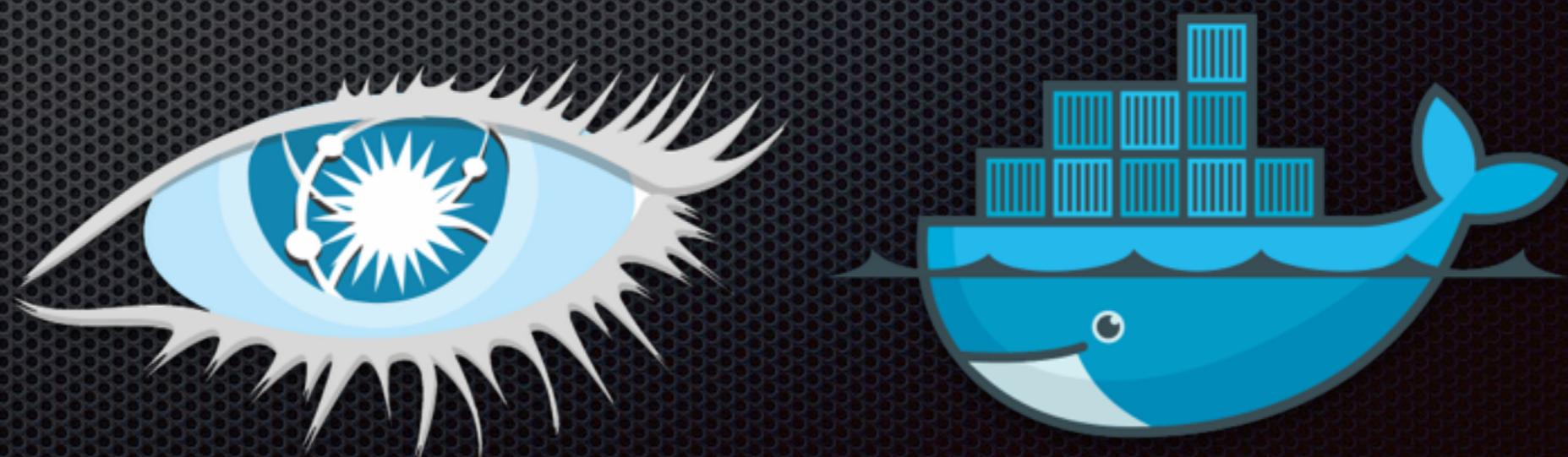


Apache Cassandra via Docker

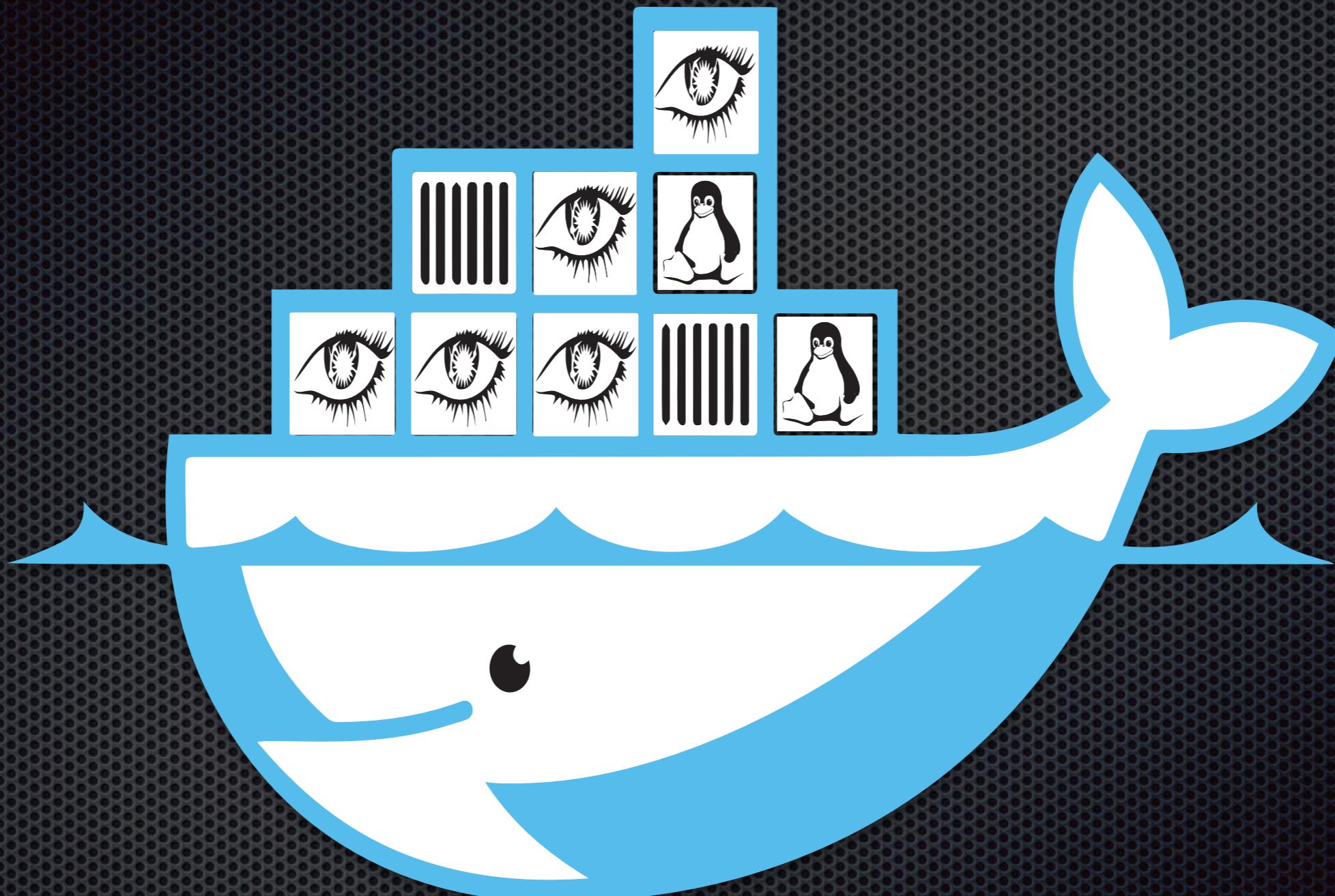
Chris Ballance
Diligent Corporation
@ballance



What we'll cover

- Docker Fundamentals
- CAP Theorem
- Cassandra Fundamentals
- Demo of Cassandra deployed with Docker

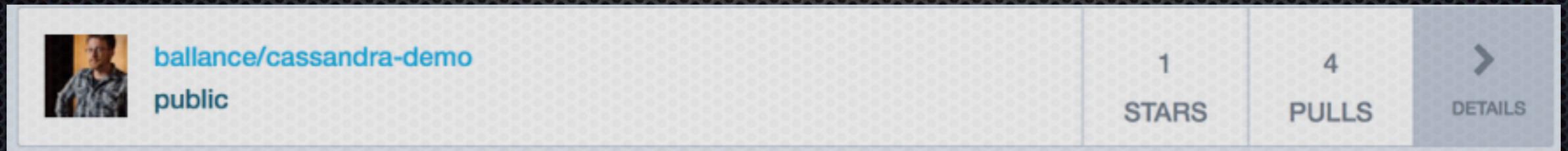
Docker



Hosting & Coordination

Docker Hub

“GIT for Virtual Machines”



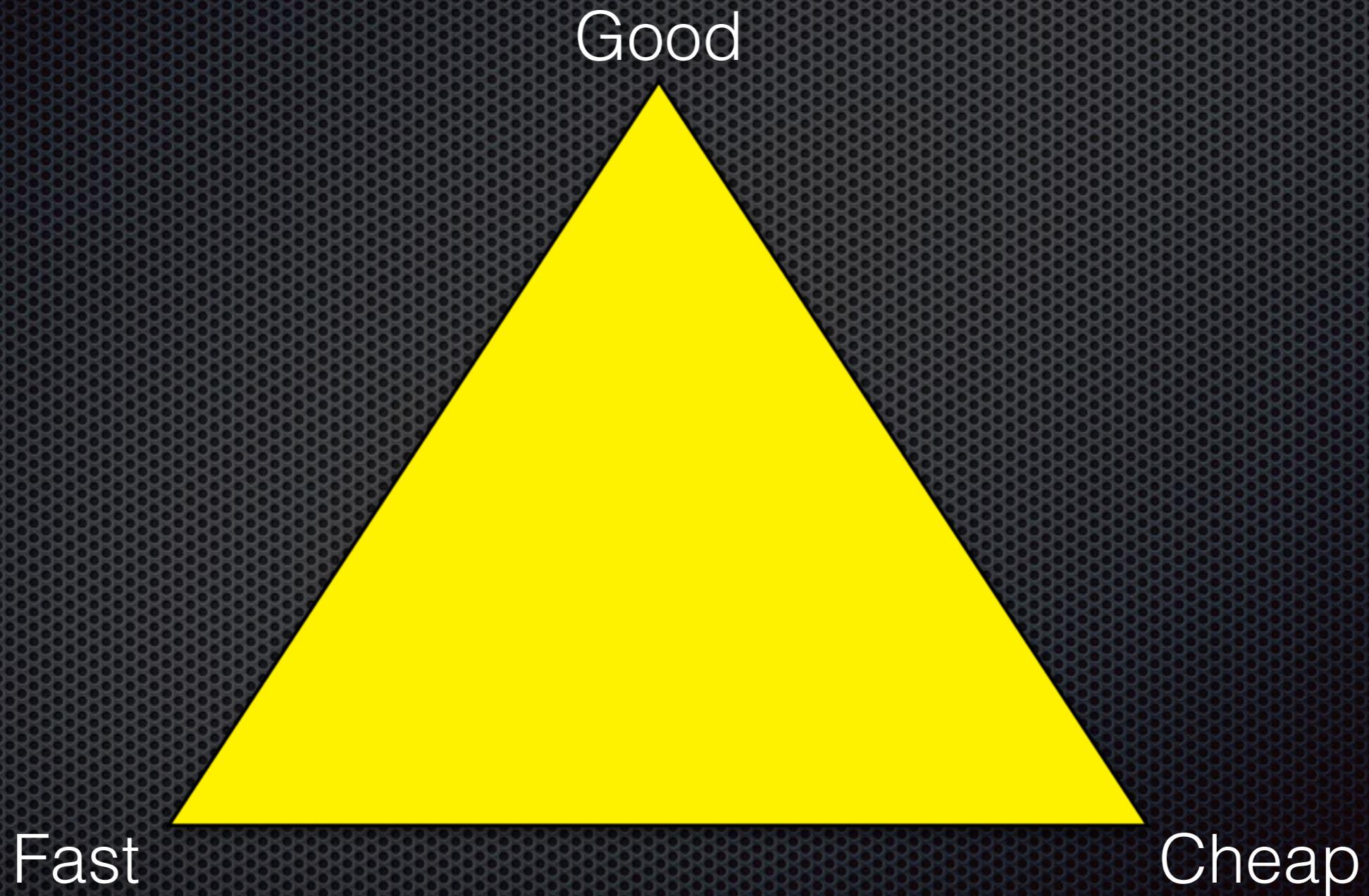
<https://hub.docker.com/>

Available Docker Images

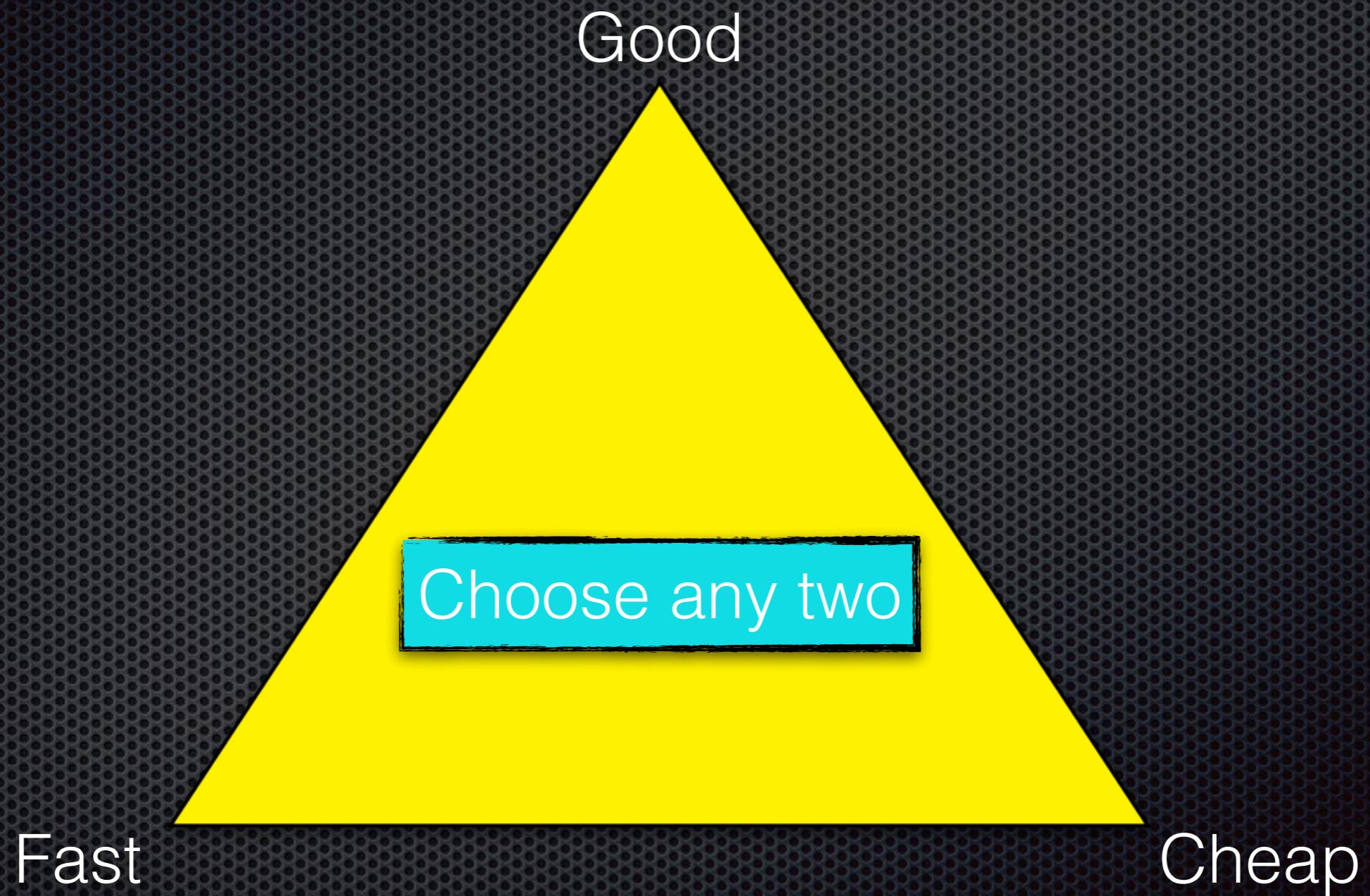
- Ubuntu, Centos, Debian, Fedora
- Cassandra, mySQL, mongoDB
- Node, Java, Erlang, Ruby, Rails
- Wordpress
- Redis
- Hipache, NGINX
- Create your own!

CAP THEOREM

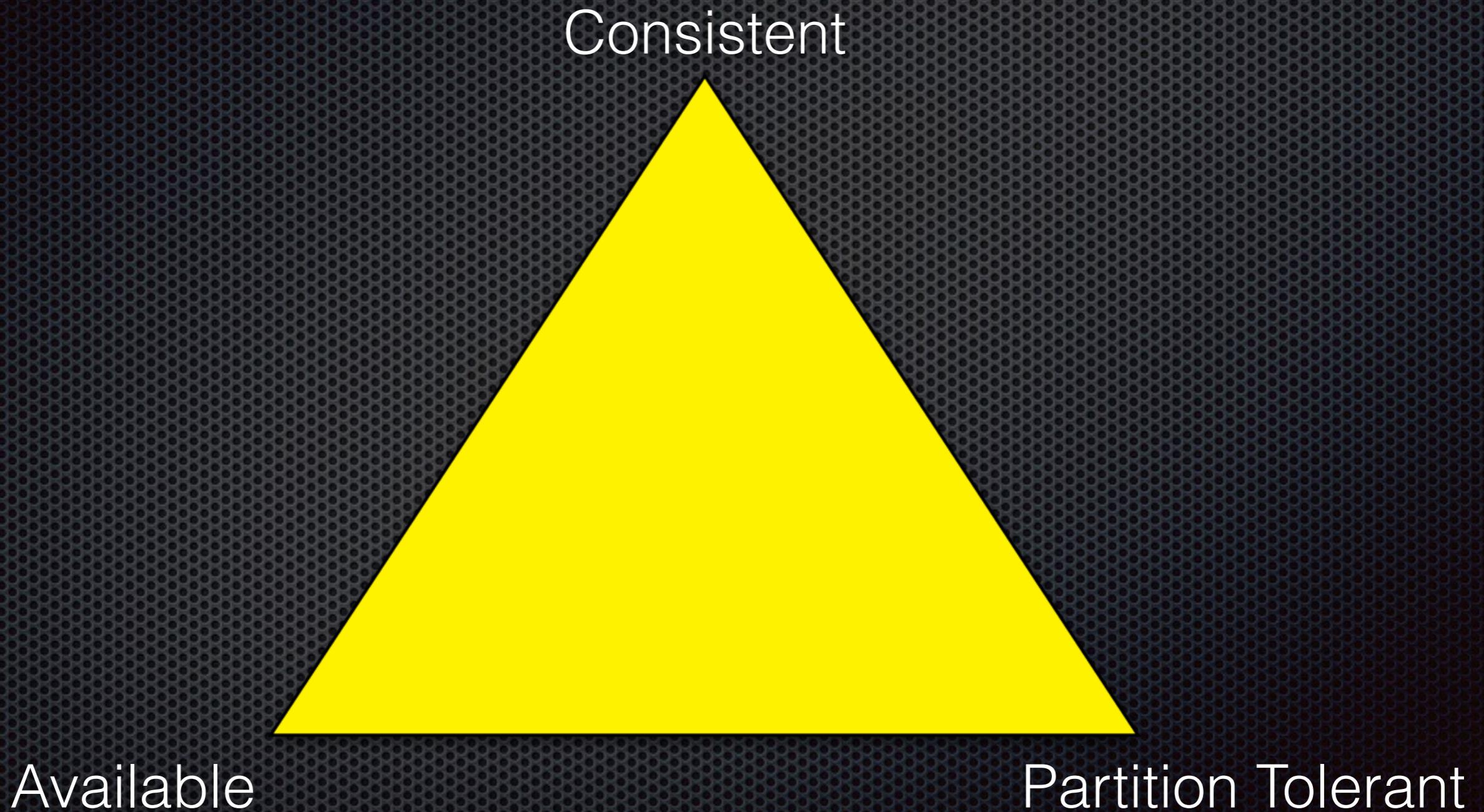
Project Delivery Triangle



Project Delivery Triangle



CAP Theorem



CAP Theorem

Consistent

Choose any two

Available

Partition Tolerant

Consistent



Available

Partition Tolerant

Available

Partition Tolerant

Consistent

Available

Partition Tolerant

Consistent

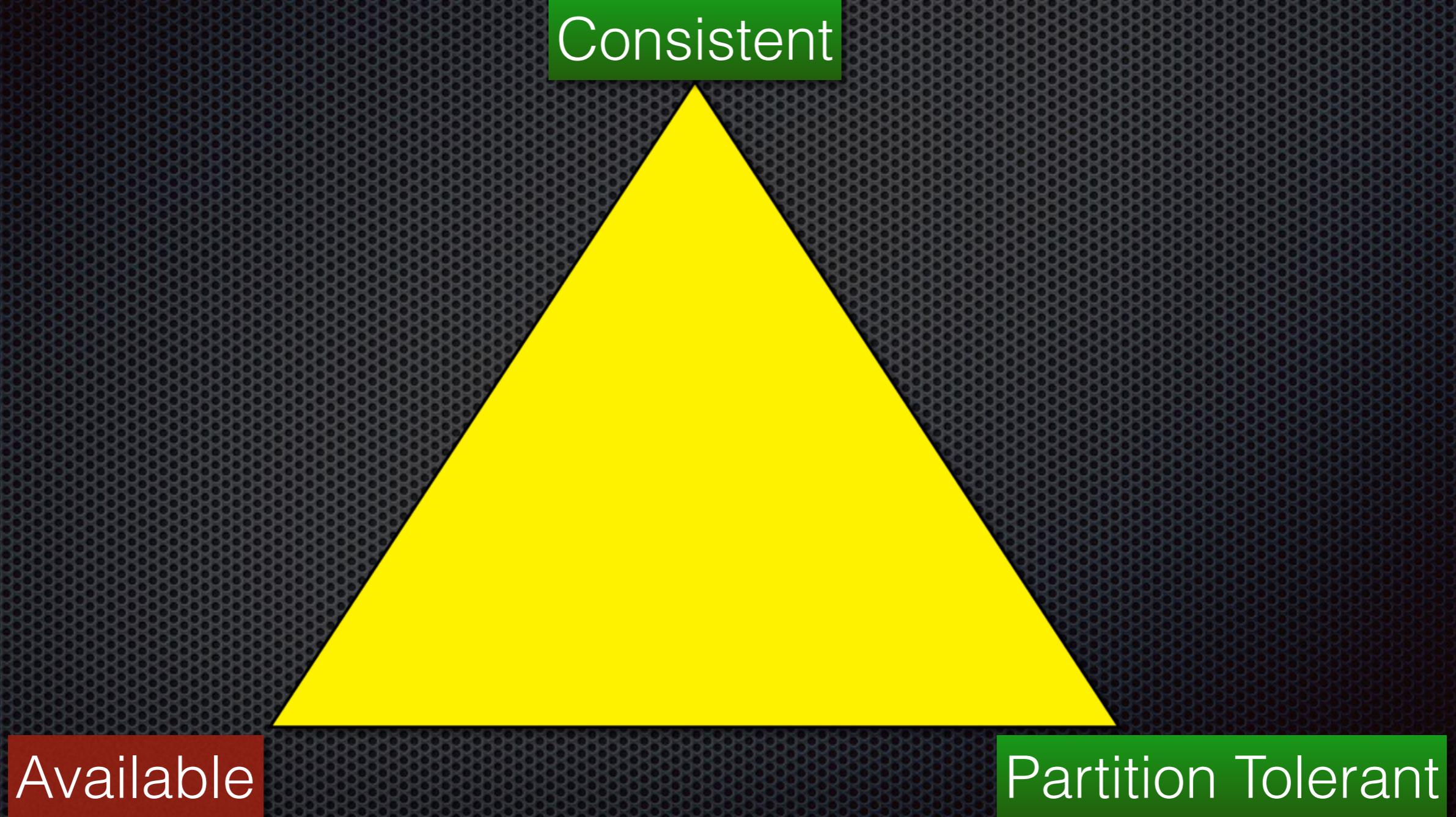
Partition Tolerance is a myth!

Consistent

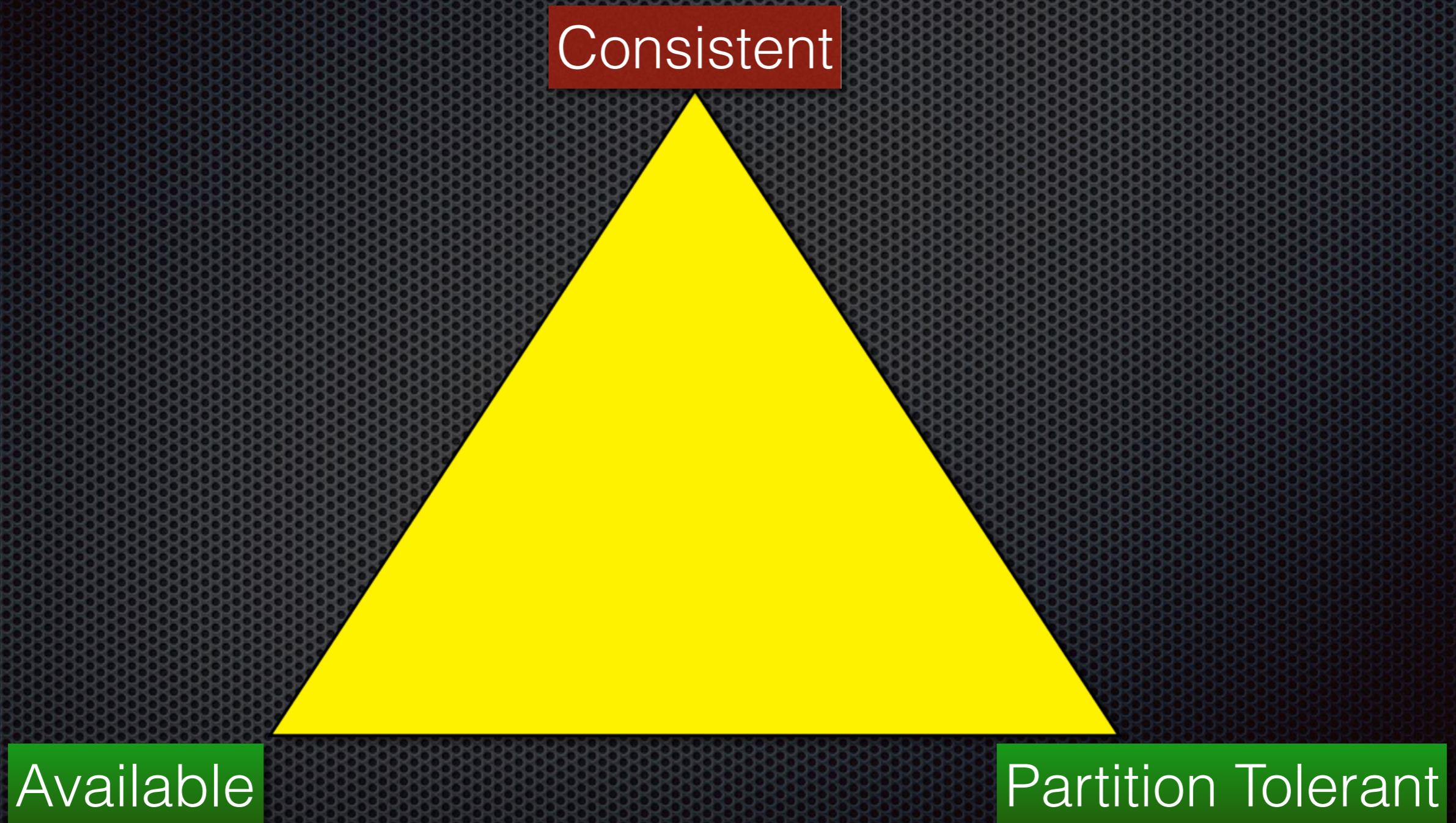
Available

Partition Tolerant

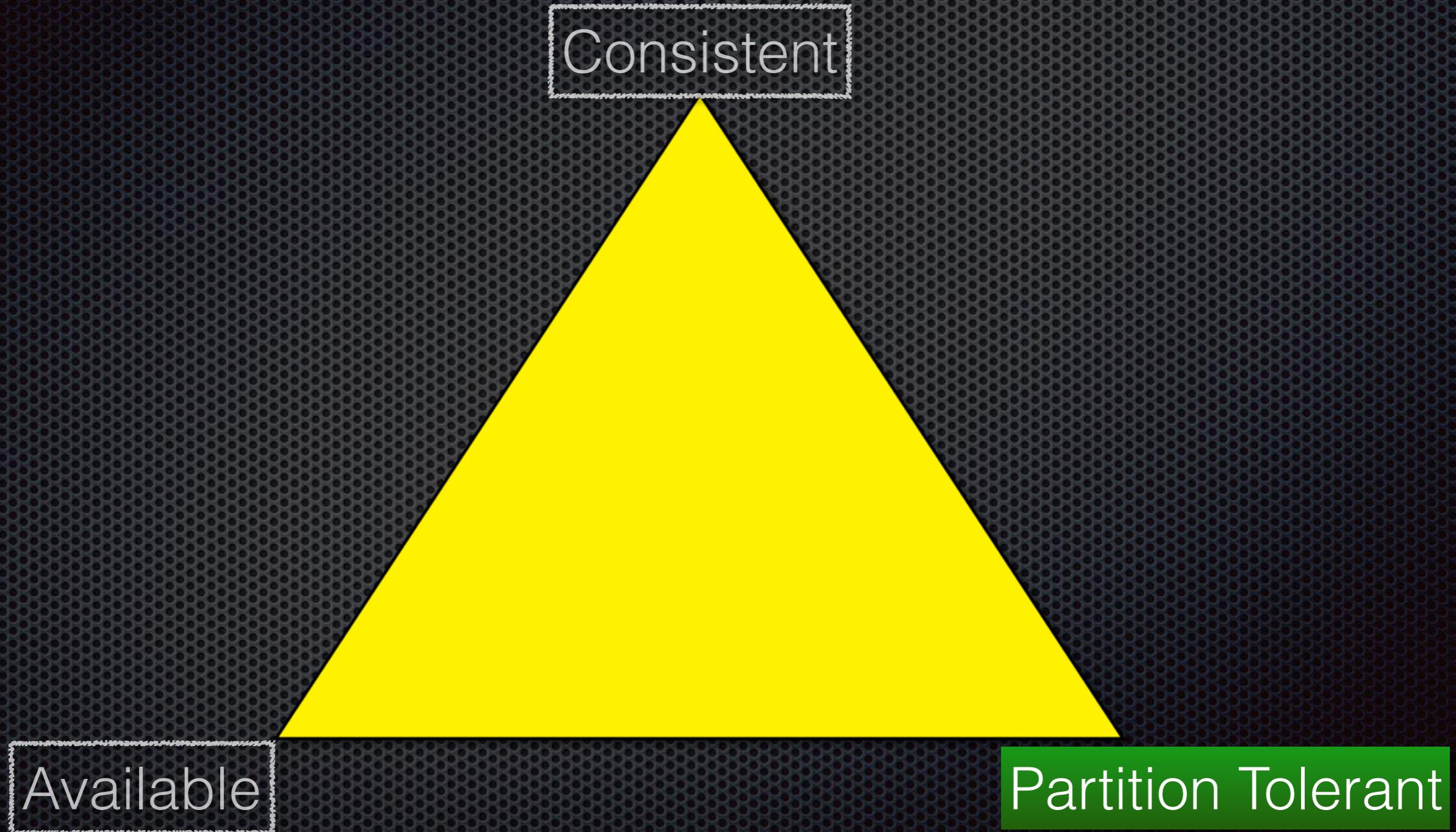
*We can guarantee Consistency, but not Availability.
We may time out or fail to return anything*



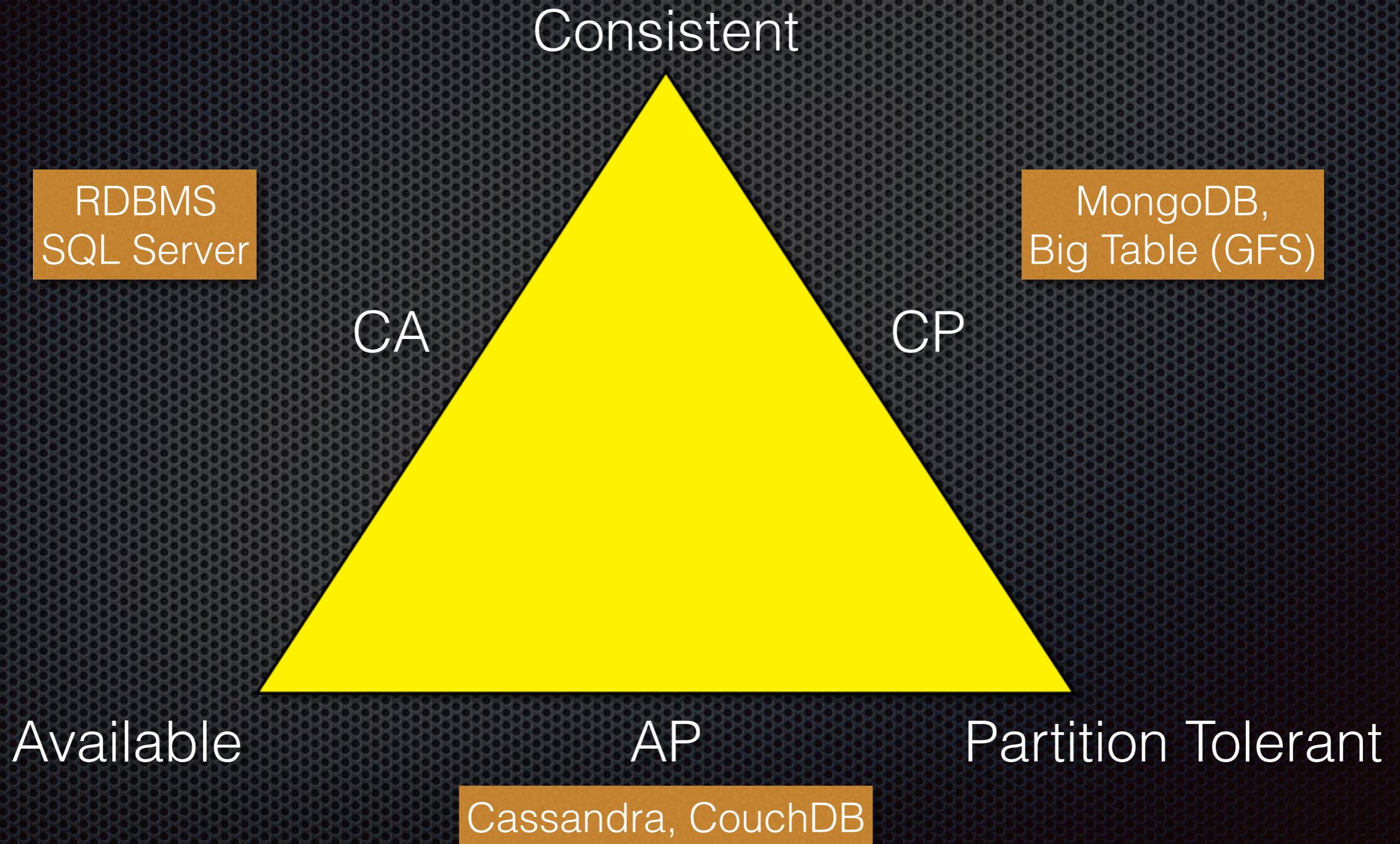
*We can guarantee Availability, but our data
may not be Consistent with other nodes*



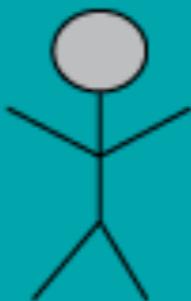
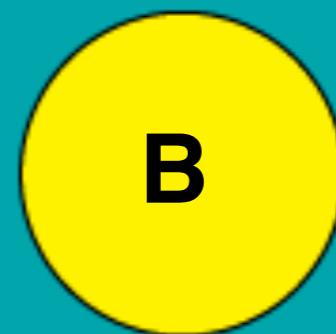
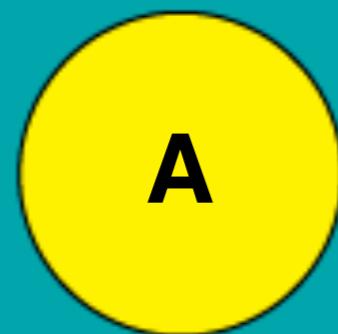
Our business needs will drive whether we choose Consistency or Availability



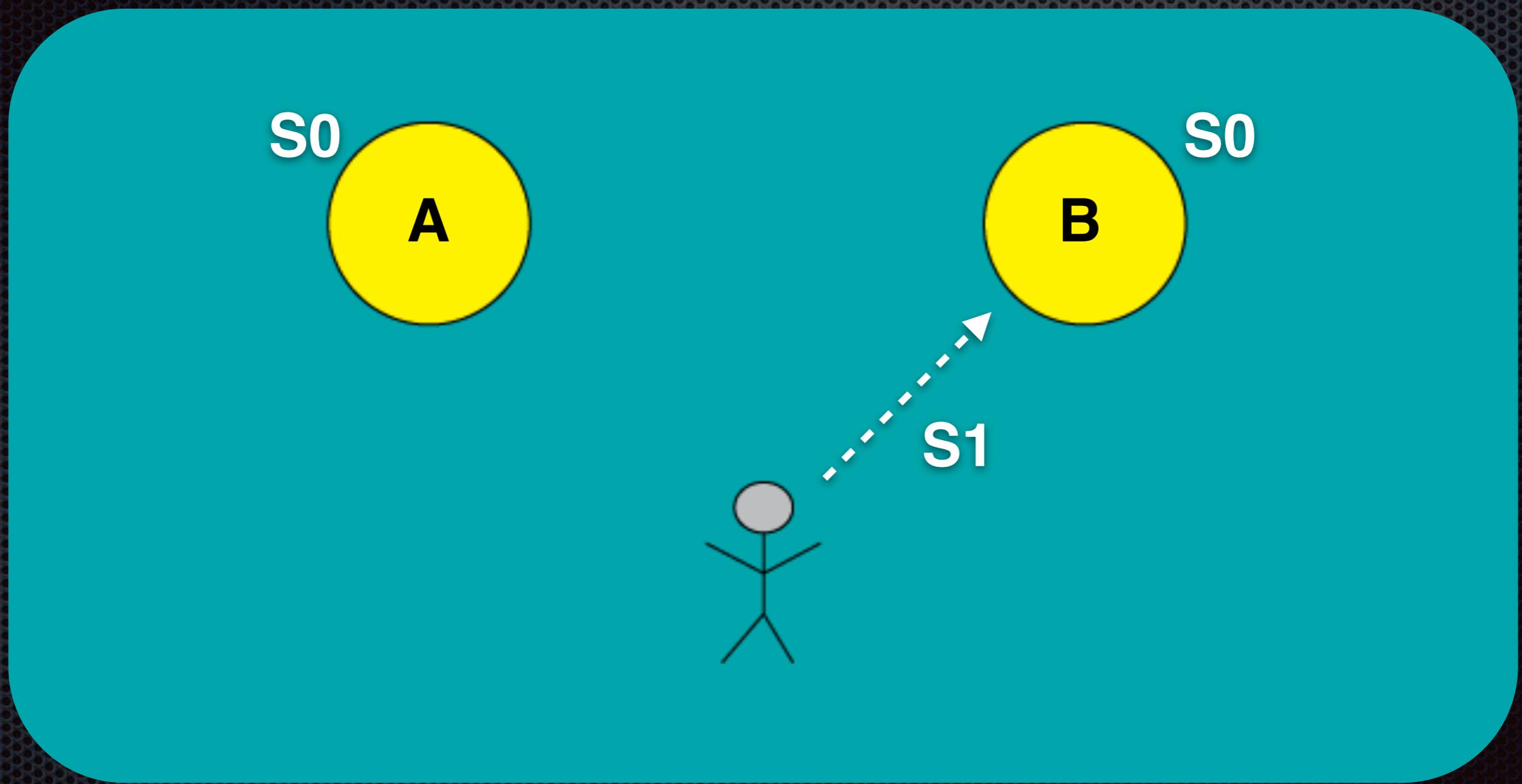
Implementations



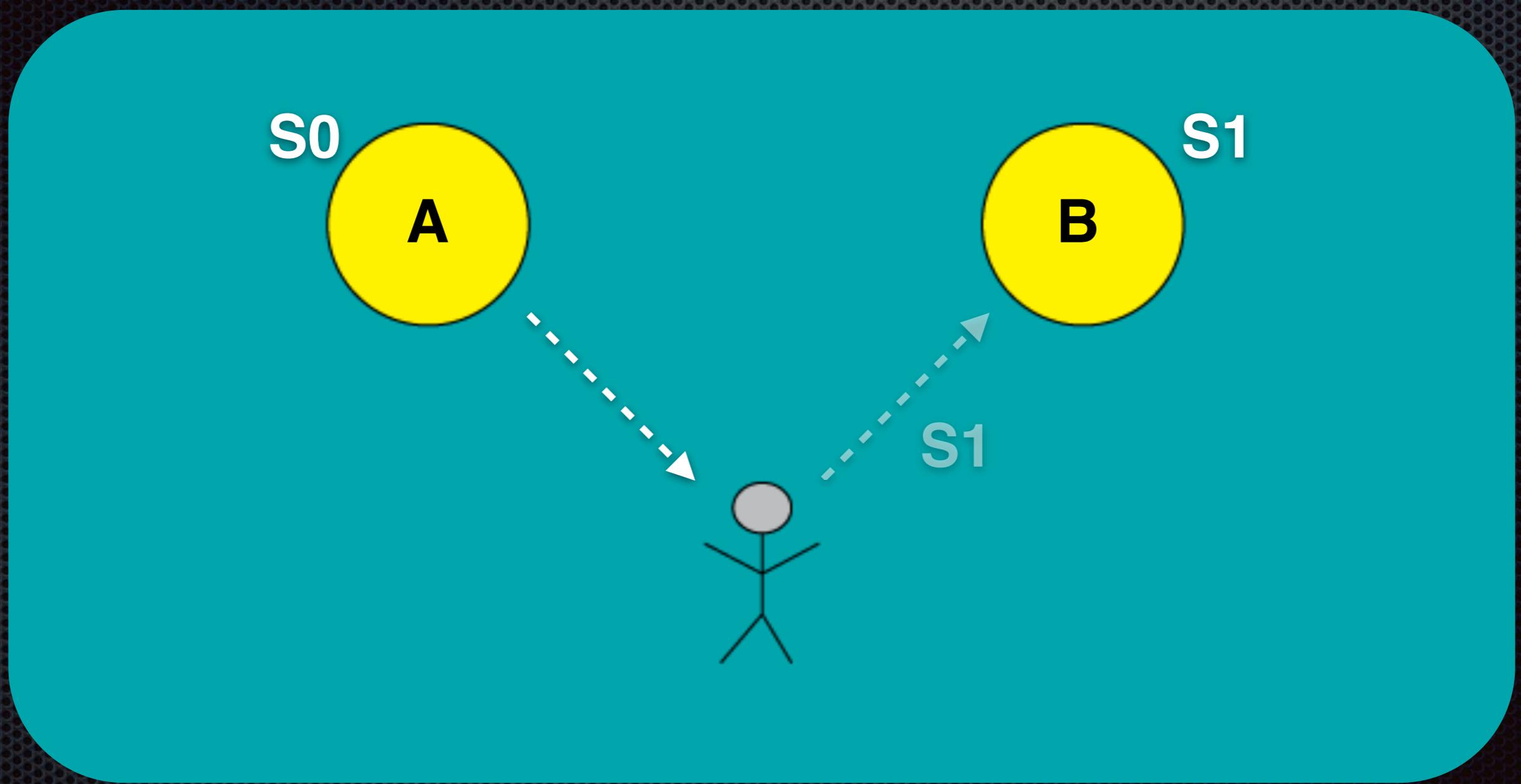
A contrived example for Consistency



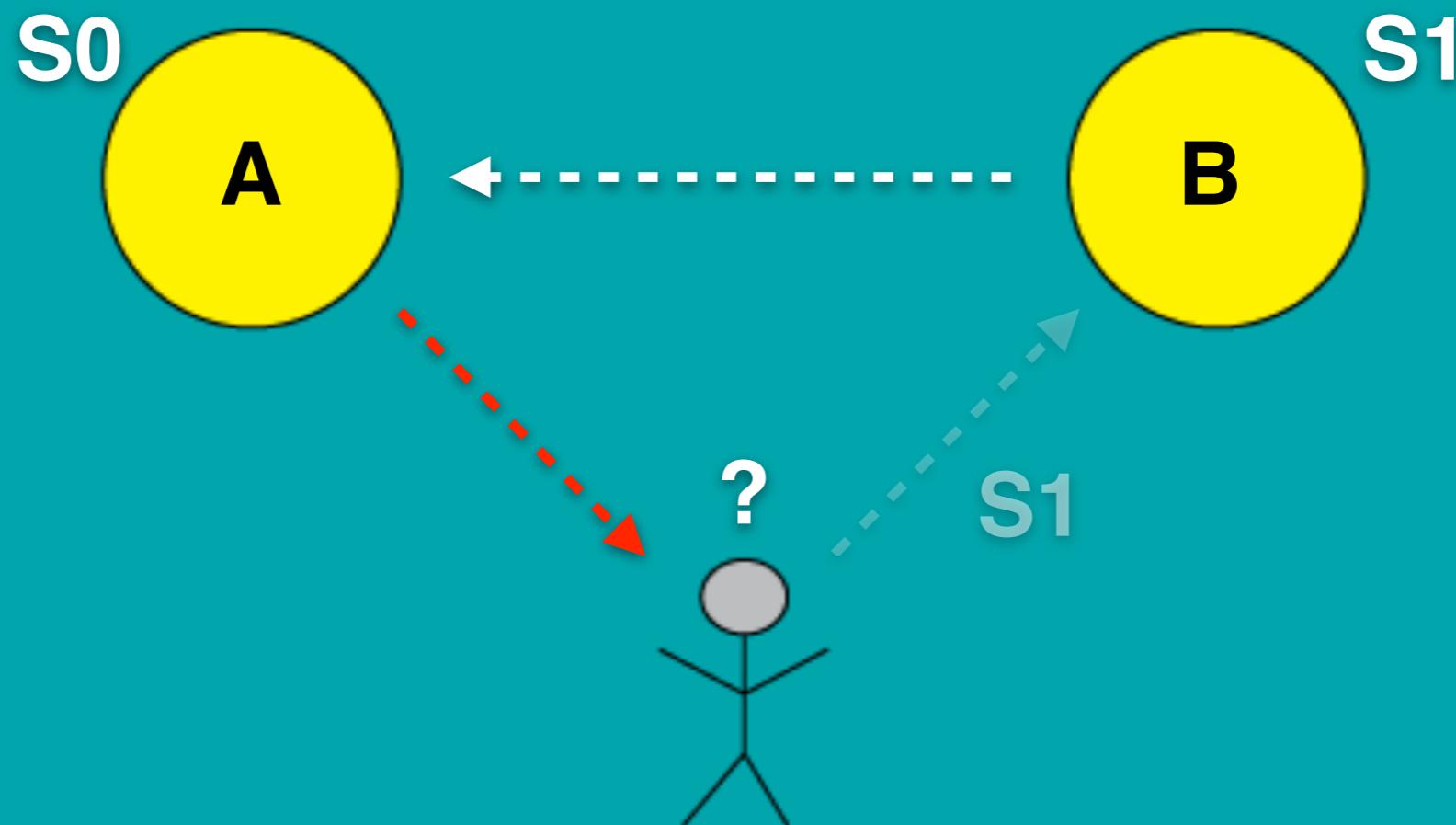
User writes S1 to Node B



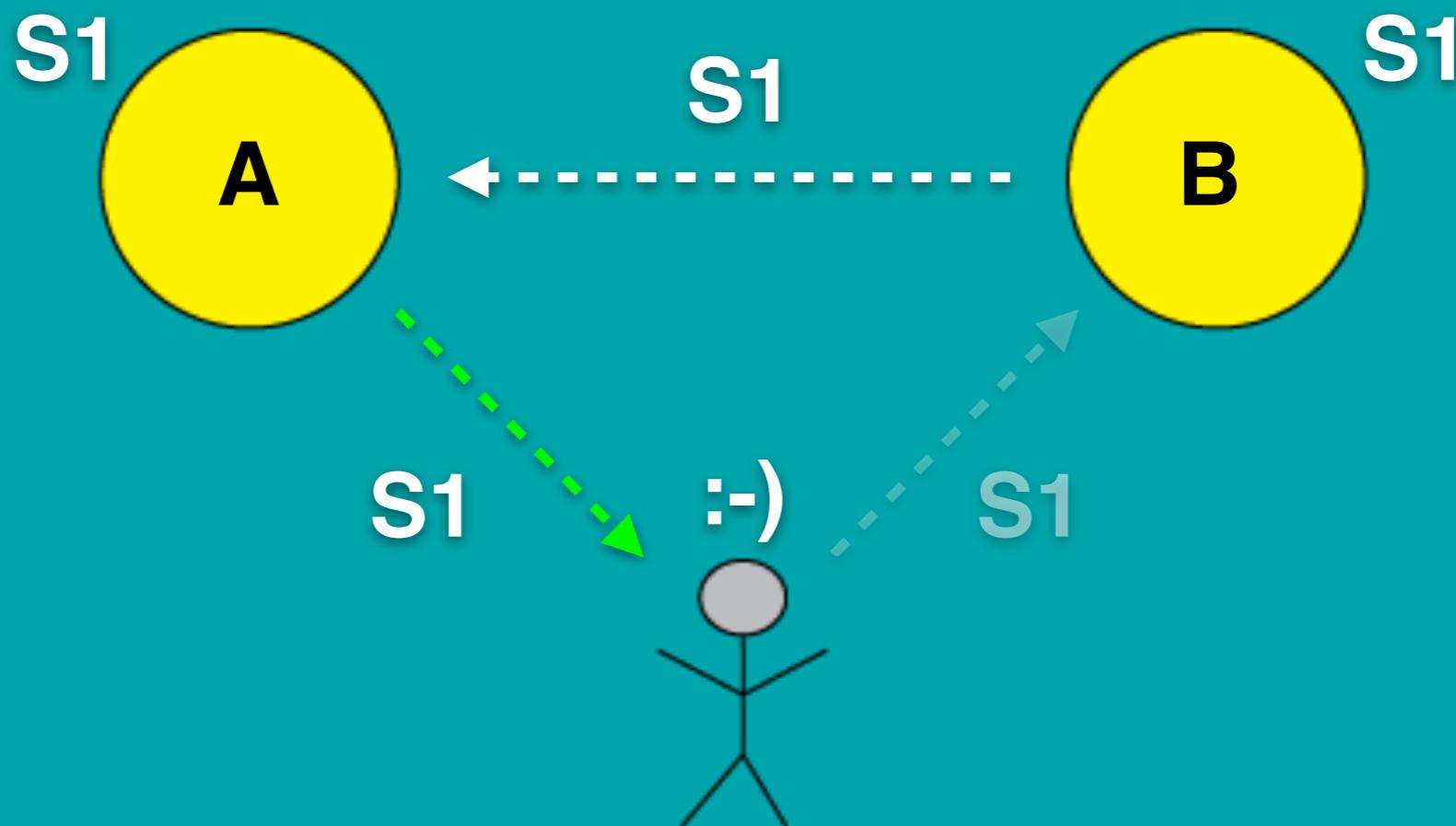
User queries Node A
Nodes B & A have not sync'd



Query is blocked
until B syncs with A

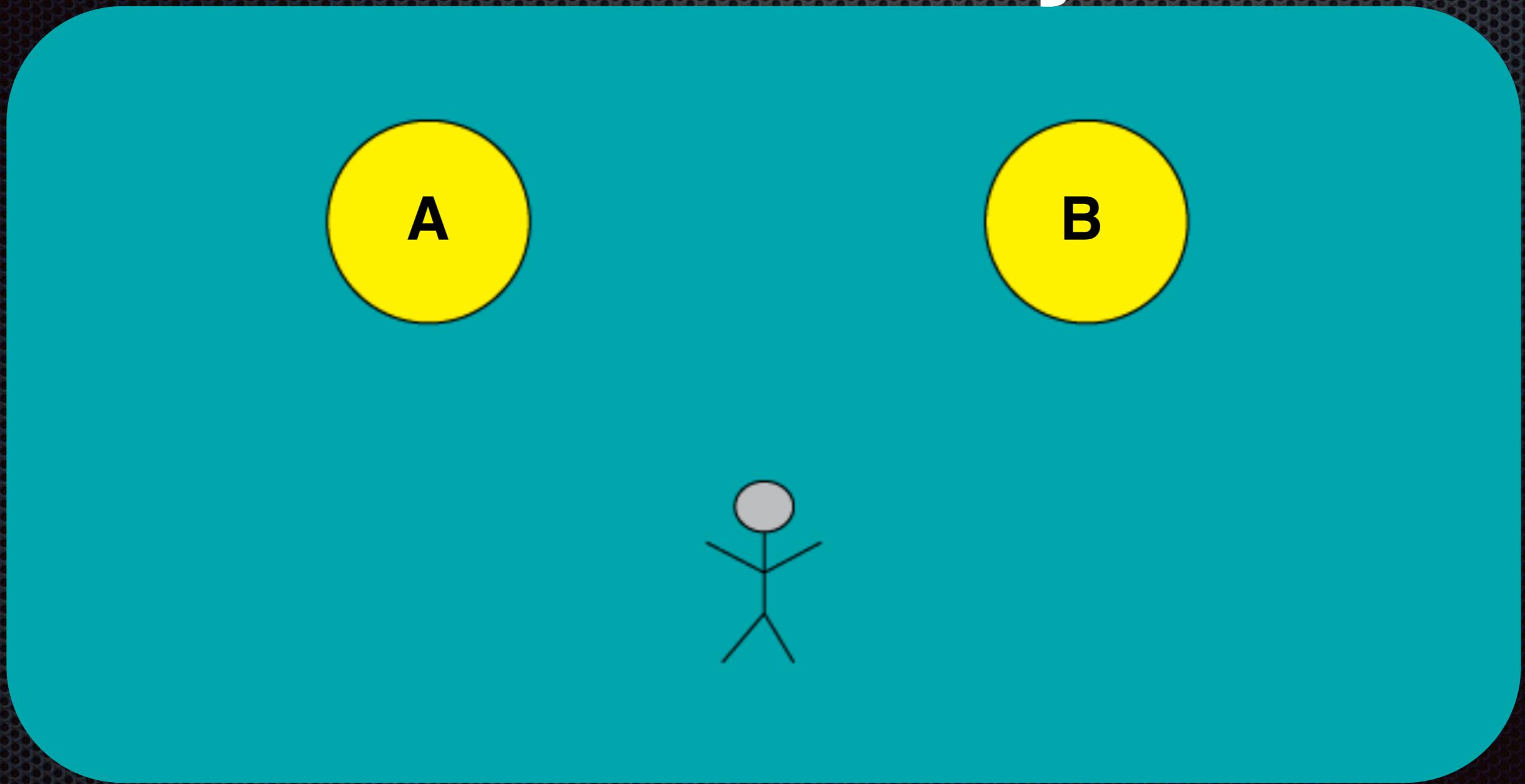


Once B syncs with A,
the query on A is unblocked
and returns S1 as expected

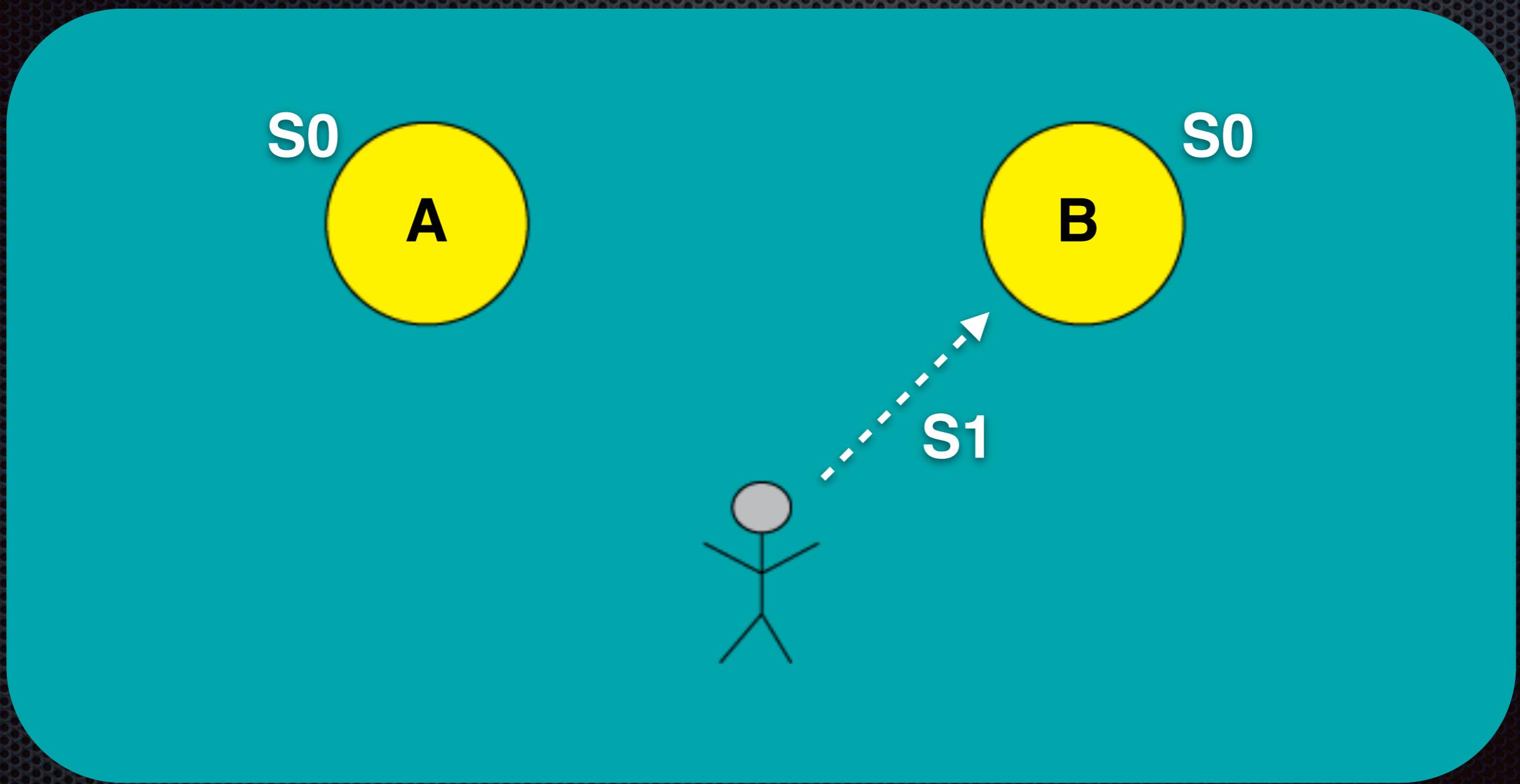


* The query could potentially time out

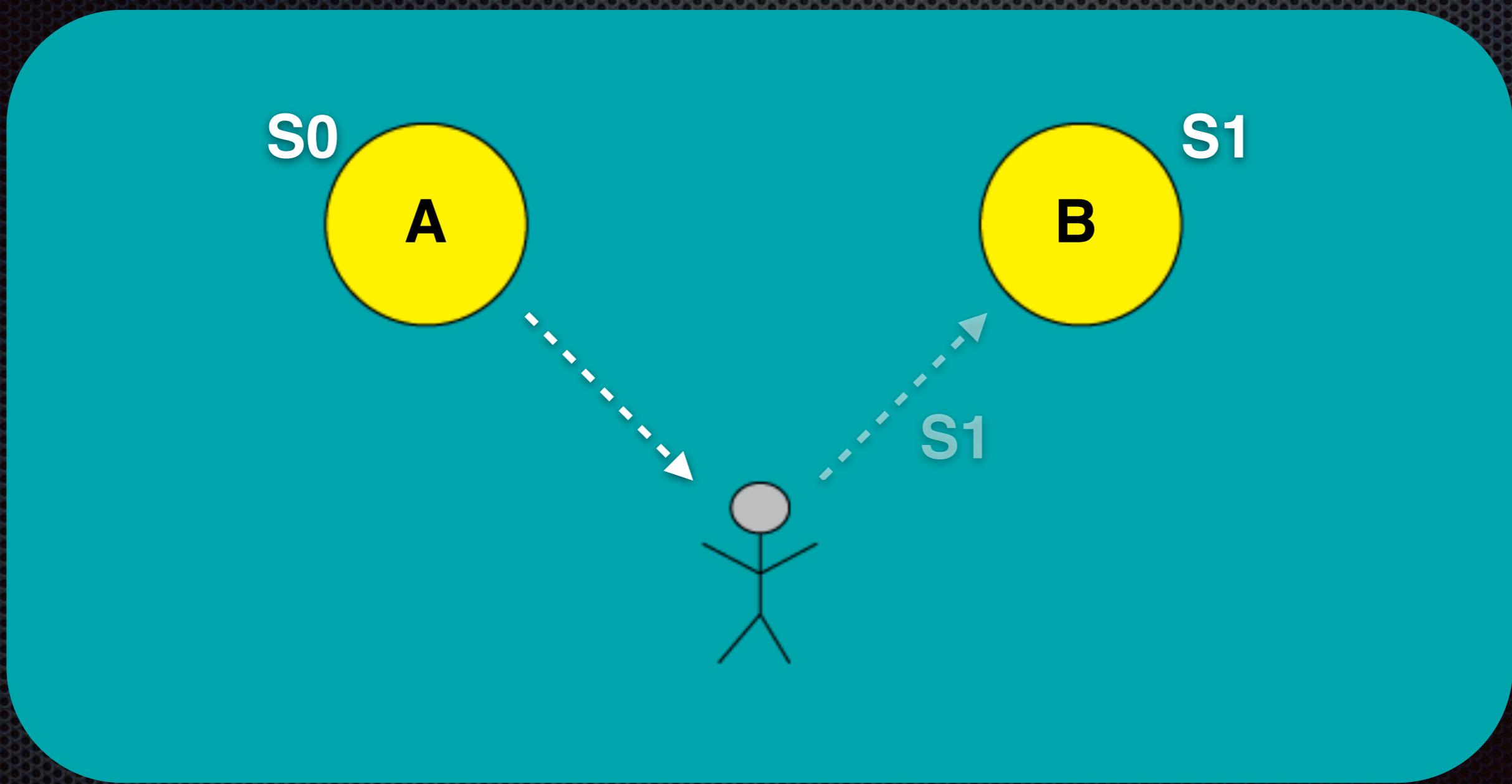
A contrived example for Availability



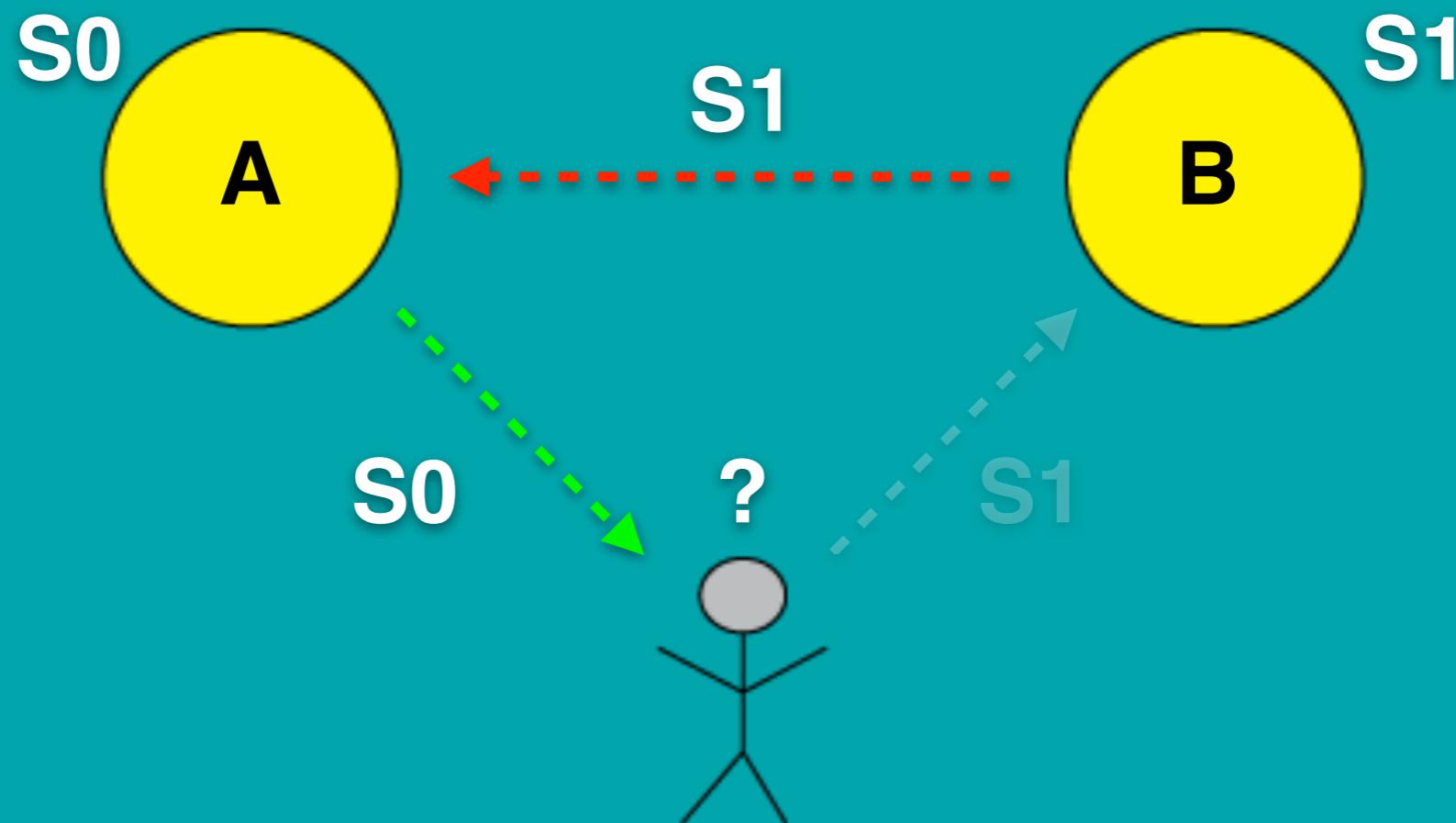
User writes S1 to Node B



User queries Node A for S1

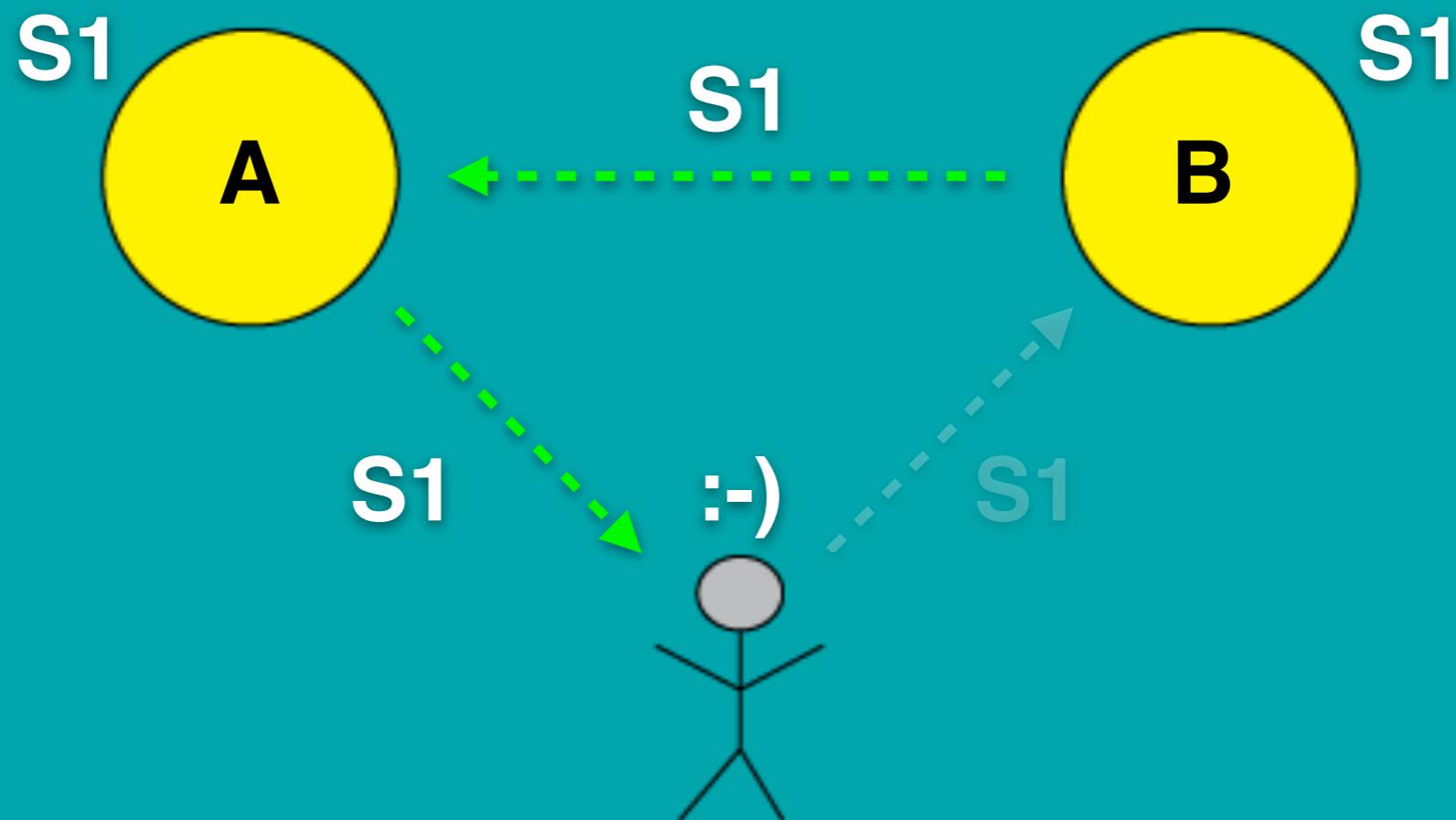


Query returns current state of A, but is not consistent with B

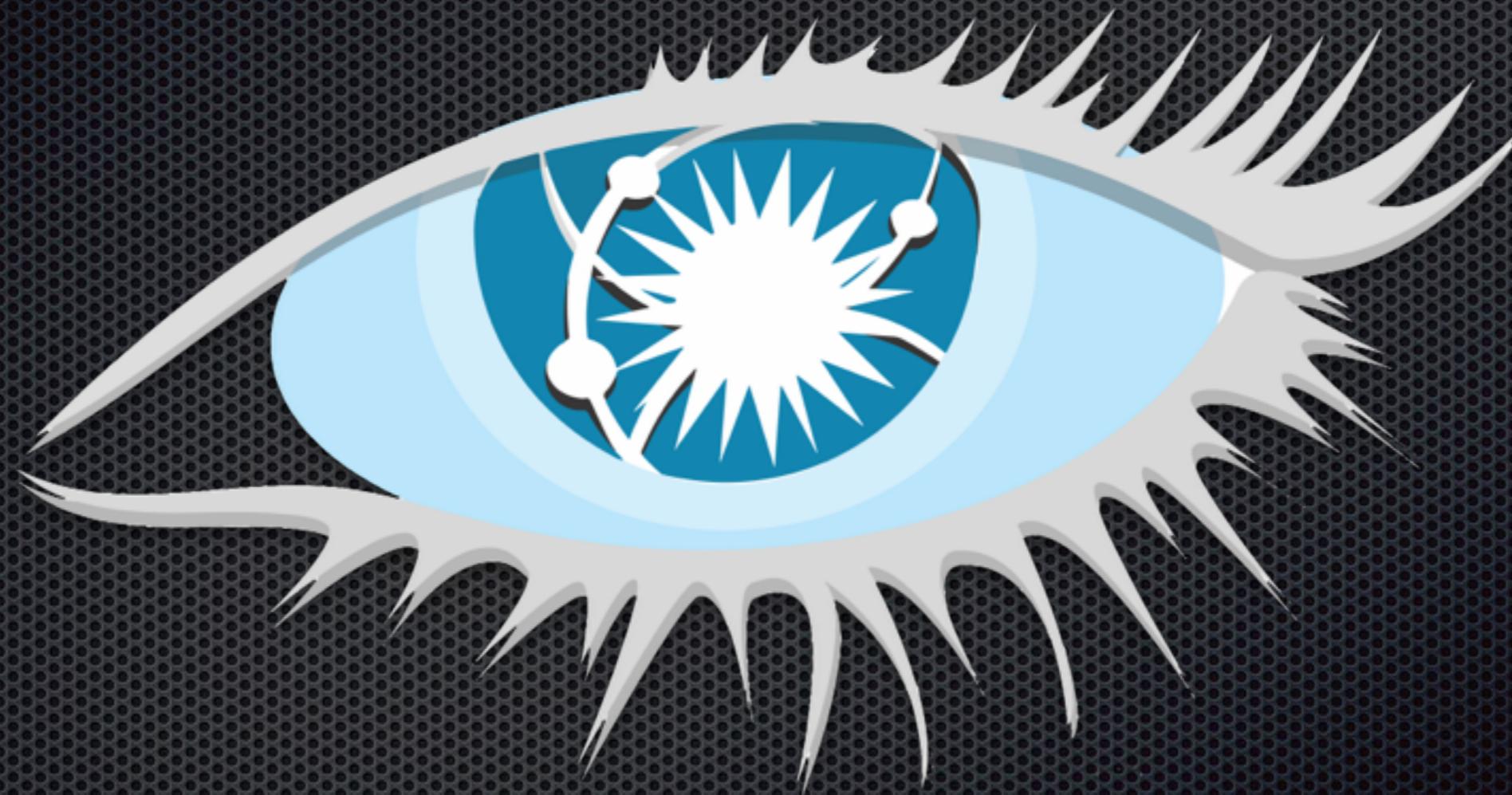


A later query of A yields S1 previously written to B.

Eventual consistency has been achieved.



Cassandra



History

- Developed for Facebook Inbox Search
- Created by one of the authors of Amazon Dynamo
- Released as Open Source in 2008
- Became an Apache Incubator project in 2009
- Graduated to an Apache Top Level Project

Who is using Cassandra in production today?

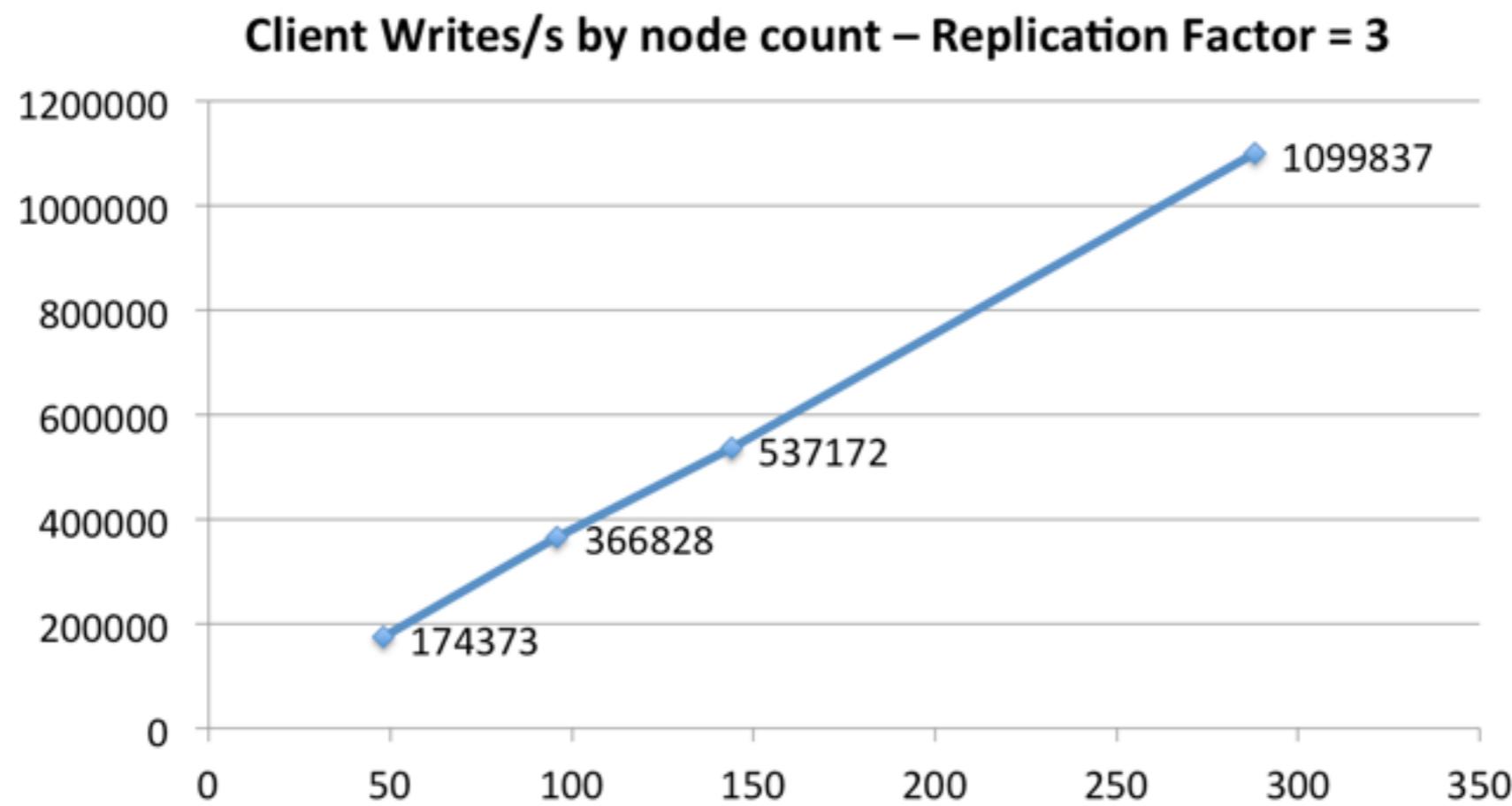
- Twitter
- Netflix
- Credit Suisse
- Cisco
- Many more...
- <http://PlanetCassandra.org/companies/>

Benefits

- Linear Scalability
- Data Sets can be larger than available memory
- Multi-master
- Built-in support for handling multiple data centers
- Decentralized & Distributed - No single point of failure
- Integrated caching
- Consistency options can be tuned through configuration
- Supports MapReduce
- Familiar query syntax - CQL (Cassandra Query Language)
- Designed for sparse loading of loosely typed data

Linear Scalability

Scale-Up Linearity



Challenges

- JOINs are not supported! (not unique to Cassandra)
- Not for financial data (eventual consistency)
- Tooling is not yet as mature as Oracle or SQL Server

Why NOSQL?

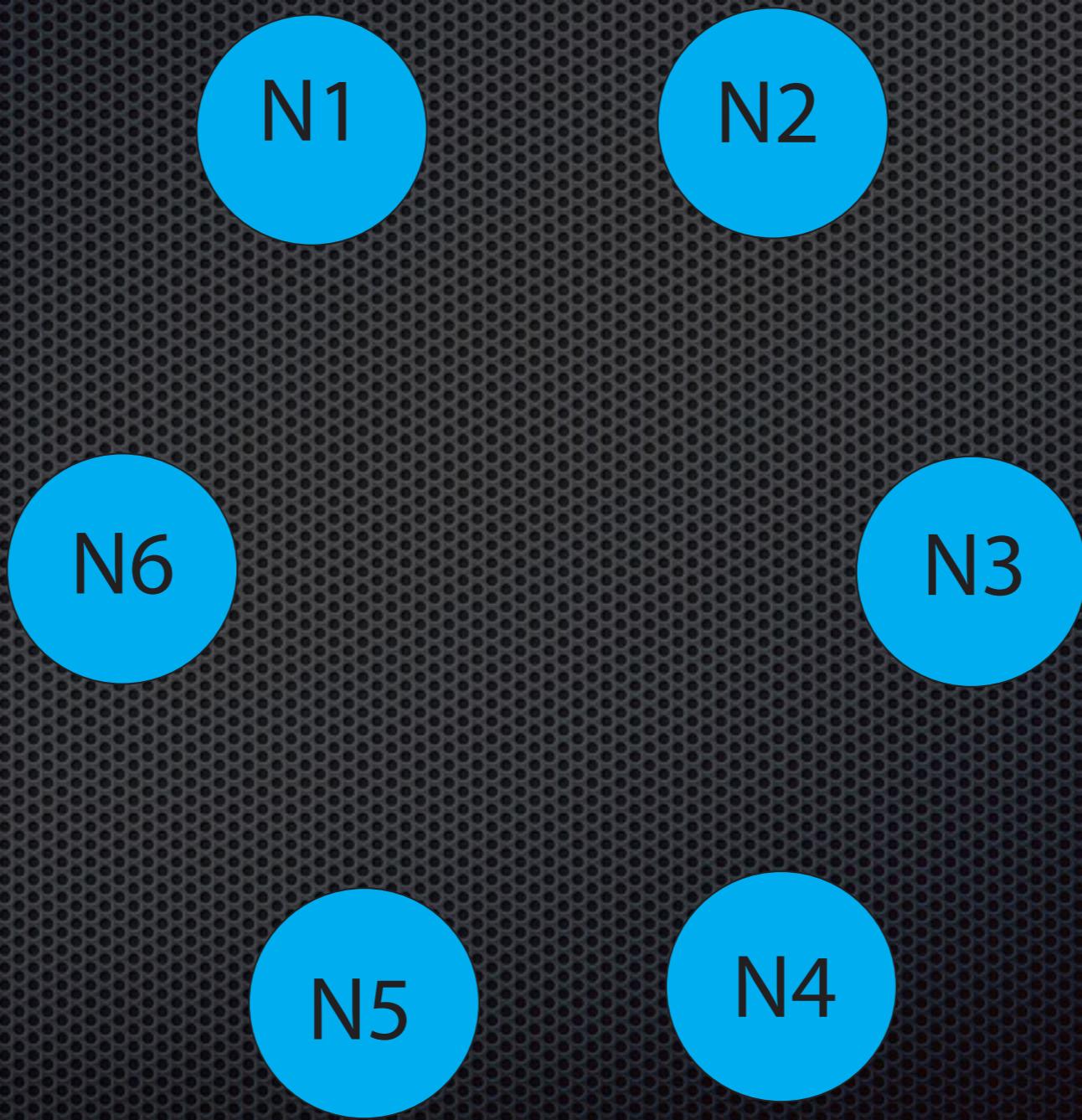
Multiple persistence strategies

- Use the right tool for the job. Sometimes more than one tool is appropriate. NoSQL can work in conjunction with SQL solutions. For example, you might have transactions you continue to store in a Relational Database, and build your new user social graph data in a NoSQL solution. Can't decide on a local database for a mobile app? Sometimes the best route is to just persist the whole thing as JSON on disk until you need something faster.

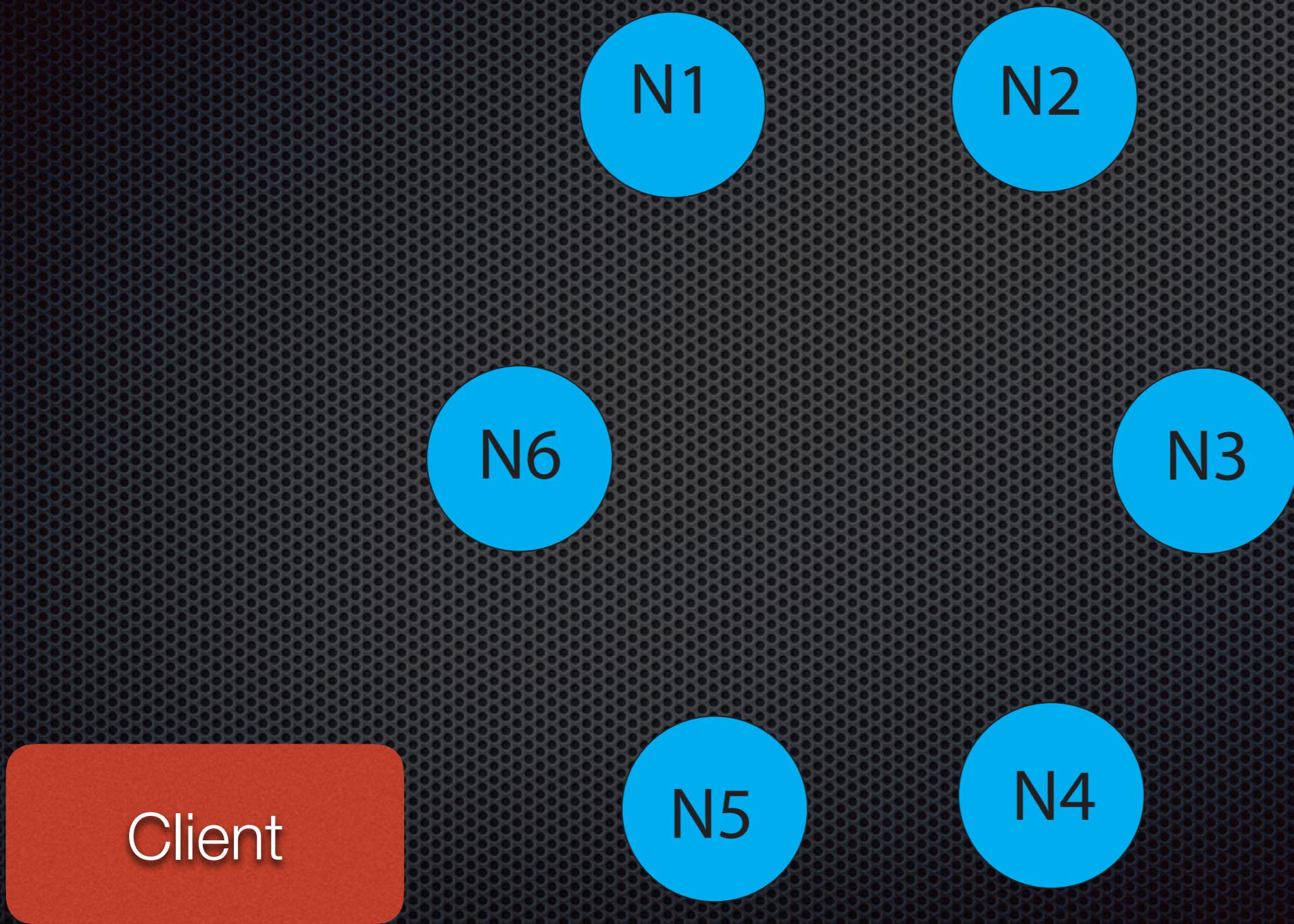
Total Cost of Ownership (TCO)

- Let's face it, SQL Server is expensive. If you need it, and can fully justify the cost, then it might be right for you. But as we've discussed, RDBMS can be a crutch and a default choice for persistence layers. Defaults are just that, default. They're a catch-all that is rarely the best choice unless you're resolving generic default problems

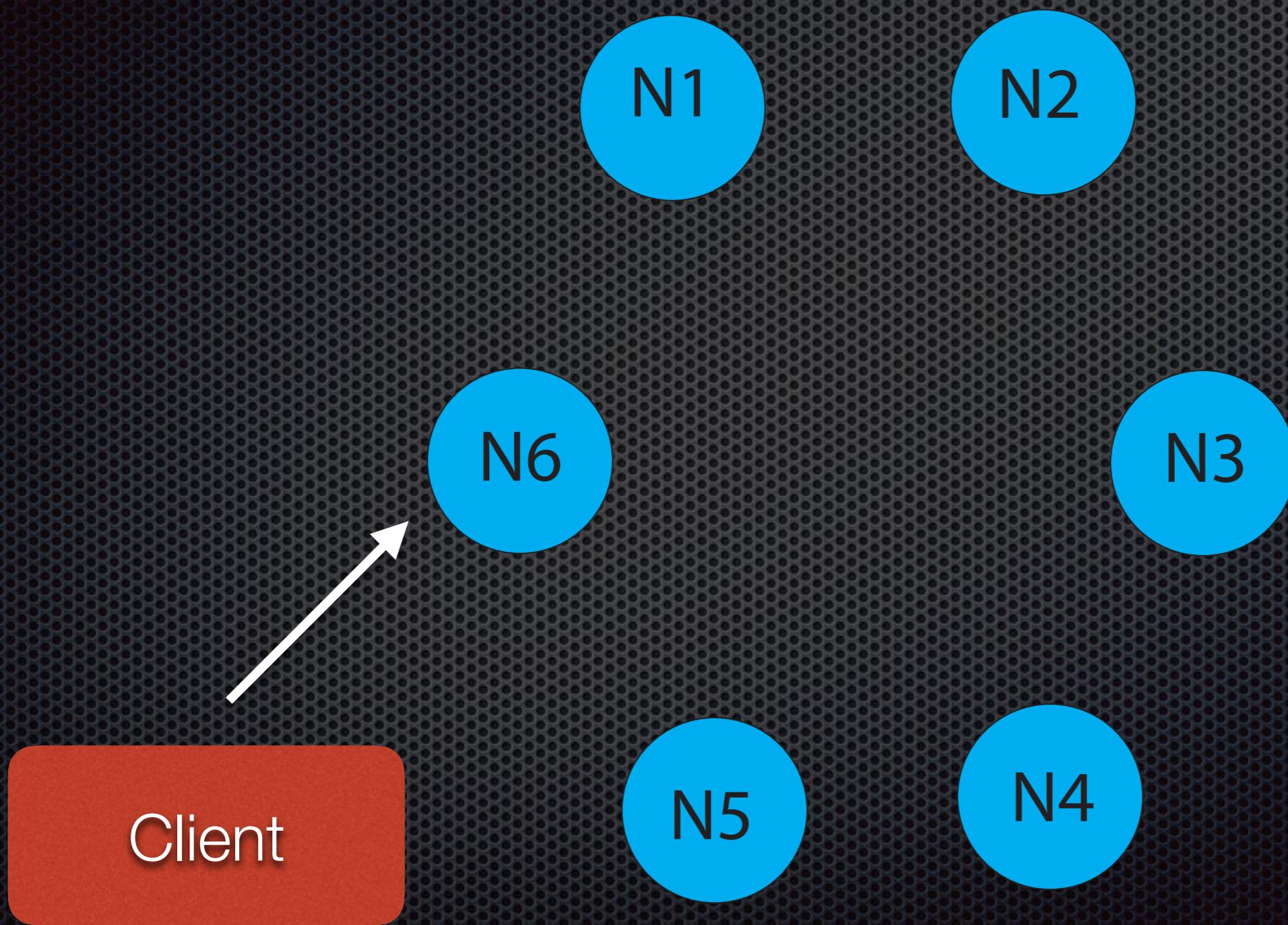
Write (consistency two)



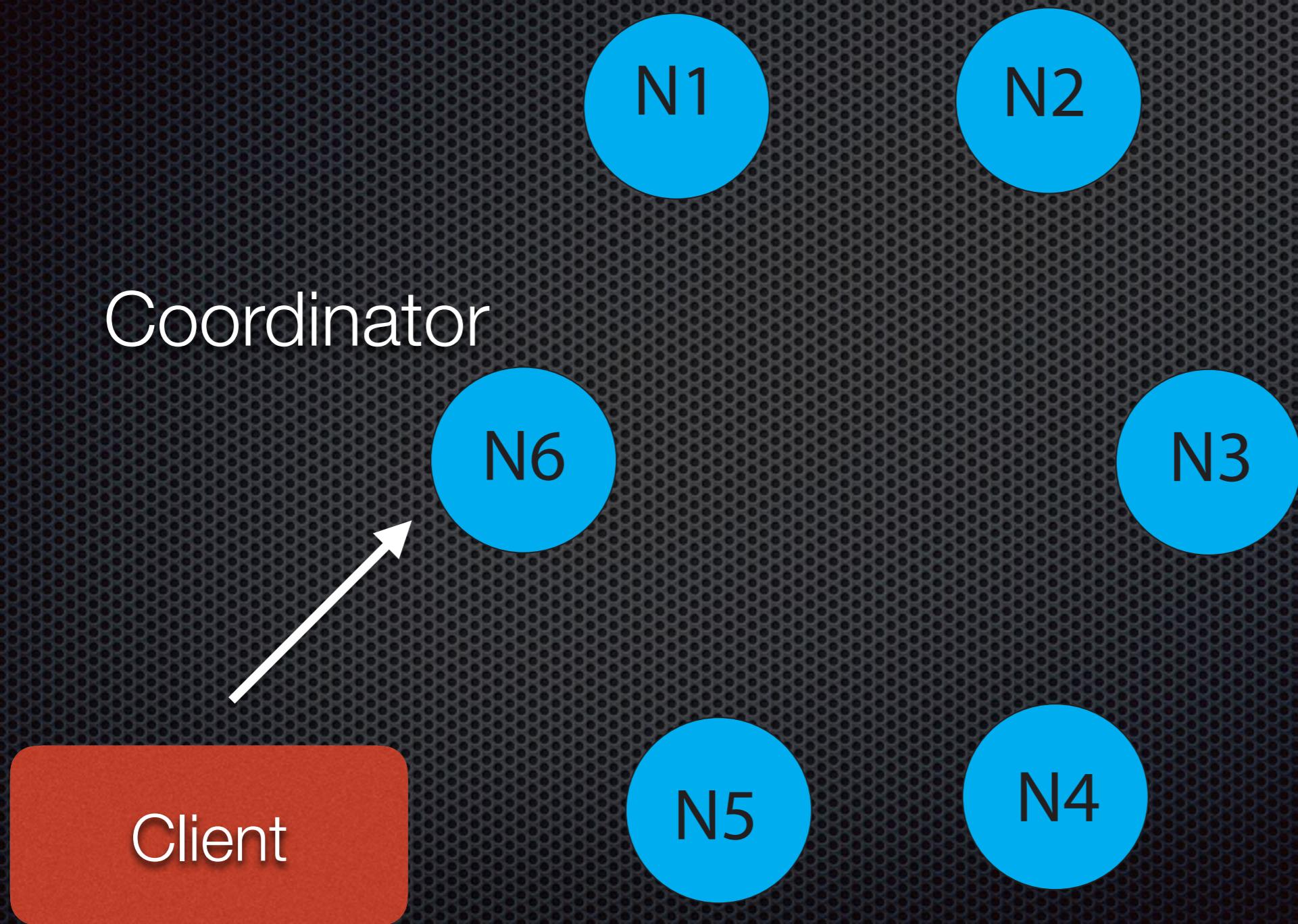
Write (consistency two)



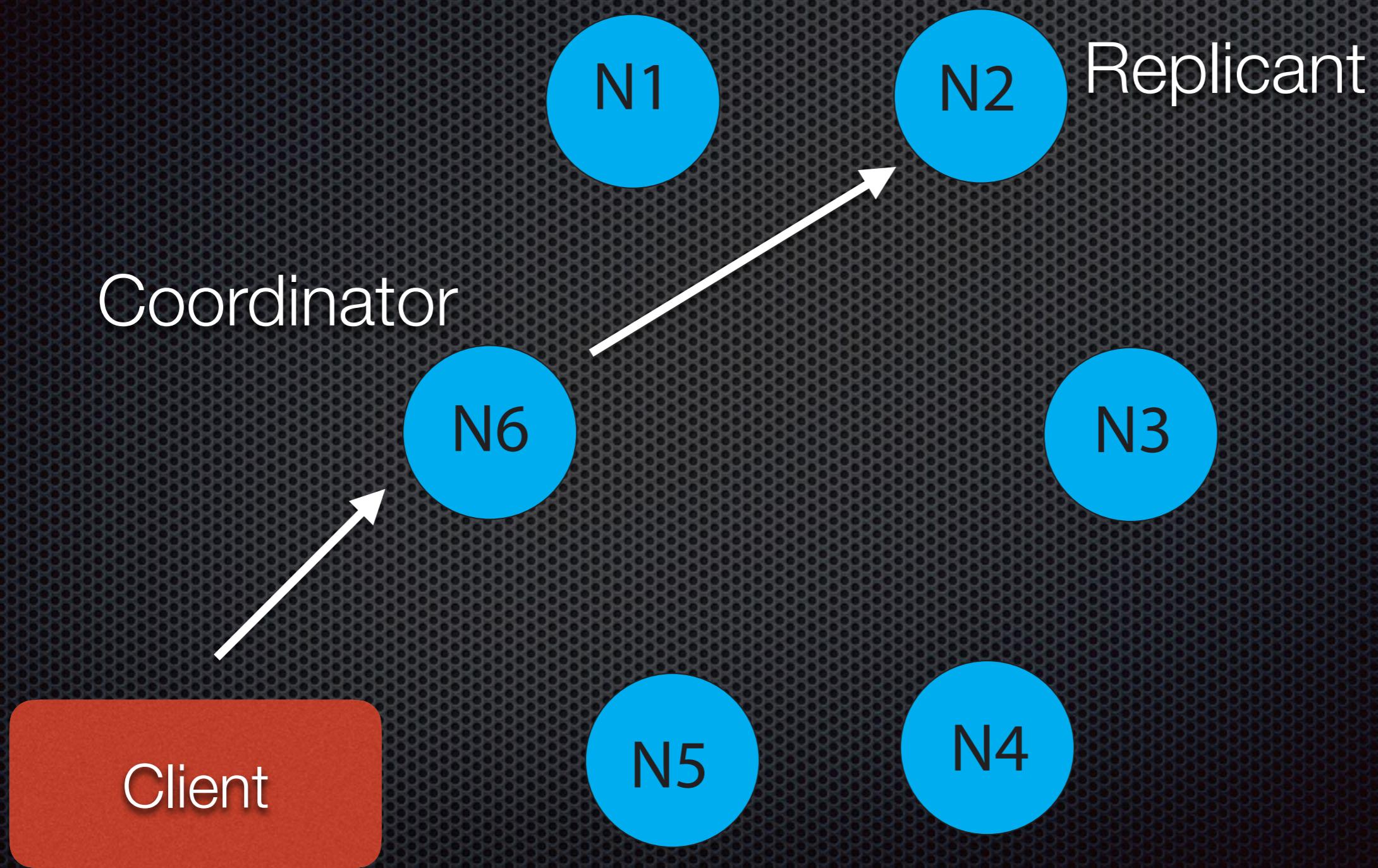
Write (consistency two)



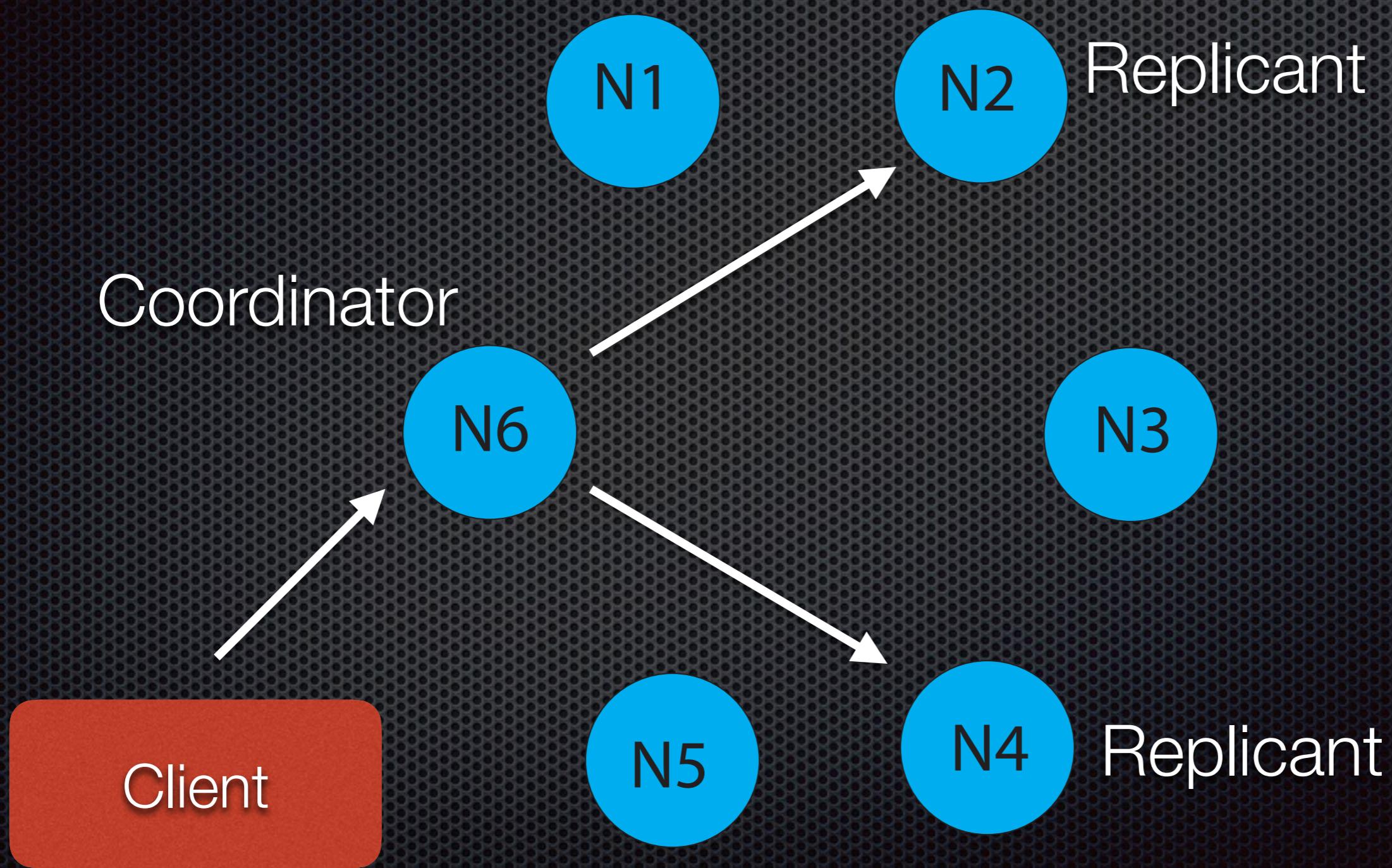
Write (consistency two)



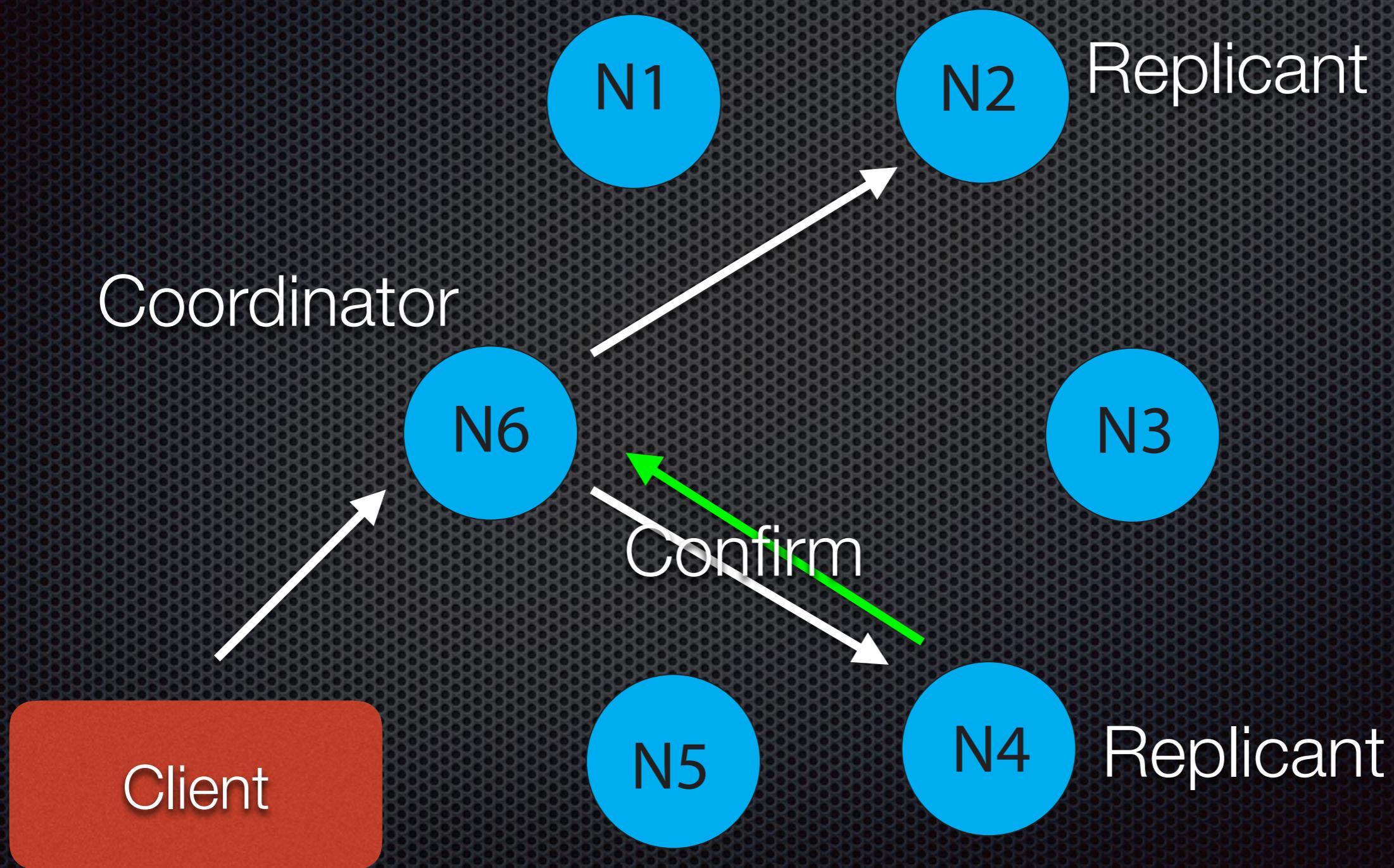
Write (consistency two)



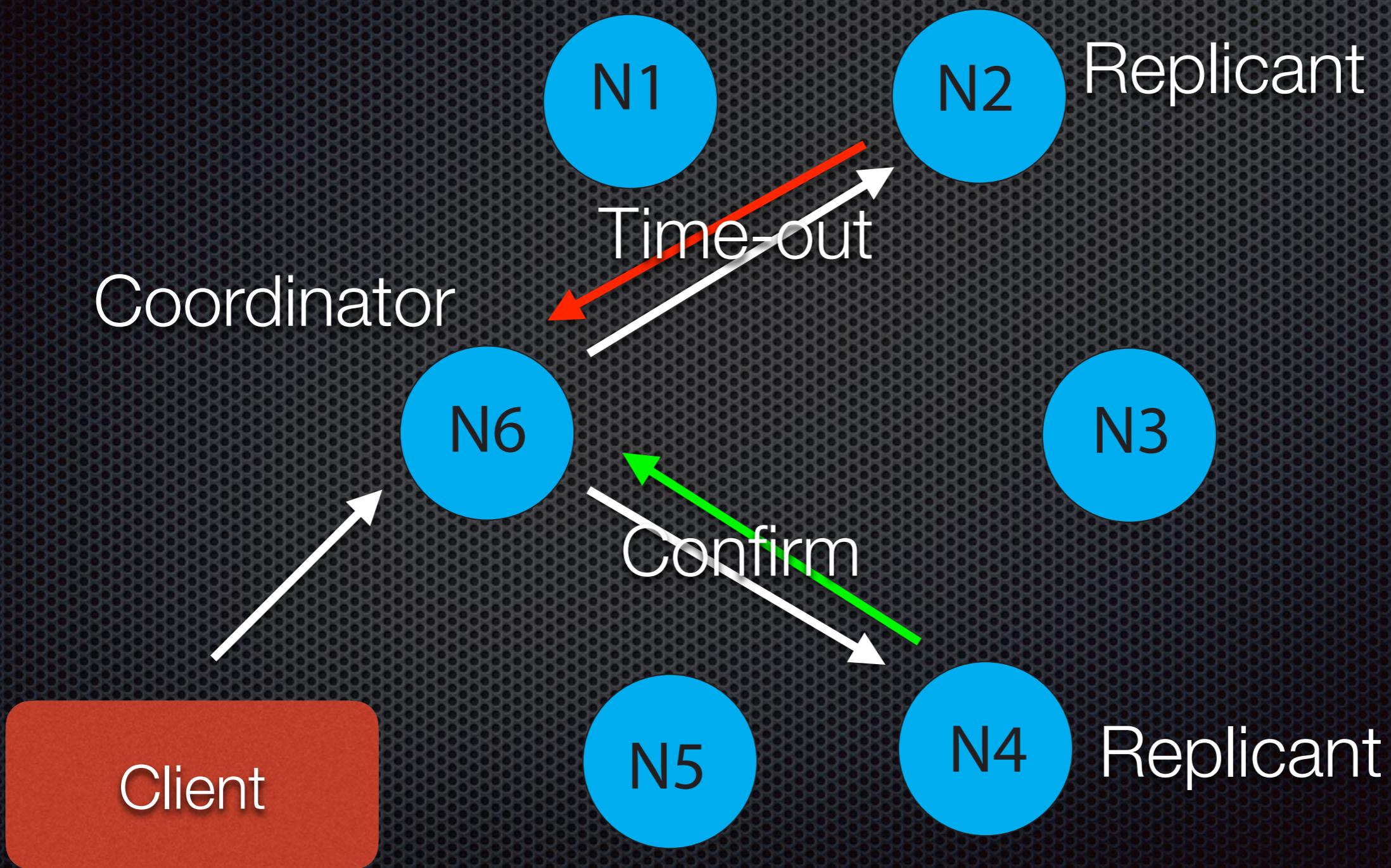
Write (consistency two)



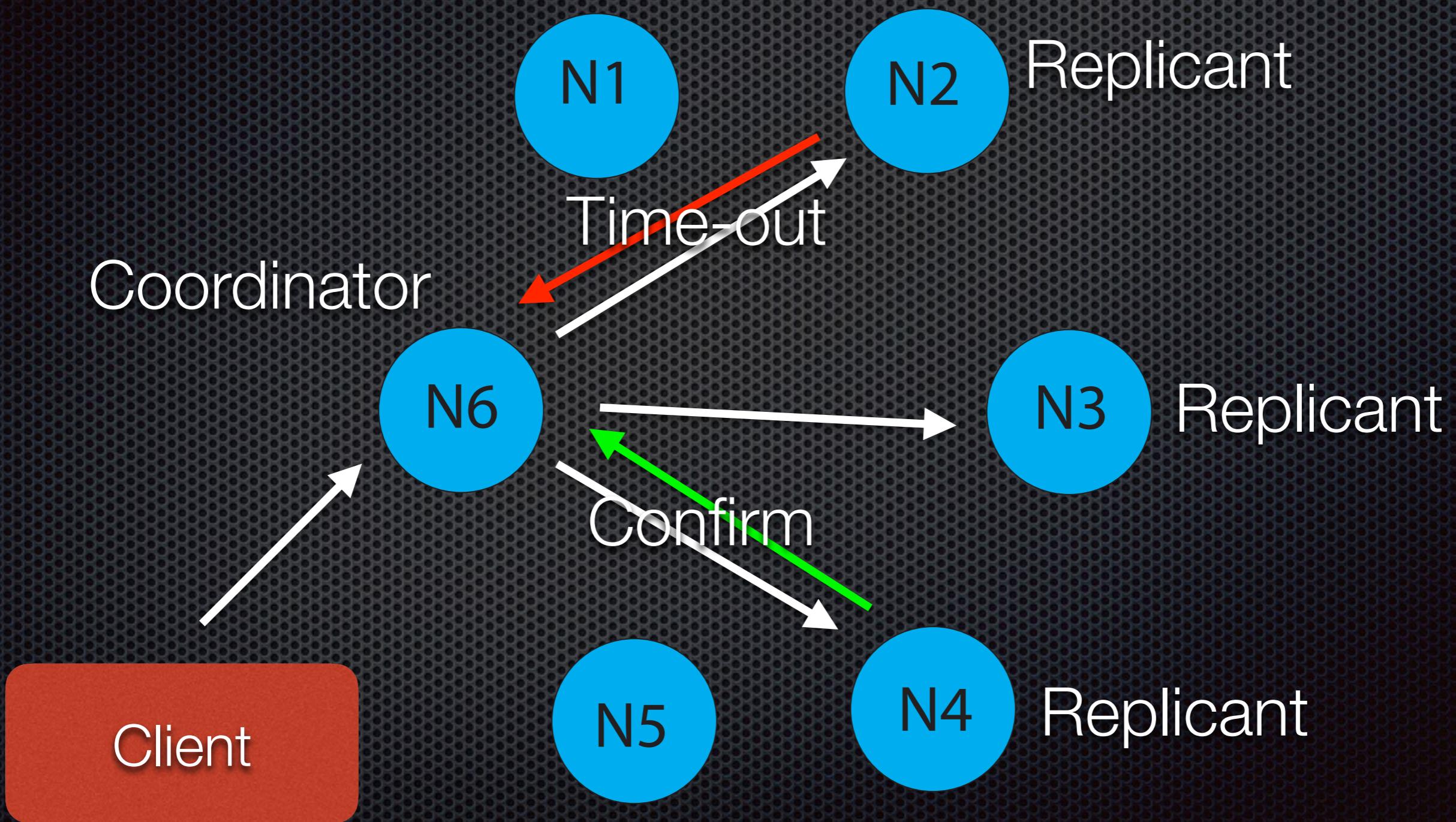
Write (consistency two)



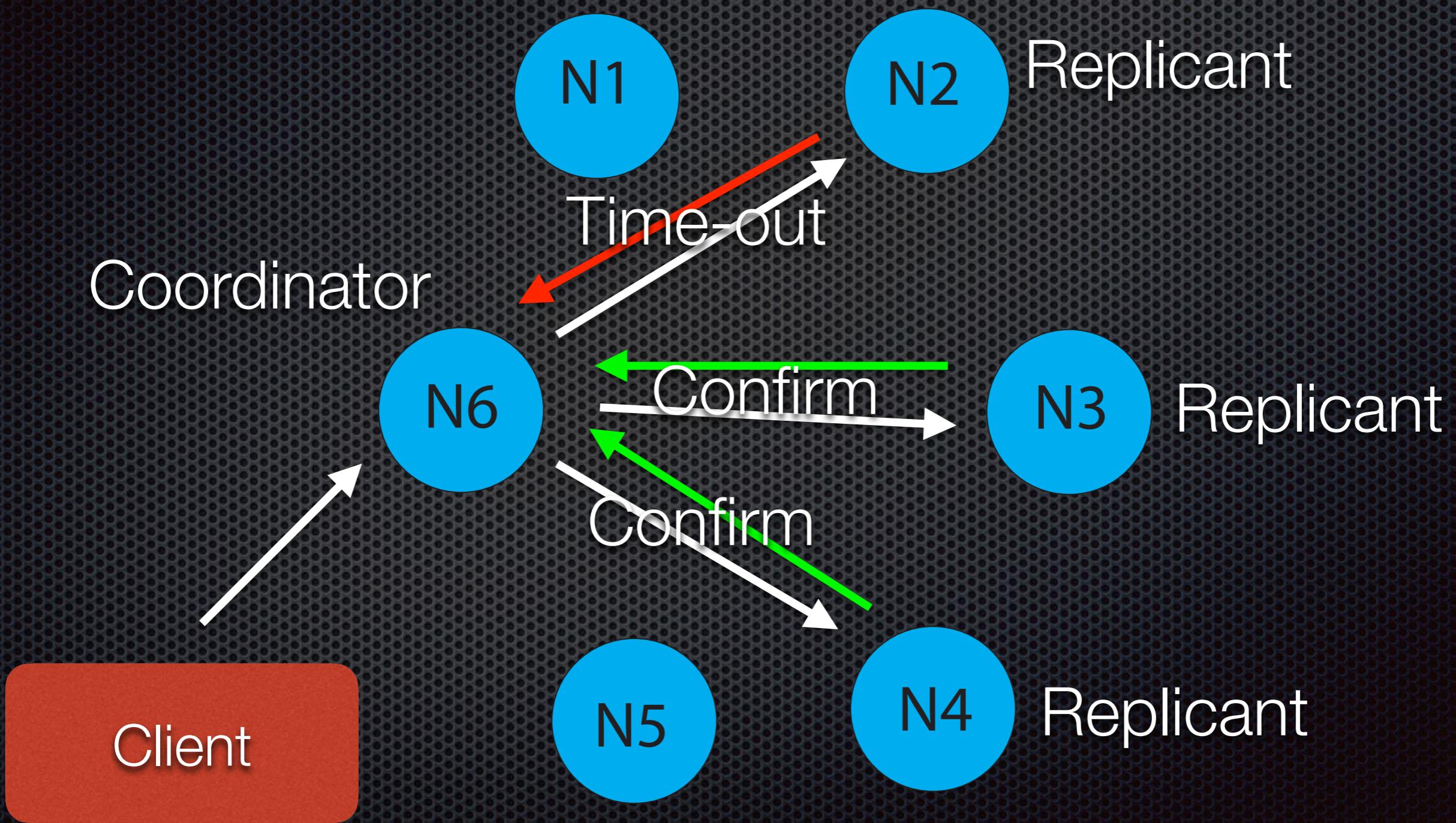
Write (consistency two)



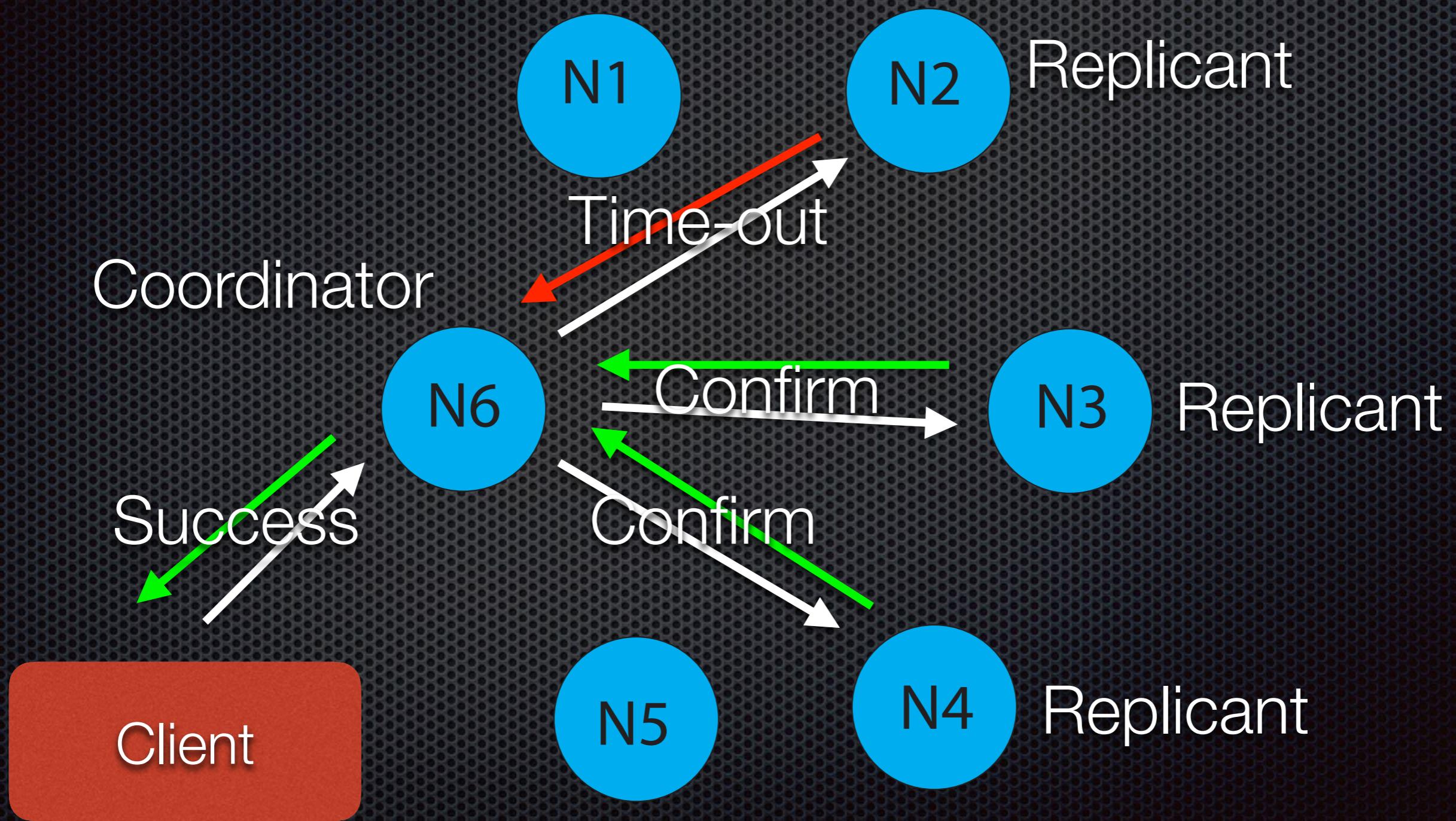
Write (consistency two)



Write (consistency two)



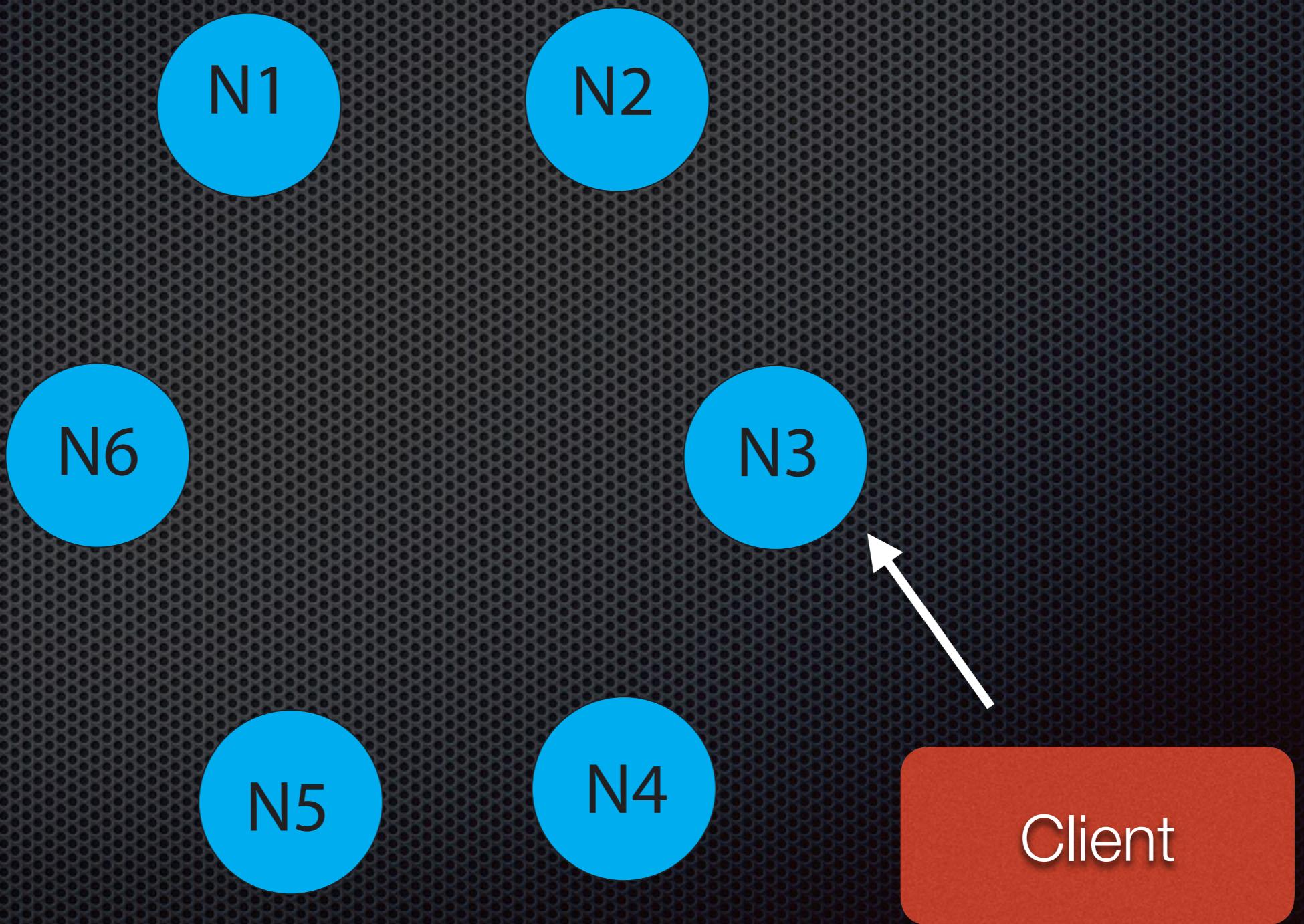
Write (consistency two)



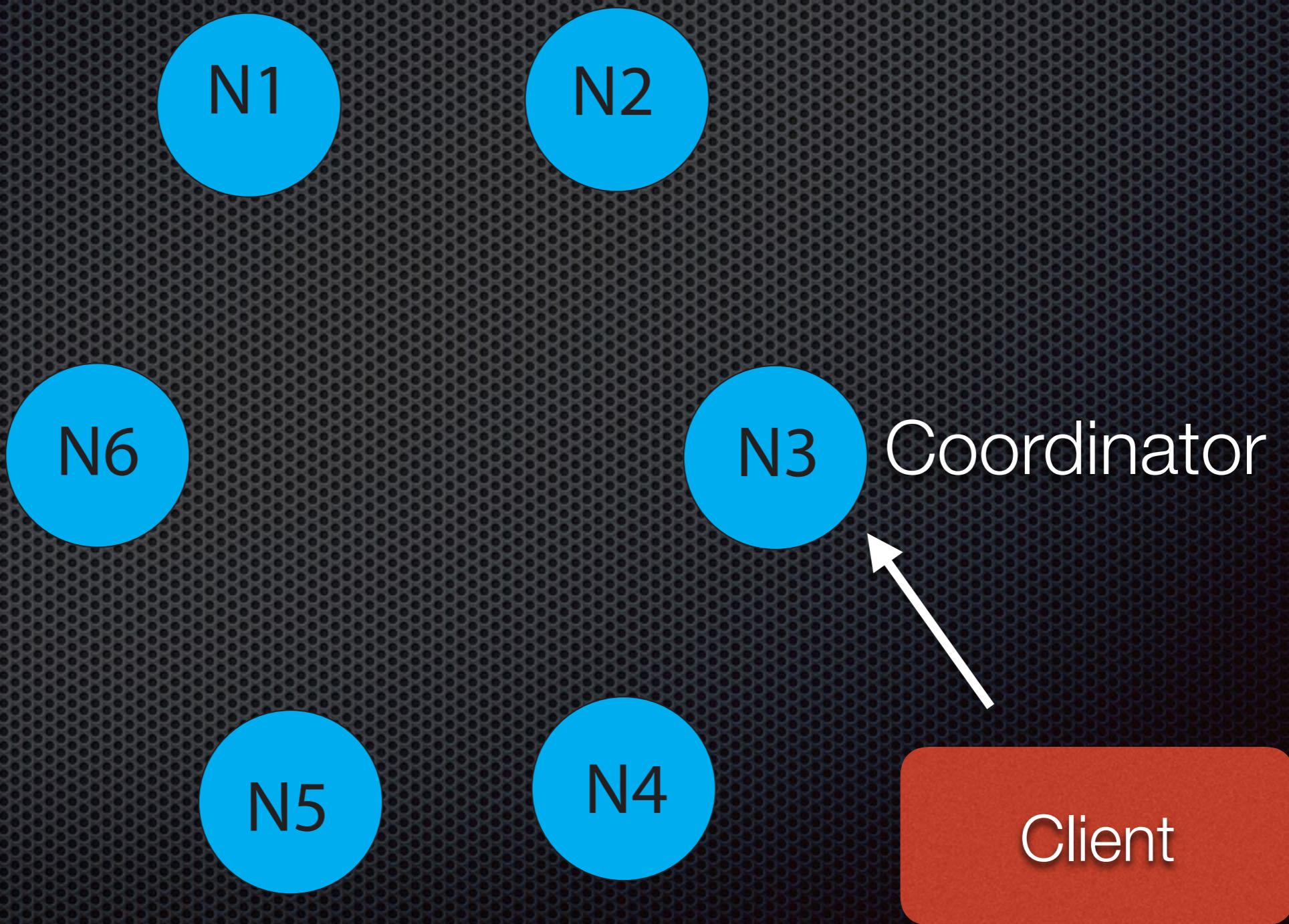
Read (consistency two)



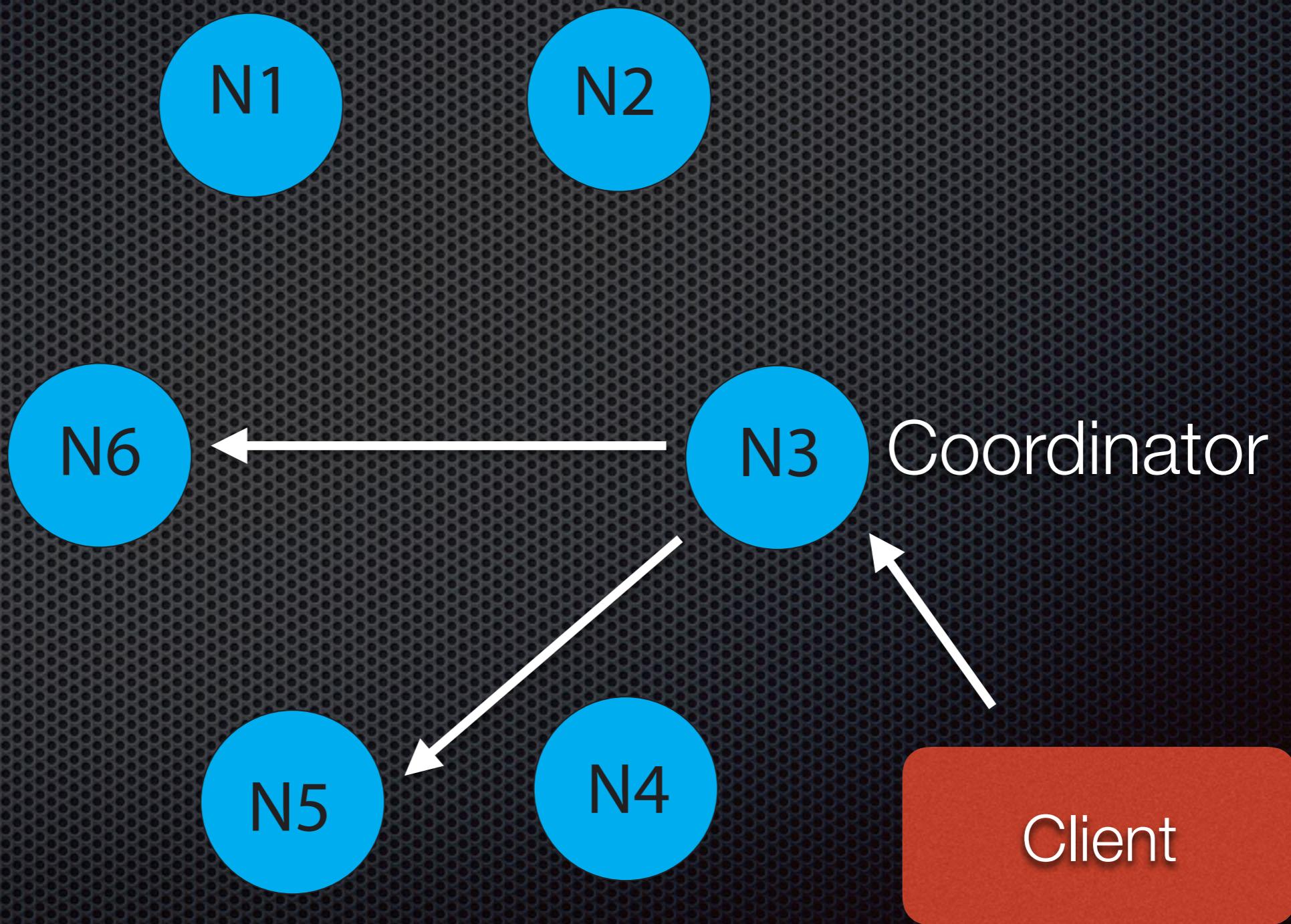
Read (consistency two)



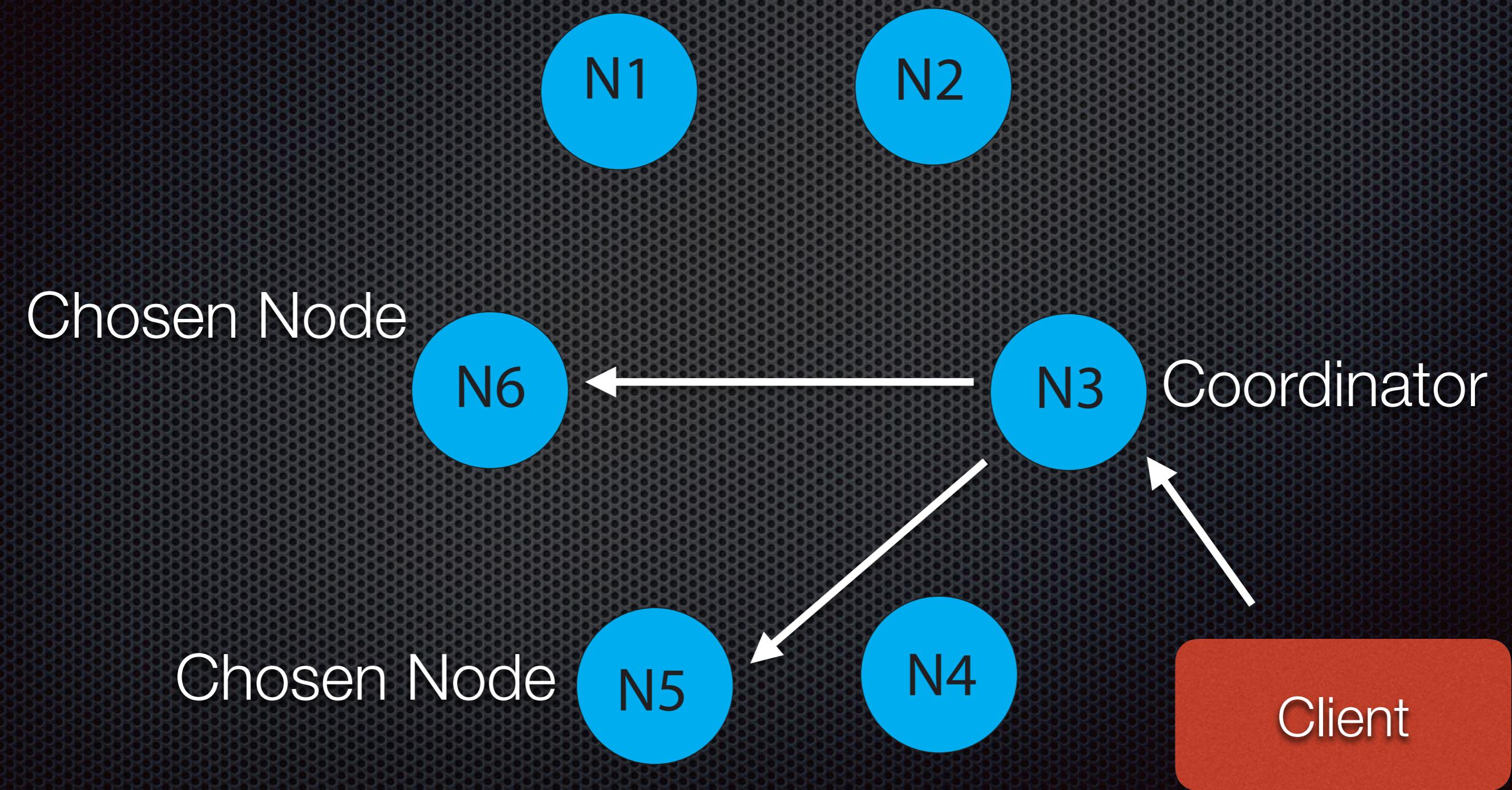
Read (consistency two)



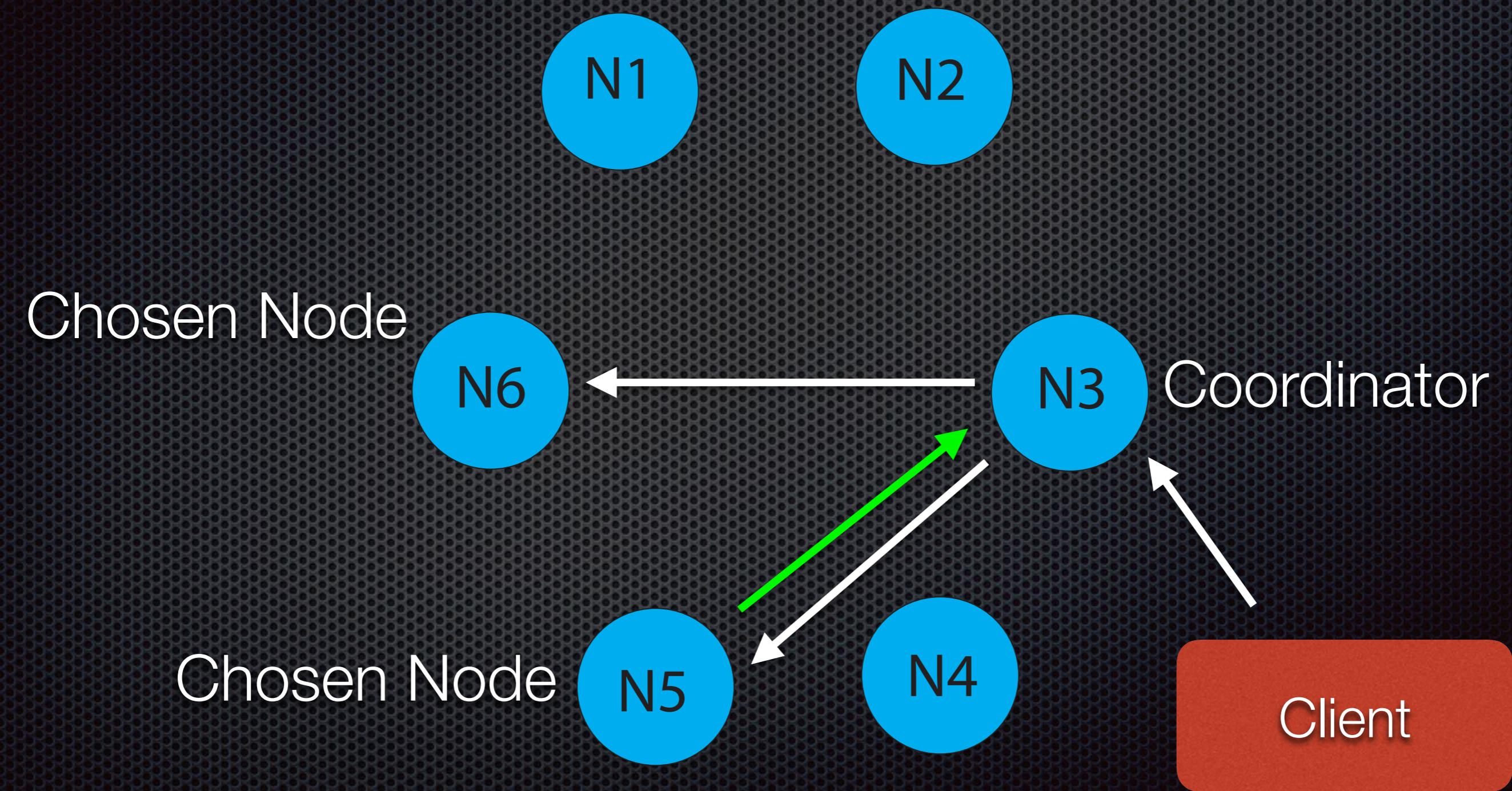
Read (consistency two)



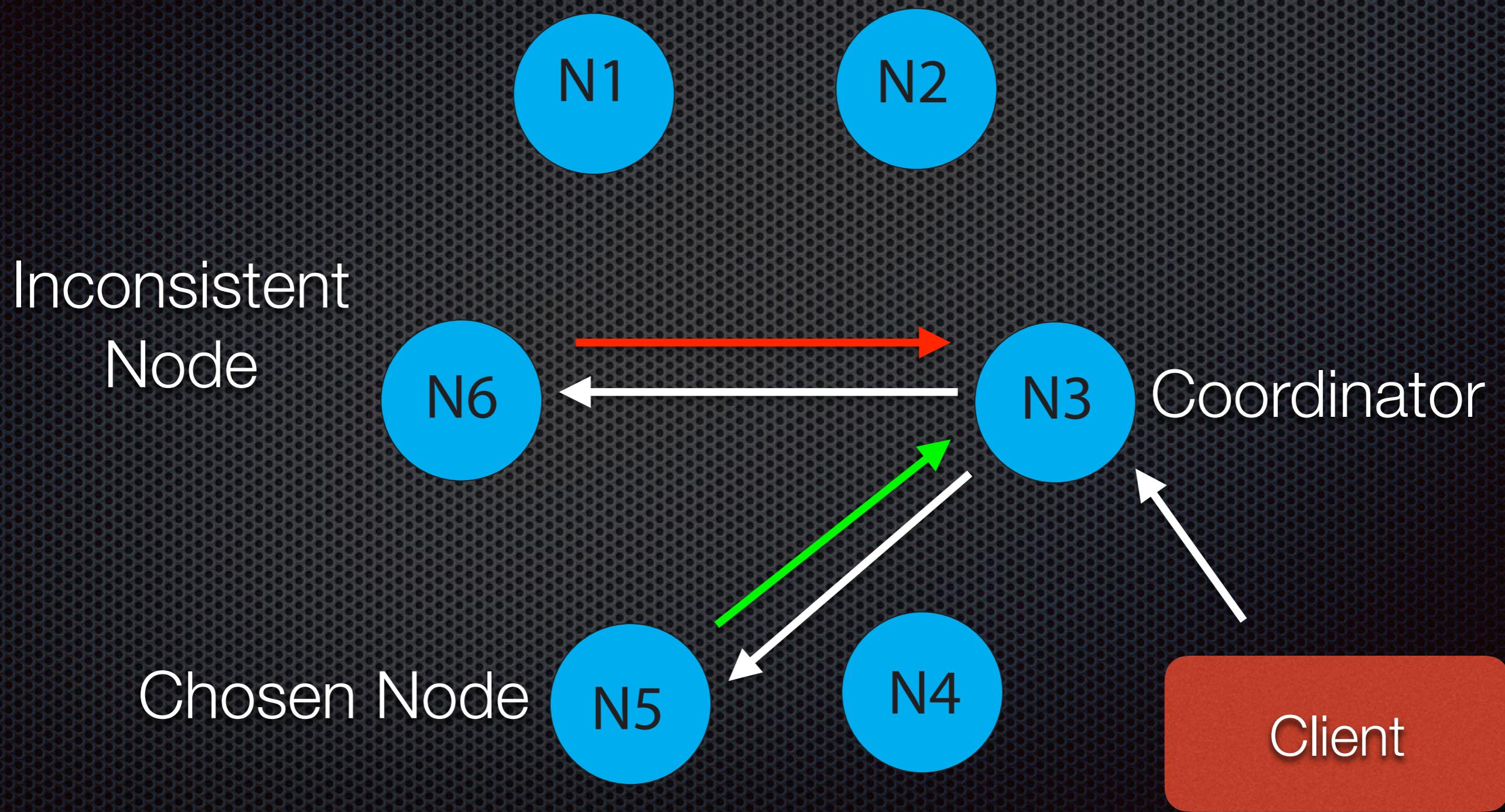
Read (consistency two)



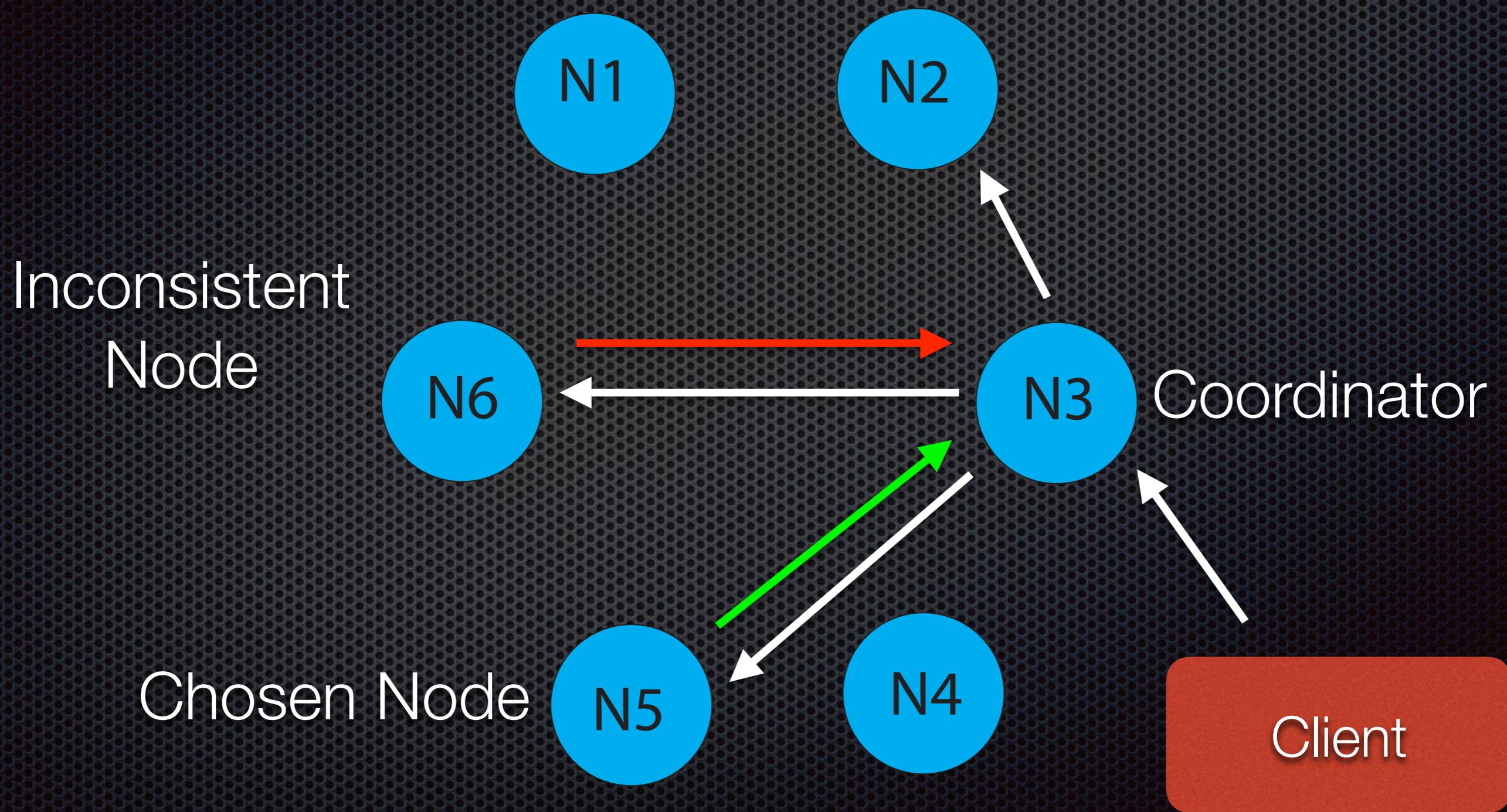
Read (consistency two)



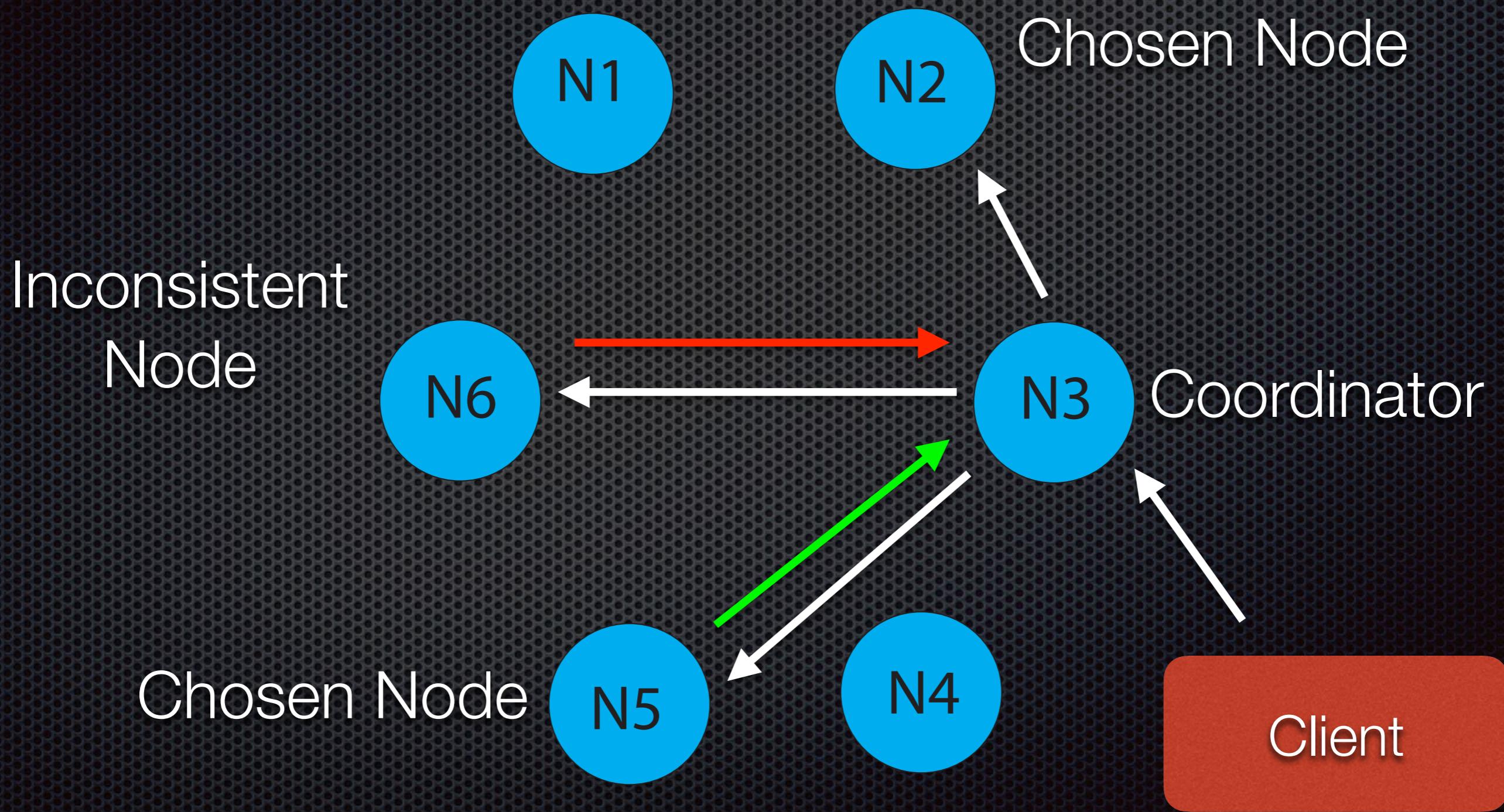
Read (consistency two)



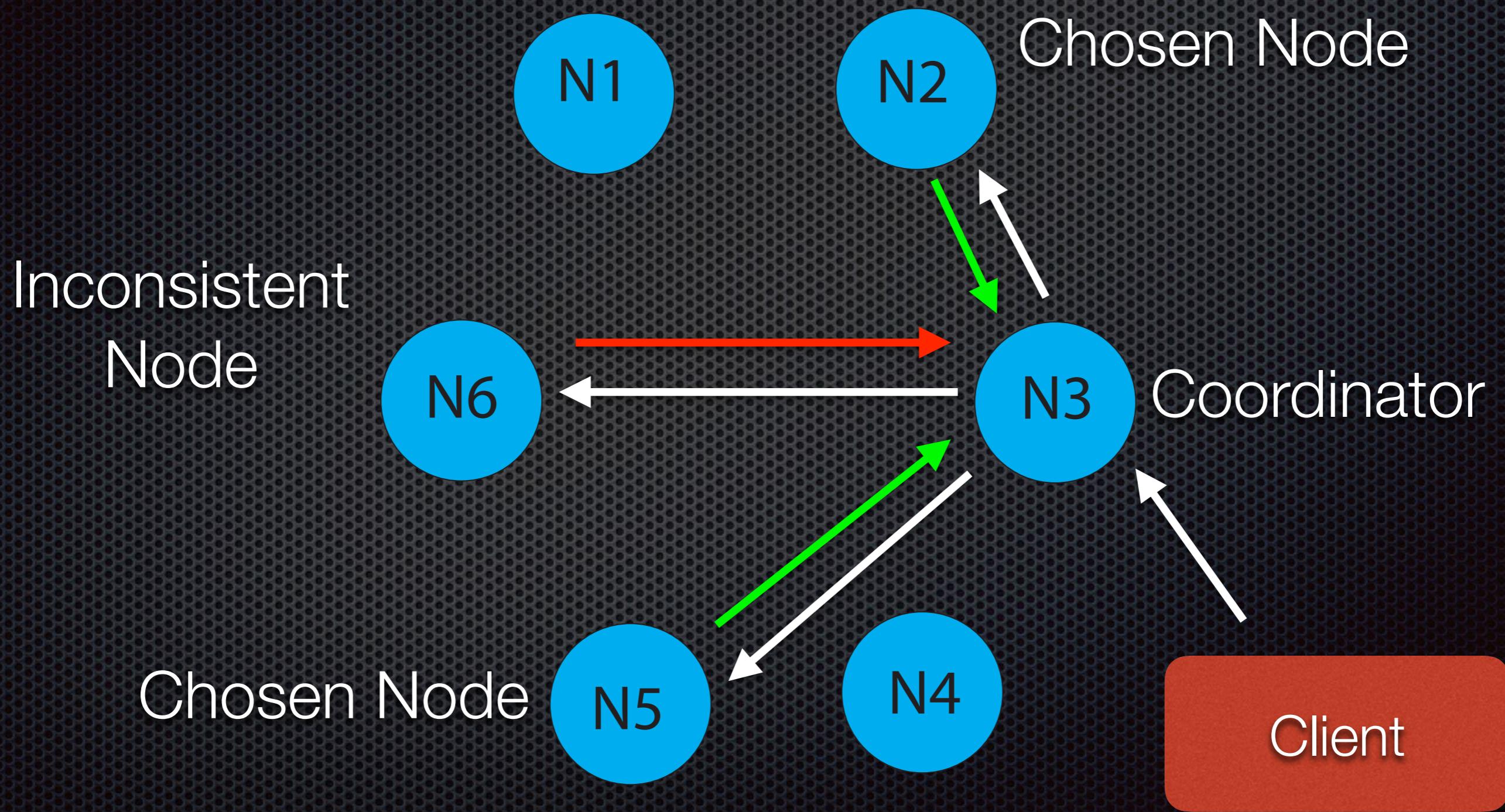
Read (consistency two)



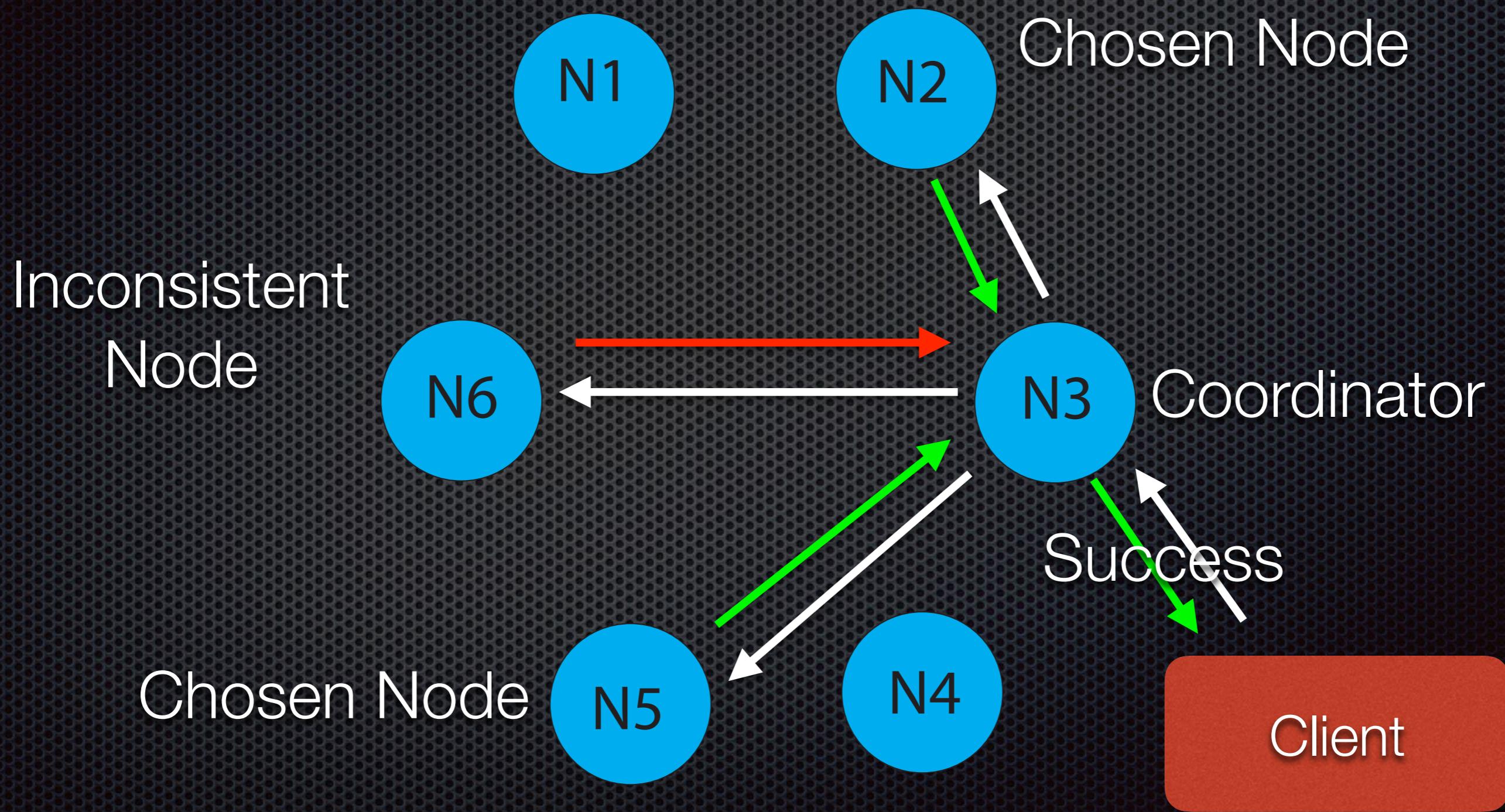
Read (consistency two)



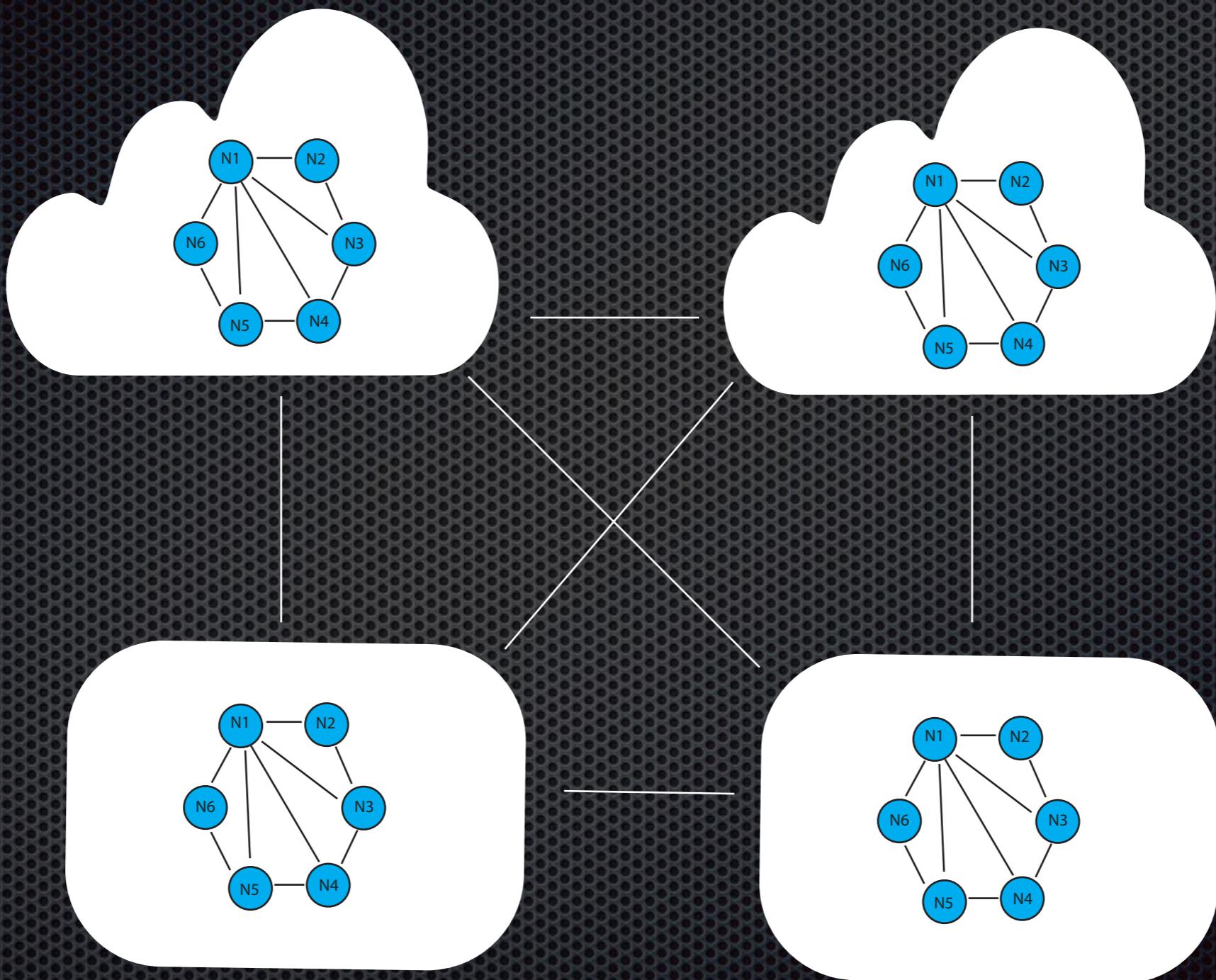
Read (consistency two)



Read (consistency two)



On-premise + Cloud



Consistency Modes

- **ALL** - Every node must have the data
- **QUORUM** - Most nodes must have the data
- **ONE** - At least one node must have the data
- **TWO** - At least two nodes must have the data
- **THREE** - At least three Nodes must have the data
- **ANY** - Any node has the data
- **EACH_QUORUM** - Each datacenter must have a quorum
- **LOCAL_QUORUM** - Each node in the datacenter handling the request must have a quorum

*Quorum = (replication_factor / 2) + 1

Calculating Consistency

$$R + W > N$$

- **R → Read level consistency**
- **W → Write level consistency**
- **N → Number of replicas of the data**

Data Replication

- **SimpleStrategy** - Single data center
- **NetworkTopologyStrategy** - Recommended strategy for multiple data centers. Provides Cassandra with info about the location of nodes by rack and datacenter

Deploying Cassandra with Docker Demo

Additional Resources

- Apache Cassandra
<http://cassandra.apache.org/>
- Docker
<https://www.docker.com/what-docker>
- Docker Hub
<https://hub.docker.com/explore/>
- Cassandra for Developers - Paul O'Fallon
<https://www.pluralsight.com/courses/cassandra-developers>
- DBA's Guid to NoSQL: Apache Cassandra (Free eBook)
<http://is.gd/CassandraFreeEbook>
- Cassandra 3.0 - DataStax PDF
<http://docs.datastax.com/en/cassandra/3.0/pdf/cassandra30.pdf>

Questions?