REDIS CLUSTER

or: distributed systems are hard

Jan-Erik Rediger

28. Mai 2015

Hi, I'm Jan-Erik

Student of Computer Science, RWTH

first Redis-related project: 2010 (an Erlang client)

Maintainer of

- try.redis.io
- hiredis
- hiredis-rb, hiredis-py, hiredis-node

REDIS

Redis *is an open source, BSD licensed, advanced* ***key-value cache*** *and* ***store****. It is often referred to as a* ***data structure server*** *since keys can contain strings, hashes, lists, sets, sorted sets, bitmaps and hyperloglogs.*

```
SET redis rocks

HSET meetup.42 name PHPUGDUS

SADD meetups-in-dus 42:PHPUGDUS

ZADD meetups-in-nrw 20150628 PHPUGDUS

LPUSH trivago "meetup host"
```

LIMITS

Must fit into RAM

LIMITS

Must fit into RAM

No redundancy

LIMITS

Must fit into RAM

No redundancy

Single-threaded

Sharding

- Split data set across nodes
- e.g. via Twemproxy or Codis

OVERCOMING LIMITS

Sharding

- Split data set across nodes
- e.g. via Twemproxy or Codis

Replication

- Failover for HA
- Sentinel or another system
- Scale reads to more instances

REDIS CLUSTER

It's still Redis

It's still Redis

must be fast

It's still Redis

- must be fast
- must scale

It's still Redis

- must be fast
- must scale
- must be simple to use

It's still Redis

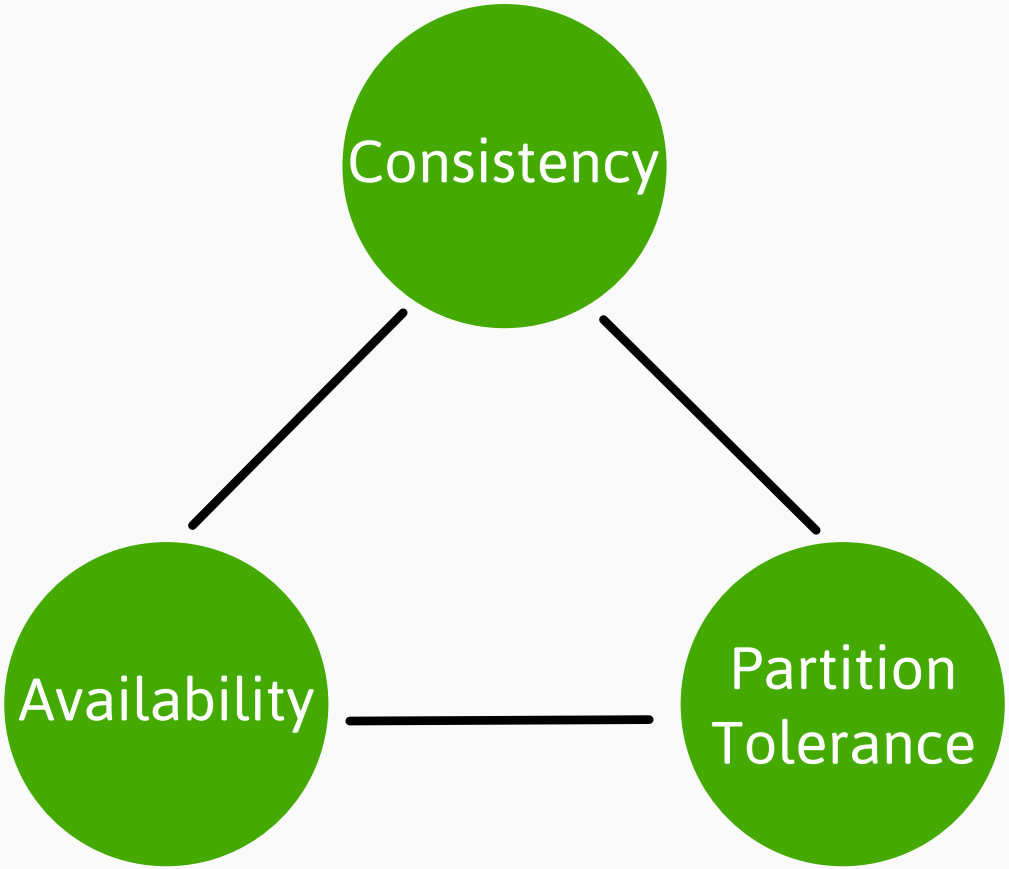
- must be fast
- must scale
- must be simple to use
- must give *some* guarantees

It's still Redis

- must be fast
- must scale
- must be simple to use
- must give *some* guarantees

but it's a Distributed System

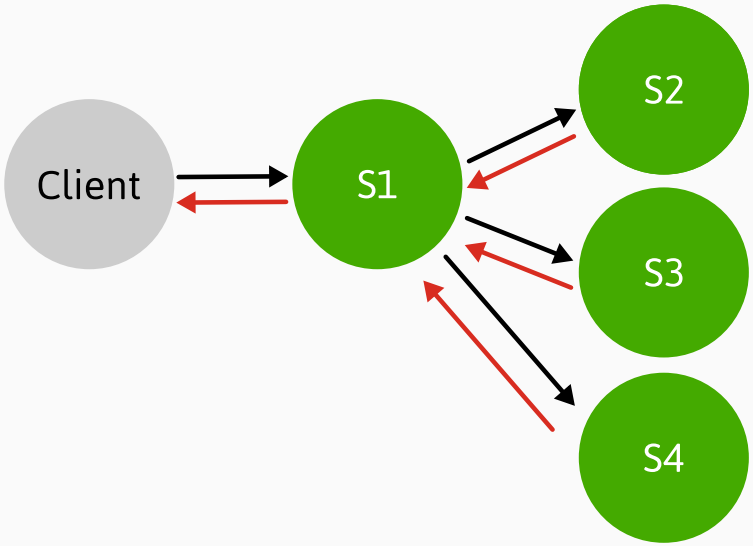
CAP - PICK TWO



CP - CONSISTENT & PARTITION TOLERANT

Higher Latency

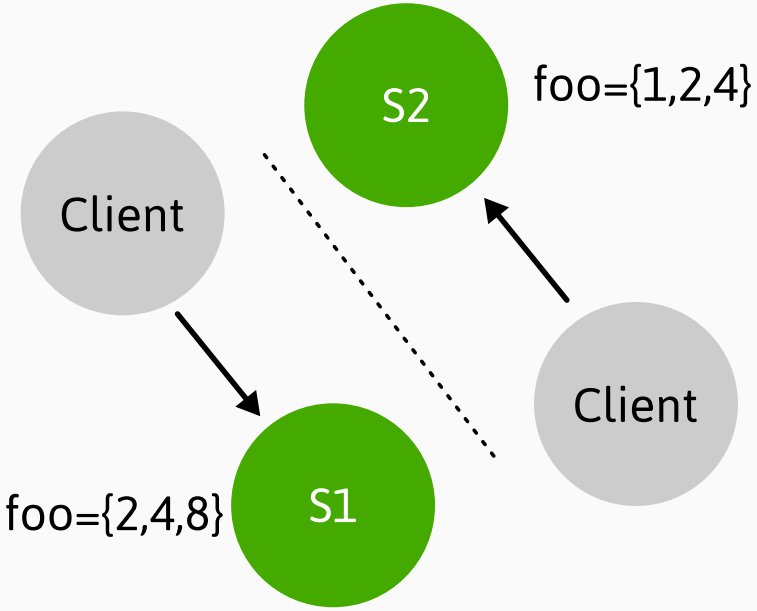
Wait for majority of ACKs



AP - AVAILABLE & PARTITION TOLERANT

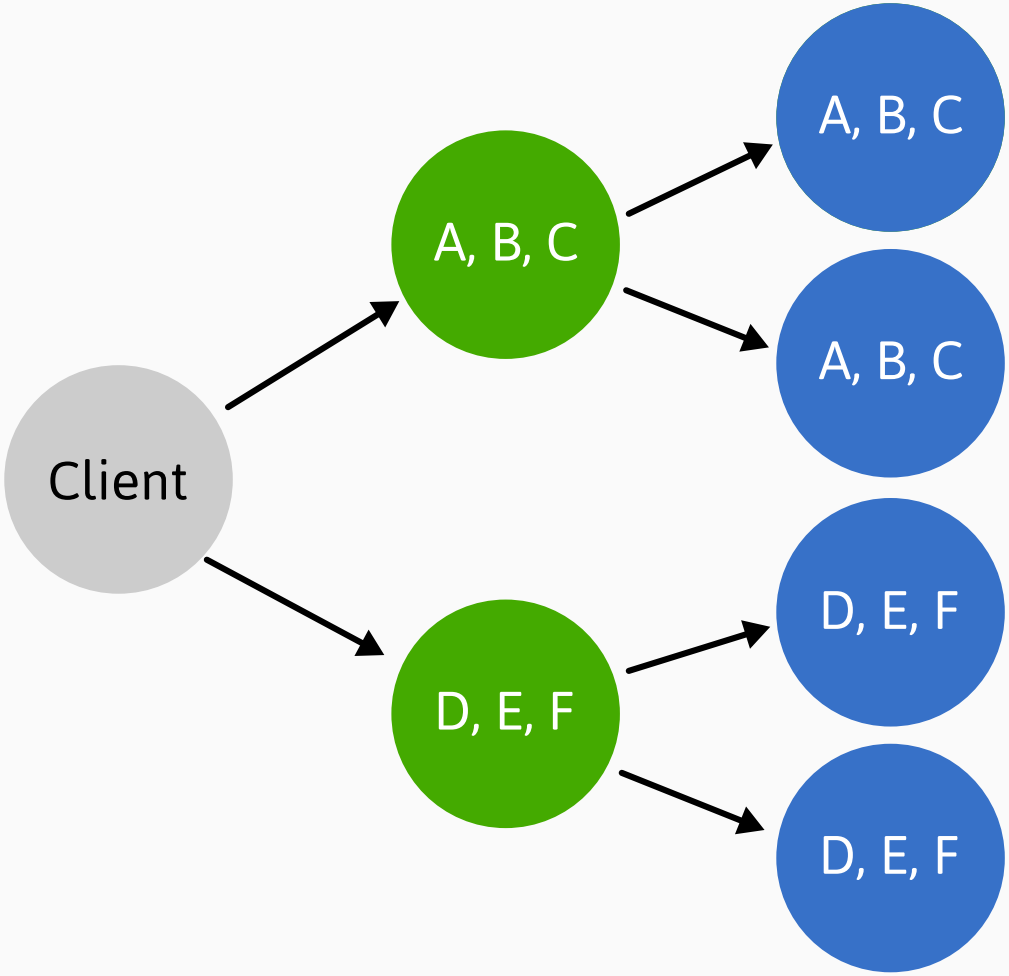
Eventual Consistency

might require merges

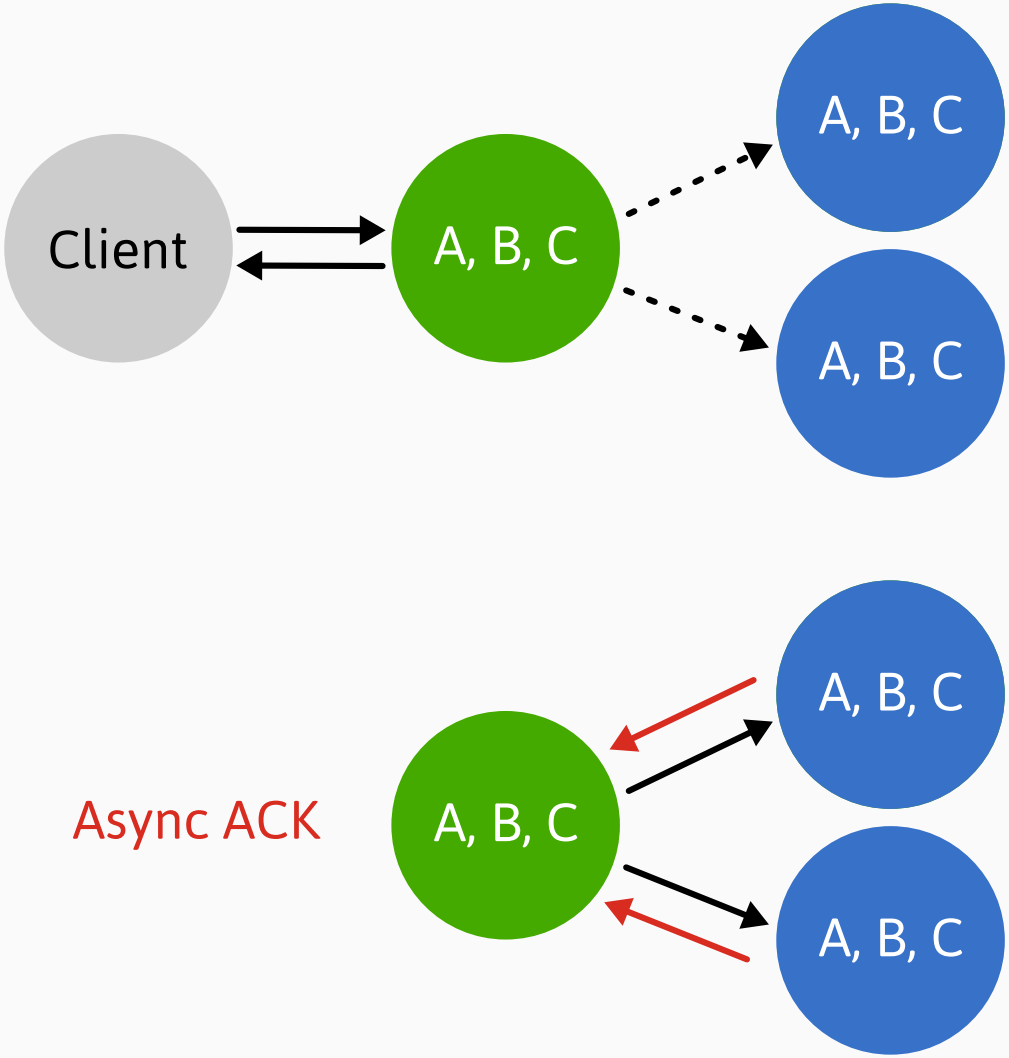


SO WHAT IS IT?

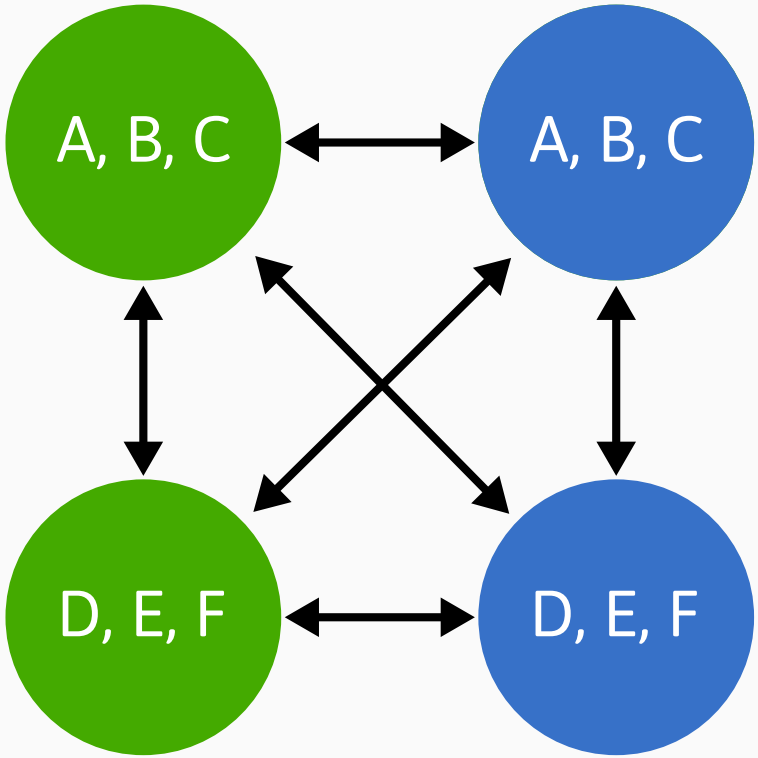
SHARDING + REPLICATION



REPLICATION: ASYNCHRONOUS



FULL MESH



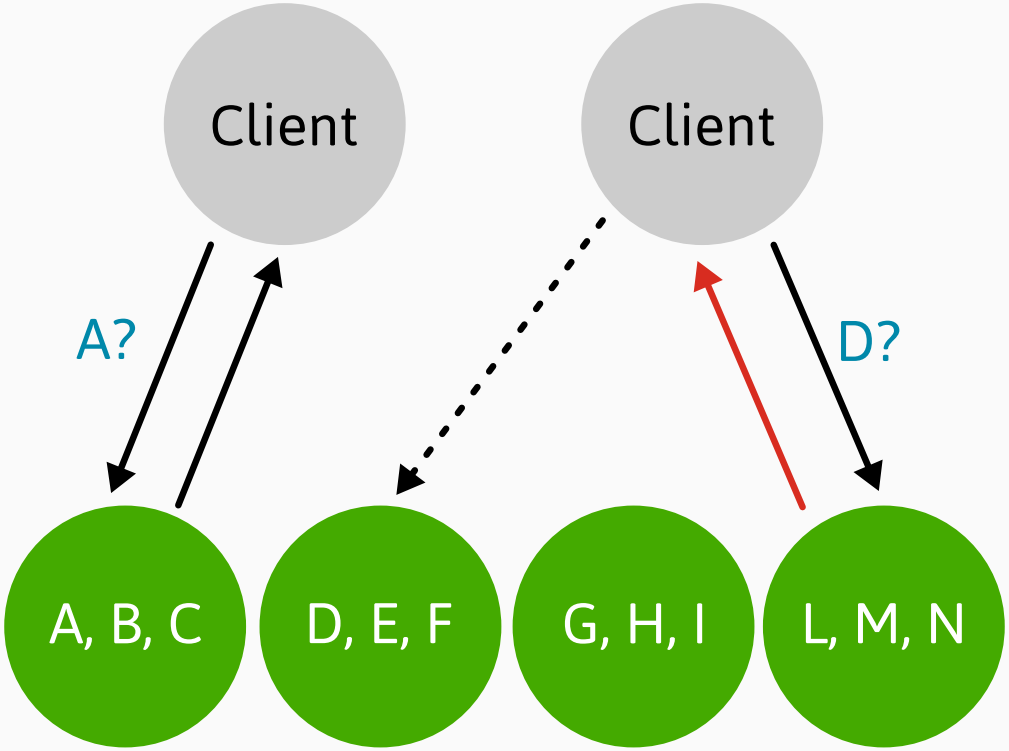
Heartbeats

Gossip

Failover

Config update

REDIRECTIONS



SLOT CONFIGURATION

Key space split into 16384 slots

Every node serves a subset of the slots

Every node knows the node slot mapping

SINGLE-KEY OPERATIONS

```
> GET redis
1) "rocks"
```

```
> GET kjdopiqh
(error) MOVED 12182 127.0.0.1:7002
```

All keys in same slot: it's fine

```
> MGET foo10 foo5406
1) "hello"
1) "world"
```

MULTI-KEY OPERATIONS

Keys in different slots: sorry, not possible

```
> MGET foo bar
(error) CROSSSLOT Keys in request don't hash
  to the same slot
```

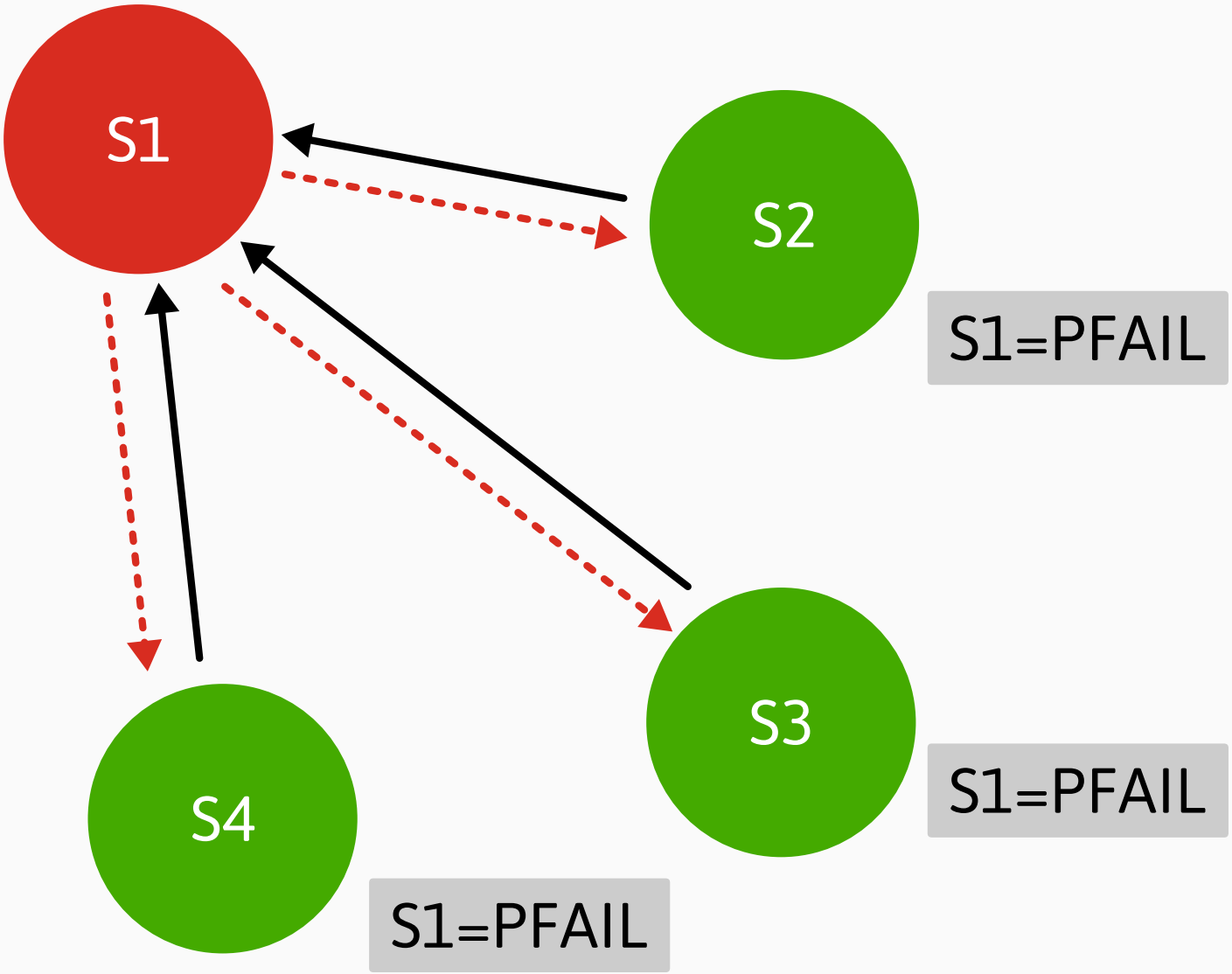
MULTI-KEY OPERATIONS WITH HASH TAGS

Ensure keys map to the same slot

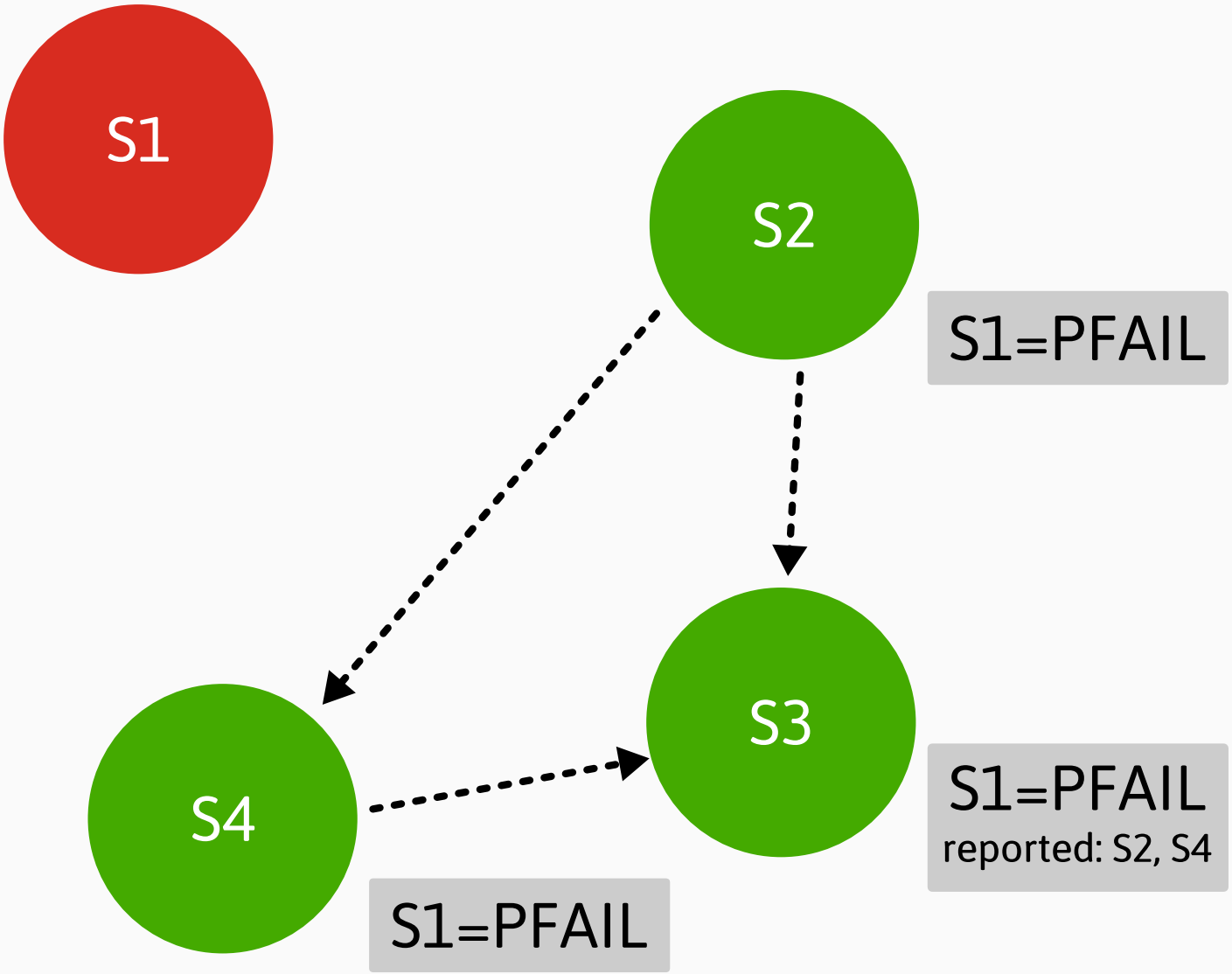
```
> MGET {foo}.meetup {foo}.city
1) "PHPUG"
2) "Düsseldorf"
```

FAILURE DETECTION

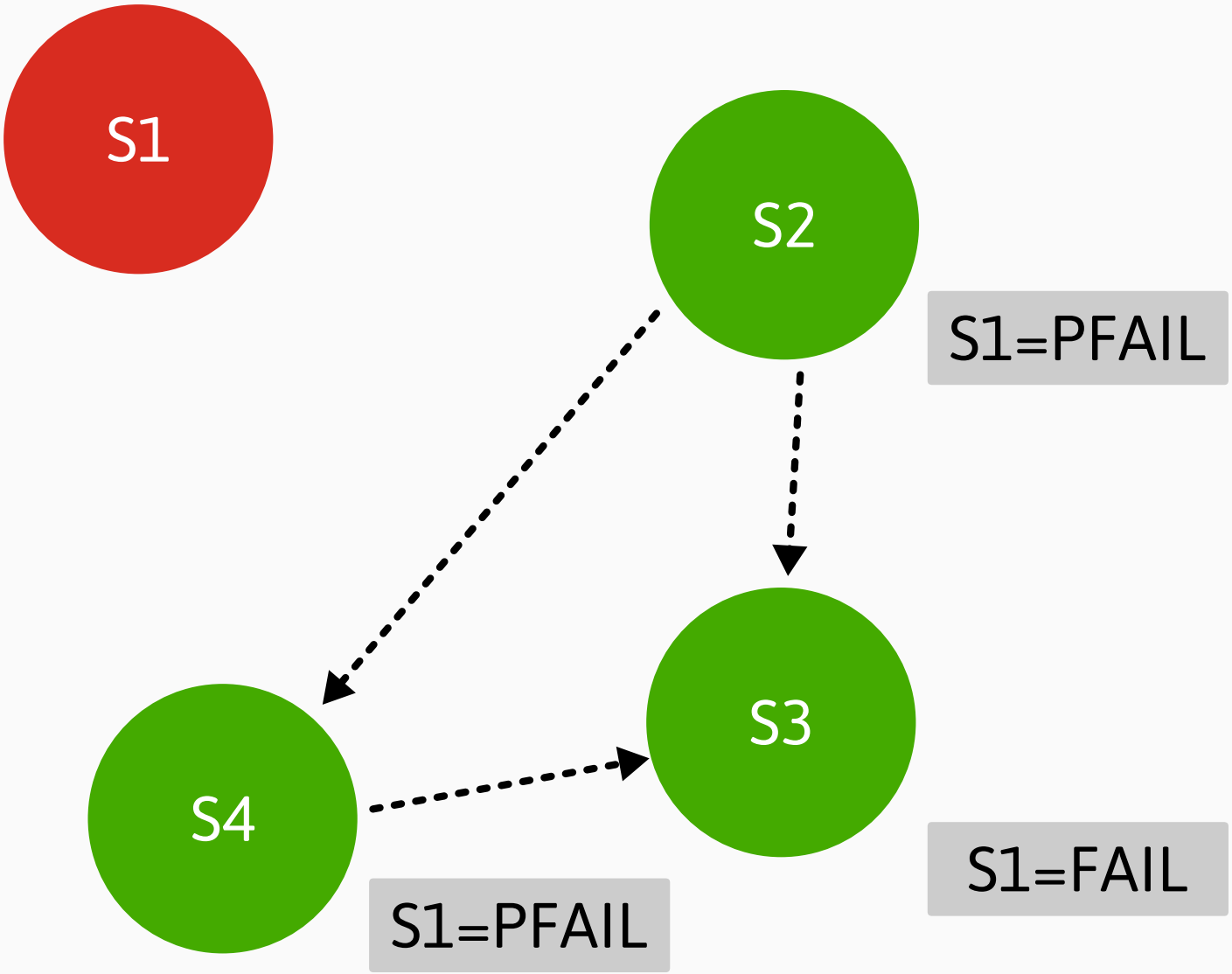
NODE TIMES OUT: PFAIL



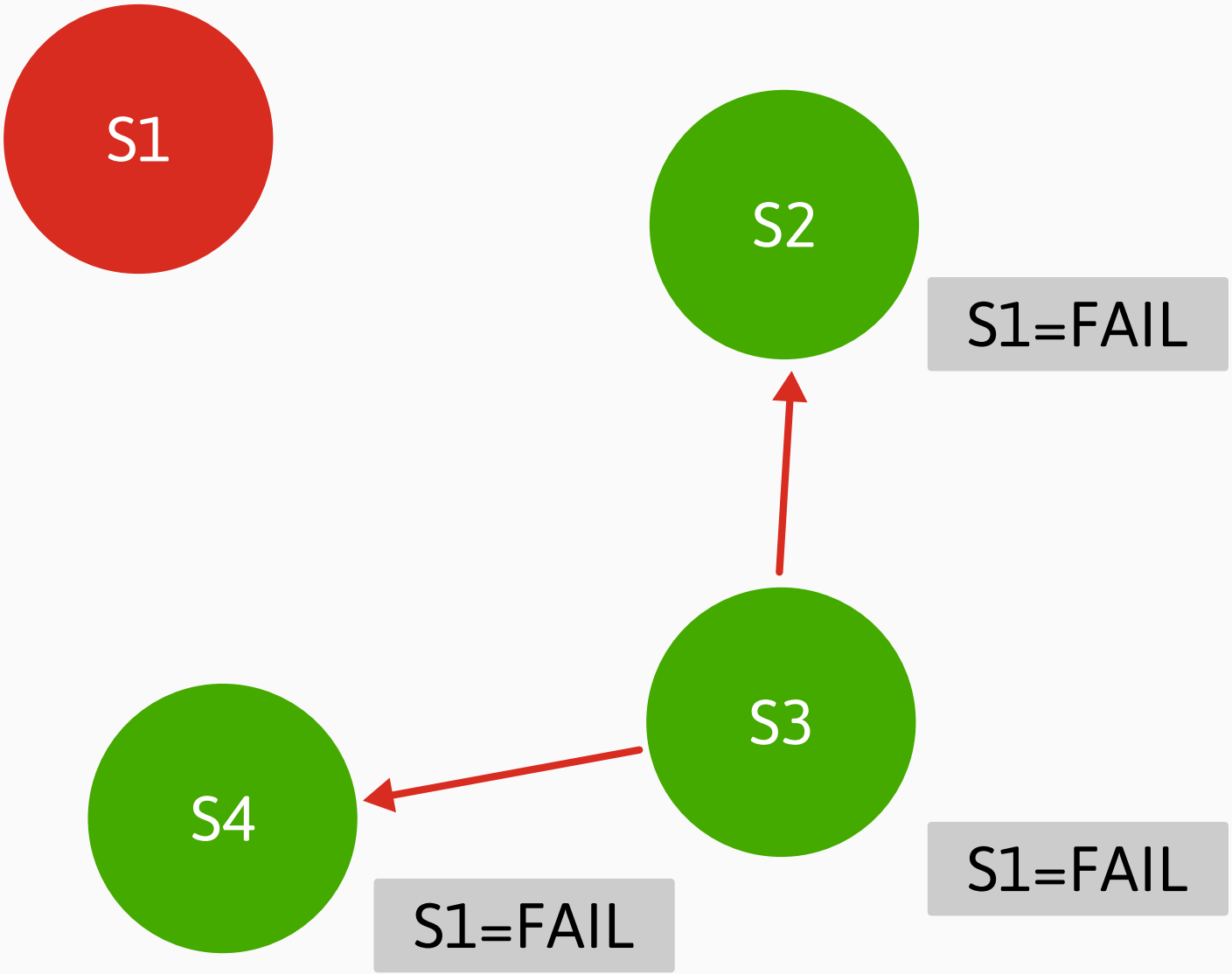
ALL REMAINING NODES SEE PFAIL



PFAIL → FAIL

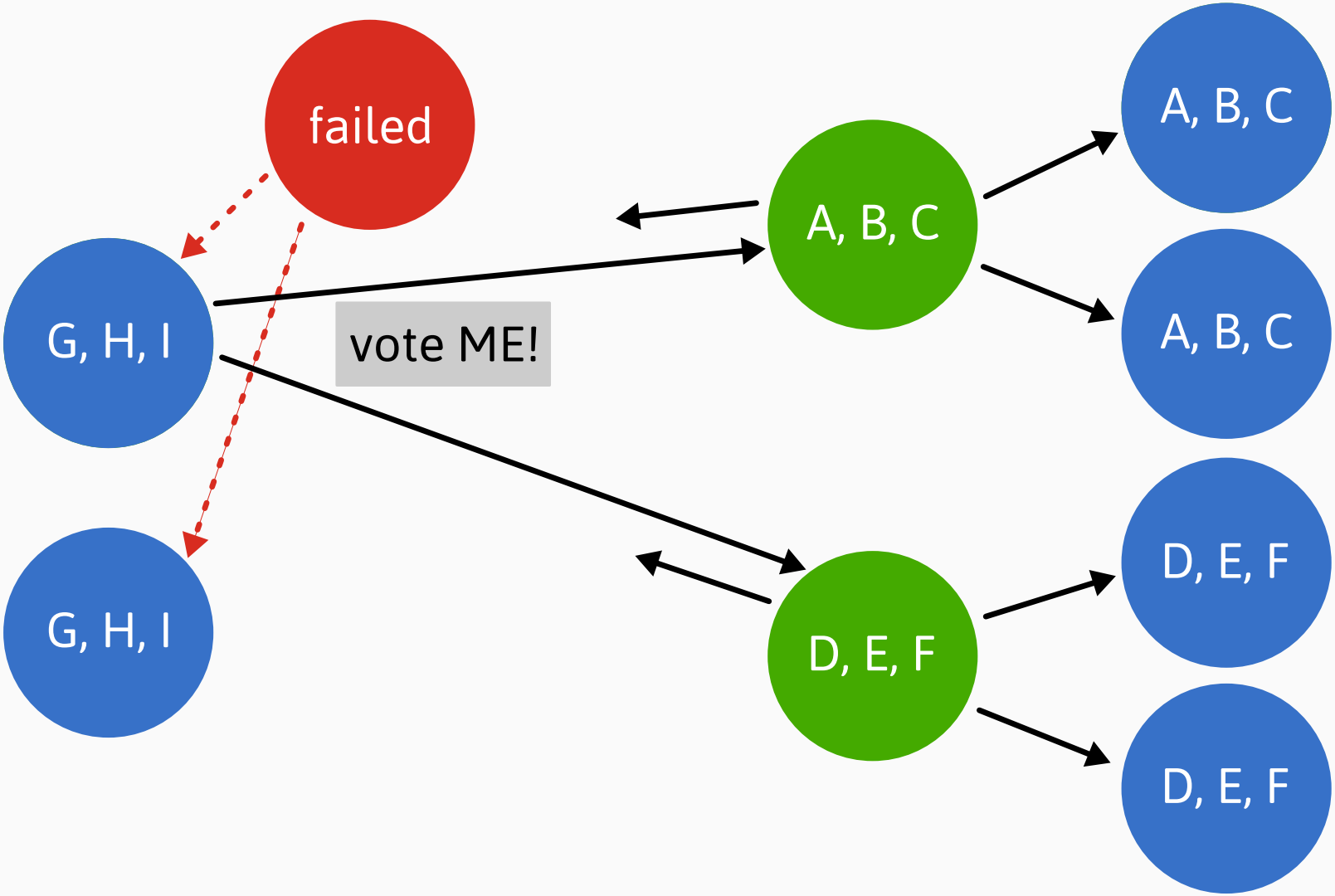


FORCE FAIL, TRIGGER FAILOVER

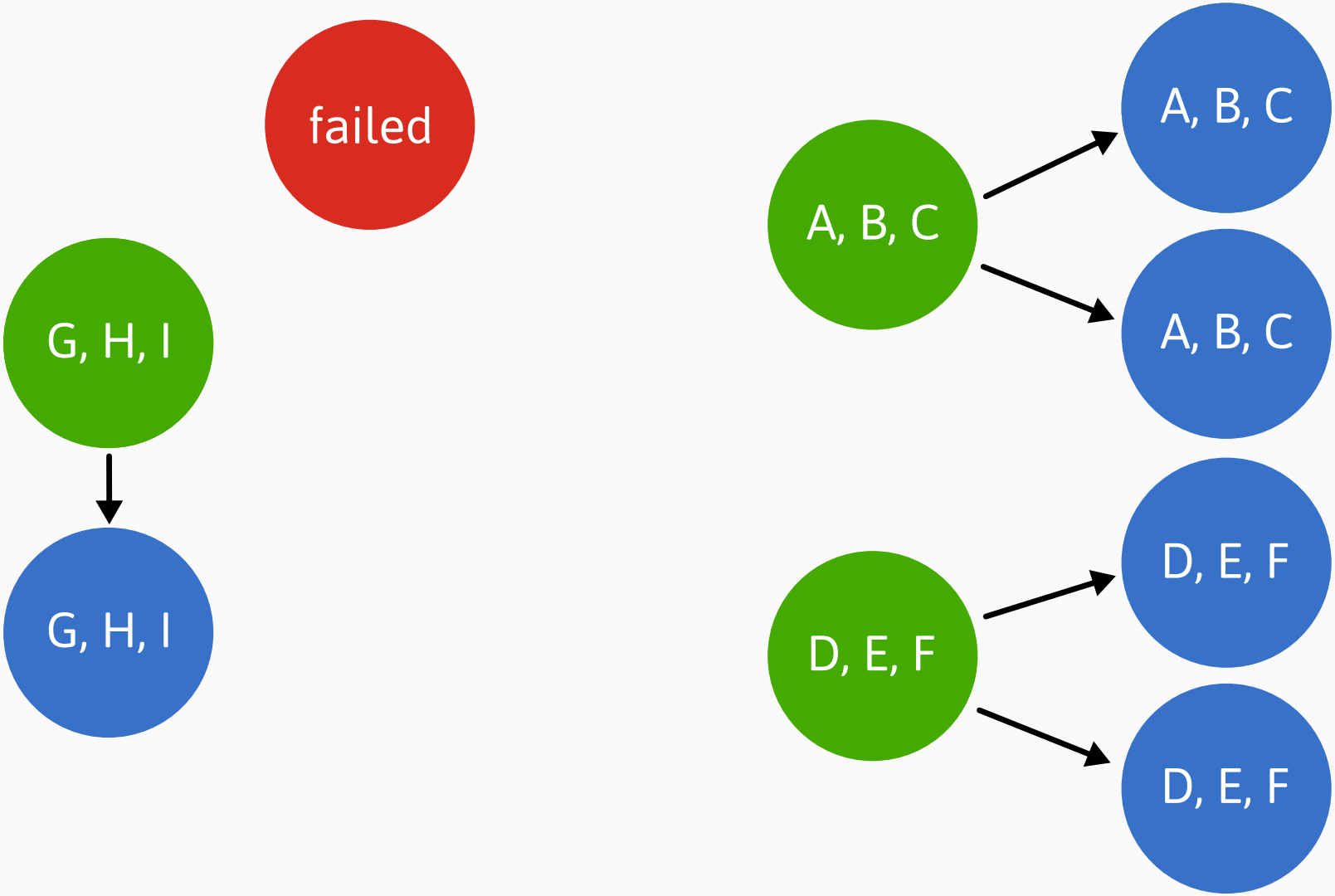


FAILOVER

ACTUAL FAILOVER



ACTUAL FAILOVER



IS IT CONSISTENT?

IS IT CONSISTENT?

Eventually...

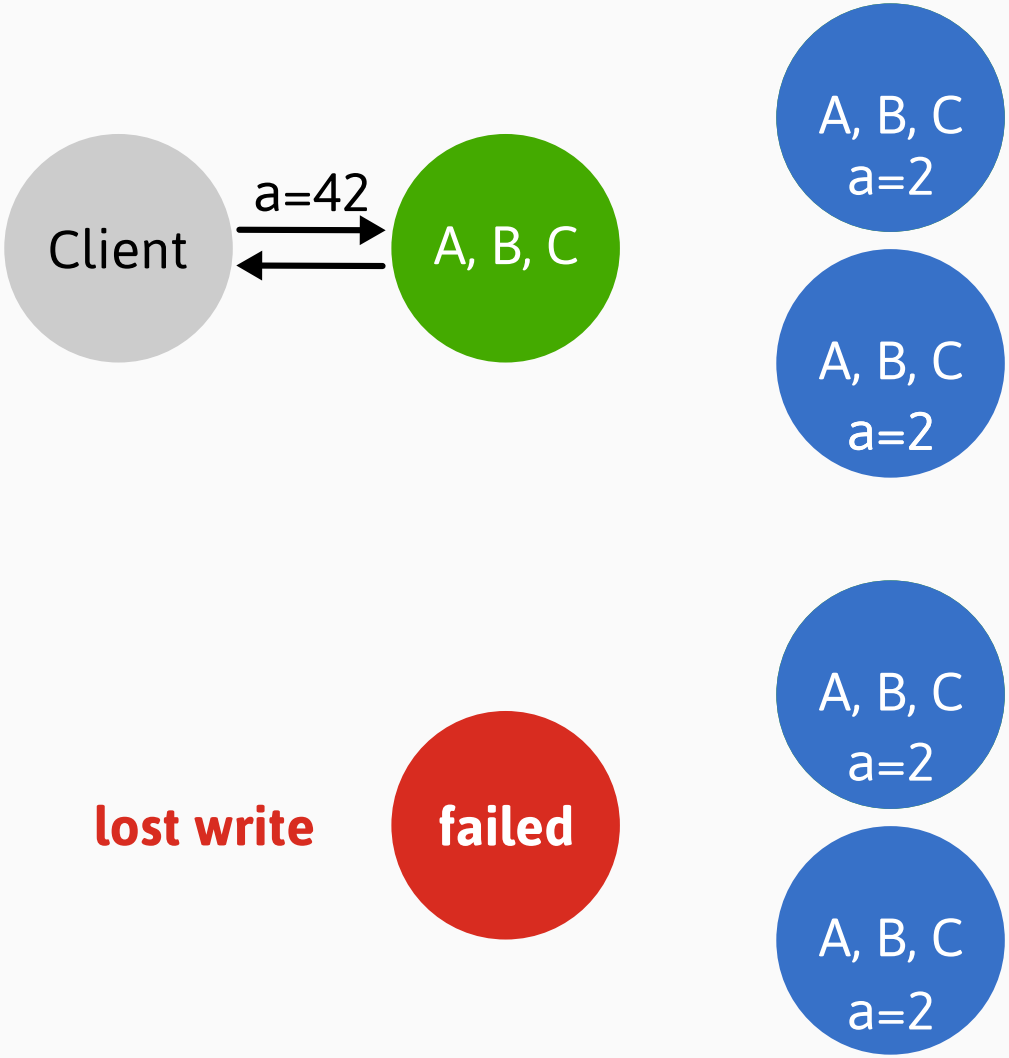
IS IT CONSISTENT?

Eventually...

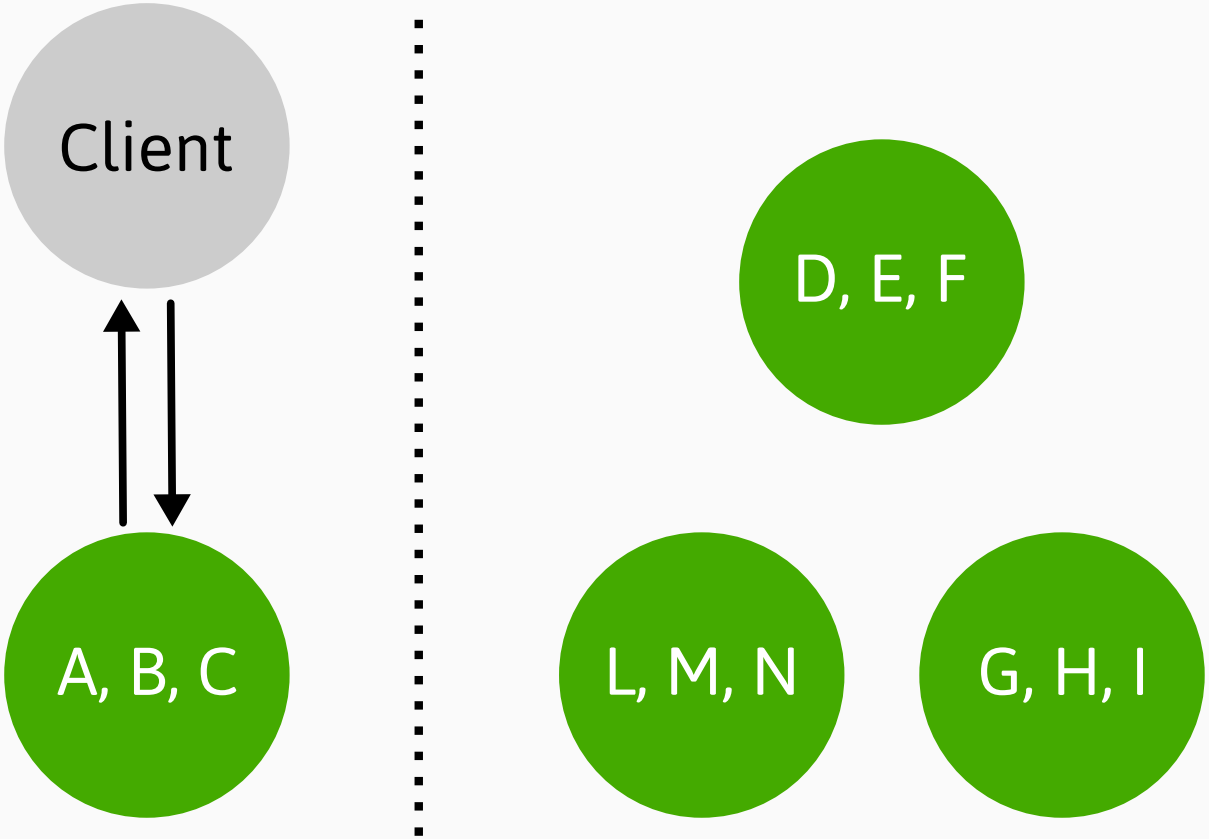
- Last Failover wins
- Mechanisms to avoid unbound data loss

FAILURE MODES

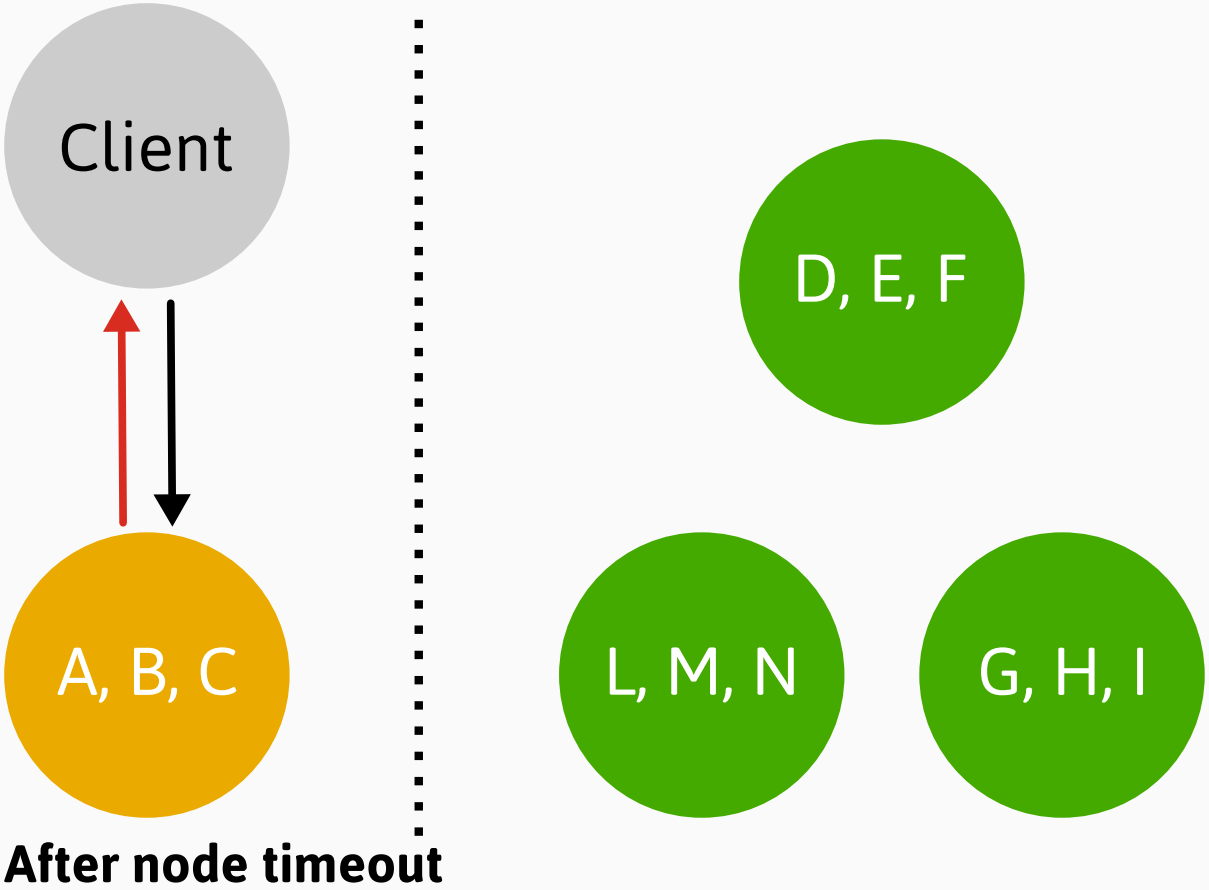
FAILURE: CRASHES



FAILURE: NETWORK SPLIT



FAILURE: NETWORK SPLIT



CREATING A CLUSTER

CONFIGURATION

```
port 7001
cluster-enabled yes
cluster-config-file nodes.conf
cluster-node-timeout 5000
```

LET THEM MEET

```
CLUSTER MEET 127.0.0.1 7001
```

* on every node ;)

MORE CONVENIENT

```
./redis-trib.rb create --replicas 1 \
    127.0.0.1:7000 127.0.0.1:7001 \
    127.0.0.1:7002 127.0.0.1:7003 \
    127.0.0.1:7004 127.0.0.1:7005
```

MORE COMMANDS YOU WANT TO KNOW

```
CLUSTER ADDSLOTS 42 43 44

CLUSTER SETSLOT 42 NODE ff7831dfe

CLUSTER SETSLOT 42 MIGRATING ff7831dfe

CLUSTER SETSLOT 42 IMPORTING ff7831dfe
```

AND SOME MORE

```
CLUSTER NODES

CLUSTER FAILOVER

CLUSTER REPLICATE ff7831dfe
```

ALL CLUSTER COMMANDS

redis.io/commands#cluster

NODES.CONF

```
ff7831dfe7fc73f741d5c4663a8020e654f88f22
  127.0.0.1:7001 myself,master - 0 0 1
  connected 0-5460
94f5710dab37058784824dc9d2ddb27a693f1336
  127.0.0.1:7013 slave
 a46ff091f49bd28149594dfd2272ebb3aedcdd59 0
  1415873239710 6 connected
vars currentEpoch 6 lastVoteEpoch 0
```

AVAILABLE CLIENTS

Ruby: redic-cluster

Ruby: redis-rb-cluster

Python: redis-py-cluster

PHP: Predis

Java: Jedis

JavaScript: thunk-redis

.NET: StackExchange.Redis

SOURCES

redis.io - official site & documentation

cluster-spec - official spec docu

cluster-tutorial - a tutorial

mattsta/redis-cluster-playground - easy to play around with cluster


OpenTechSchool
Conference
AUGUST 15/16, 2015
OTSCONF.COM

THE END

Get the slides here:

http://slidr.io/badboy/redis-cluster

Reach me on Twitter: **@badboy_**

Jan-Erik Rediger - 28. Mai 2015 - PHPUGDUS