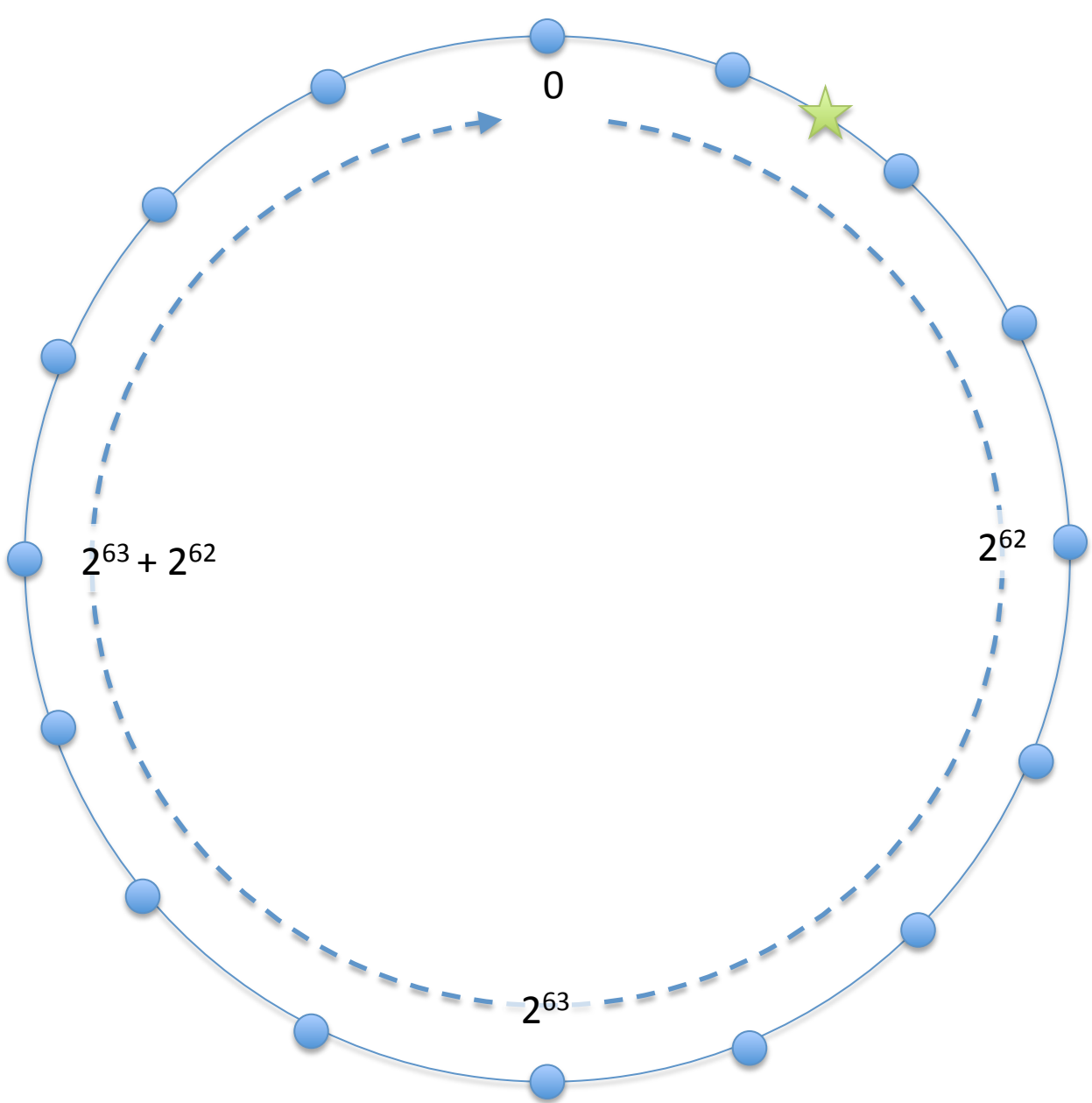# A Robust Distributed Hash Table
based on Chord

Amit Levy, Jeff Prouty, Rylan Hawkins

donut

Chord

- Overlay Network
- Stoica, Morris, Karger, Kaashoek, and Balakrishnan 2001 Paper
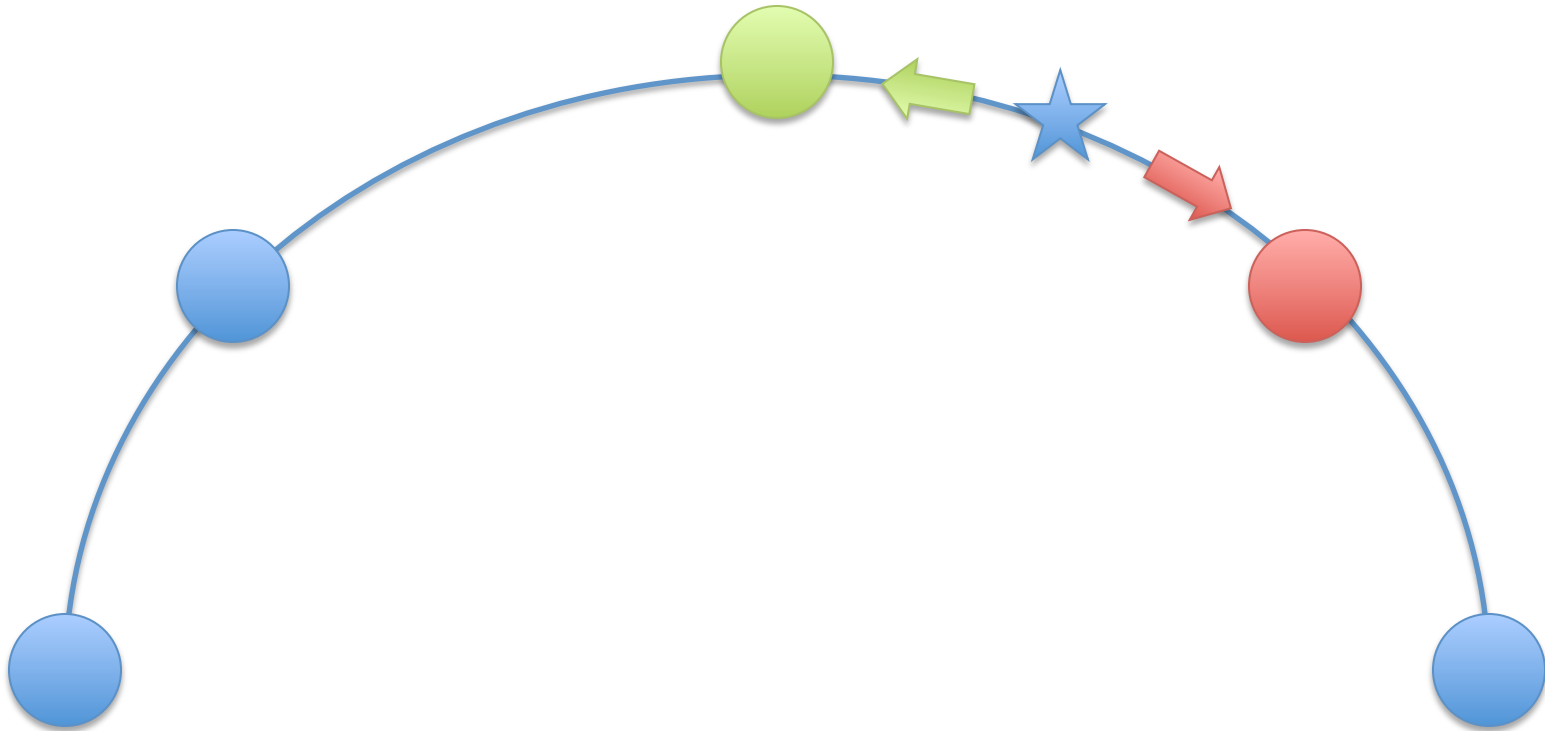- Scalable
- Nodes in a "ring" of keys

0

$2^{63} + 2^{62}$
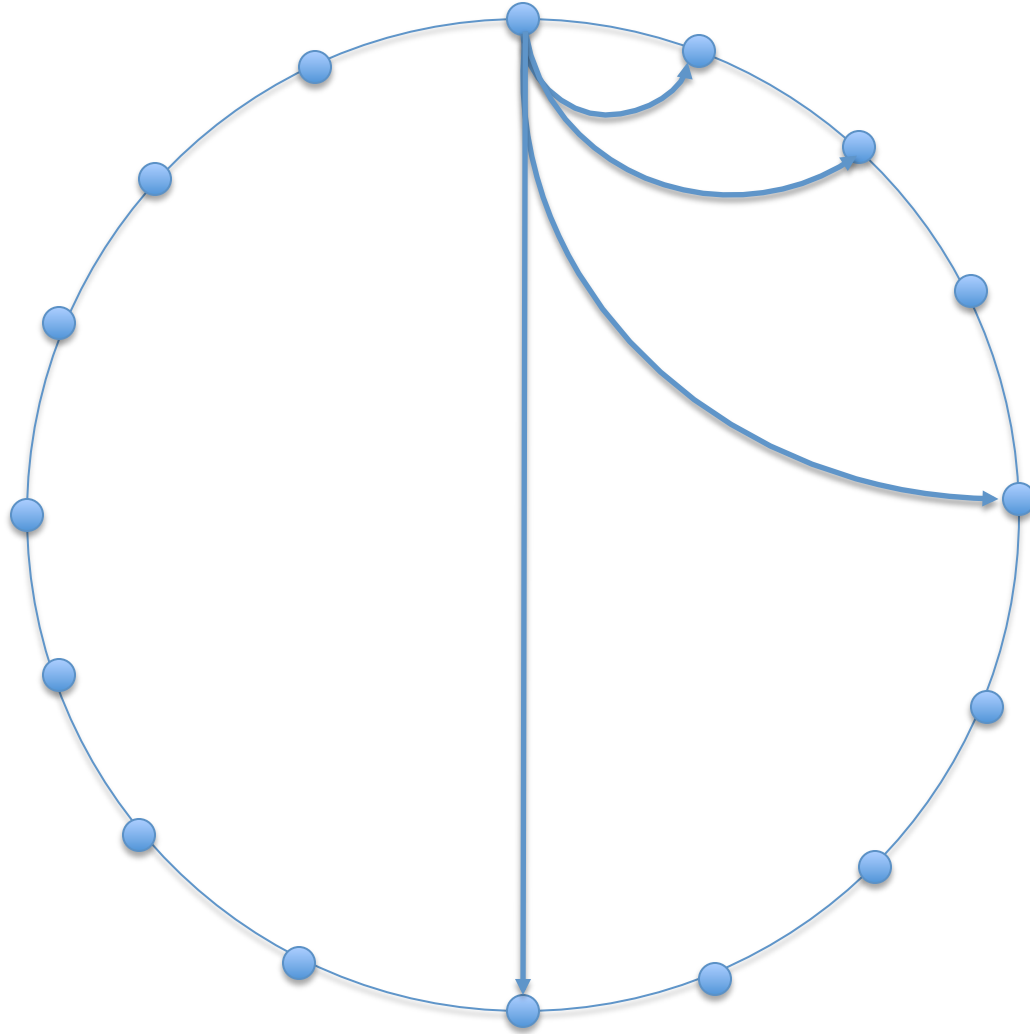
2^{62}

2^{63}
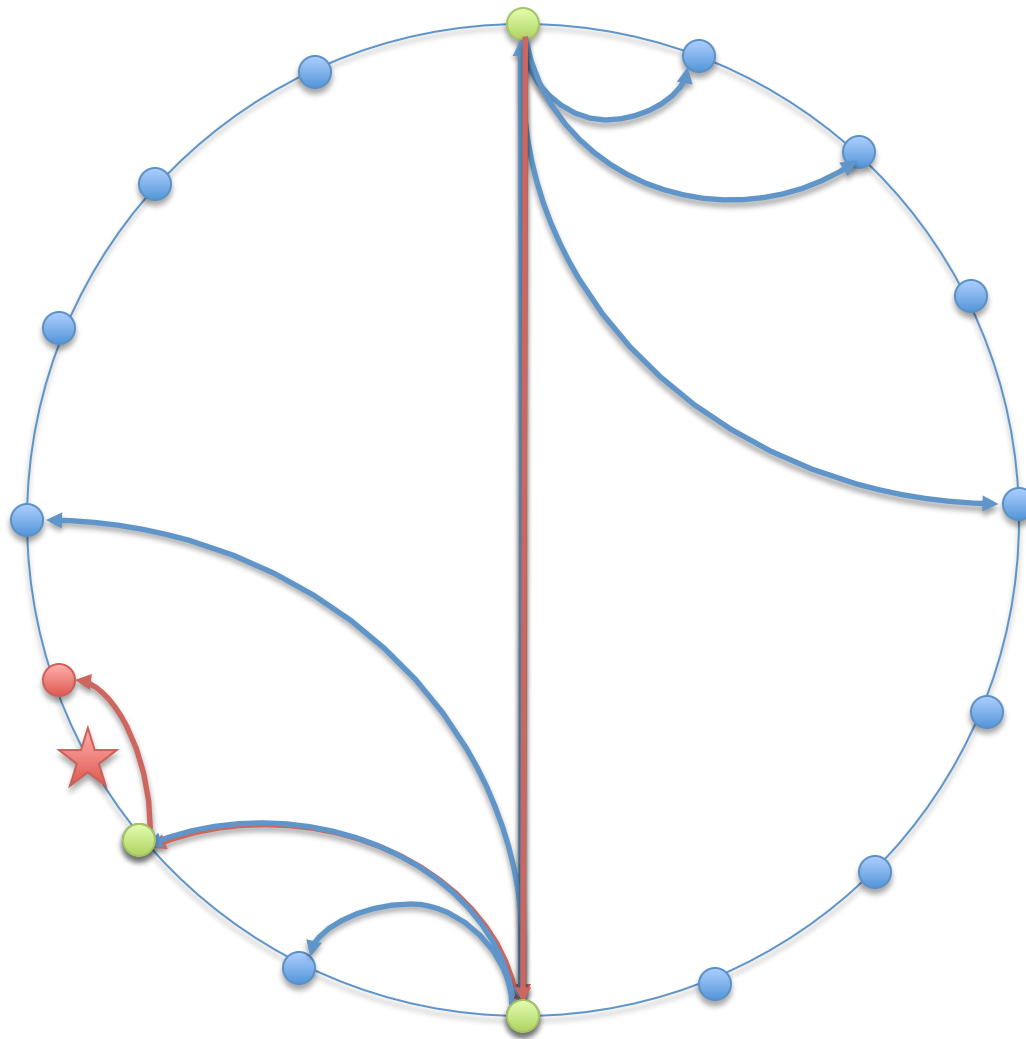
node

key

Successor and Predecessor

Fingers

- f(n, i) = *findSuccessor*(n.k + 2^i)
- 64 fingers
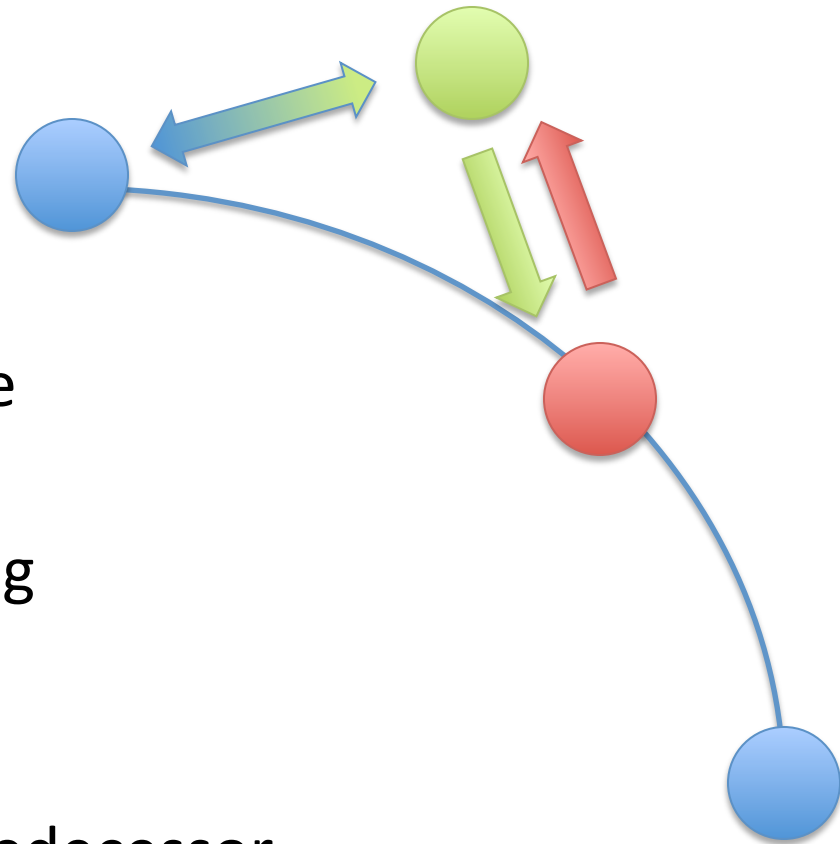- Routing table
- Analogous to binary search

Fingers

findSuccessor()
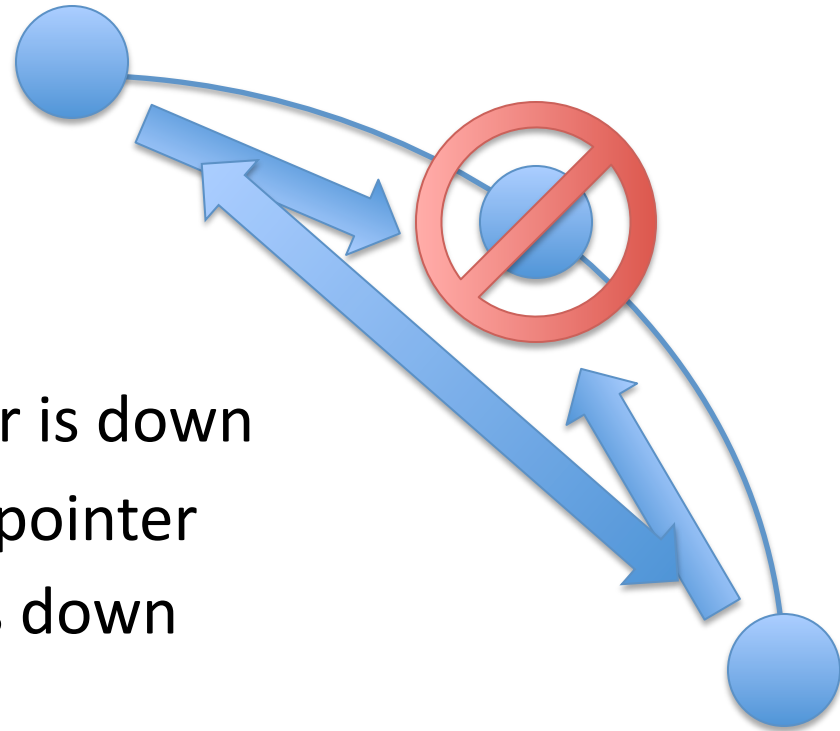
Join

Bootstrap using existing node

Process
1. Find location within ring

    n.findSuccessor(n')
2. Notify new successor

    new successor sets predecessor
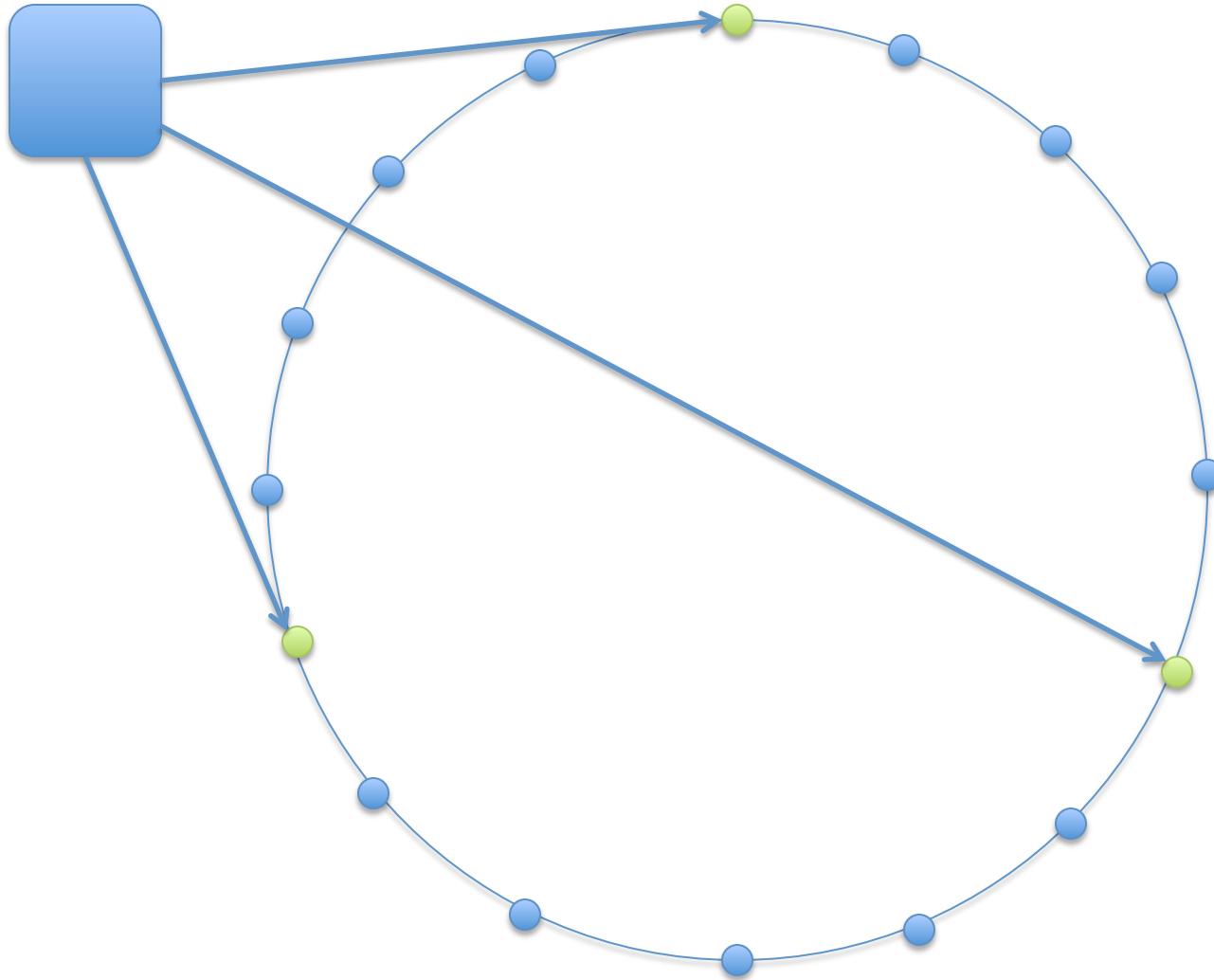3. Predecessor finds new node

Leaves

1. Node notices predecessor is down
   Invalidates predecessor pointer
2. Node notices successor is down
   Pops successor off list
3. Predecessor notifies new successor
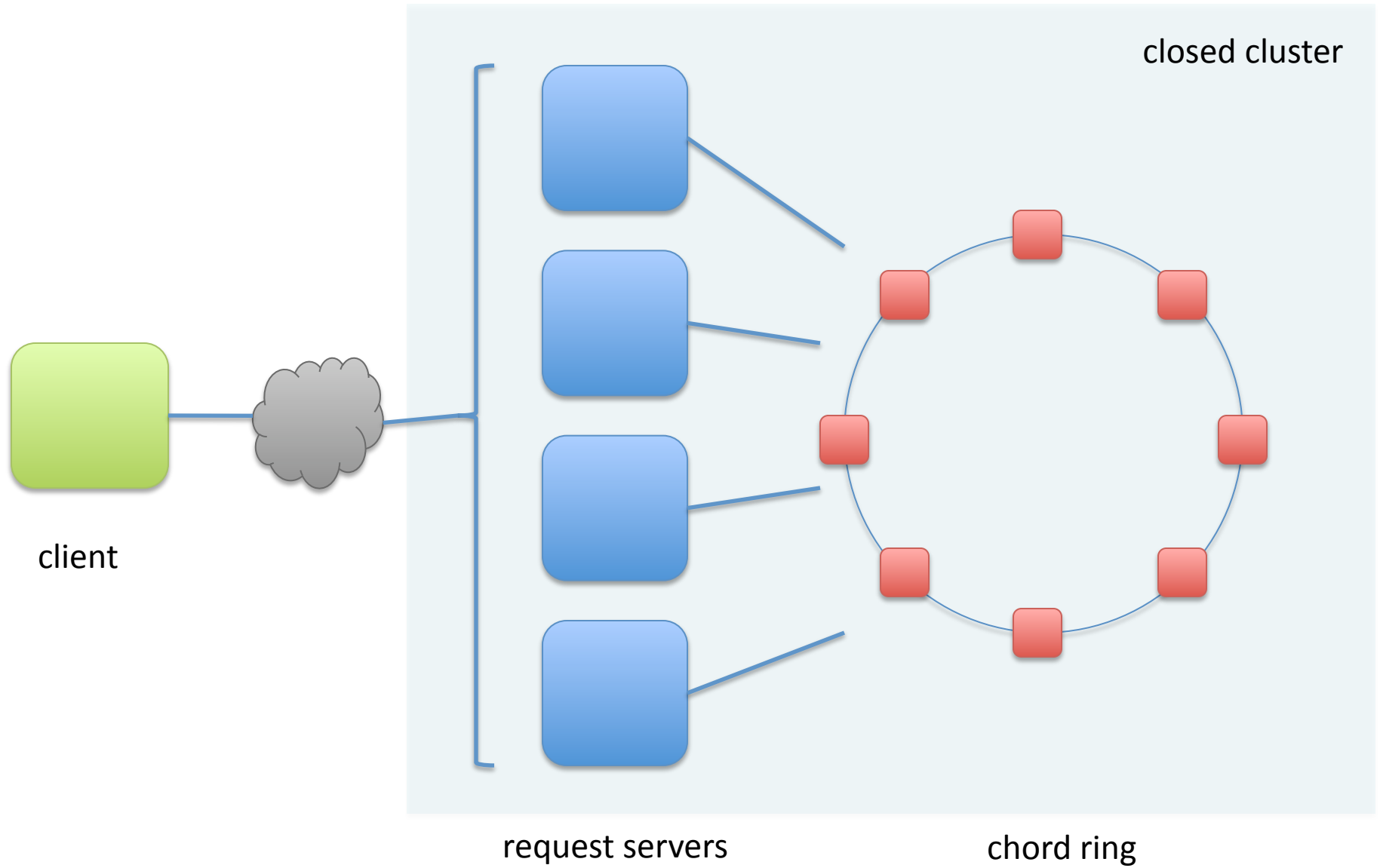   New successor fixes predecessor

Chord Replication
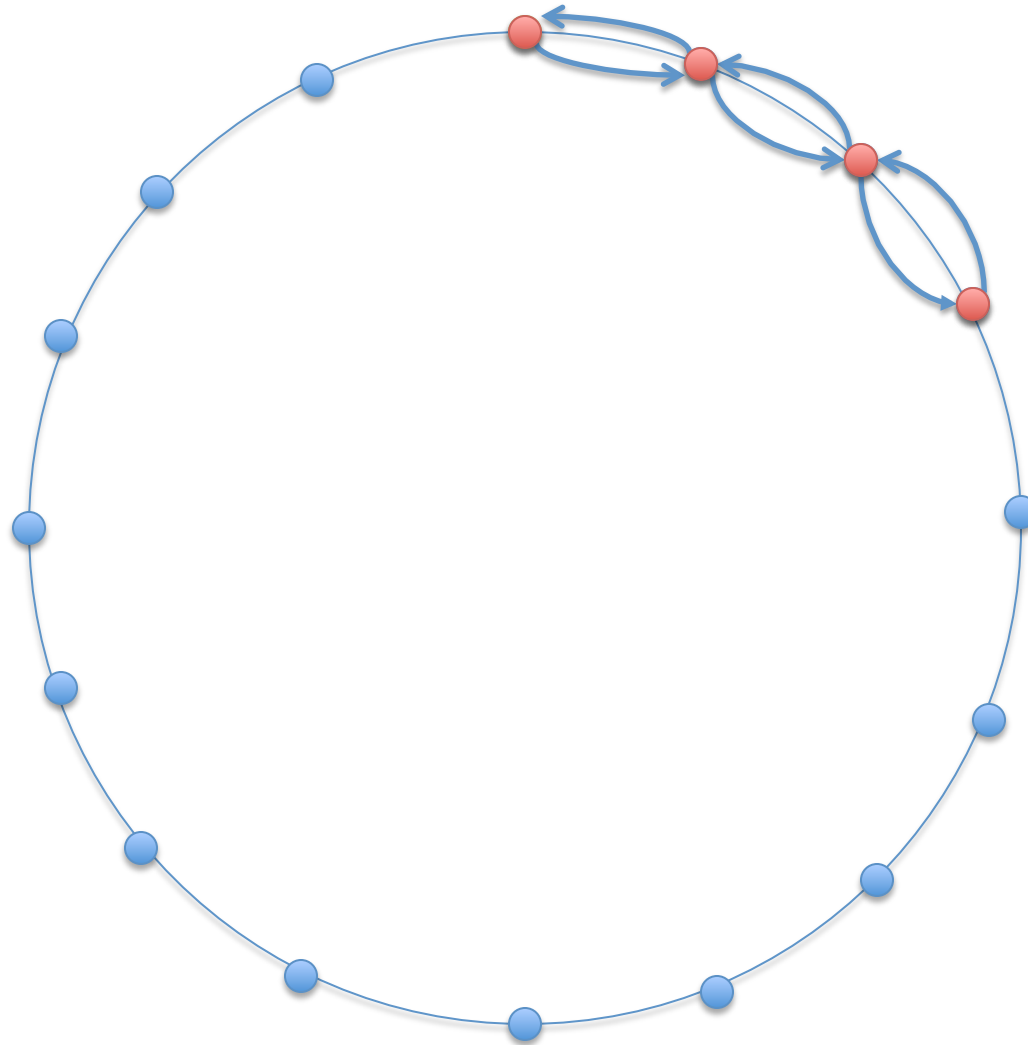
Service Replication

Our Architecture

closed cluster

client

request servers

chord ring

Conclusions

- Initial File System became Robust DHT
- Scope was unclear early
- Mentors focused us