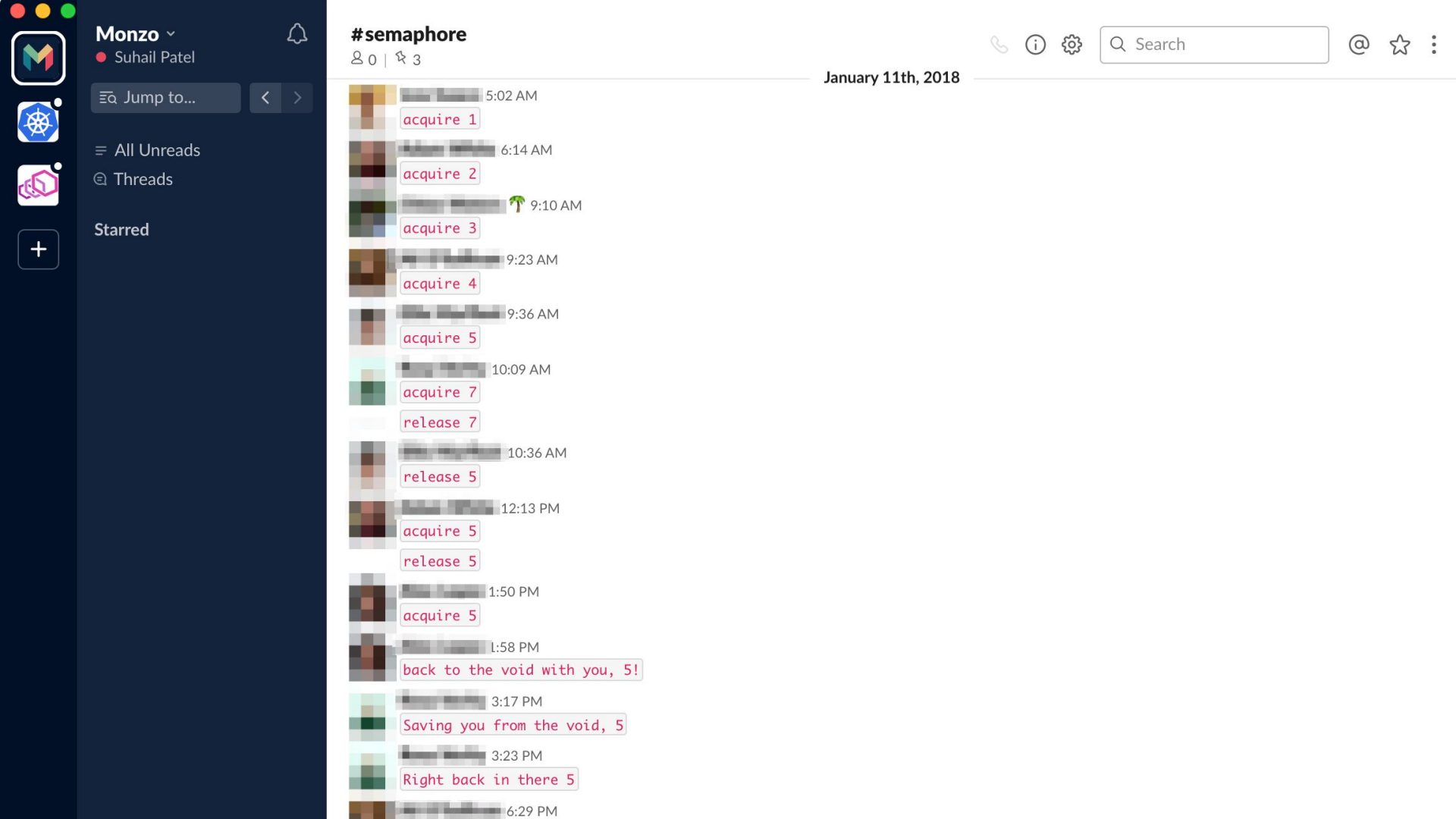
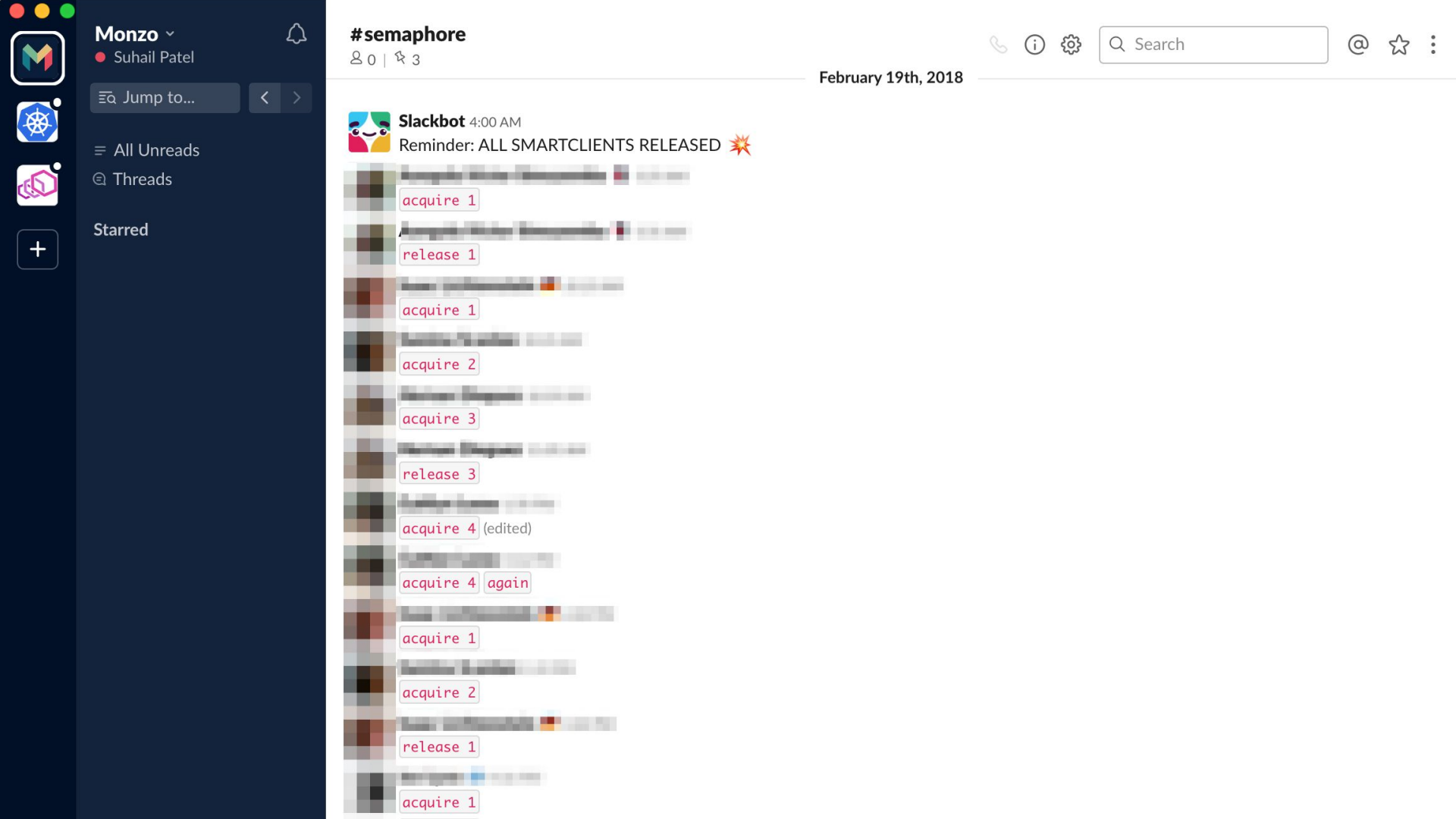
Eventually Consistent Service Discovery

@suhailpatel - SRECon 2019


Monzo
Suhail Patel
Jump to...
All Unreads
Threads
Starred
#semaphore
0 | 3
Search
January 11th, 2018
5:02 AM
acquire 1
6:14 AM
acquire 2
9:10 AM
acquire 3
9:23 AM
acquire 4
9:36 AM
acquire 5
10:09 AM
acquire 7
release 7
10:36 AM
release 5
12:13 PM
acquire 5
release 5
1:50 PM
acquire 5
1:58 PM
back to the void with you, 5!
3:17 PM
Saving you from the void, 5
3:23 PM
Right back in there 5
6:29 PM


Monzo
Suhail Patel
Jump to...
All Unreads
Threads
Starred
#semaphore
0 | 3
Search
February 19th, 2018
Slackbot 4:00 AM
Reminder: ALL SMARTCLIENTS RELEASED
acquire 1
release 1
acquire 1
acquire 2
acquire 3
release 3
acquire 4 (edited)
acquire 4 again
acquire 1
acquire 2
release 1
acquire 1

Eventually Consistent Service Discovery

@suhailpatel - SRECon 2019

Hi, i’m Suhail

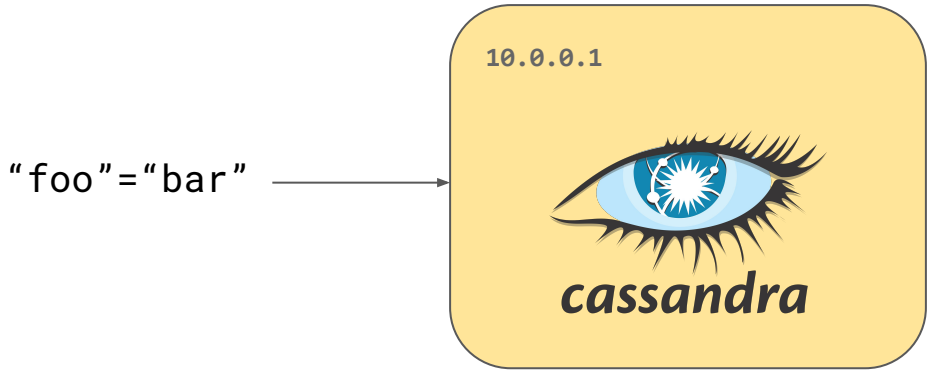
I’m an Engineer at Monzo on the Platform squad.
We help build the base so other engineers can
ship their services and applications.

Email: **hi@suhailpatel.com**
Twitter: **@suhailpatel**

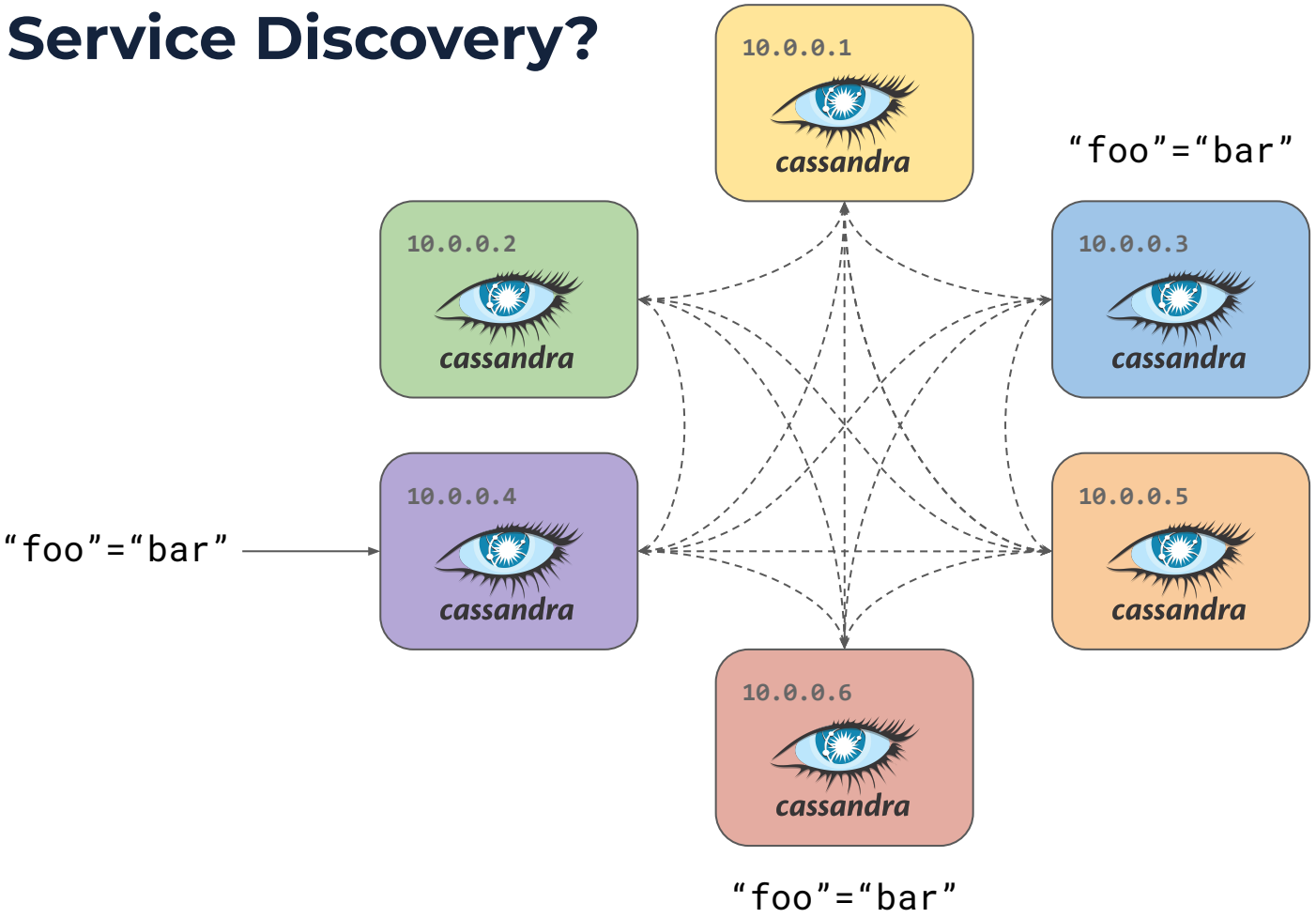

monzo
5354 2000 0000 0000
EXP
END 08/1
469-624-882

Service Discovery?

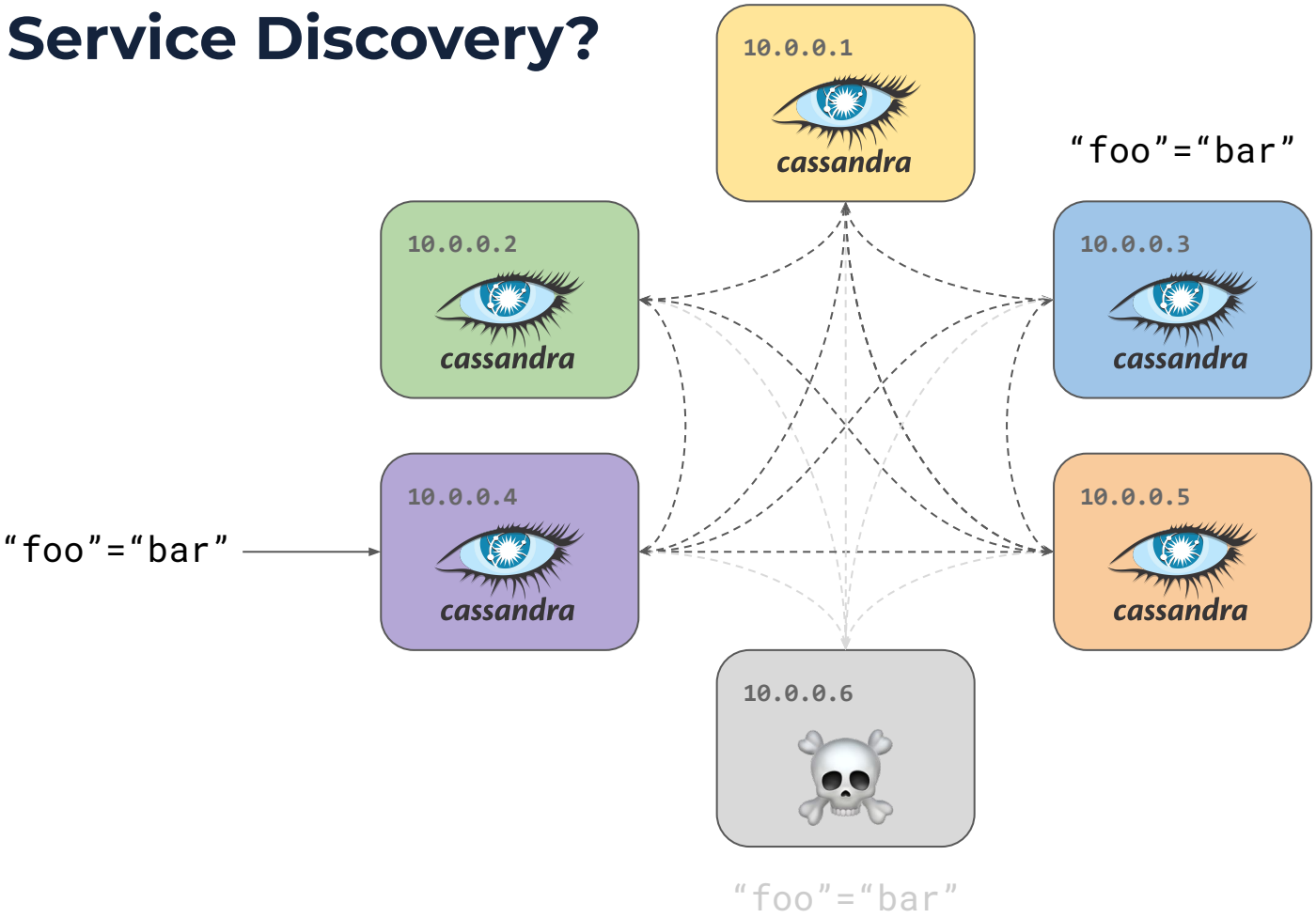
Service Discovery?



Service Discovery?



Service Discovery?



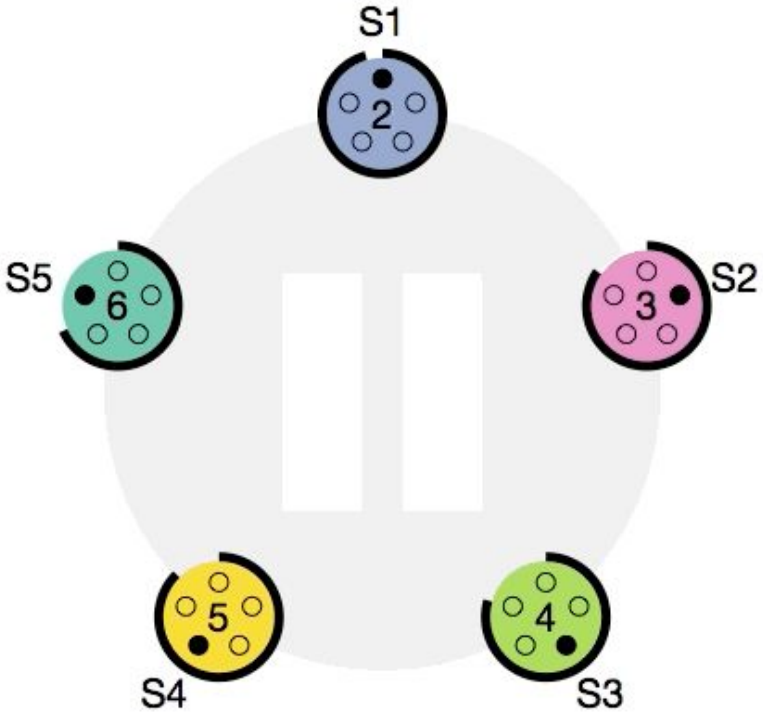
Strong Consistency

https://etcd.io/

Strong Consistency

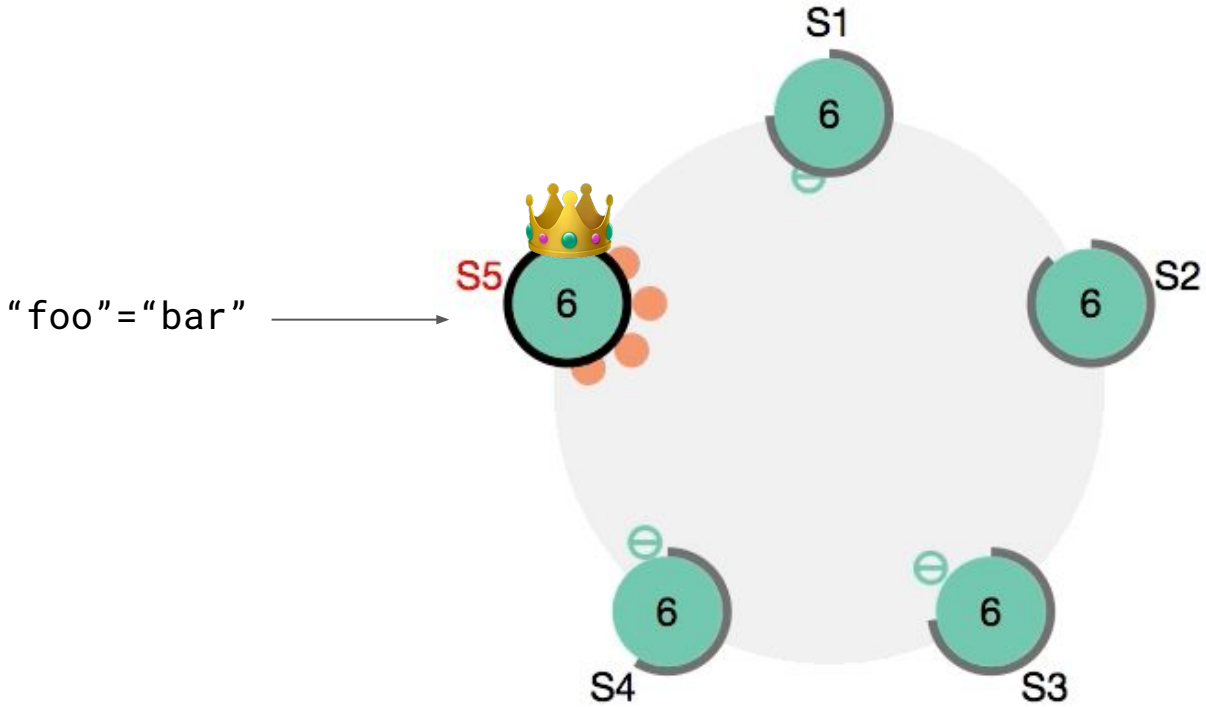
https://raft.github.io

Strong Consistency

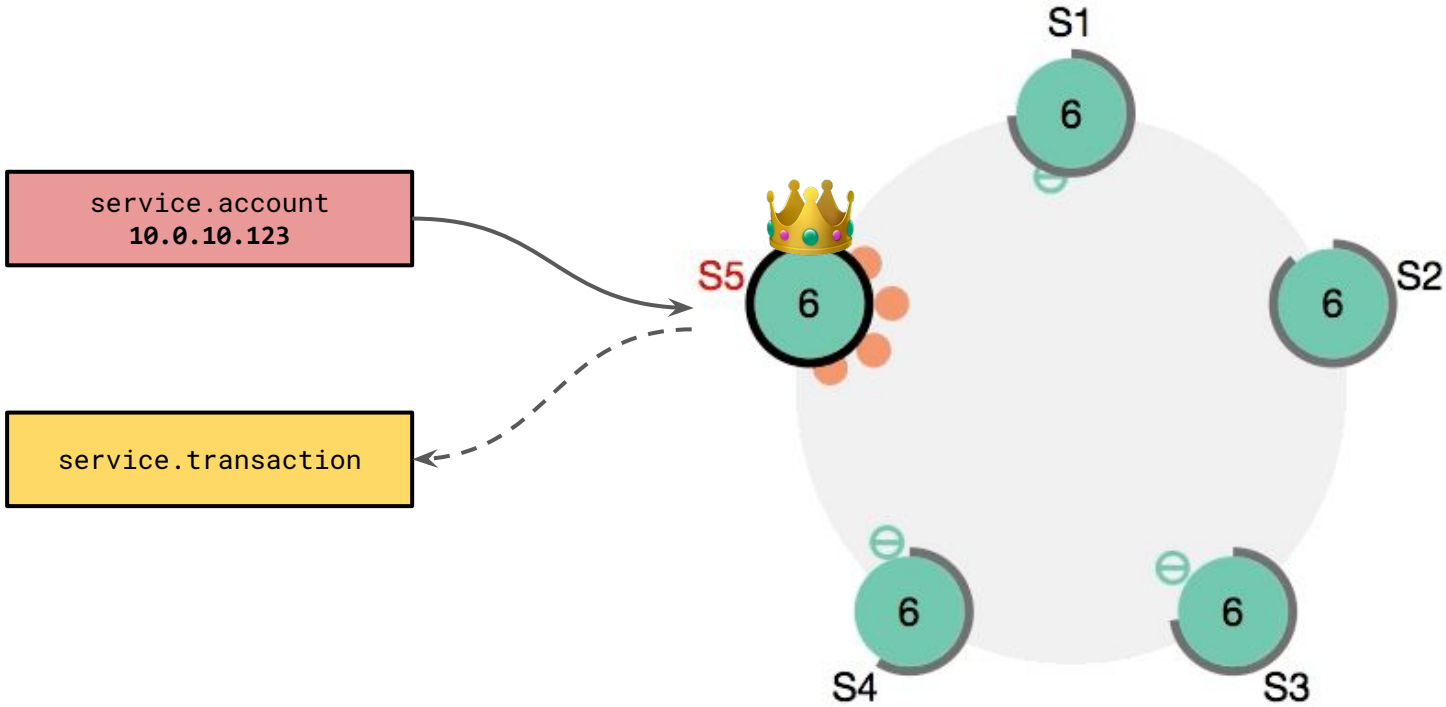


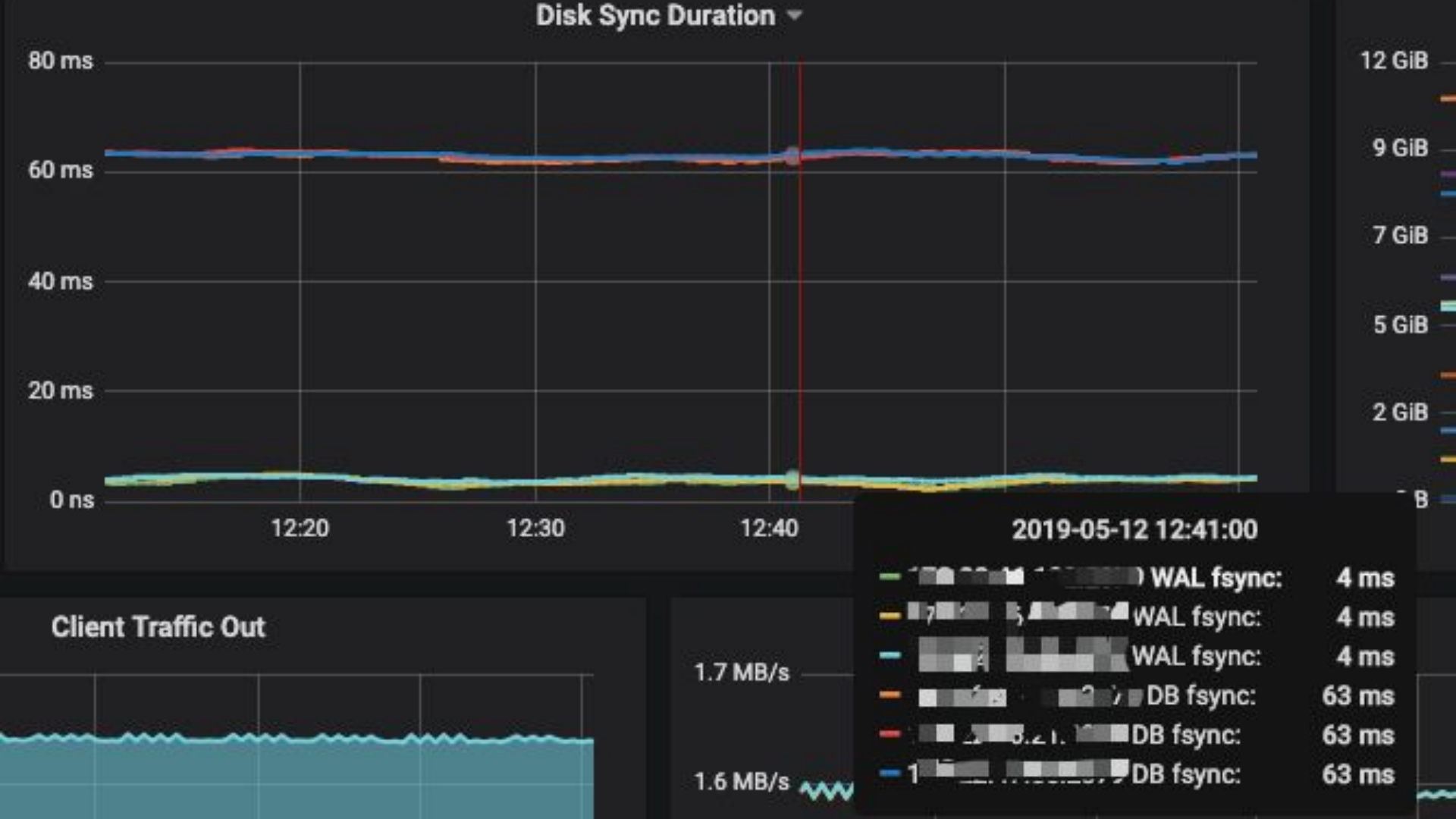
https://raft.github.io

Strong Consistency



Strong Consistency

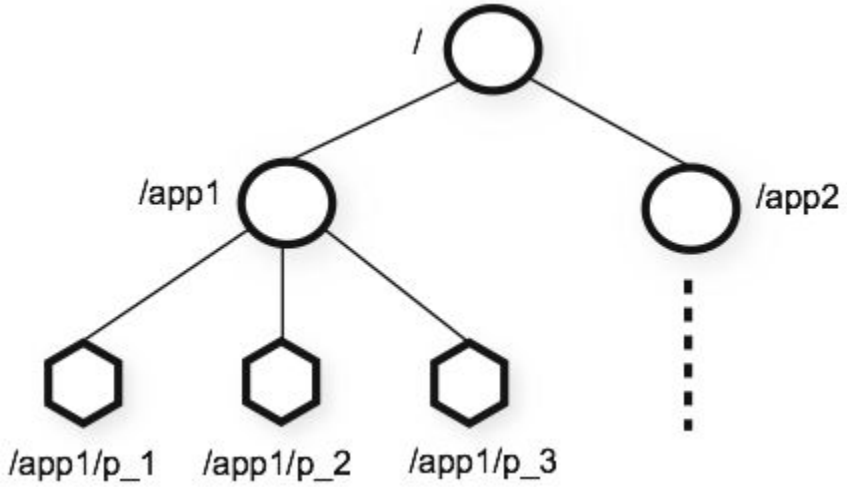



Disk Sync Duration
80 ms
60 ms
40 ms
20 ms
0 ns
12:20
12:30
12:40
12 GiB
9 GiB
7 GiB
5 GiB
2 GiB
2019-05-12 12:41:00
WAL fsync: 4 ms
WAL fsync: 4 ms
WAL fsync: 4 ms
DB fsync: 63 ms
DB fsync: 63 ms
DB fsync: 63 ms
Client Traffic Out
1.7 MB/s
1.6 MB/s

Sequential Consistency

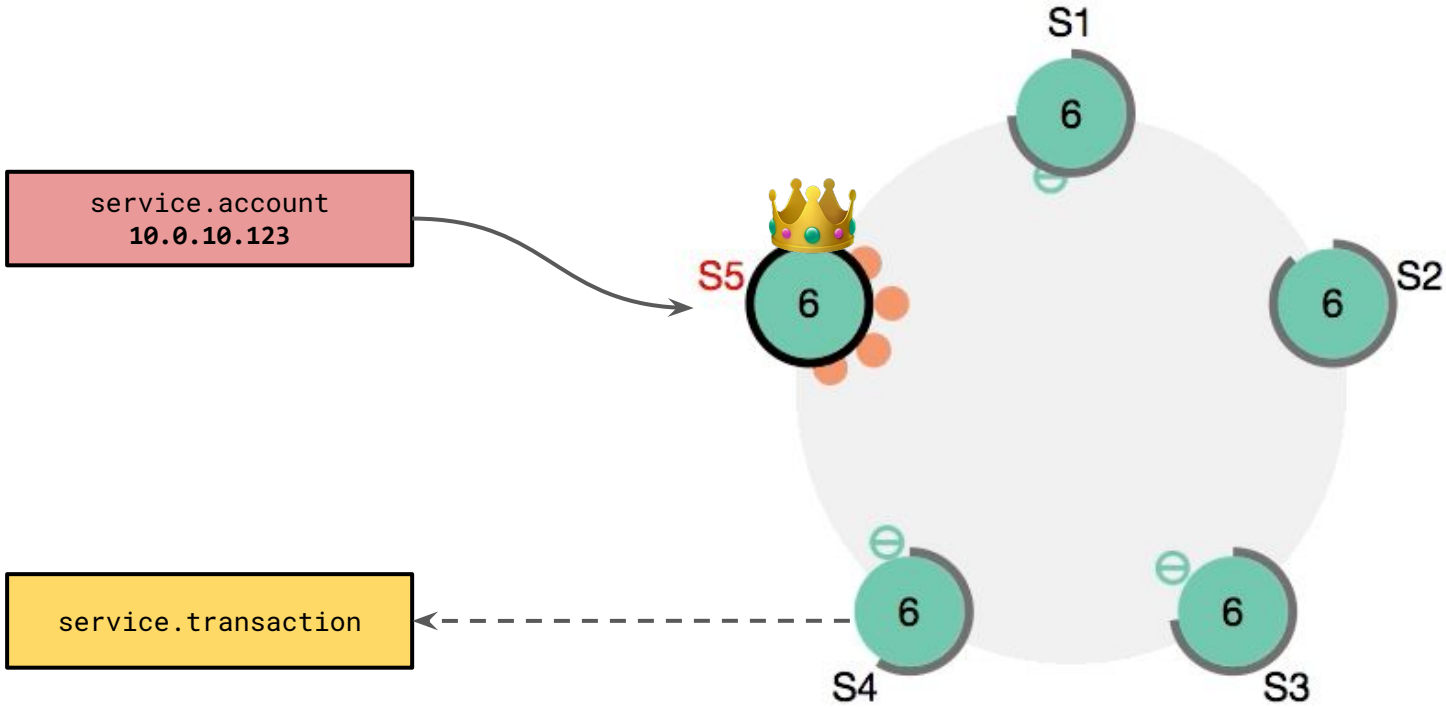
APACHE ZooKeeper™

Sequential Consistency



https://zookeeper.apache.org/doc/r3.1.2/zookeeperOver.html

Sequential Consistency



usenix
THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

SRECON AMERICAS

Open Access Sponsor

salesforce

Summary

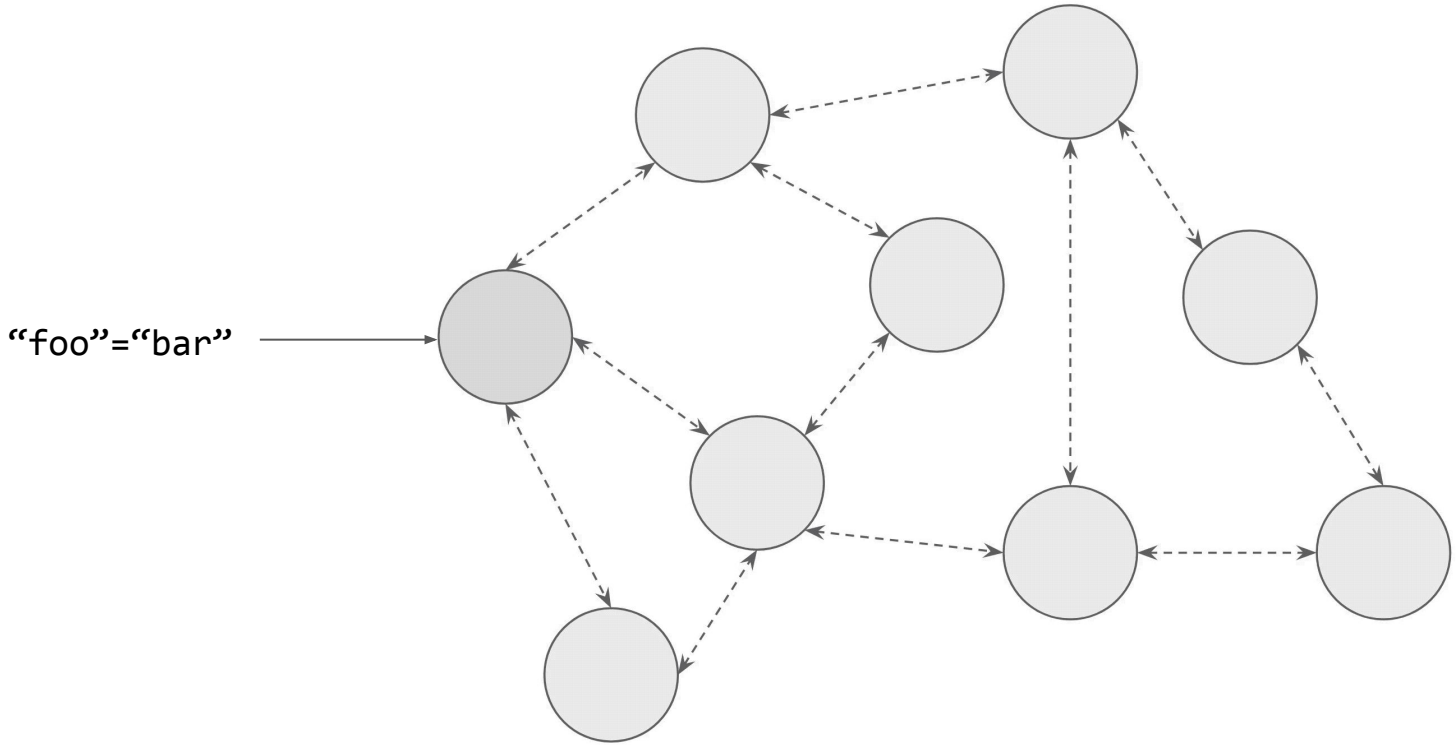
Remaining issues:

- Positive feedback loop
- Write cost depends on number of backend instances
- Read cost depends on write cost multiplied by number of clients

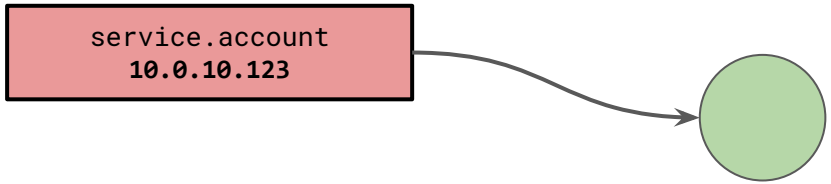
https://www.usenix.org/conference/srecon19americas/presentation/nigmatullin

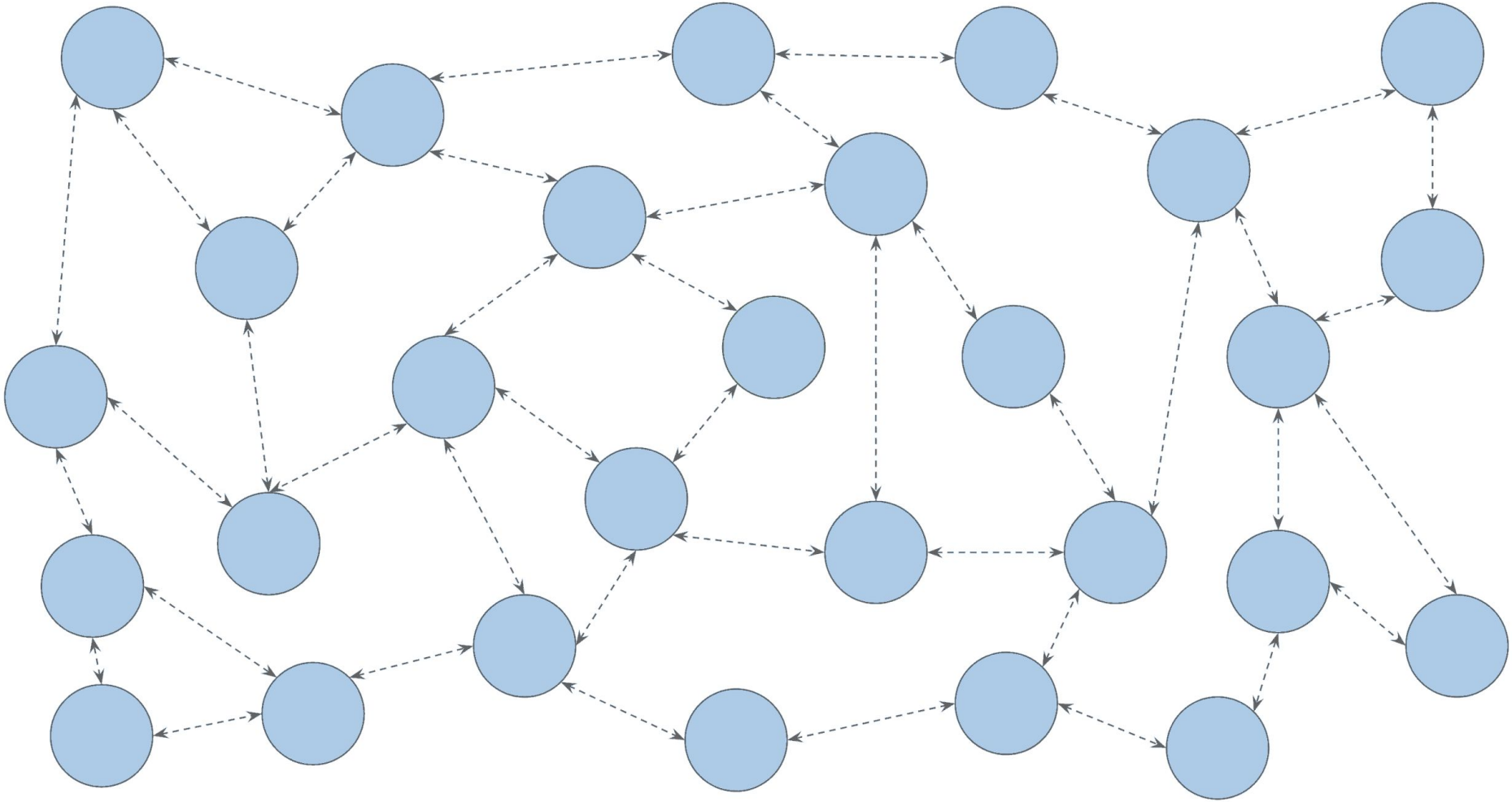

NEWS
starts cure research
9-4-2015
World
INFECTED
//4,005,270,166
DEAD
//114,463
DISEASE
39
WORLD
1%

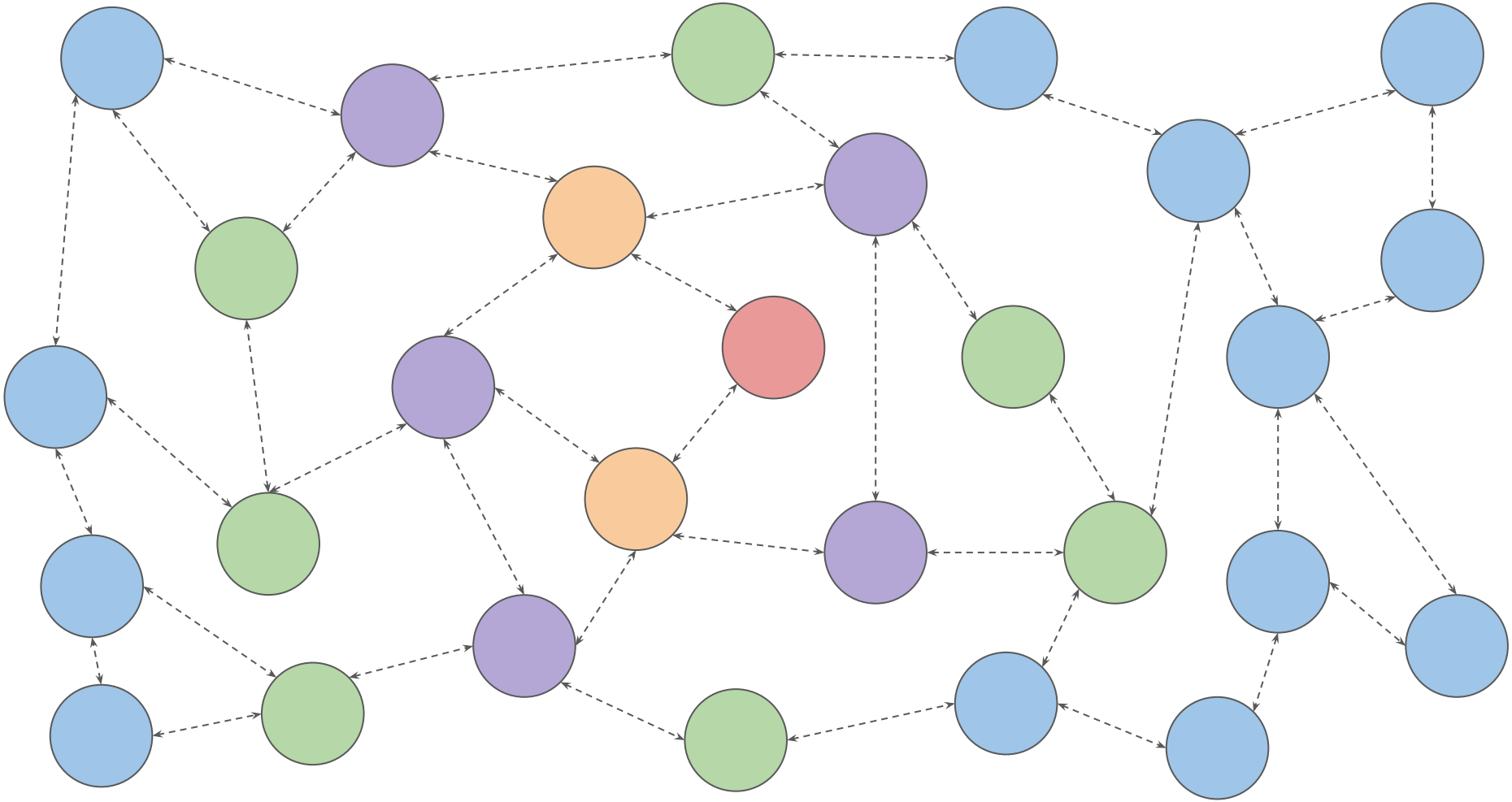
Gossip Protocols

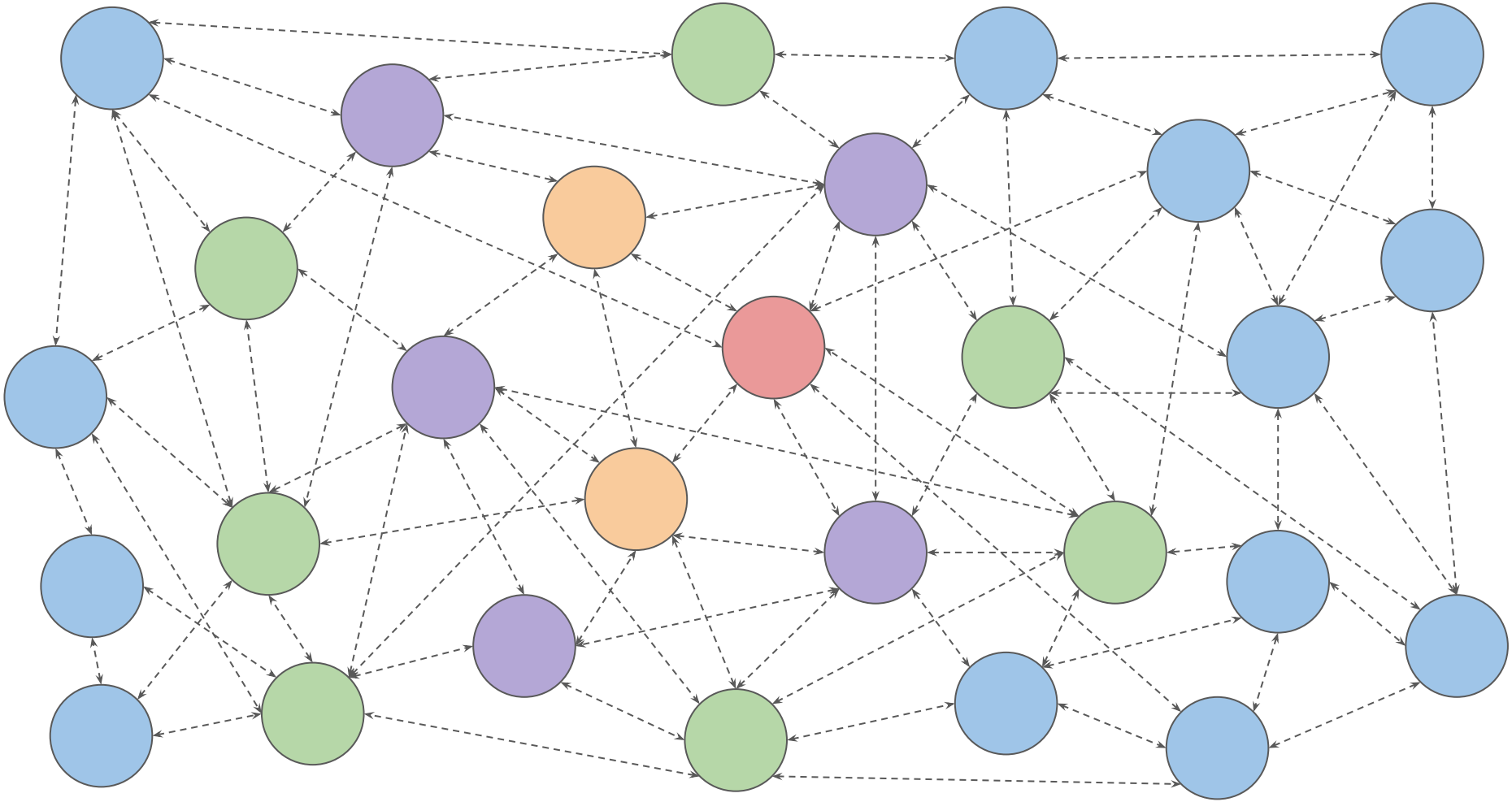


Gossip Protocols









SWIM: *S*calable *W*eakly-consistent *I*nfection-style Process Group *M*embership Protocol

Abhinandan Das, Indranil Gupta, Ashish Motivala*
Dept. of Computer Science, Cornell University
Ithaca NY 14853 USA
{asdas,gupta,ashish}@cs.cornell.edu

Abstract

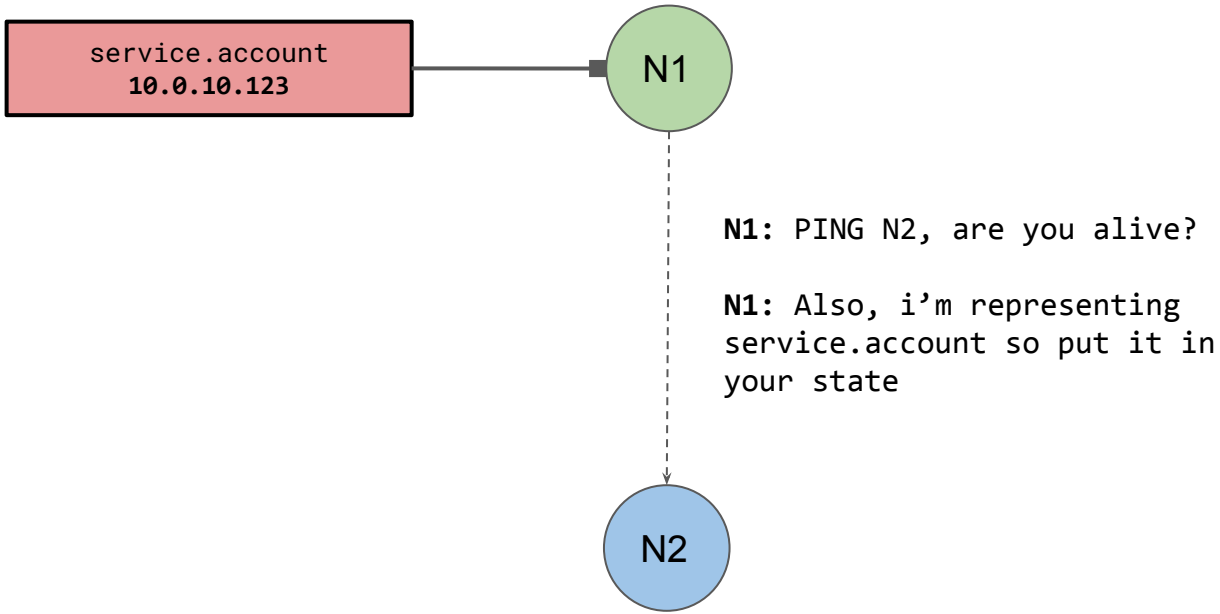
Several distributed peer-to-peer applications require weakly-consistent knowledge of process group membership information at all participating processes. SWIM is a generic software module that offers this service for large-scale process groups. The SWIM effort is motivated by the unscalability of traditional heart-beating protocols, which either impose network loads that grow quadratically with group size, or compromise response times or false positive frequency w.r.t. detecting process crashes. This paper re-

1. Introduction

As you swim lazily through the milieu,
The secrets of the world will infect you.

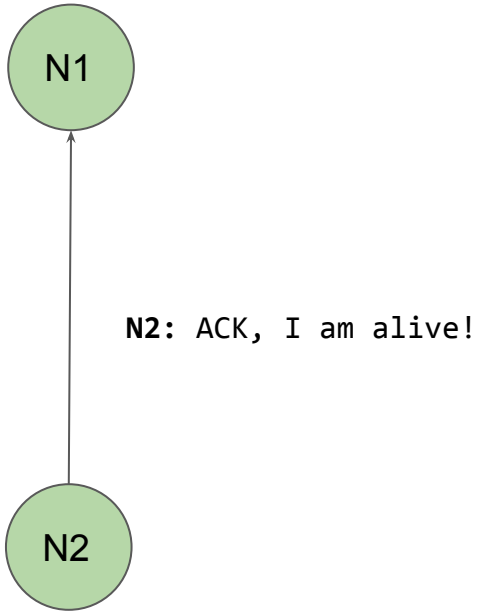
Several large-scale peer-to-peer distributed process groups running over the Internet rely on a distributed membership maintenance sub-system. Examples of existing middleware systems that utilize a membership protocol include reliable multicast [3, 11], and epidemic-style information dissemination [4, 8, 13]. These protocols in turn find use in applications such as distributed databases that need to reconcile re-

(Scalable) Gossip Protocols



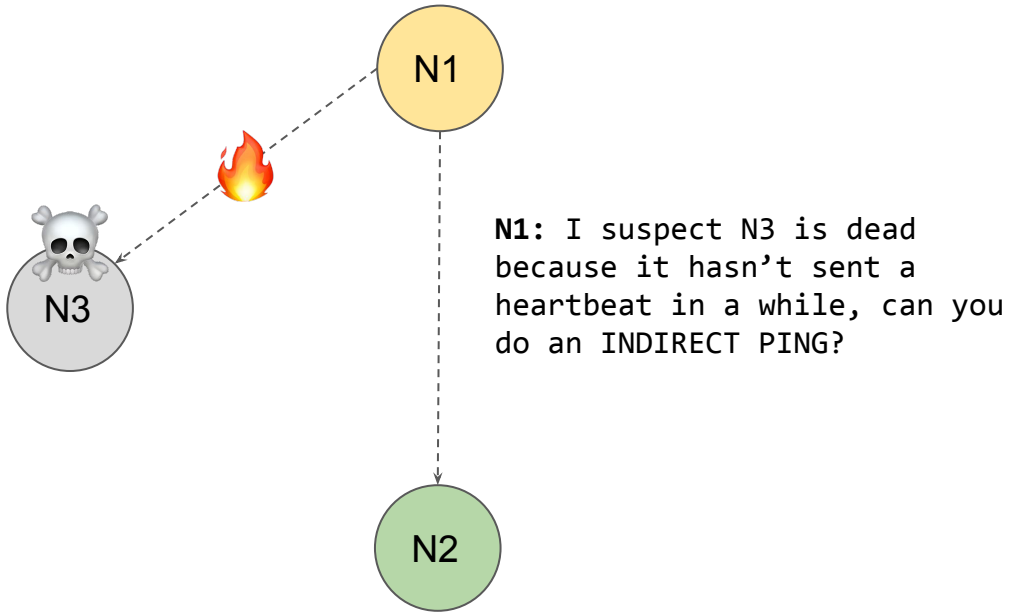
https://www.cs.cornell.edu/projects/Quicksilver/public_pdfs/SWIM.pdf

(Scalable) Gossip Protocols



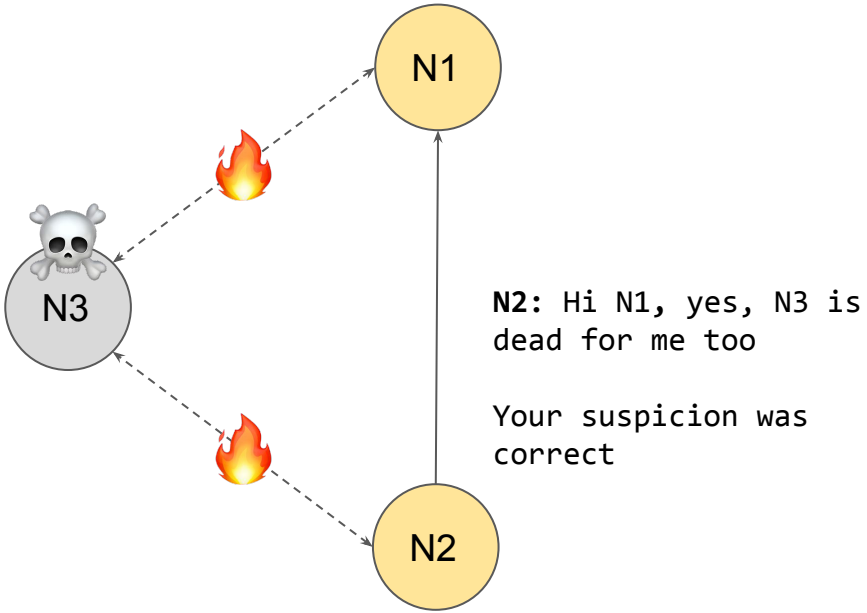
https://www.cs.cornell.edu/projects/Quicksilver/public_pdfs/SWIM.pdf

(Scalable) Gossip Protocols



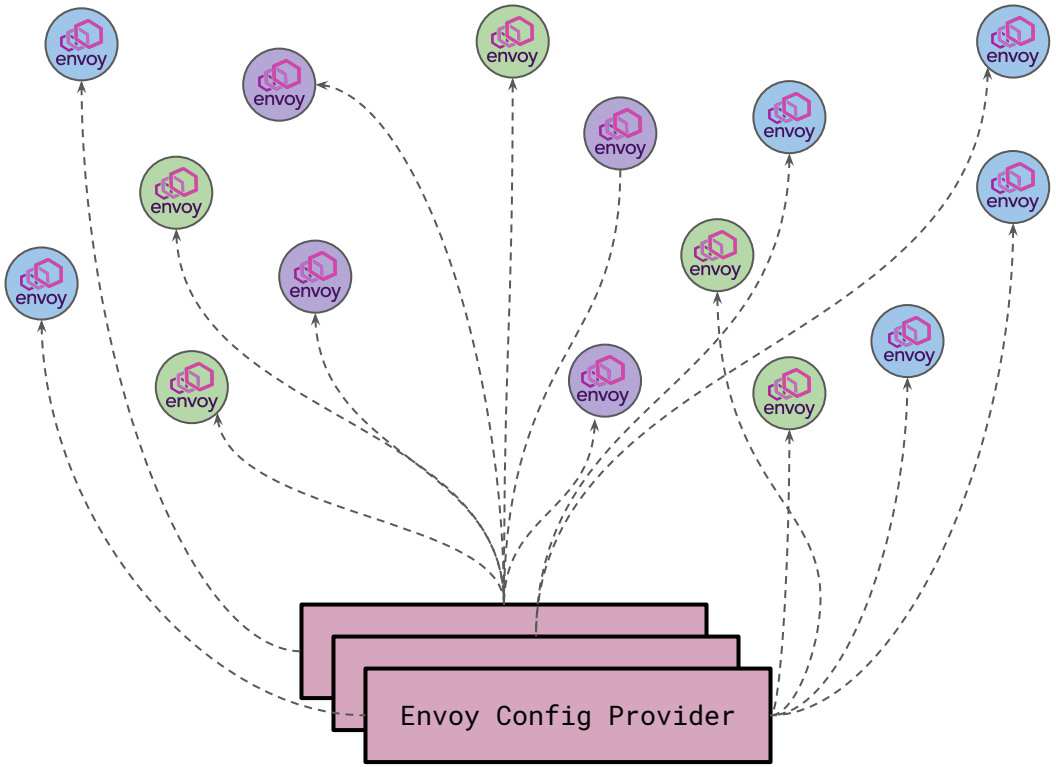
https://www.cs.cornell.edu/projects/Quicksilver/public_pdfs/SWIM.pdf

(Scalable) Gossip Protocols



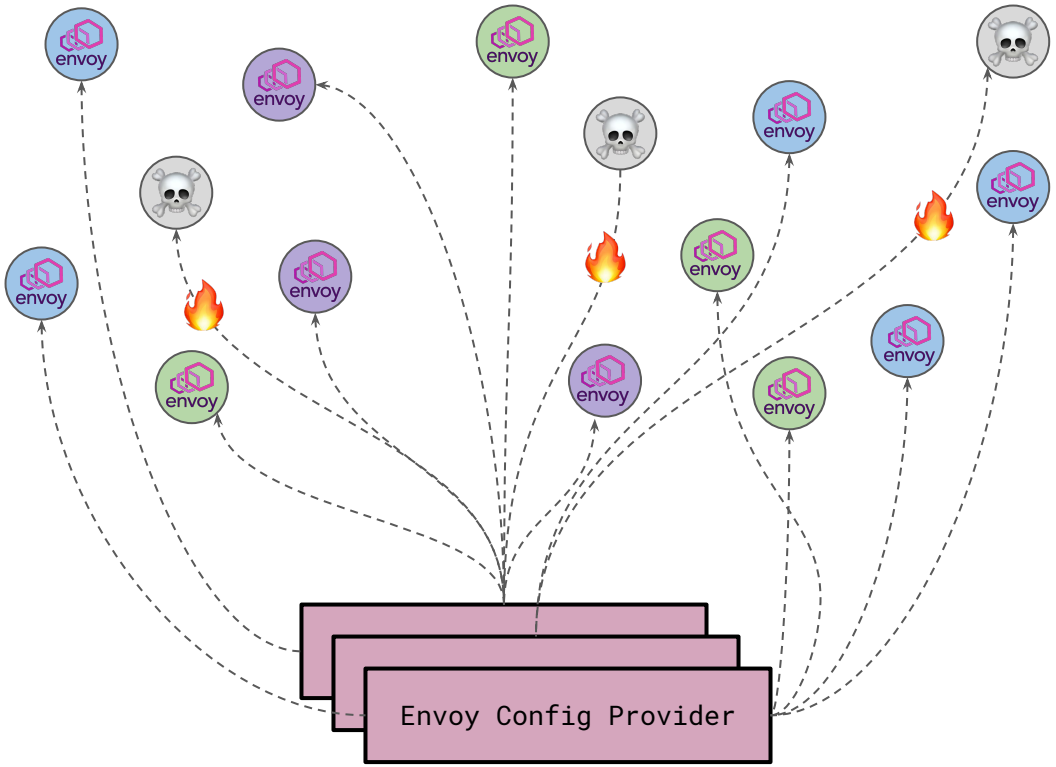
https://www.cs.cornell.edu/projects/Quicksilver/public_pdfs/SWIM.pdf

Eventual Consistency



Eventual Consistency

Eventual Consistency



Summary

Need Agreement?

- A strongly consistent system may be ideal for this use case

Scalability?

- Eventual consistency will work as long as you acknowledge in your applications that it’s not always perfect

Thanks!

Email: **hi@suhailpatel.com**
Twitter: **@suhailpatel**