

# Bases de datos distribuidas III

**Alex Di Genova**

**23/11/2023**

# Resumen

- **Bases de datos distribuidas**
  - Estrategias de Fragmentación
  - Estrategias de Asignación
  - Modelo de replicas
  - Sincronización distribuida (algoritmo RAFT)
  - Motor RQLITE

# RQlite

- rqlite es una base de datos relacional distribuida, ligera y fácil de usar, que utiliza SQLite como motor de almacenamiento.
- rqlite es simple de implementar, operar y sus capacidades de distribución brindan tolerancia a fallas y alta disponibilidad.
- <https://github.com/rqlite/rqlite>
- La versión actual es v7.11.0

```
$ curl -XPOST 'localhost:4001/db/execute?pretty&timings' -H "Content-Type: application/json" -d '[
  ["INSERT INTO foo(name, age) VALUES(?, ?)", "fiona", 20]
]'
{
  "results": [
    {
      "last_insert_id": 1,
      "rows_affected": 1,
      "time": 0.00886
    }
  ],
  "time": 0.0152
}
```

```
$ curl -G 'localhost:4001/db/query?pretty&timings' --data-urlencode 'q=SELECT * FROM foo'
```

# SQLite

- Base de datos relacional sólida, dentro de un solo archivo fuente C
- Con la replicación obtienes confiabilidad.
- Instalación e implementación súper fáciles gracias a Go
- Operación ligera.
- Un sistema distribuido proporciona confiabilidad
  - Los datos se encuentran en varios lugares
  - El cálculo está disponible desde varios lugares.
- Un sistema distribuido proporciona escalabilidad.
  - Los sistemas distribuidos pueden ser más potentes.



eSoftner

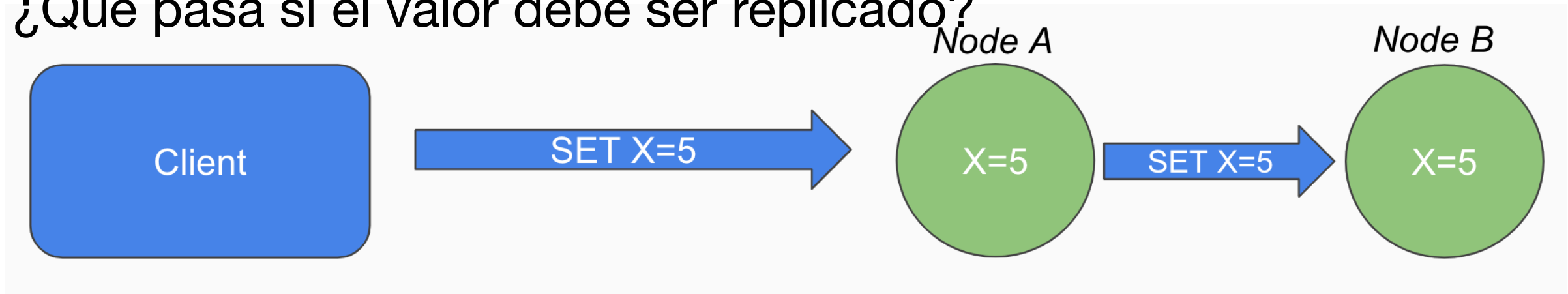
# BDD

## Replicación

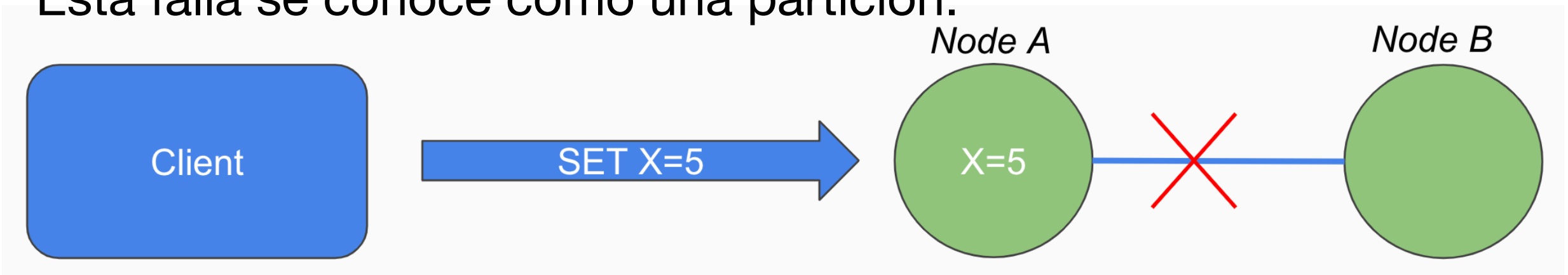
Es fácil establecer el valor de un solo nodo



¿Qué pasa si el valor debe ser replicado?



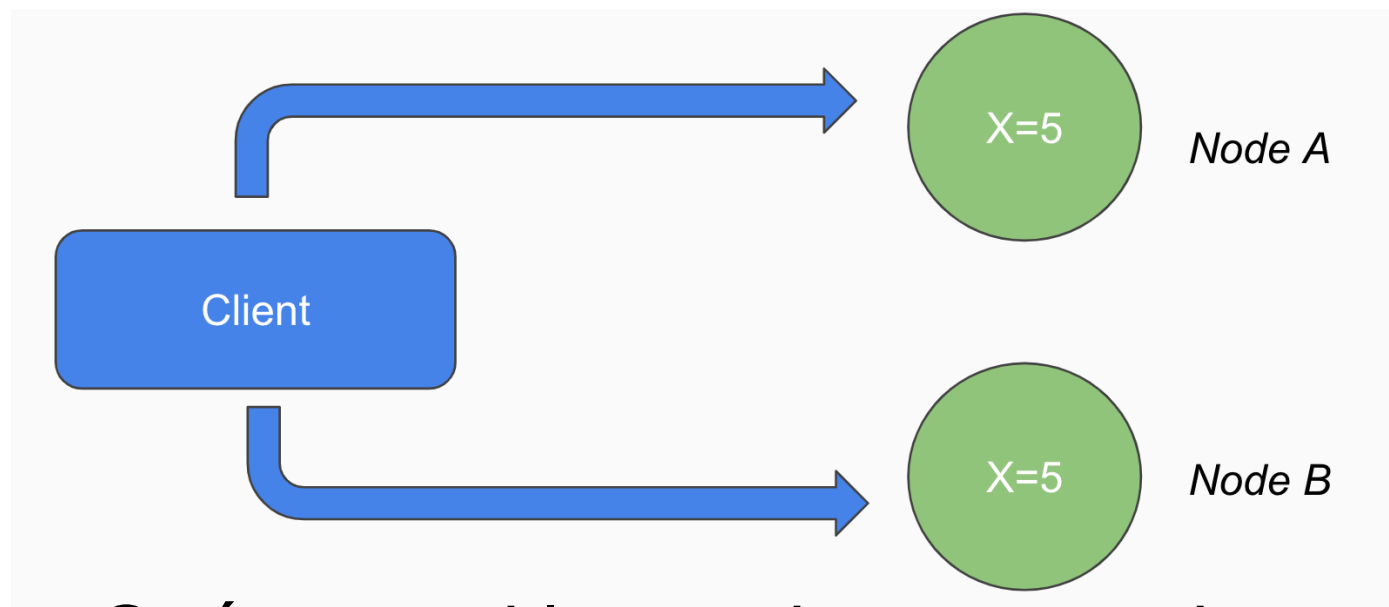
Esta falla se conoce como una partición.



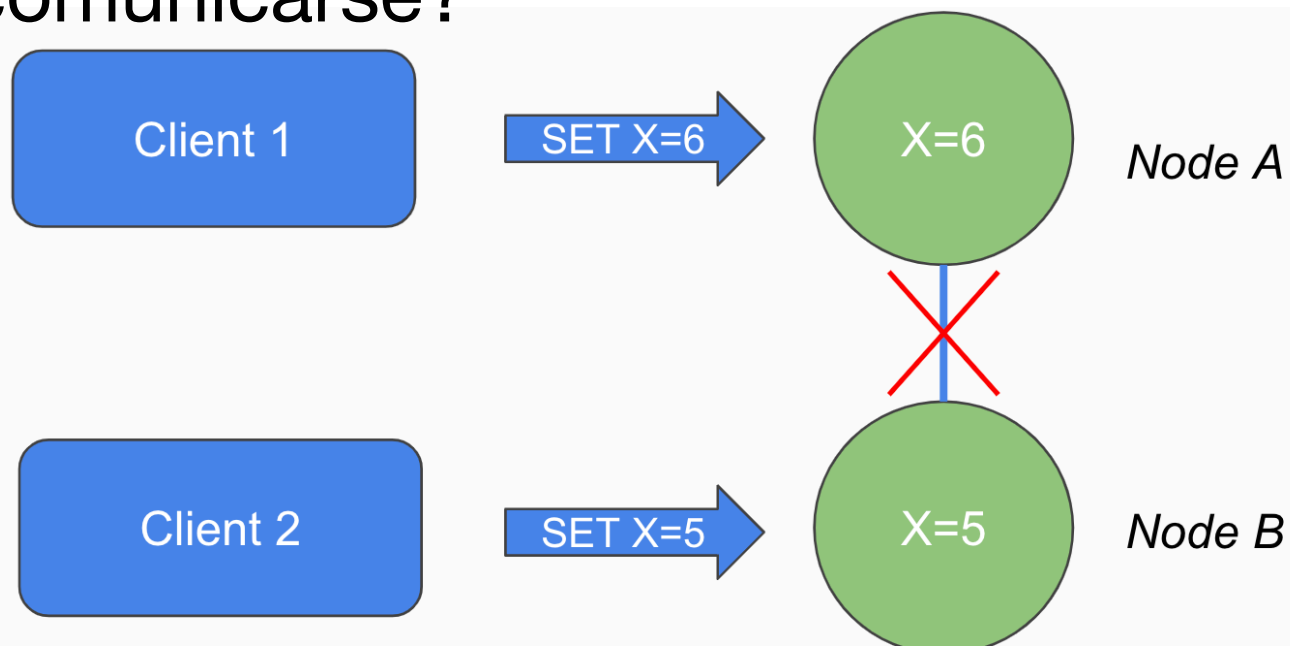
# BDD

## Replicación

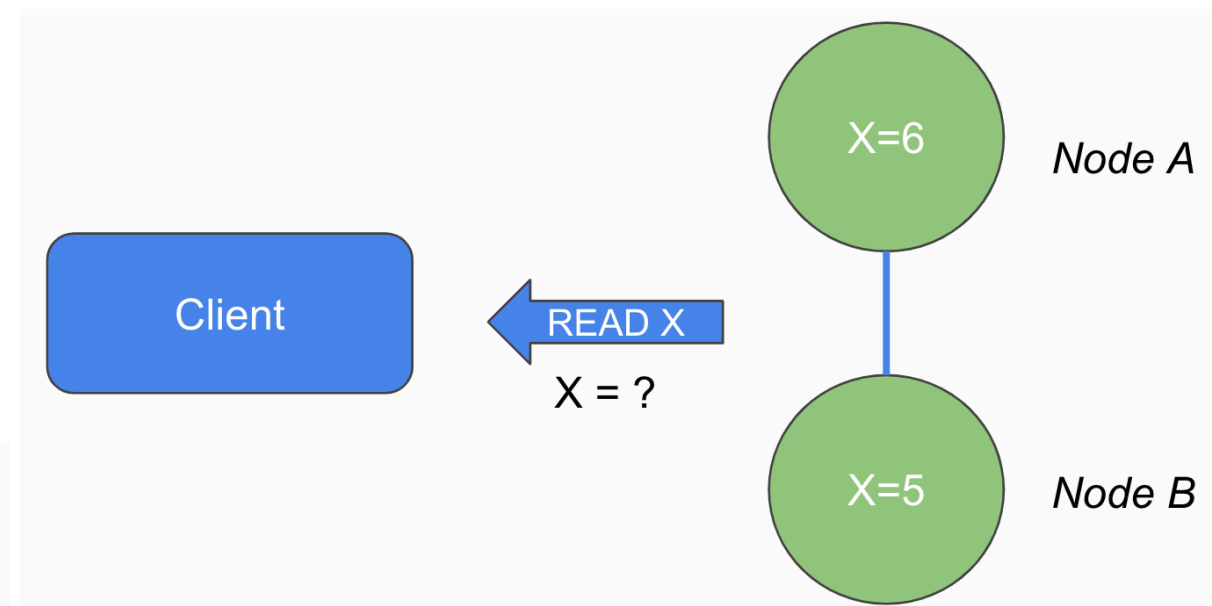
- Replicación de clientes: cada nodo debe admitir cambios de estado



¿Qué pasa si los nodos no pueden comunicarse?



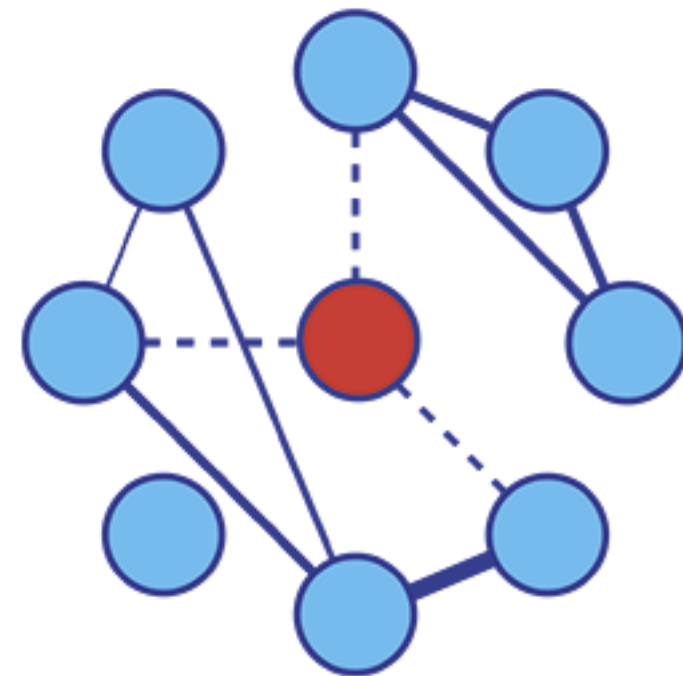
¿Qué valor se debe leer el cliente?



**Este problema se conoce como Consenso Distribuido.**

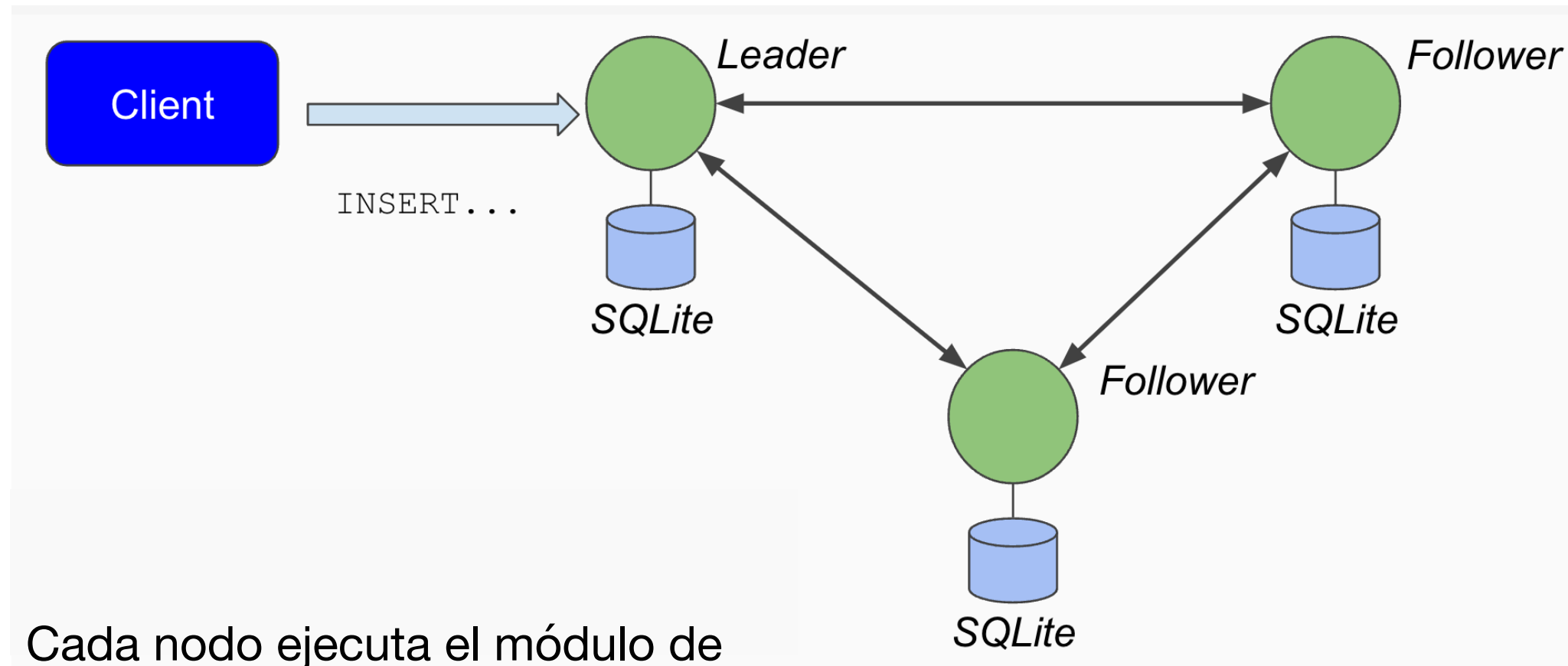
# Que es Raft?

- Raft es un protocolo de consenso distribuido.
- Dichos protocolos se utilizan para garantizar que varios nodos diferentes (servidores) siempre estén de acuerdo en un conjunto de valores determinado.
- Nos permite construir un clúster de servidores, de modo que para un quórum de servidores dentro del clúster, cada uno de esos servidores tiene el mismo estado.
- Dentro de rqlite ese estado es una base de datos SQLite.
- <http://thesecretlivesofdata.com/raft/>

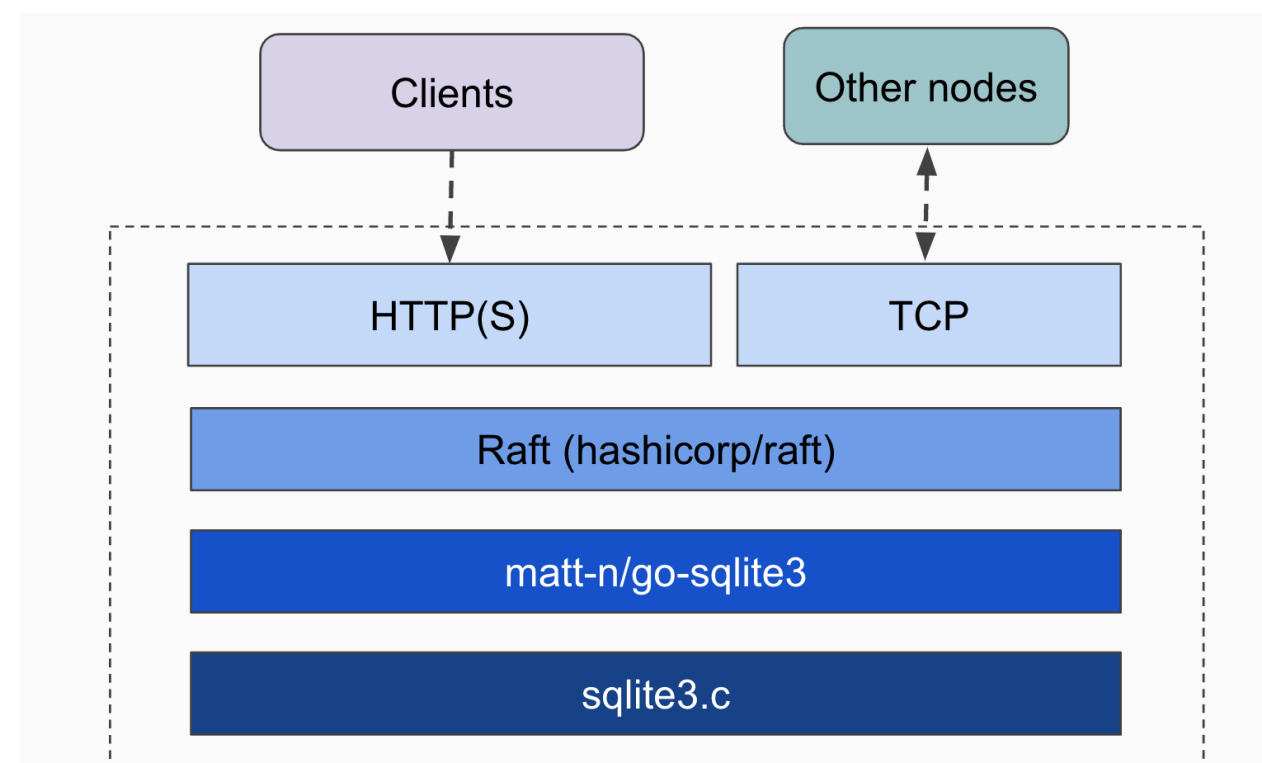


# Rqlite

## Arquitectura



- Cada nodo ejecuta el módulo de consenso Raft
- Una vez confirmados en los registros de Raft, cada nodo aplica los cambios a su base de datos **SQLite local**.
- Arquitectura a nivel de nodo.





# Rqlite

## Integración con raft

- La integración con el módulo de consenso de Raft implica implementar cinco funciones clave.

```
Apply(l *raft.Log) interface{} // Aplicar una entrada confirmada a la máquina de estado
```

```
Snapshot() (raft.FSMSnapshot, error) // Devuelve una snapshot de la máquina de estado.
```

```
Restore(rc io.ReadCloser) error // Crear máquina de estado a partir de un snapshot
```

```
Persist(sink raft.SnapshotSink) // Escribir el snapshot en almacenamiento persistente
```

```
Release() //libera el snapshot
```

<https://github.com/otoolep/hraftd>

# Que puede hacer rqlite

- Rqlite es un sistema distribuido liviano y confiable para datos relacionales.
- Podemos usar Rqlite como parte de un sistema más grande, como un almacén central para algunos datos relacionales críticos, sin tener que ejecutar una solución más pesada como MySQL.
- Rqlite también podría ser una forma efectiva de proporcionar una pequeña cantidad de réplicas de lectura de SQLite.

# Consultas?

Consultas o comentarios?

Muchas gracias