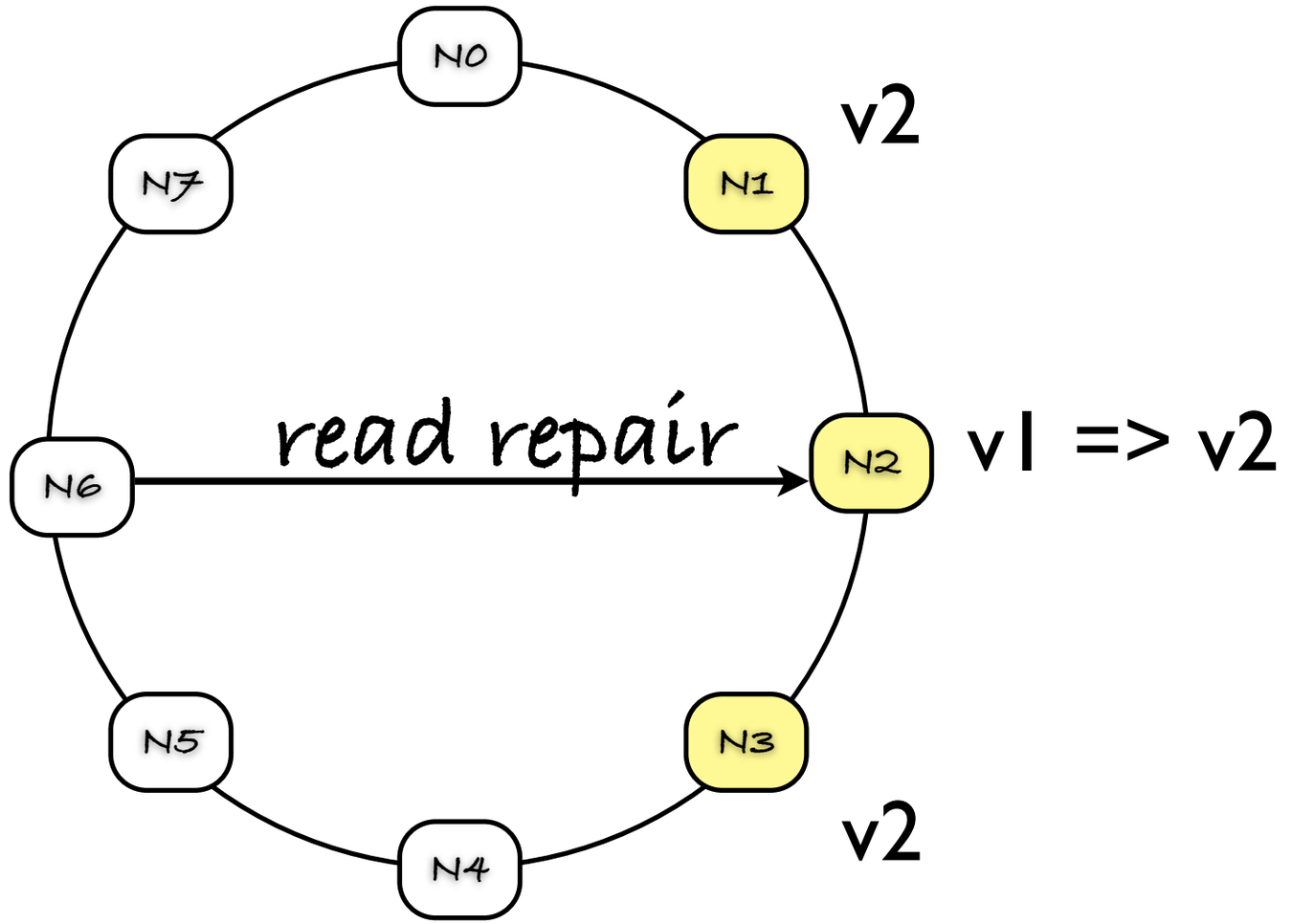
RF = 3

CL = QUORUM

Donnée la plus récente parmi 2 replicas

Exemple 1 (suite)

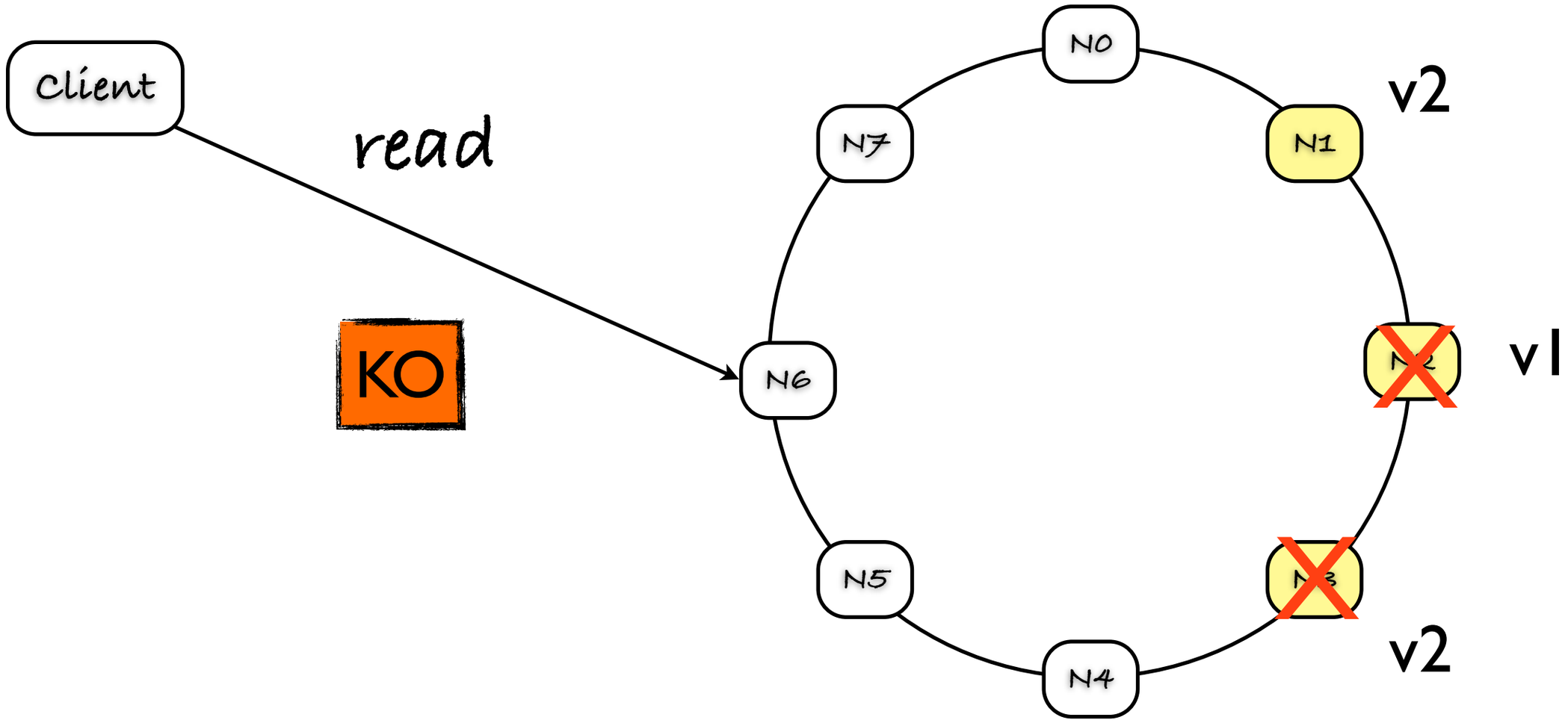
Client



RF = 3

CL = QUORUM

Exemple 2



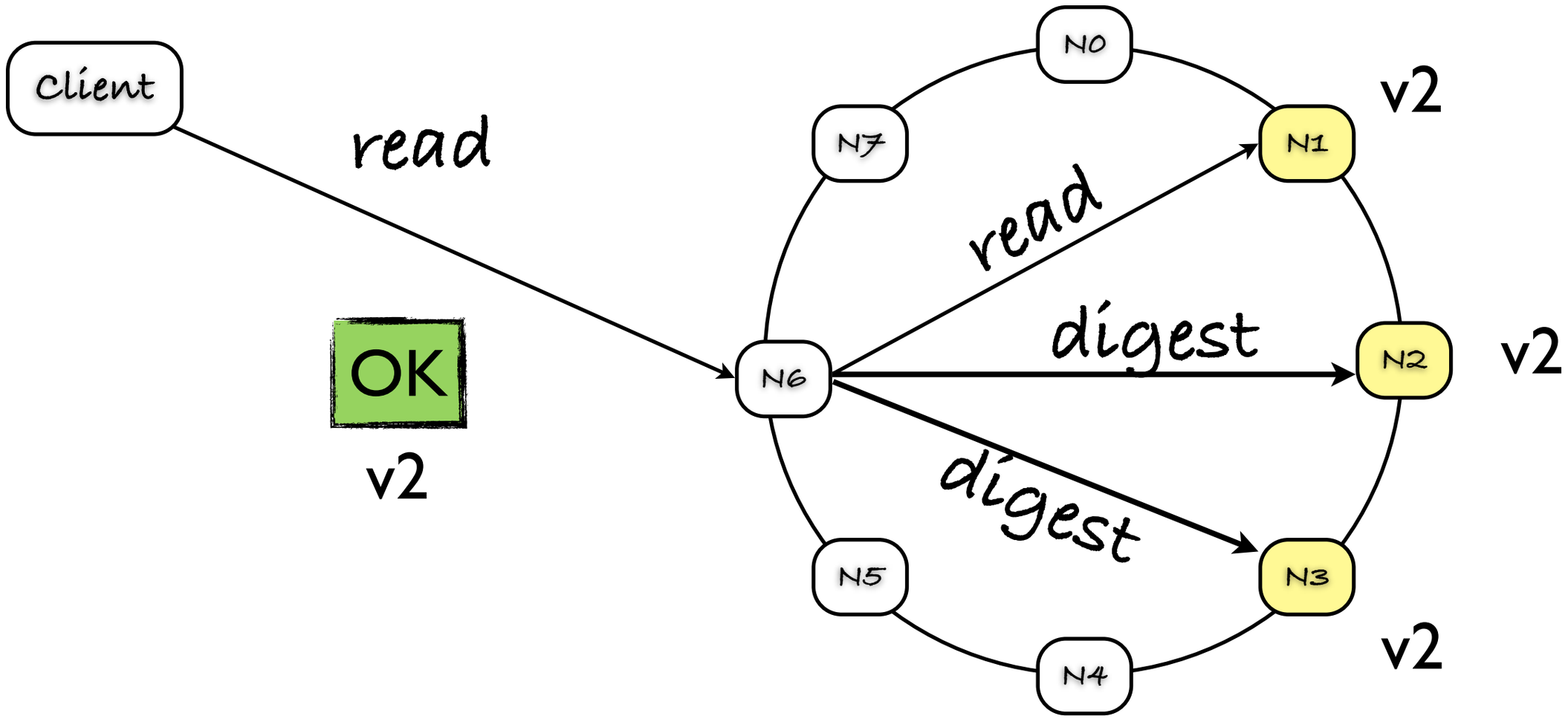
RF = 3

CL = QUORUM

CL - ALL

lecture

Exemple 1



RF = 3

CL = ALL

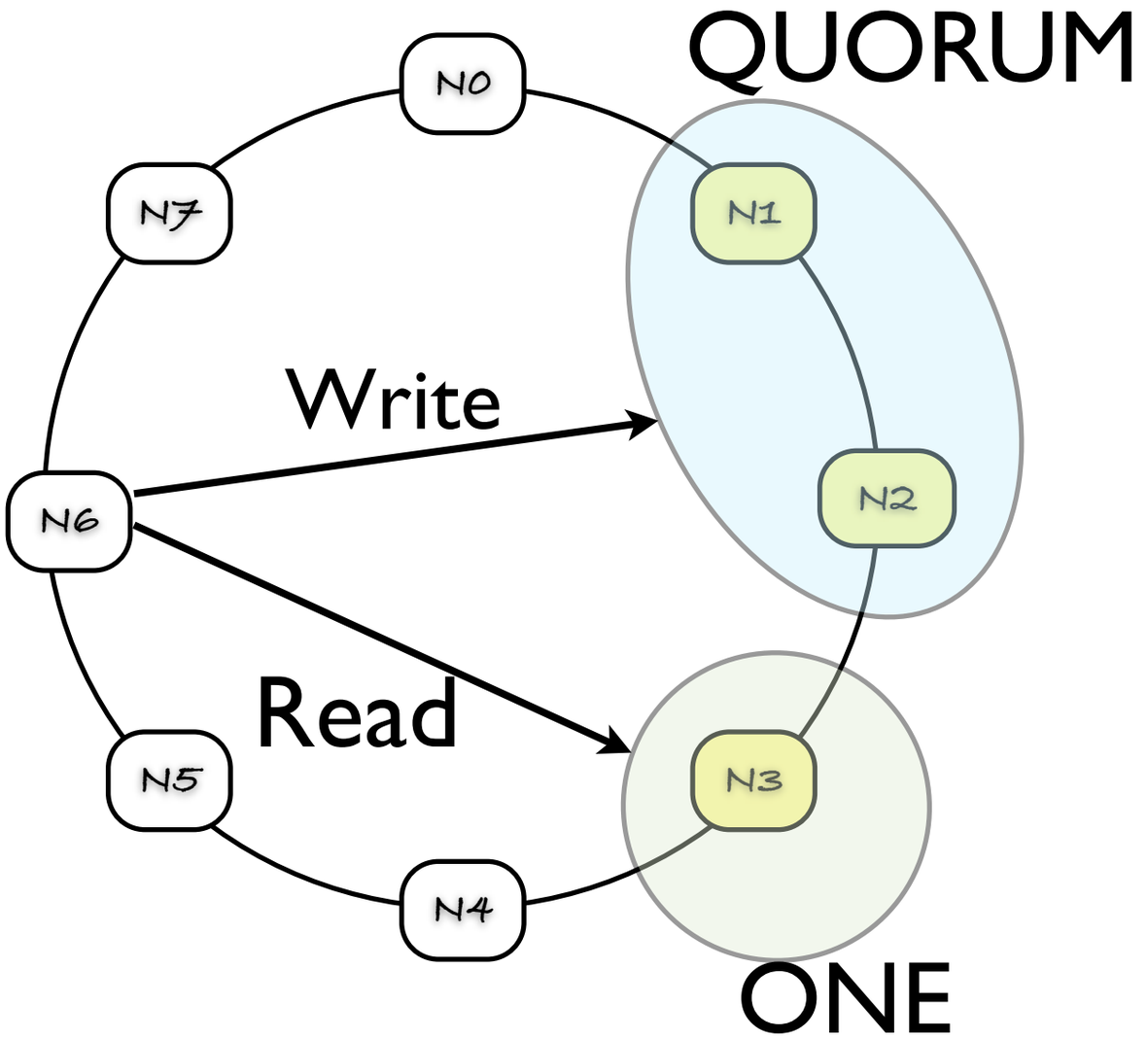
Donnée la plus récente parmi 3 replicas

Consistence Forte

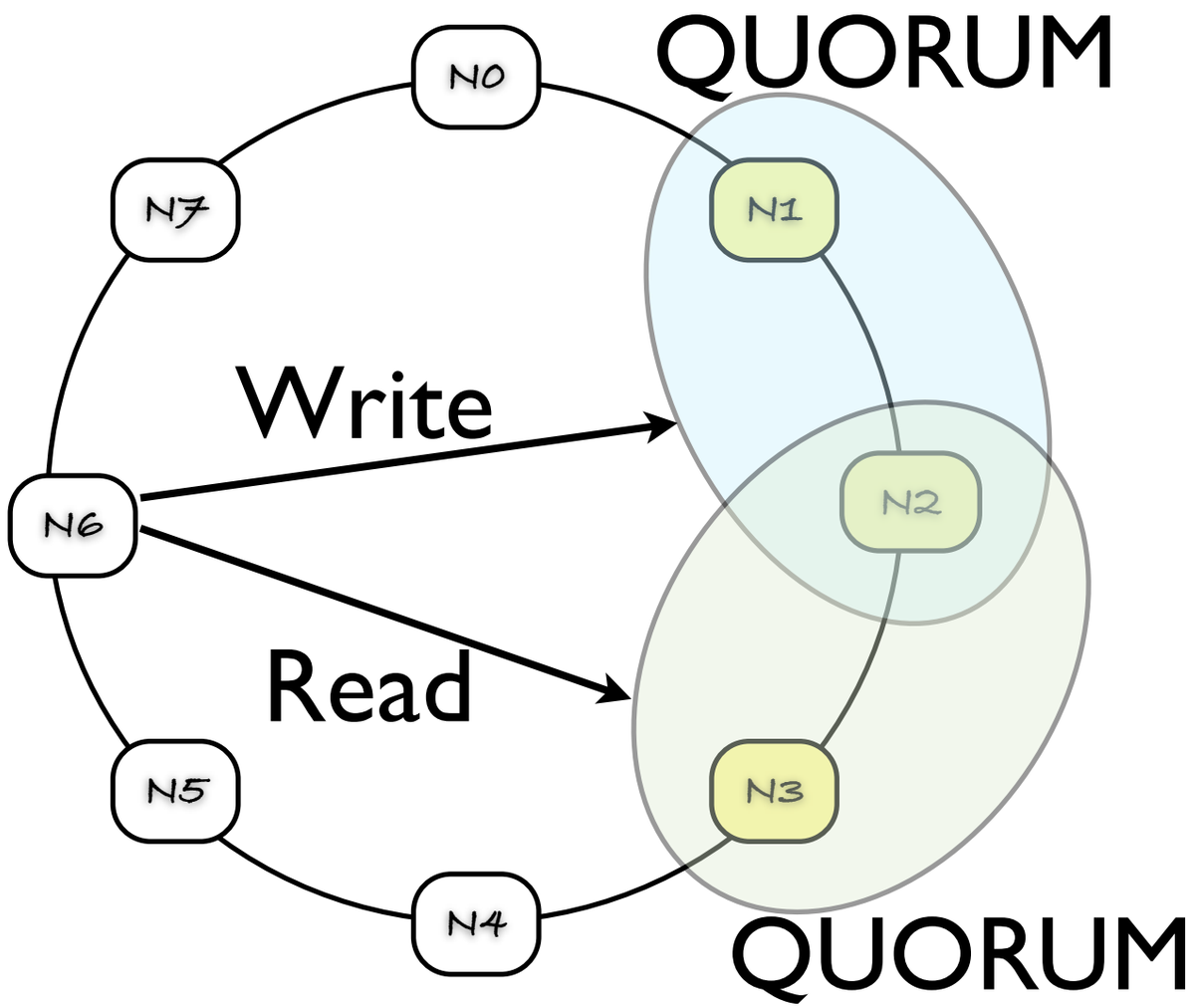
- Comment être certain que la donnée lue est la plus récente?
- formule W + R > RF
  - W: nb de replica contactées en écriture
  - R: nb de replica contactées en lecture
  - RF: replication factor
- Cas trivial: écriture en ALL et lecture en ONE

Consistency

RF = 3



«Weak»



«Strong»

Démo - calculerToken

```
import java.math.BigInteger;

public class SimpleTokenCalculator {

    public static void main(String args[]) {
        int nbNodes = Integer.parseInt(args[0]);

        for (int i = 0; i < nbNodes; i++) {
            BigInteger token = new BigInteger("2");
            token = token.pow(127);
            token = token.multiply(new BigInteger("" + i));
            token = token.divide(new BigInteger("" + nbNodes));
            System.out.println("token " + i + ": " + token);
        }
    }
}
```

Démo - création KS + CF

- `create keyspace **ks** with placement_strategy = 'SimpleStrategy' and strategy_options = {replication_factor:2};`
- `use ks;`
- `create column family **user** with key_validation_class = 'AsciiType' and comparator = 'AsciiType' and default_validation_class = 'AsciiType';`
- `set user['antonio@devoxx.fr']['nom']= 'Goncalves';`
- `get user['antonio@devoxx.fr'];`

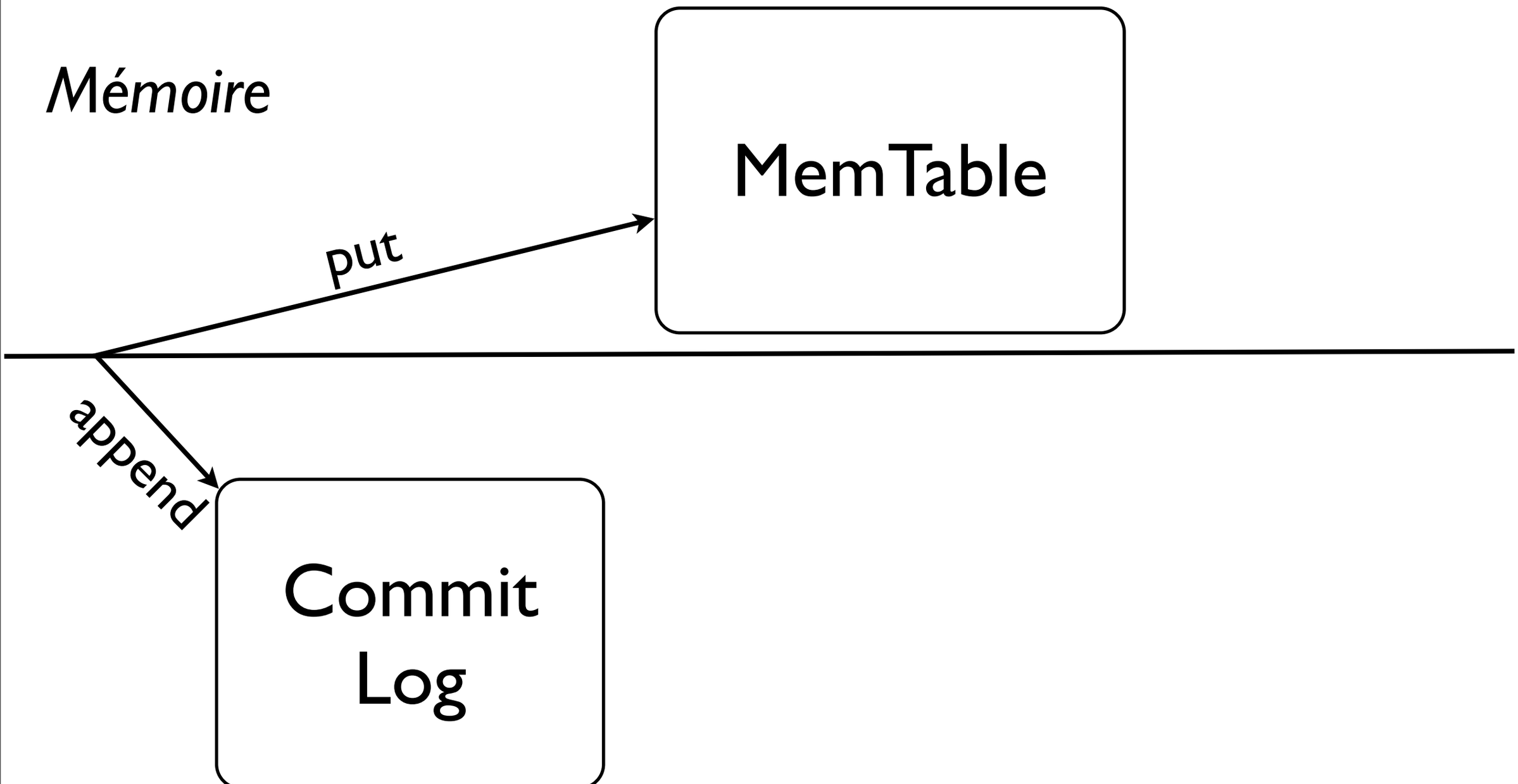
Questions?

Performance en écriture

- En écriture, Cassandra est extrêmement rapide, pourquoi?

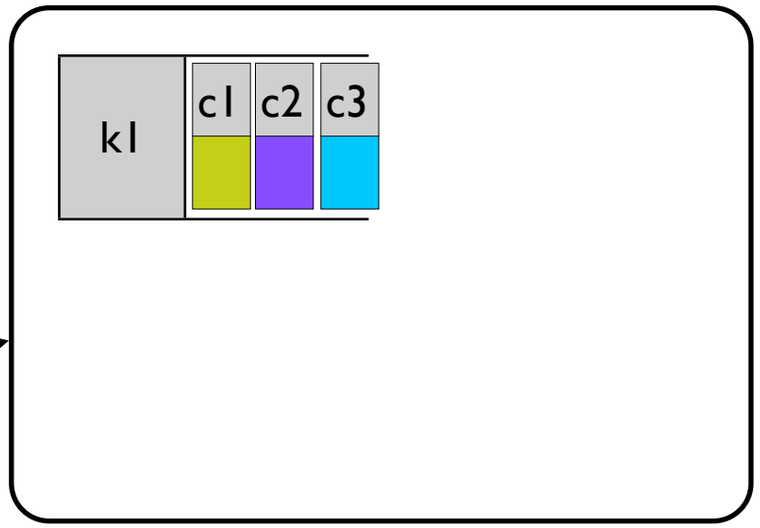
Performance en écriture

Mémoire



Fichier

Mémoire



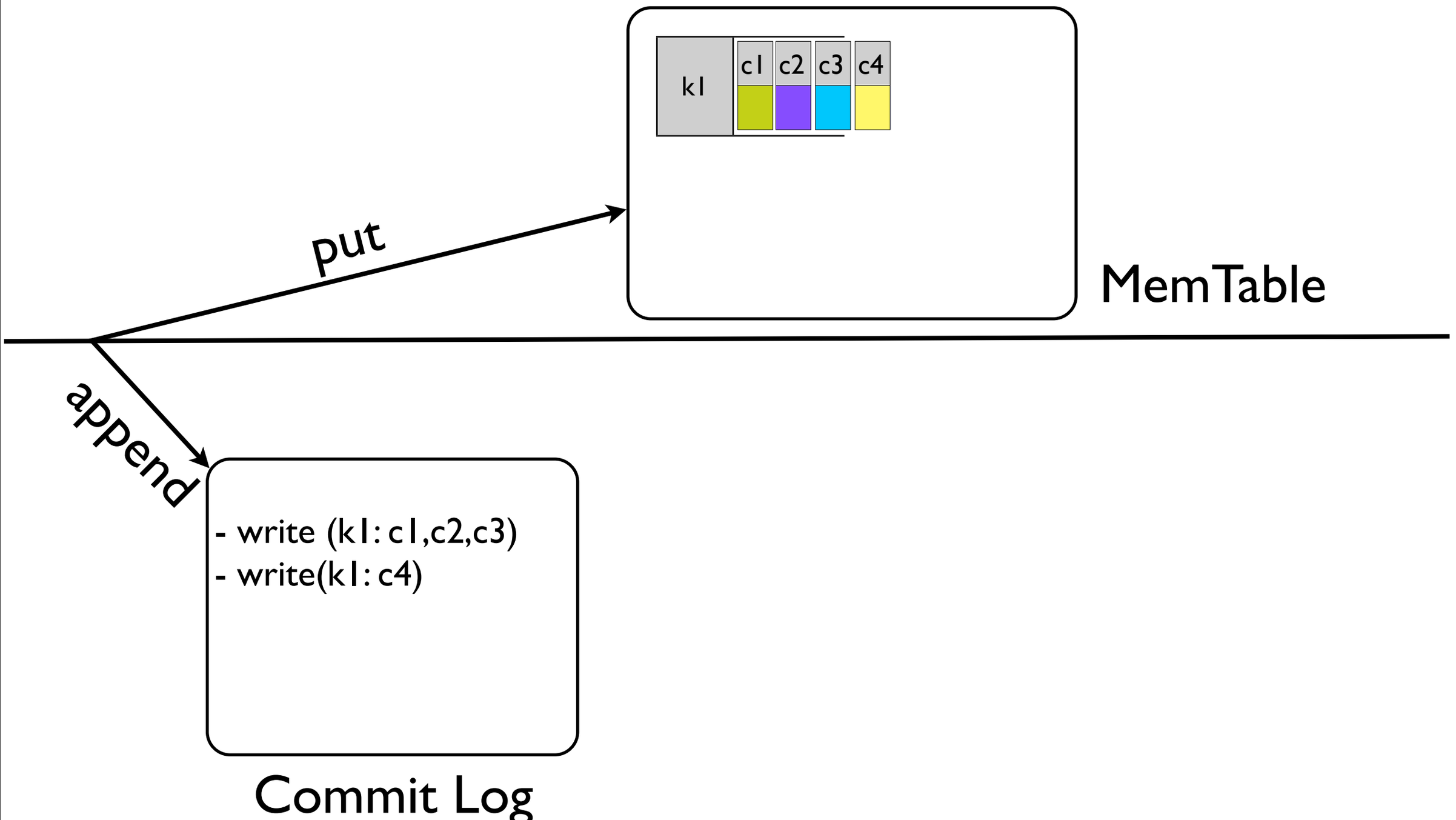
MemTable

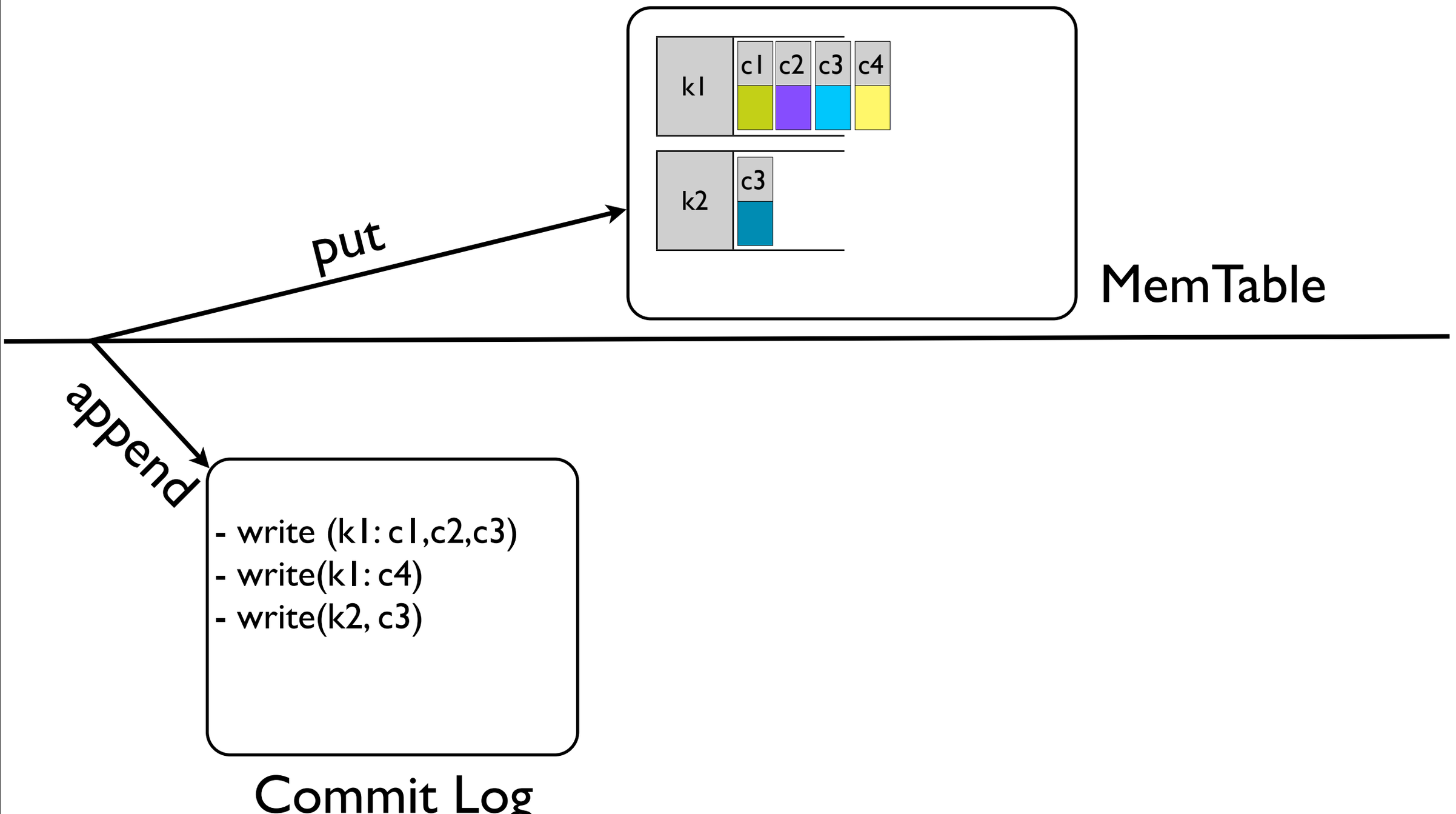
put

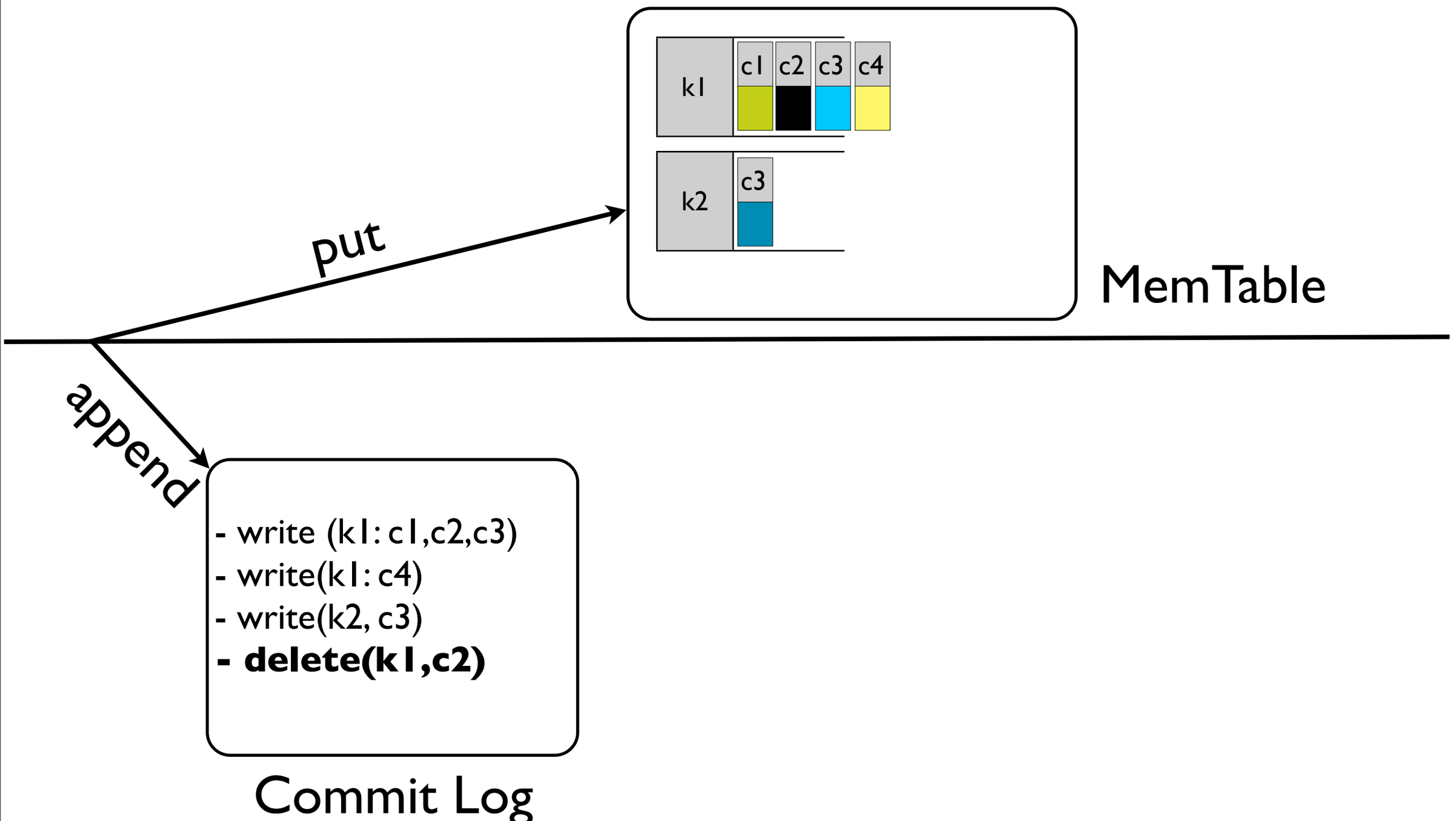
append

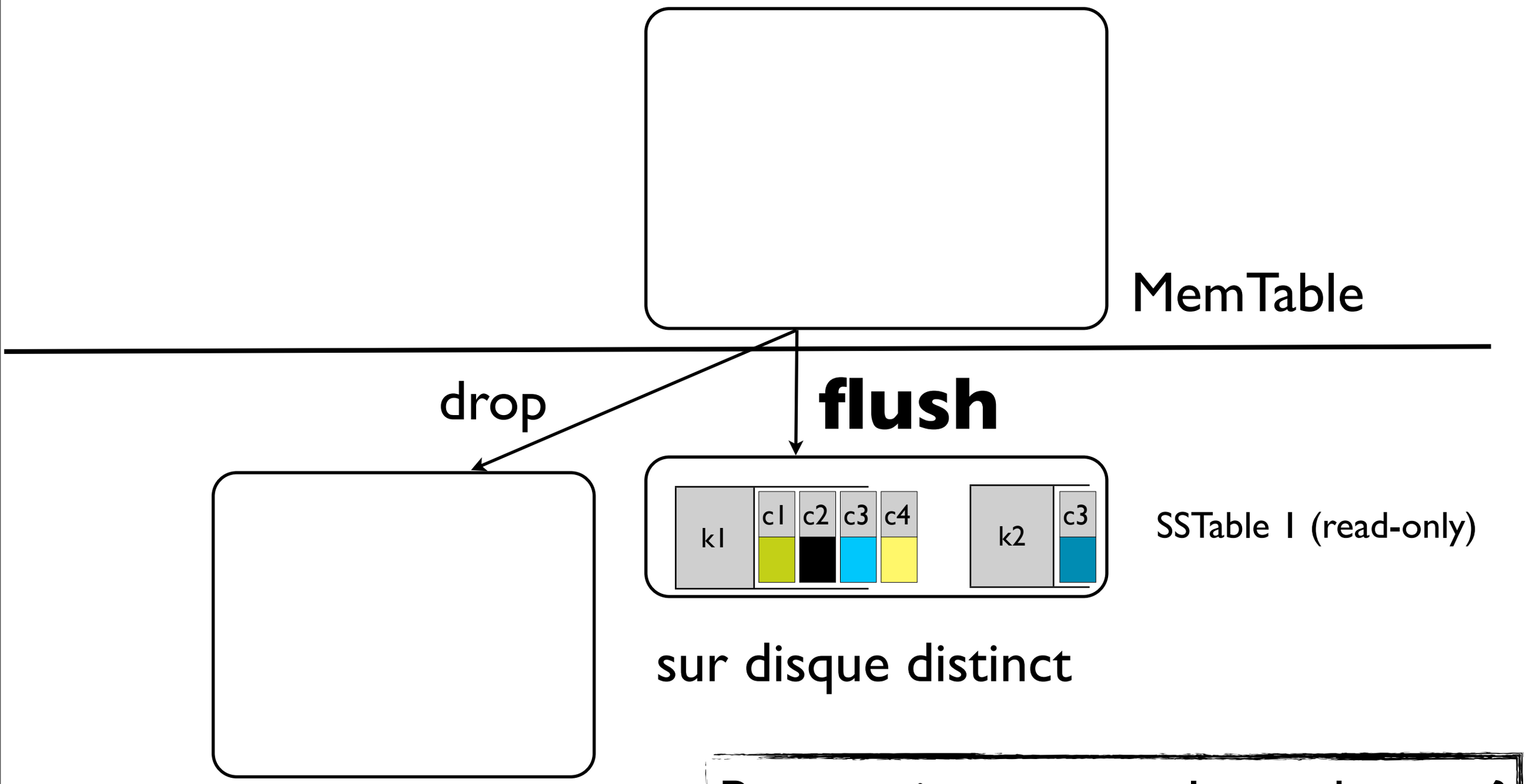
- write (k1: c1,c2,c3)

Commit Log


k1
c1
c2
c3
c4
put
MemTable
append
- write (k1: c1,c2,c3)
- write(k1: c4)
Commit Log

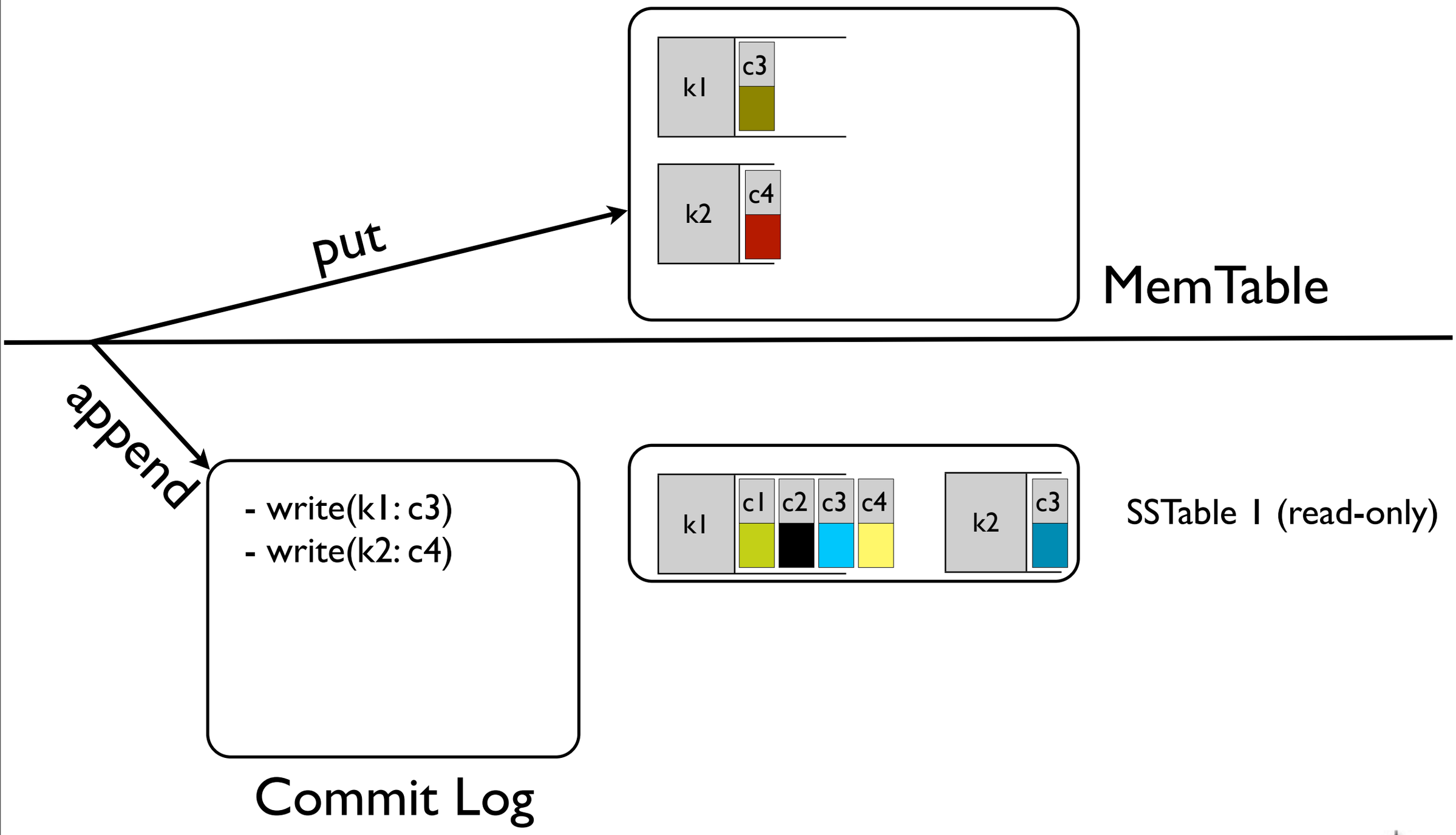

k1
c1
c2
c3
c4
k2
c3
MemTable
put
append
- write (k1: c1,c2,c3)
- write(k1: c4)
- write(k2, c3)
Commit Log


k1
c1
c2
c3
c4
k2
c3
MemTable
put
append
- write (k1: c1,c2,c3)
- write(k1: c4)
- write(k2, c3)
- delete(k1,c2)
Commit Log

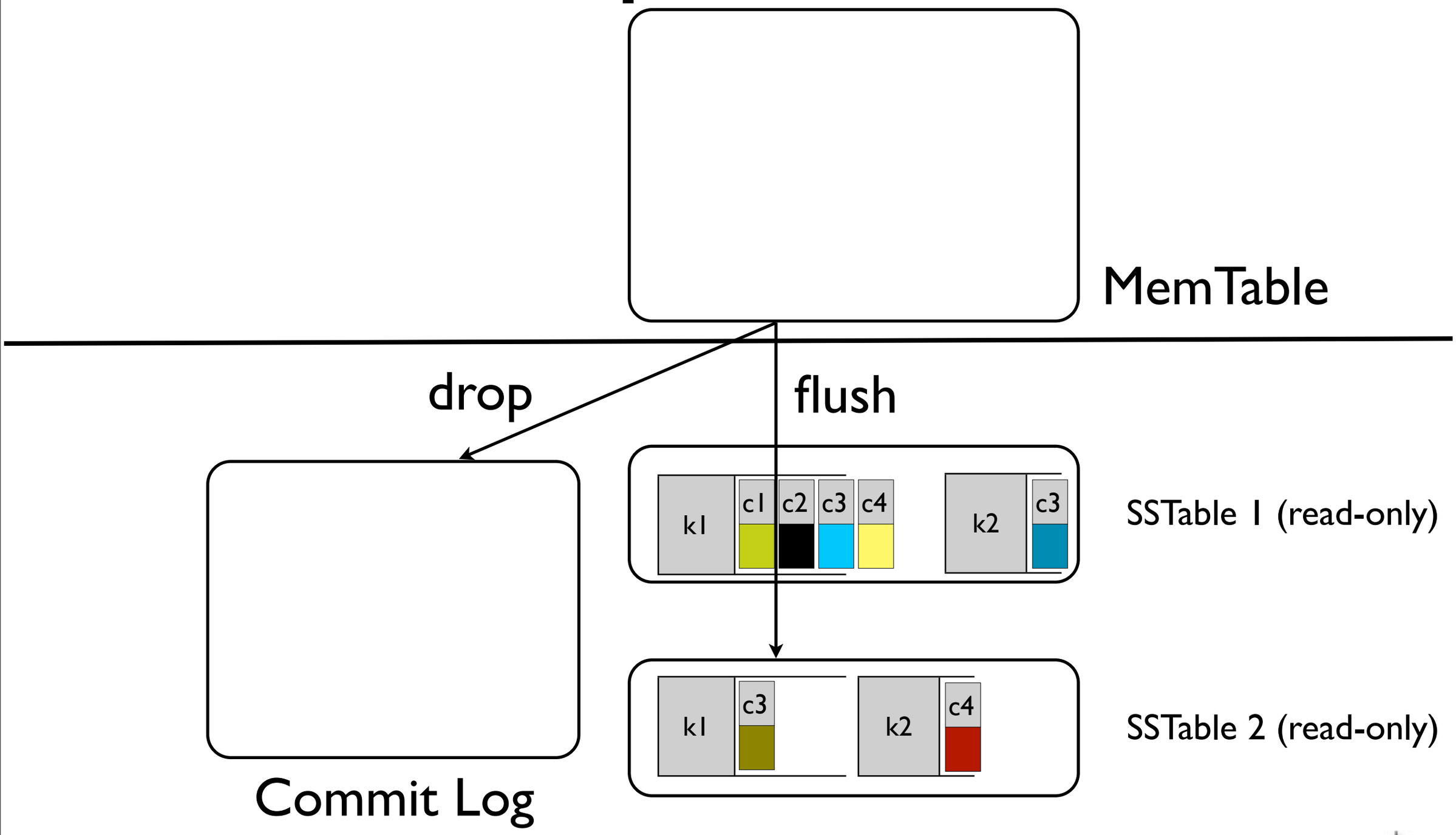

MemTable
drop
flush
k1
c1
c2
c3
c4
k2
c3
SSTable 1 (read-only)
sur disque distinct
Commit Log
Pourquoi conserver la tombstone?

Delete distribué

- Grace period: 10 jours par défaut


k1
c3
k2
c4
MemTable
put
append
- write(k1: c3)
- write(k2: c4)
Commit Log
k1
c1
c2
c3
c4
k2
c3
SSTable 1 (read-only)

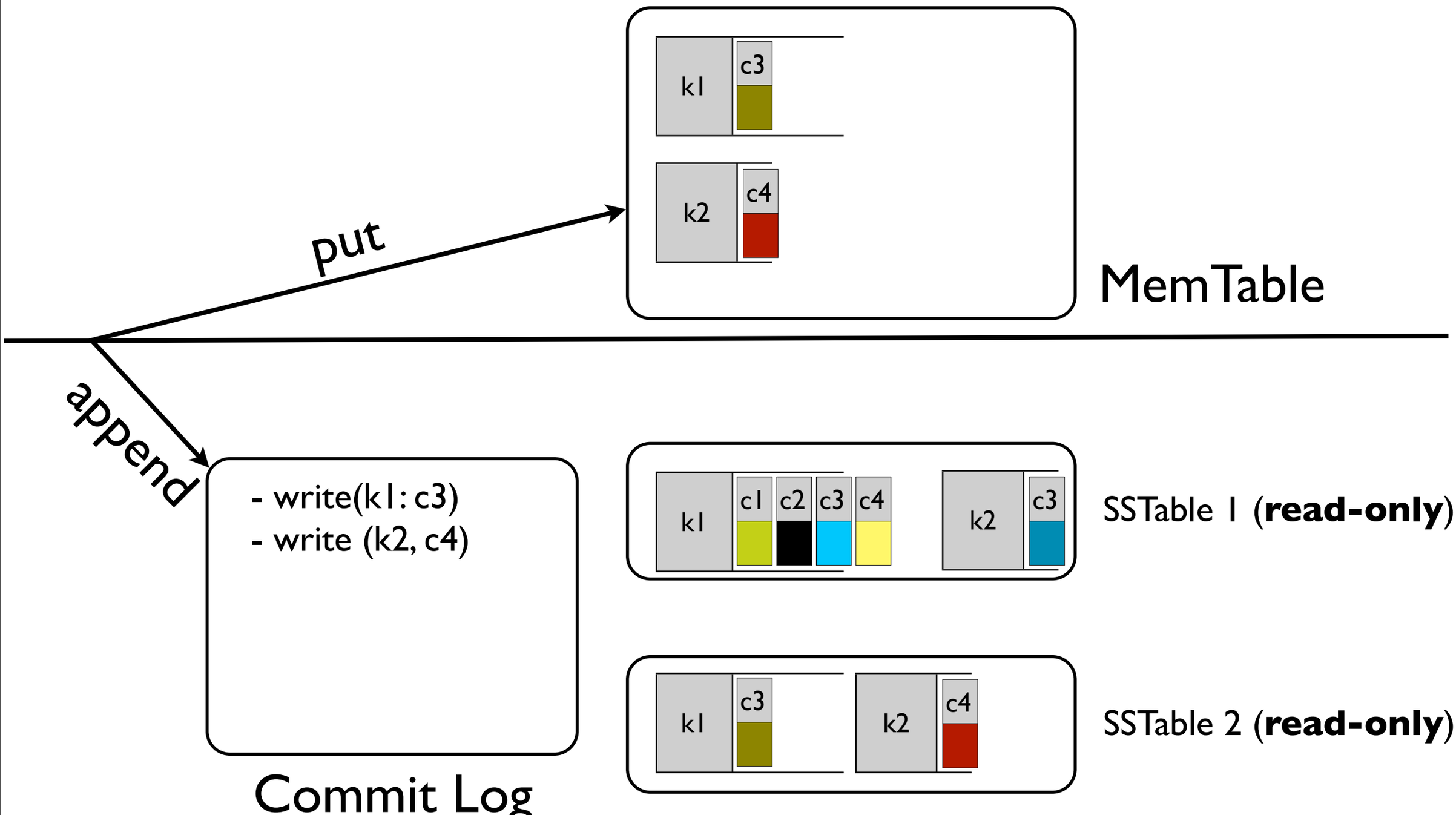
Pas d'update au flush



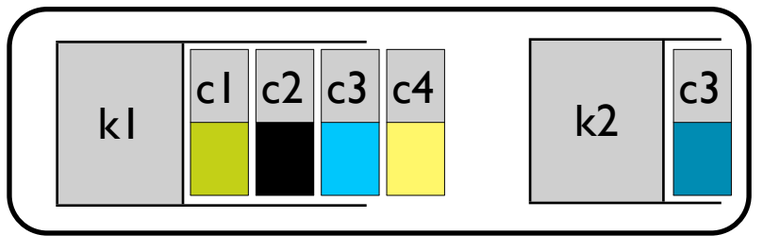
Backups?

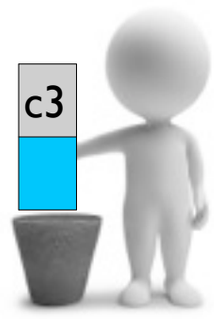
- SSTables sont readonly....

Quid de la lecture?

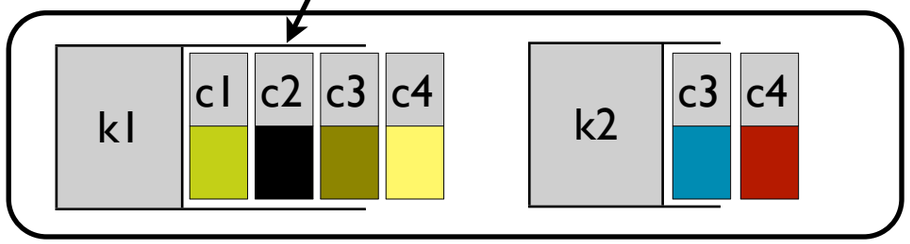


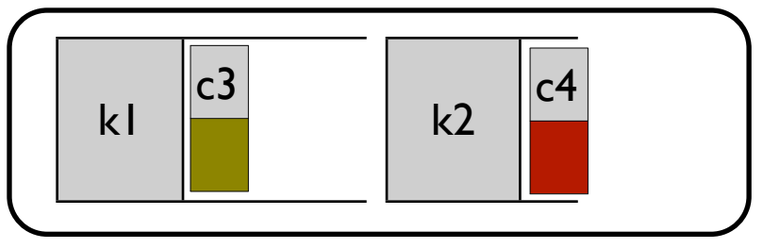
Compaction





si < grace period



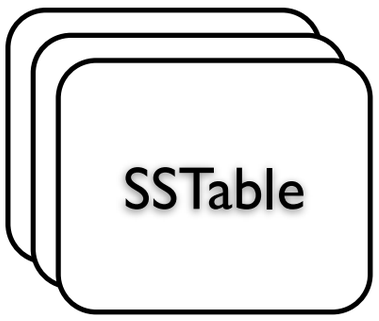


Lecture

MemTable

Mémoire

Fichiers



6- Contient la valeur...

Lecture

MemTable

Row Cache

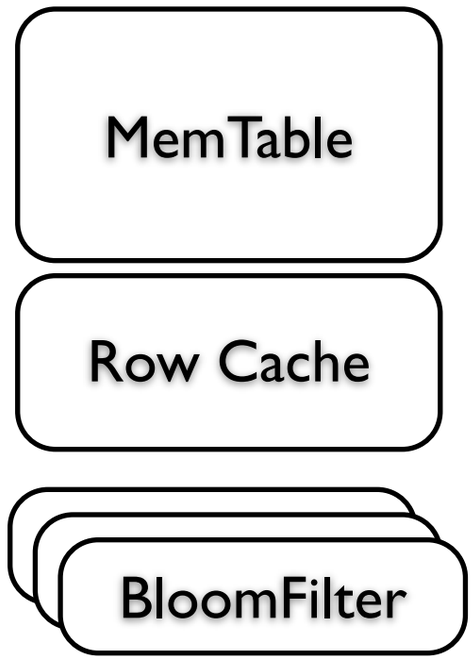
1- Pertinent lorsque le client spécifie le nom de la colonne

Mémoire

Fichiers

6- Contient la valeur...

Lecture

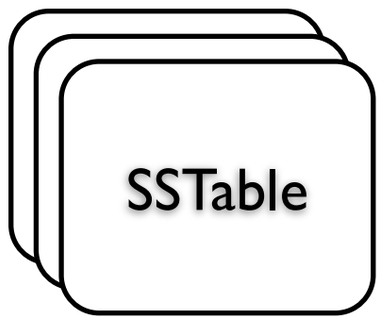


1- Pertinent lorsque le client spécifie le nom de la colonne

2- Stocke l'intégralité d'une row

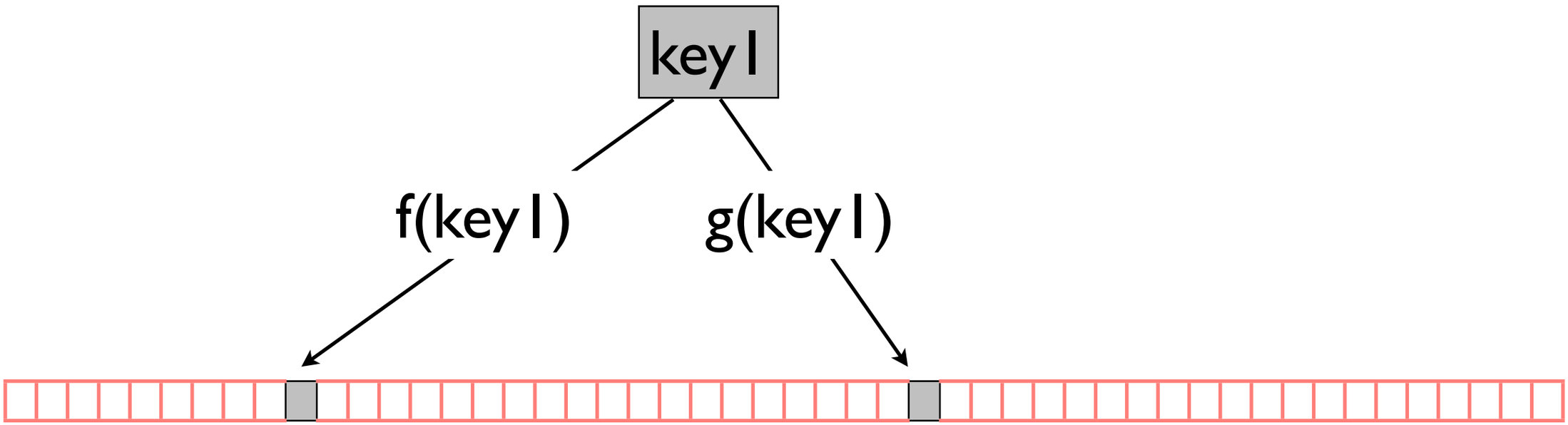
Mémoire

Fichiers

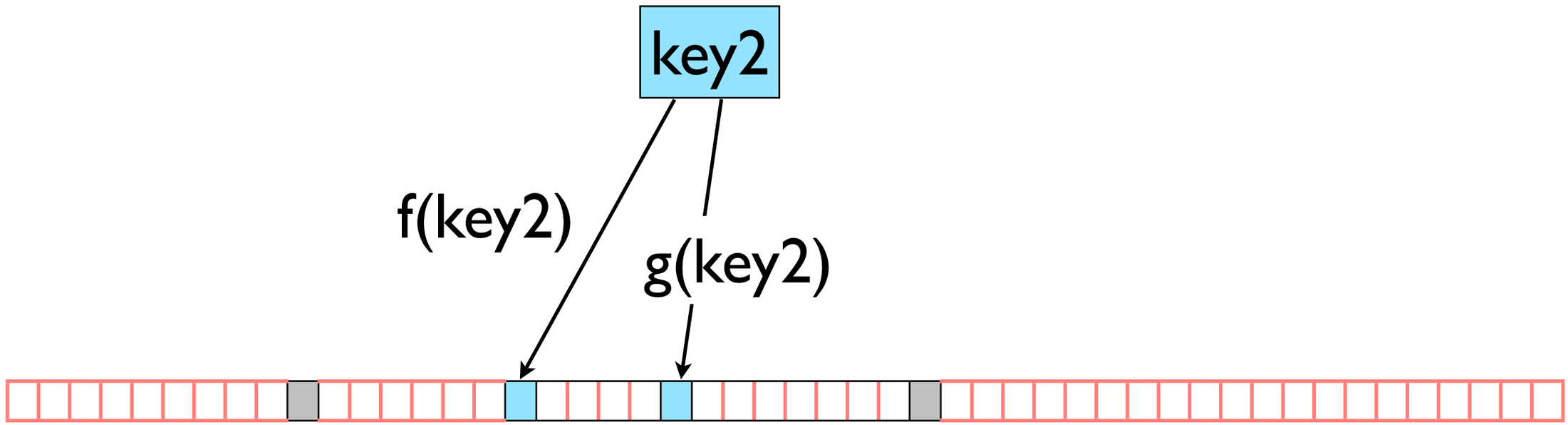


6- Contient la valeur...

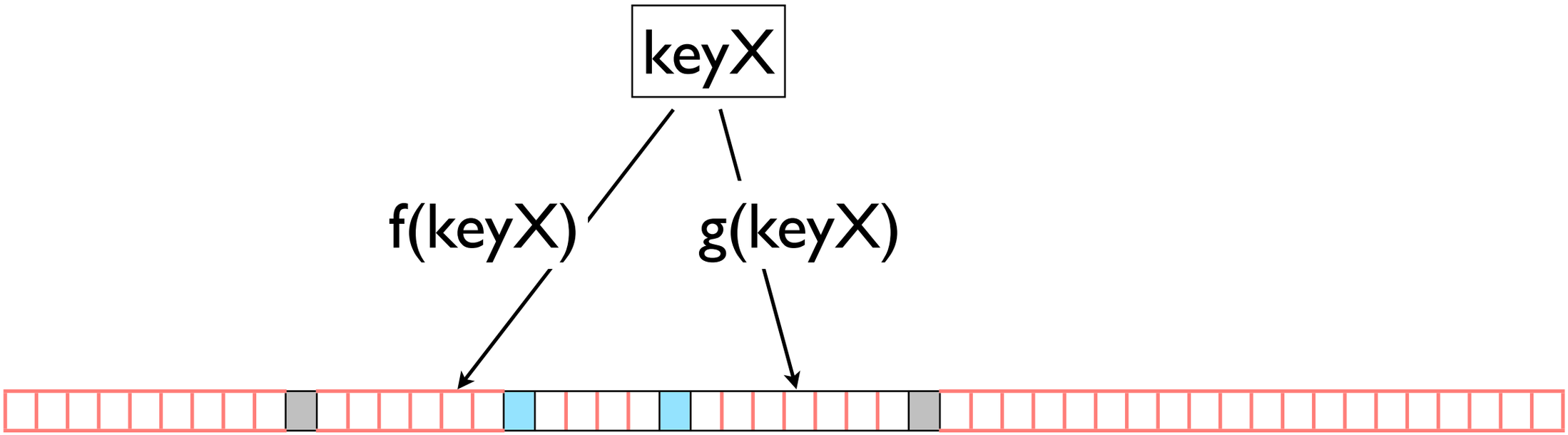
Bloom Filter (création)



Bloom Filter (création)

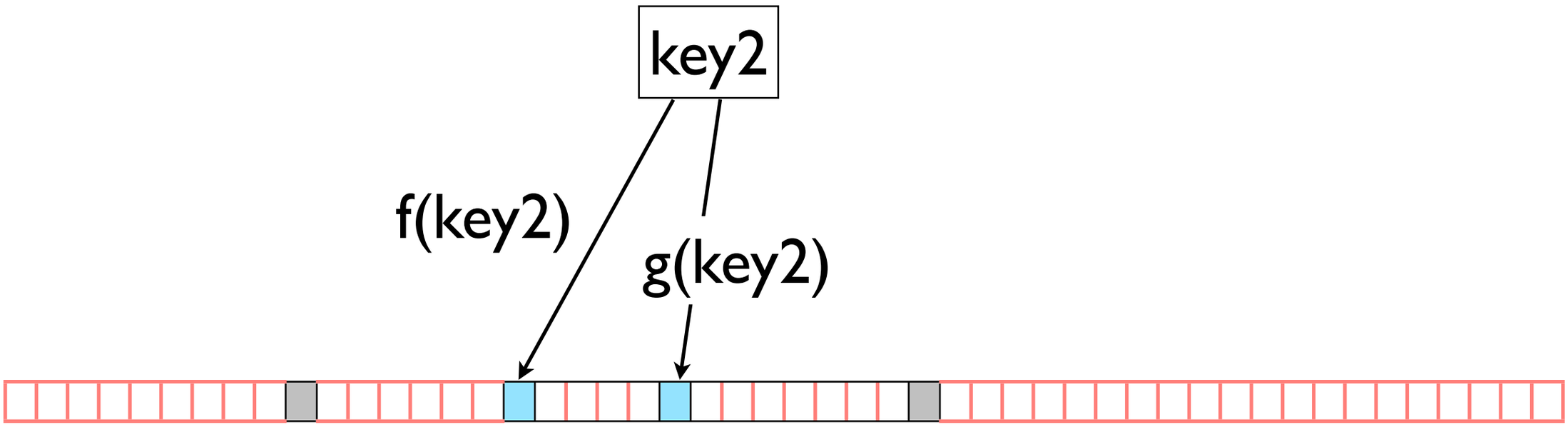


Bloom Filter (lecture)



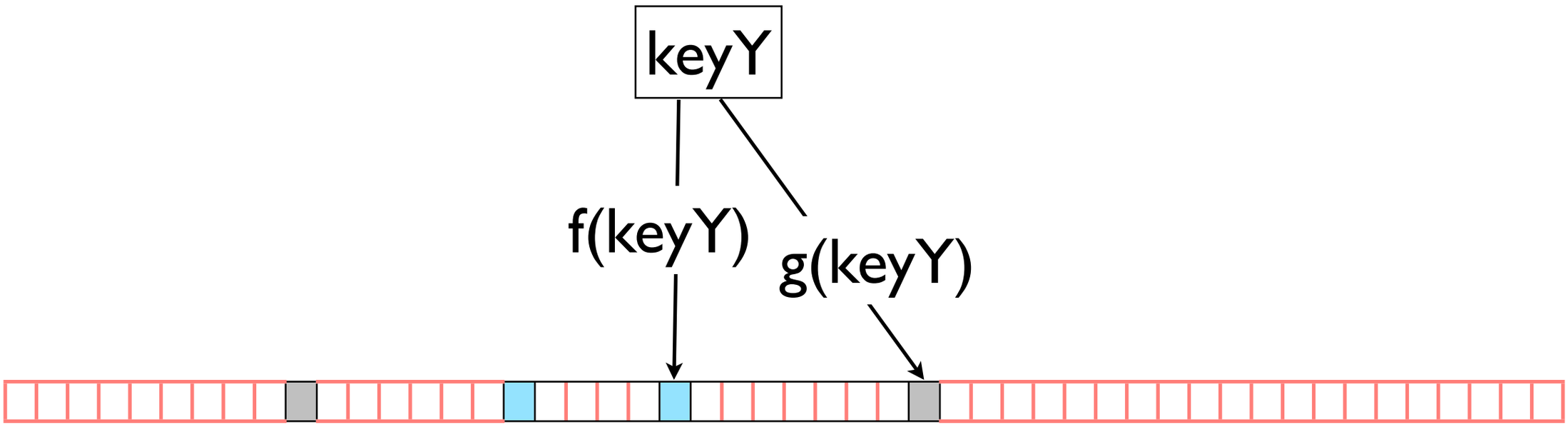
keyX présent? NON...

Bloom Filter (lecture)



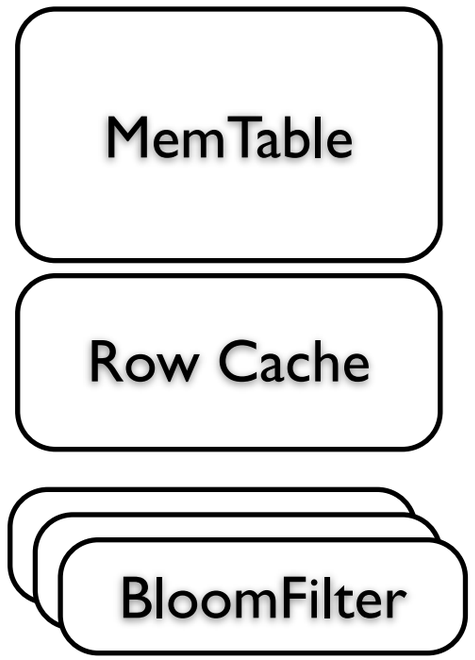
key2 présent? OUI... vrai positif

Bloom Filter (lecture)



keyY présent? OUI... mais c'est un faux positif

Lecture



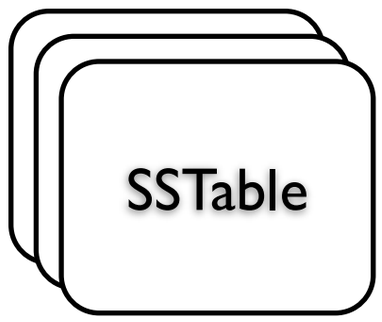
1- Pertinent lorsque le client spécifie le nom de la colonne

2- Stocke l'intégralité d'une row

3- répond à «La SSTable contient-elle un fragment de row ?»
Attention, faux positif possible.

Mémoire

Fichiers



6- Contient la valeur...

Lecture

MemTable

1- Pertinent lorsque le client spécifie le nom de la colonne

Row Cache

2- Stocke l'intégralité d'une row

BloomFilter

3- répond à «La SSTable contient-elle un fragment de row ?»
Attention, faux positif possible.

Key Cache

4- Donne la position exacte du fragment de row dans la SSTable

Index Summary

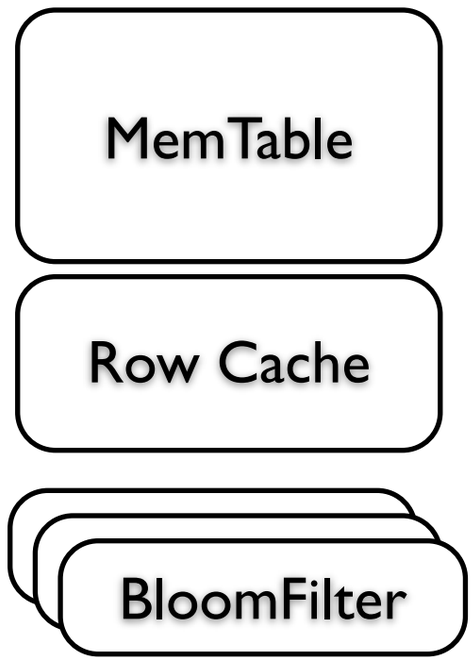
Mémoire

Fichiers

SSTable

6- Contient la valeur...

Lecture



1- Pertinent lorsque le client spécifie le nom de la colonne

2- Stocke l'intégralité d'une row

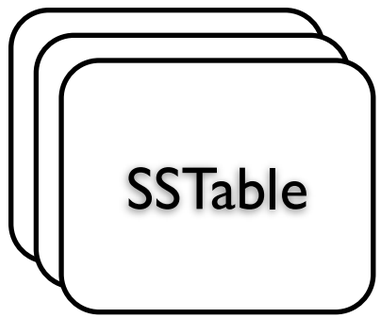
3- répond à «La SSTable contient-elle un fragment de row ?»
Attention, faux positif possible.

4- Donne la position exacte du fragment de row dans la SSTable

5- Donne la position approximative (intervalle) du fragment de row dans la SSTable

Mémoire

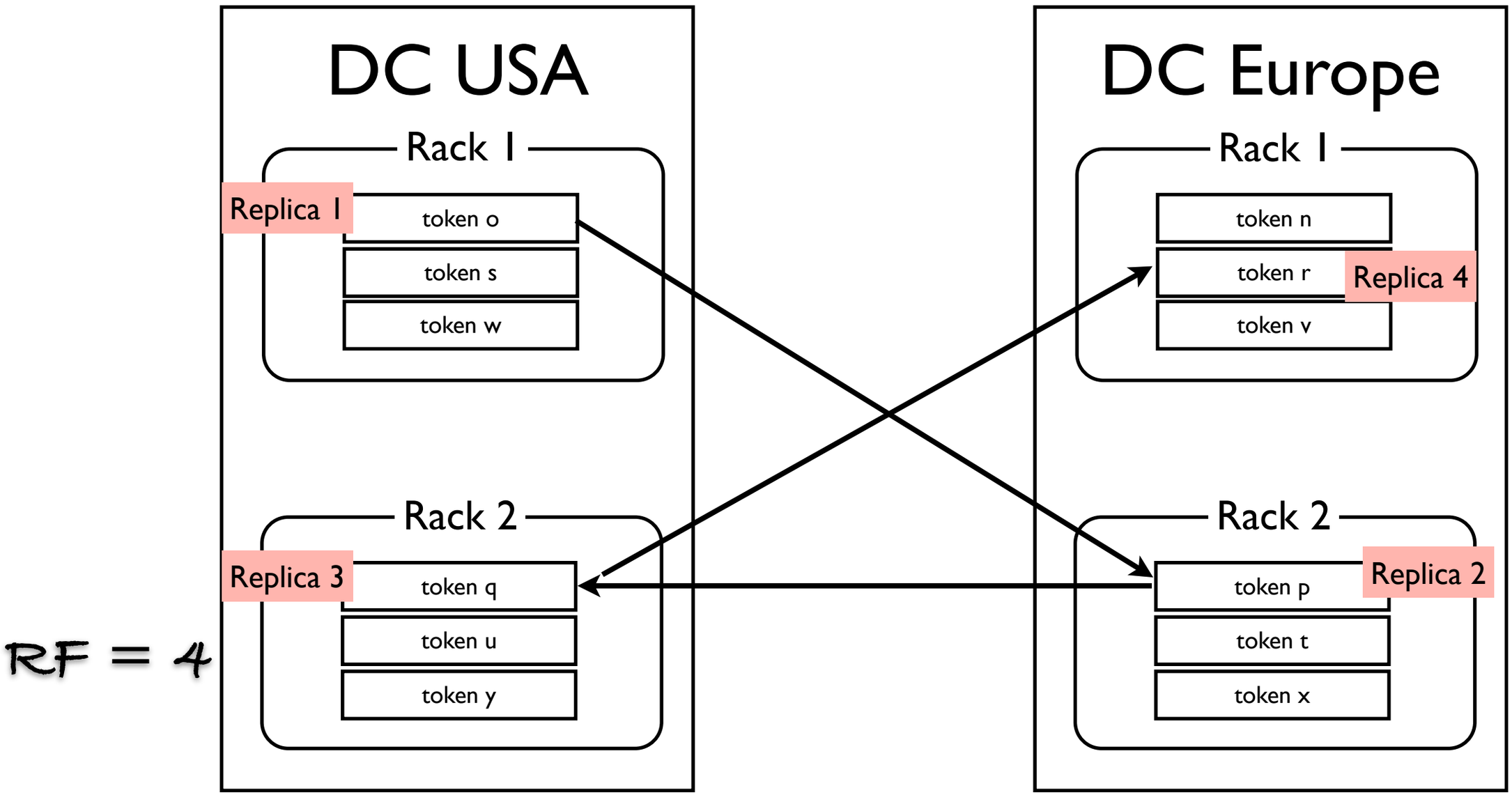
Fichiers



6- Contient la valeur...

Questions?

Multi Data Center

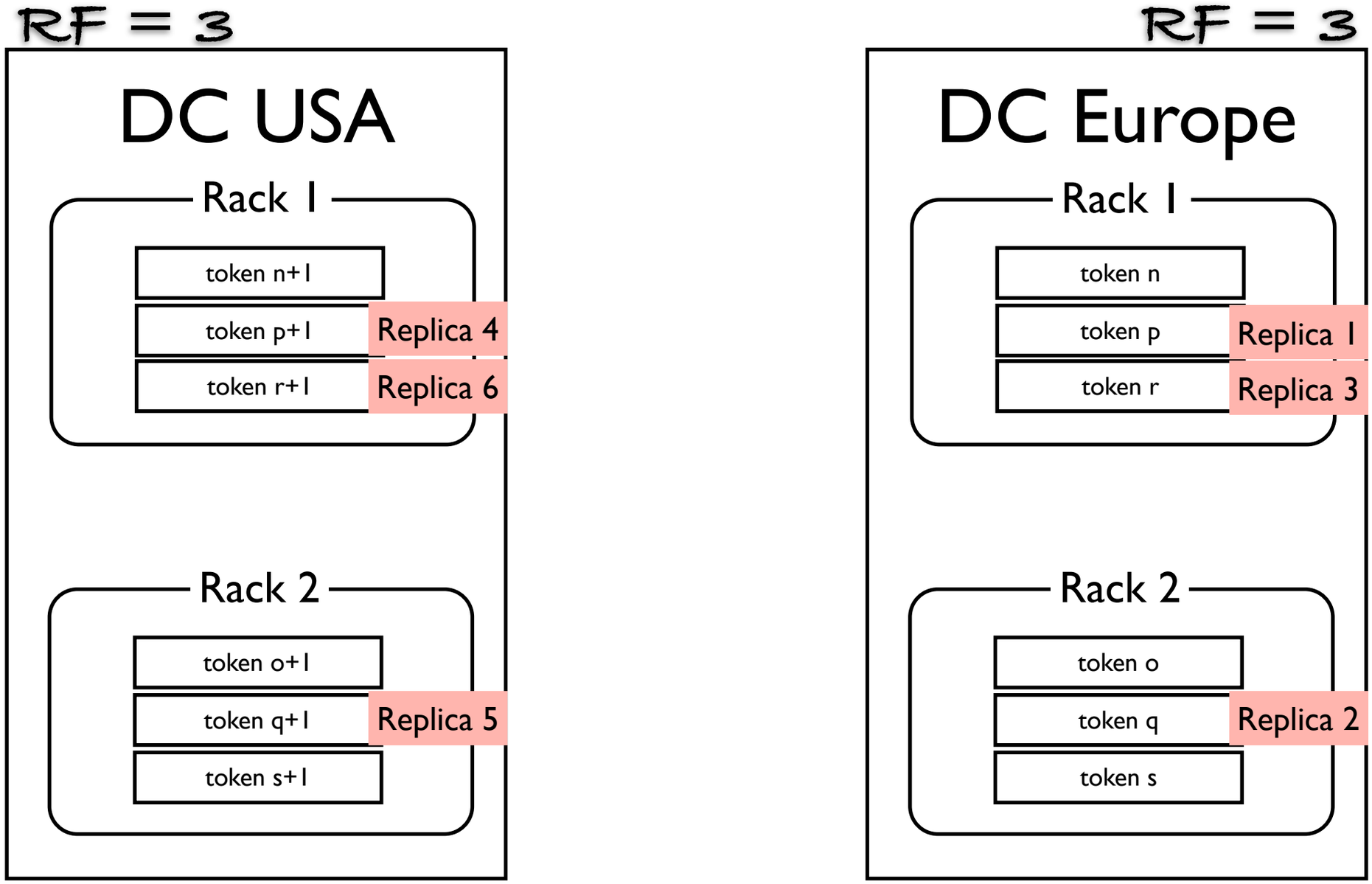


RF = 4

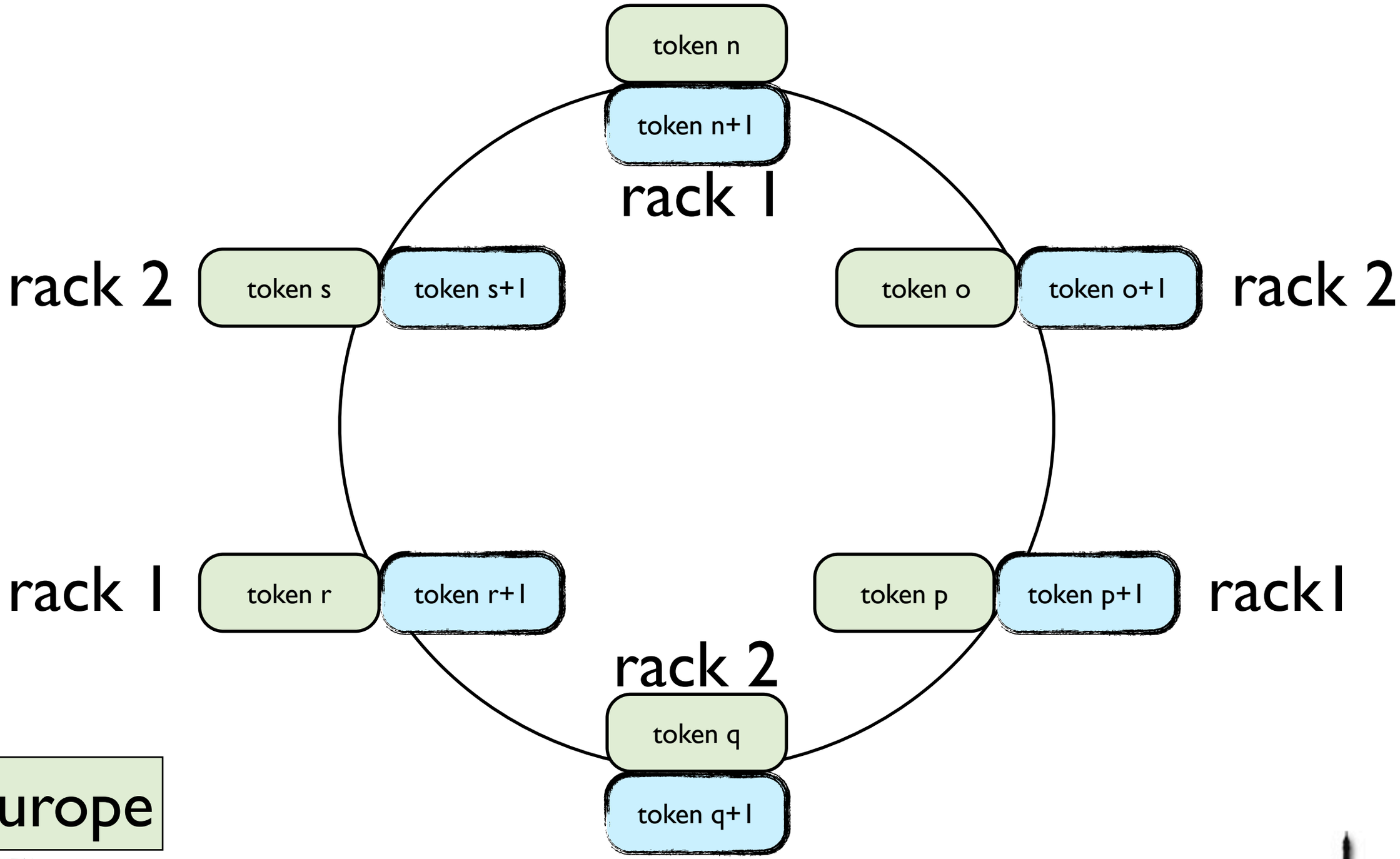
⚠ Pas de symétrie entre les DC
QUORUM implique nécessairement 2 DC => latence

DataCenter «Aware»

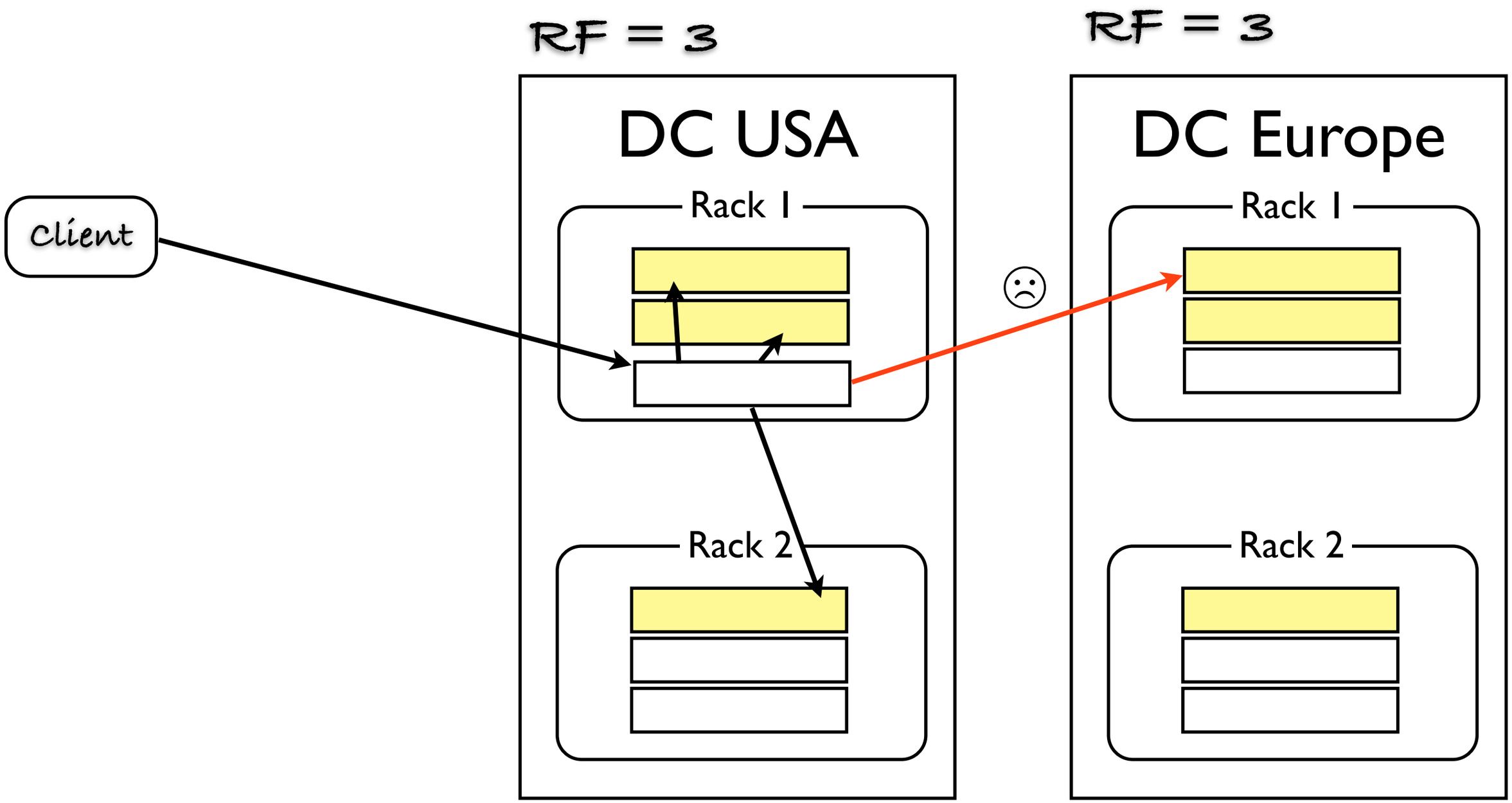
NetworkTopologyStrategy + alternance tokens



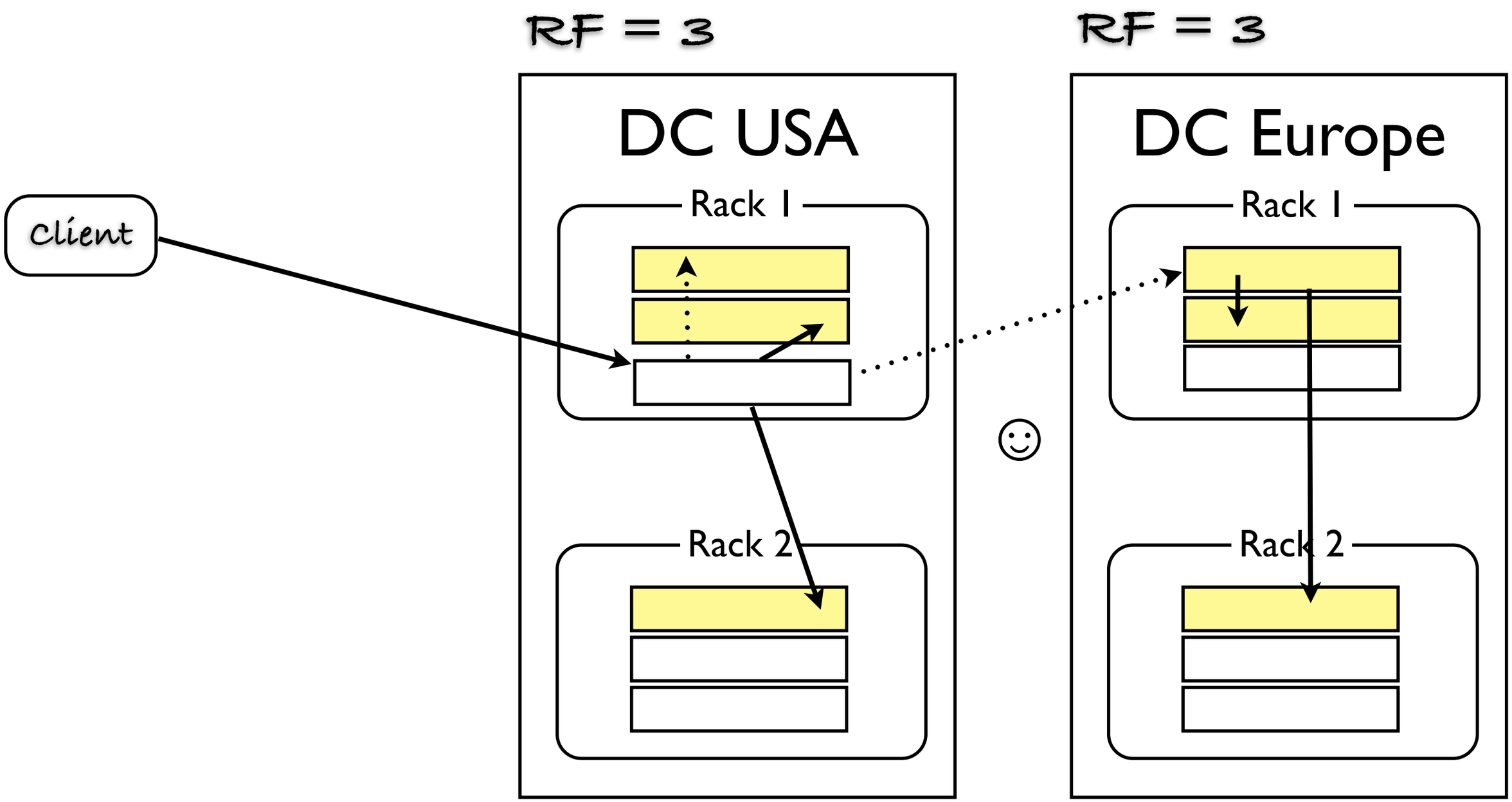
Ring 2 DataCenter (+1)



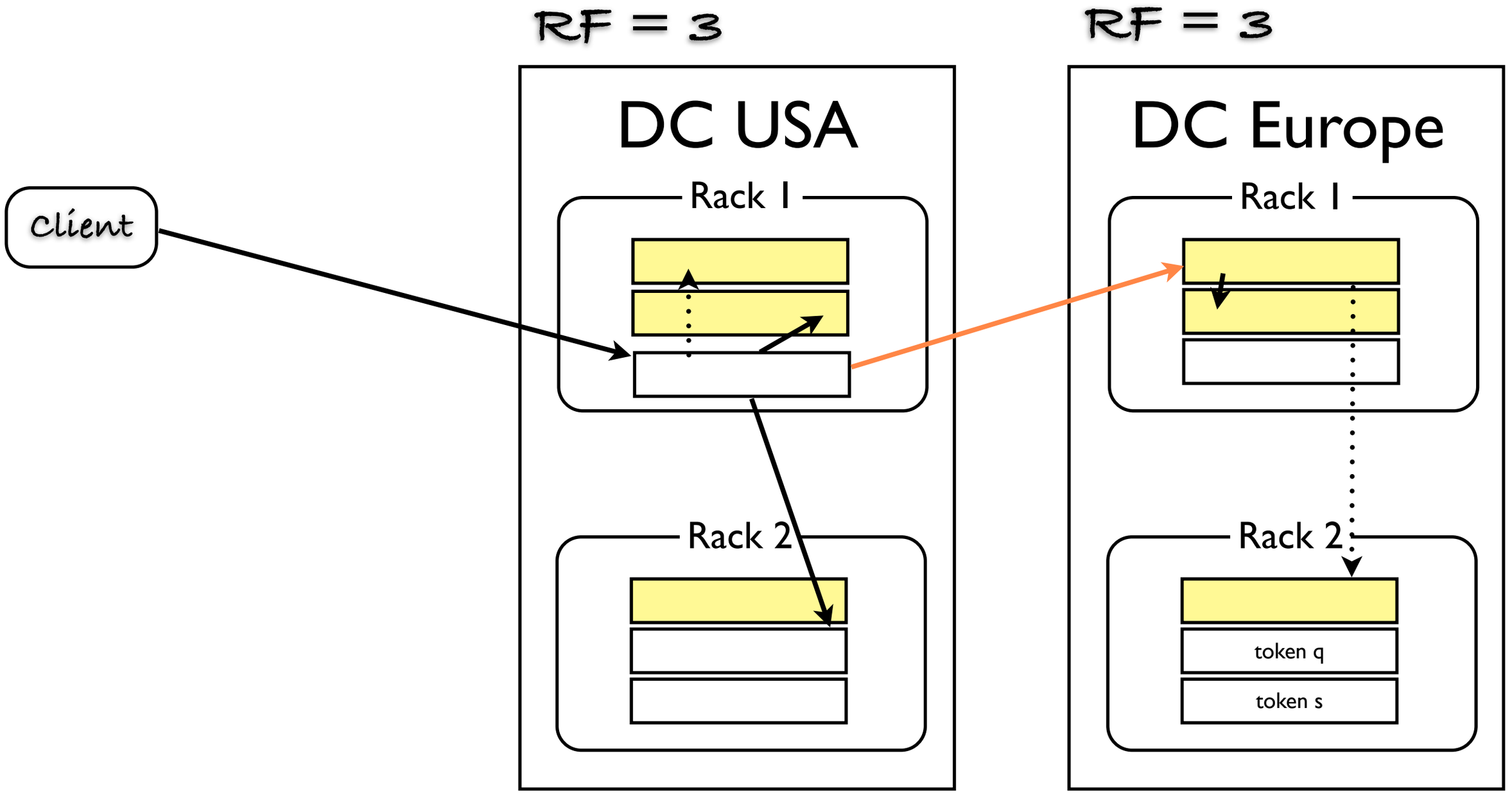
Latence du QUORUM



LOCAL_QUORUM



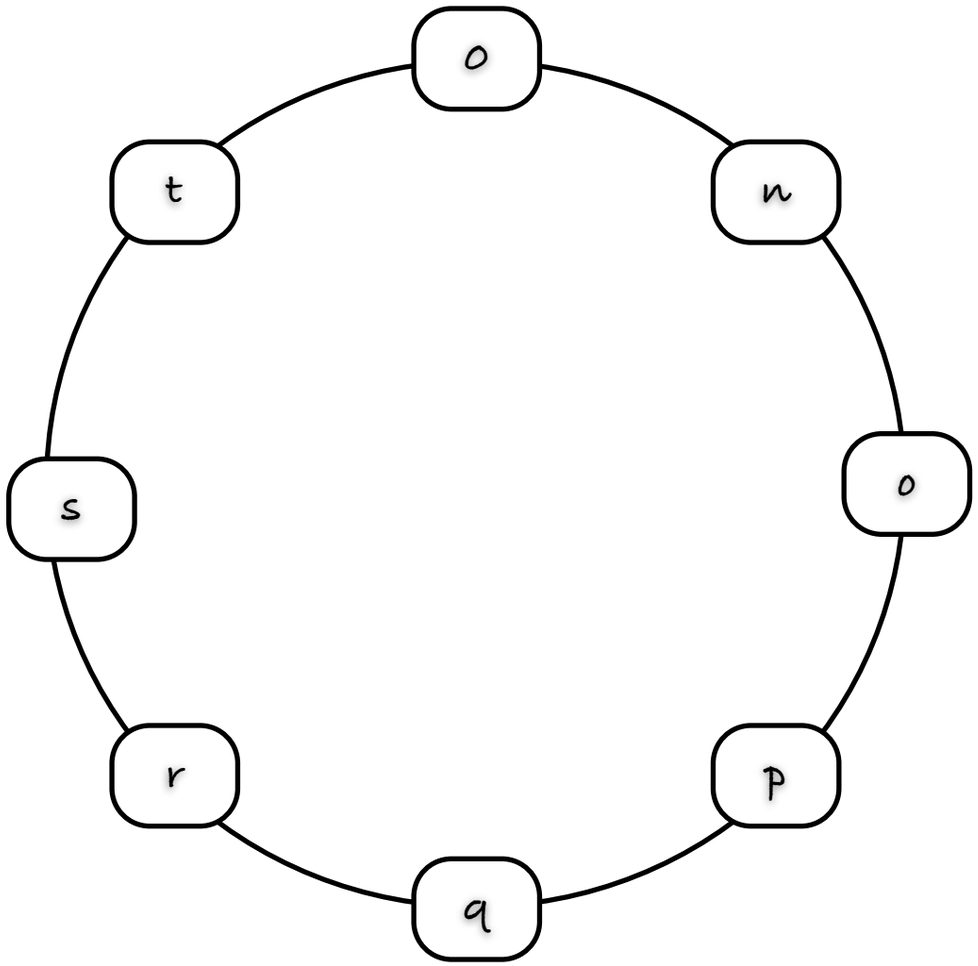
EACH_QUORUM


RF = 3
RF = 3
DC USA
DC Europe
Rack 1
Rack 1
Client
Rack 2
Rack 2
token q
token s

Autres usages Multi DC

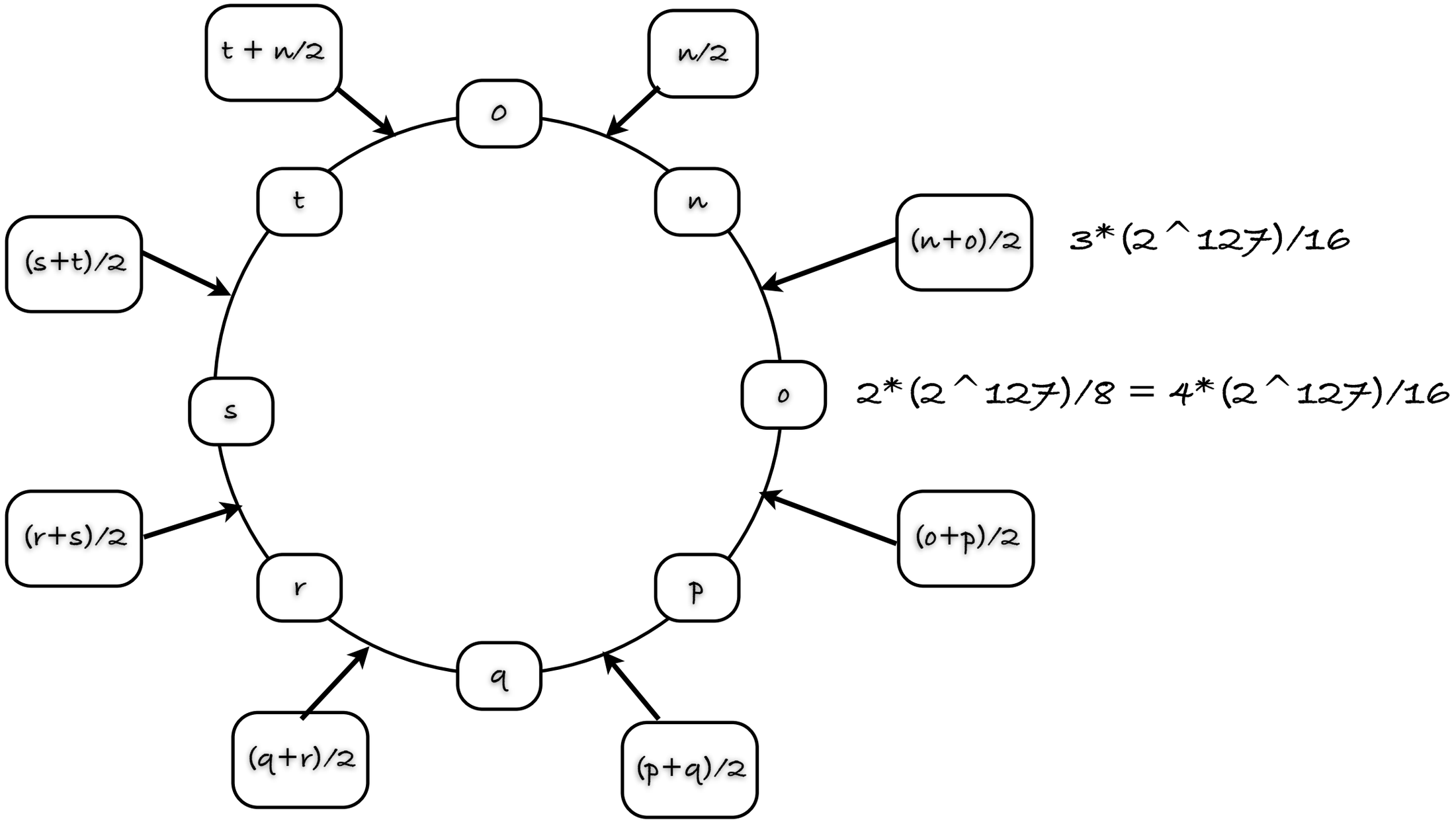
- 1 DC pour backup
- 1 DC pour traitement particulier

Doubler la taille du cluster

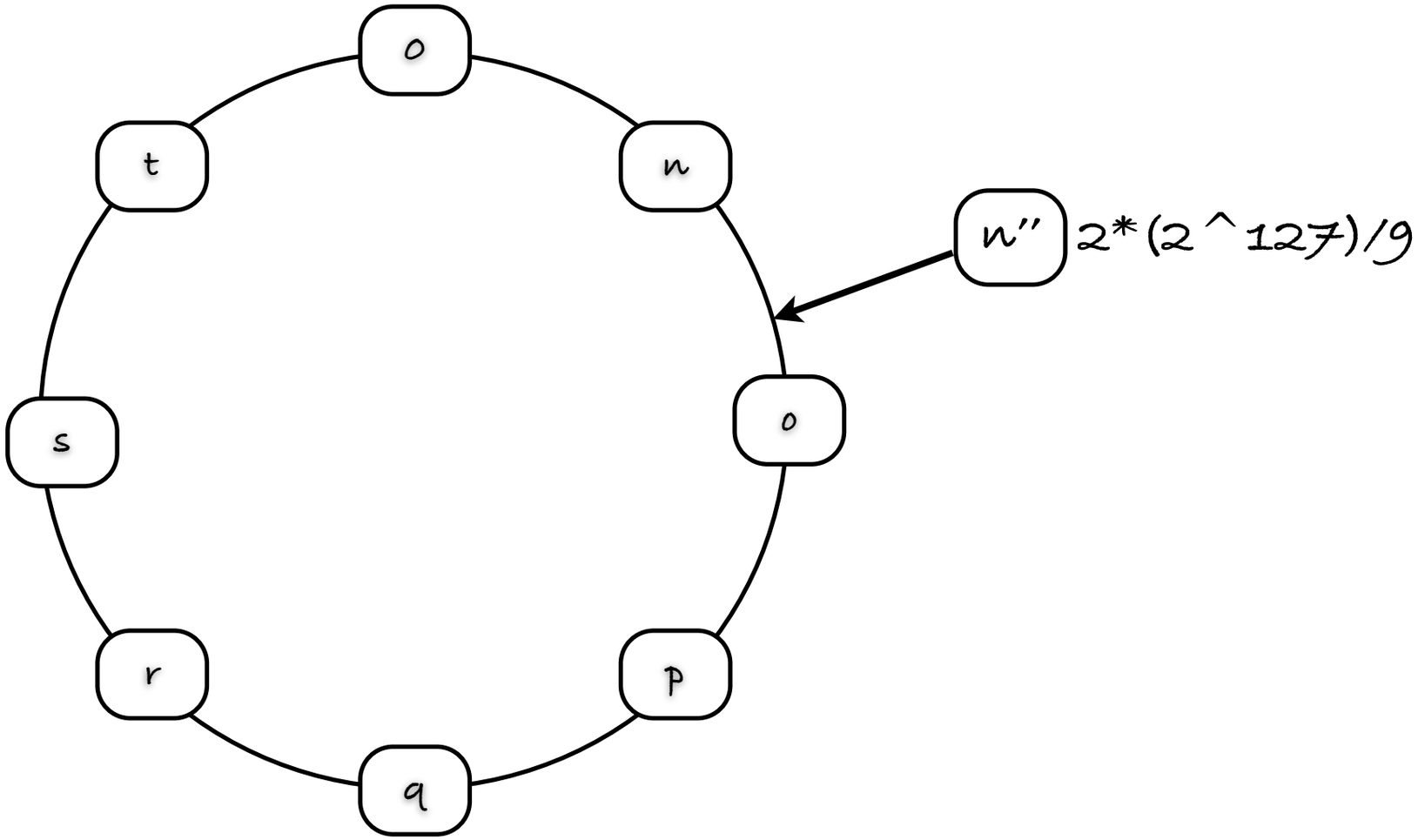


2*(2^127)/8

Doubler la taille du cluster

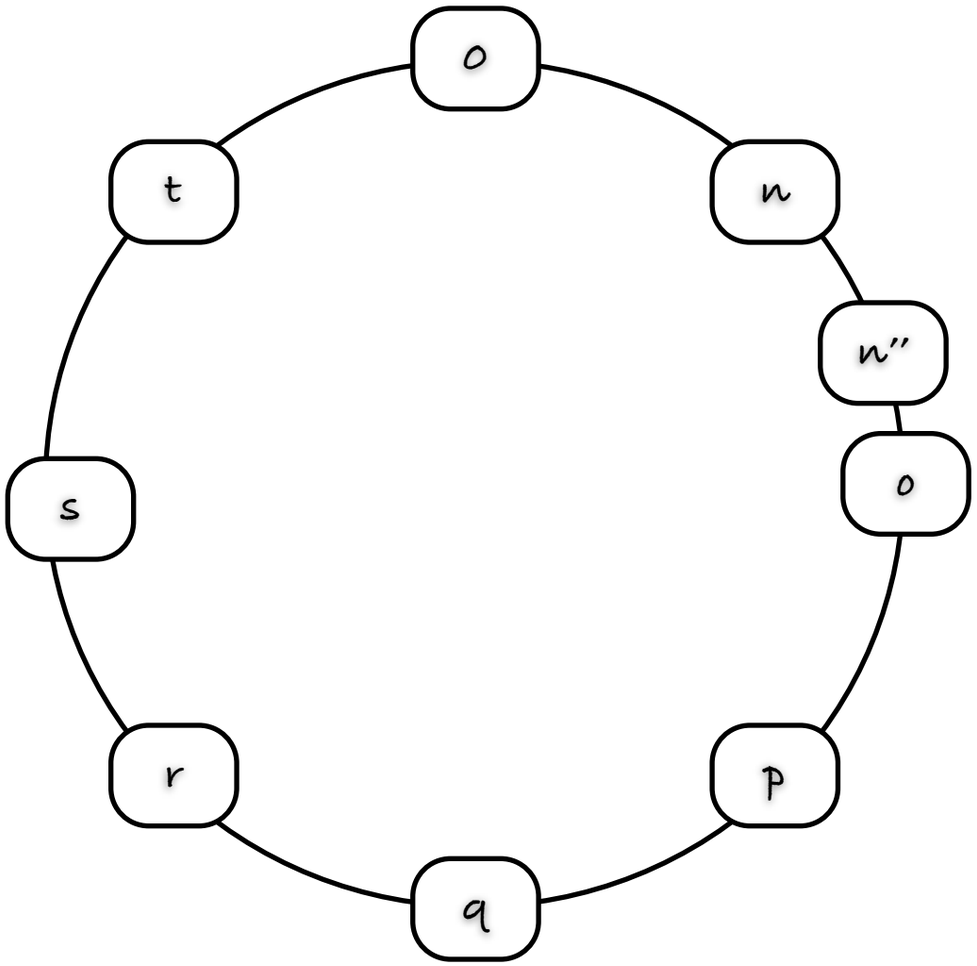


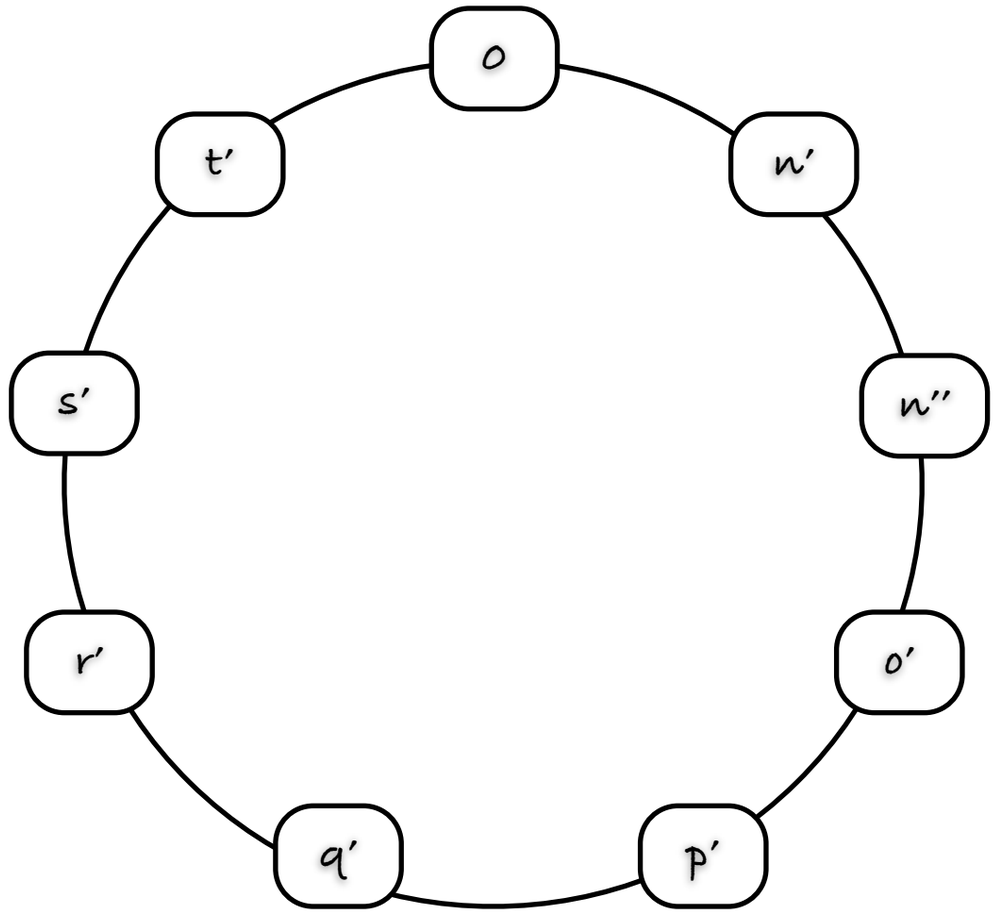
Ajouter 1 noeud



On crée un déséquilibre...

Ajouter 1 noeud



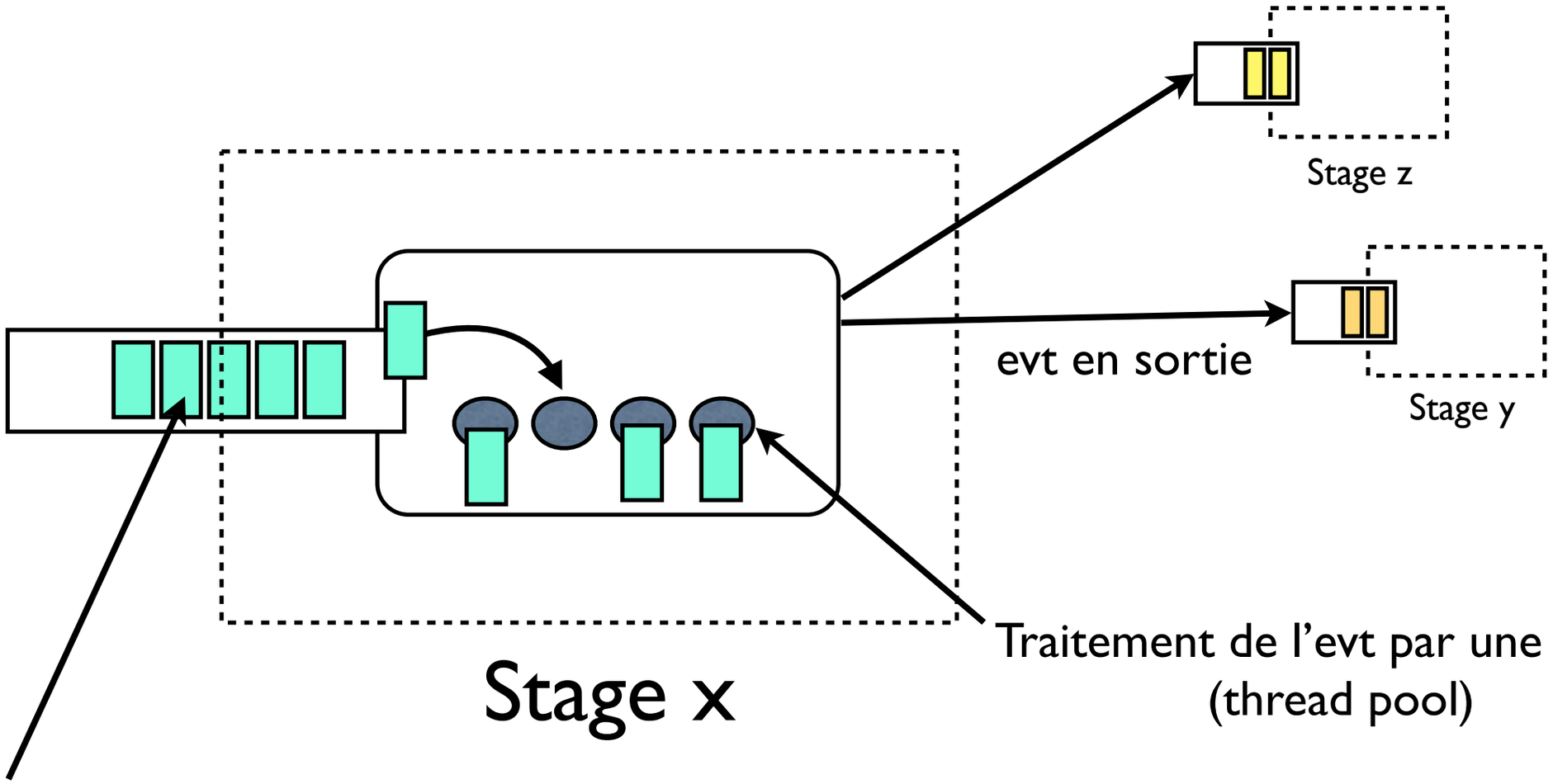


re-balancing

Ajout/Retrait/Upgrade

- Se font à chaud

Architecture Interne & Monitoring



Traitement de l'evt par une thread
(thread pool)

les evts font la queue
certains peuvent expirer avant d'être traité!

SEDA: Staged Event-Driven Architecture...

Architecture Interne & Monitoring

- Monitoring des «stages» cassandra:
  ('tp' pour thread pool)
  (1) => 1 thread seulement

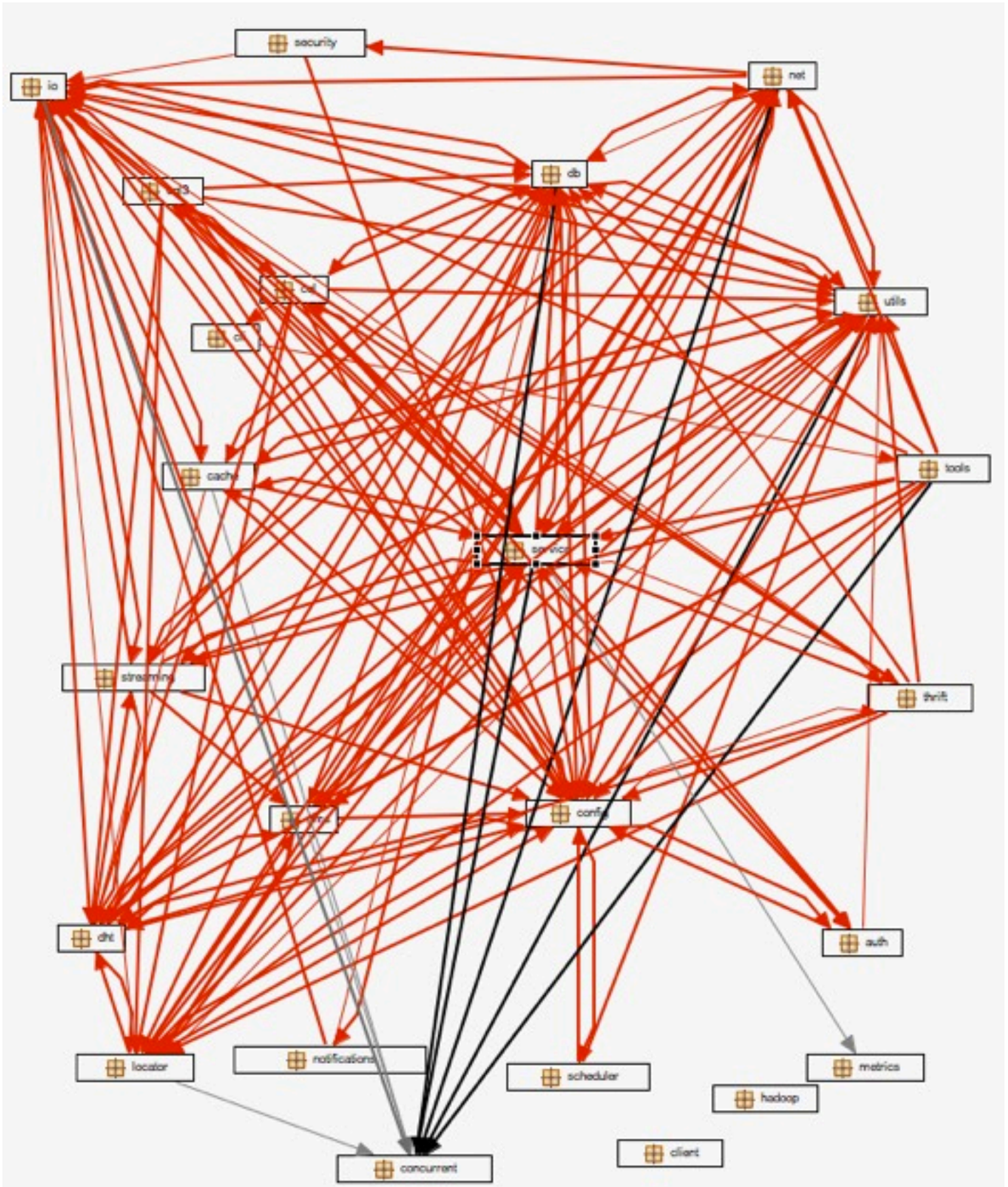
```
Pool Name                    Active   Pending      Completed   Blocked  All time blocked
ReadStage                         0         0              1         0                 0
RequestResponseStage              0         0              0         0                 0
MutationStage                     0         0             20         0                 0
ReadRepairStage (1)               0         0              0         0                 0
ReplicateOnWriteStage             0         0              0         0                 0
GossipStage (1)                   0         0          86919         0                 0
AntiEntropyStage (1)              0         0              0         0                 0
MigrationStage (1)                0         0              0         0                 0
MemtablePostFlusher               0         0              4         0                 0
StreamStage                       0         0              0         0                 0
FlushWriter                       0         0              4         0                 0
MiscStage  (1)                    0         0              0         0                 0
InternalResponseStage             0         0              0         0                 0
HintedHandoff                     0         0             13         0                 0

Message type           Dropped
RANGE_SLICE                  0
READ_REPAIR                  0
BINARY                       0
READ                         0
MUTATION                     0
REQUEST_RESPONSE             0
```

Messages expirés sont «droppés»

Une critique... si si

- Difficile de rentrer dans le code source

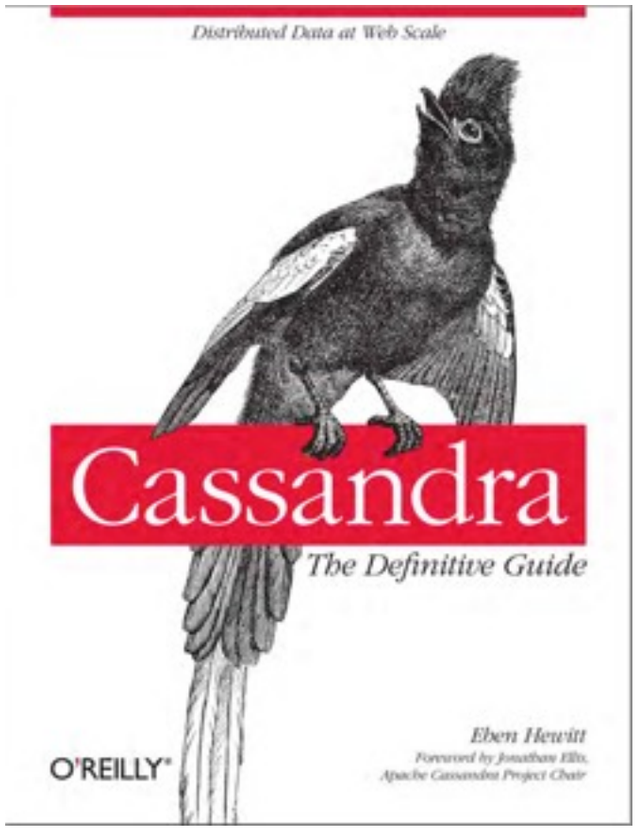
Une critique... si si

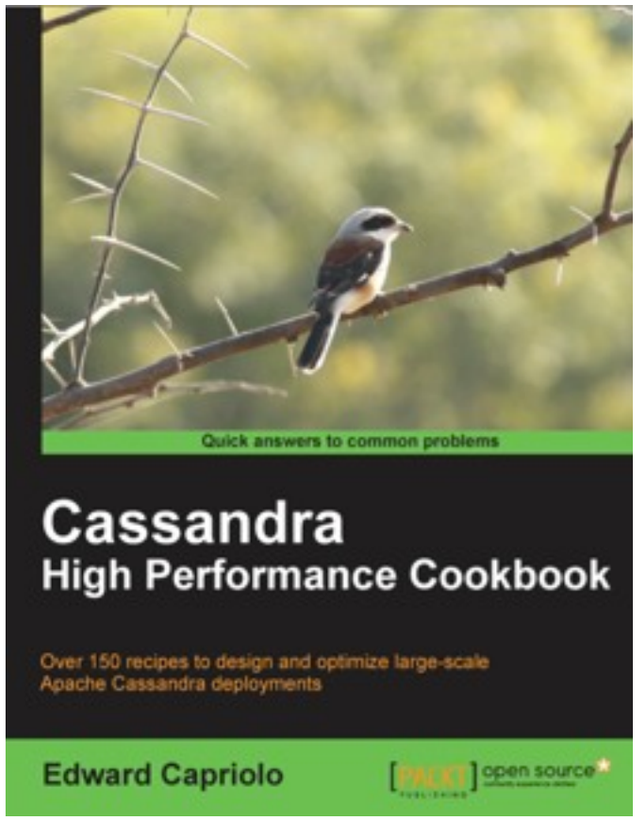


- Difficile de rentrer dans le code source

Questions?

Pour approfondir





http://www.datastax.com/docs/1.0/index

http://wiki.apache.org/cassandra/

http://ria101.wordpress.com/2010/02/24/hbase-vs-cassandra-why-we-moved/