Prague PostgreSQL Developer Day

2023 • 02 • 01

Failsafe Patroni 3.0

Presented by

Alexander Kukushkin & Polina Bungina

About us

Alexander Kukushkin

- Principal Software Engineer @Microsoft
- The Patroni guy
- akukushkin@microsoft.com
- Twitter: @cyberdemn

Polina Bungina

- Software Engineer @ZalandoTech
- polina.bungina@zalando.de
- Twitter: @hugh_capet

Agenda

Introduction to Patroni

Observer problem

Demo 1

DCS dailsafe feature

Demo 2

Conclusion

Do we need it at all?

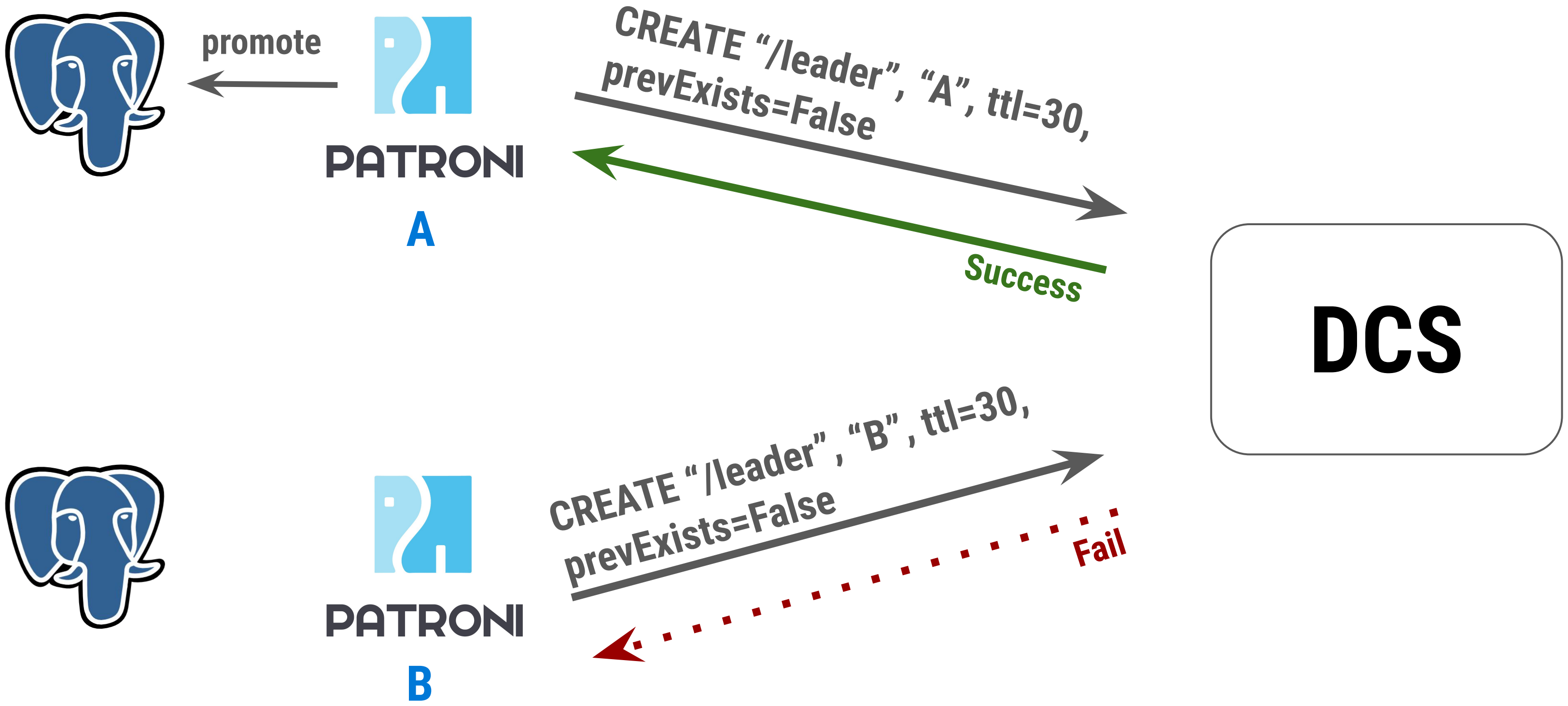
- Service-Level Agreement (**SLA**)

- Recovery point objective (**RPO**)
- Recovery time objective (**RTO**)

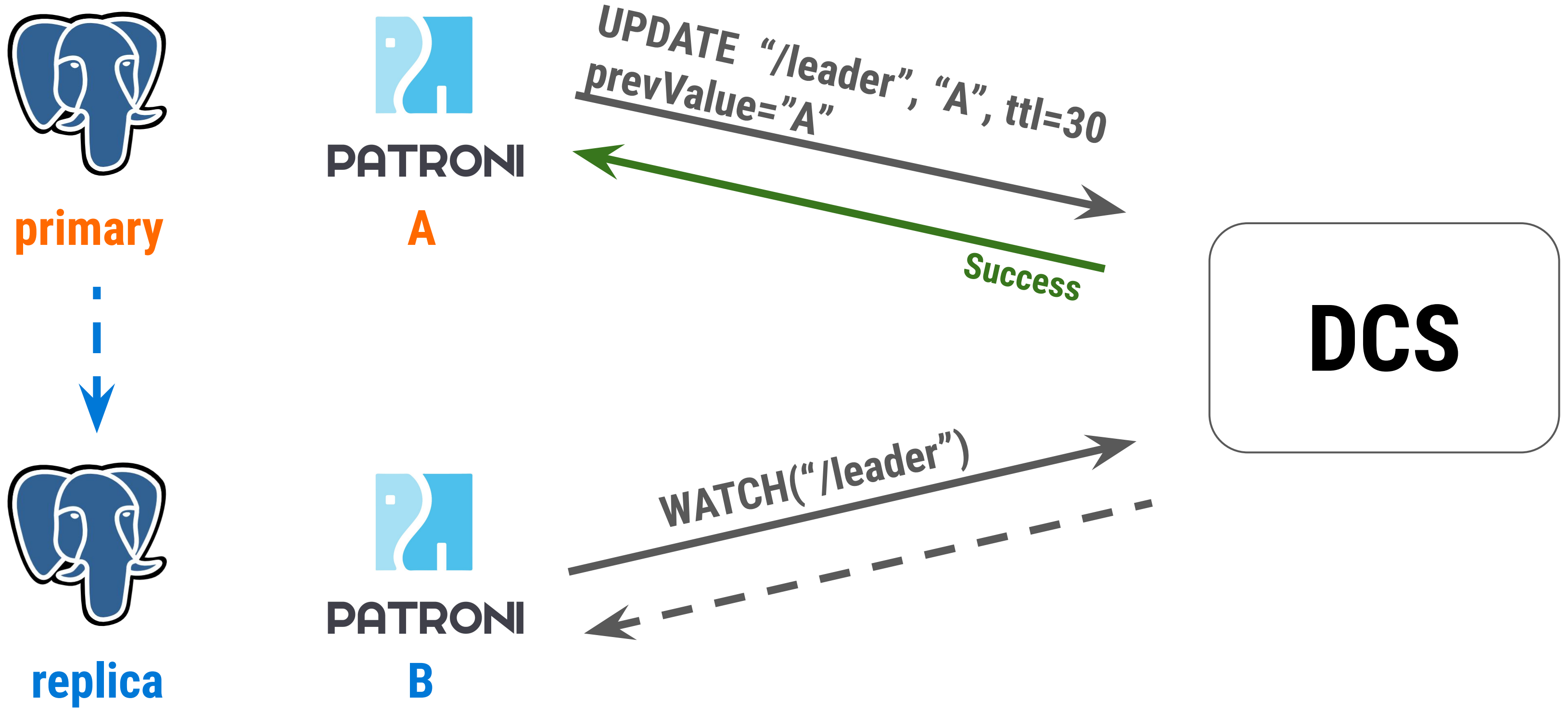
Architecture overview

- Cluster state stored in Distributed Configuration Store (**DCS**)
    - ZooKeeper
    - Etcd
    - Consul
    - Kubernetes control-plane
- **Session/TTL** to expire data (i.e. leader key)
- **Atomic CAS** operations
- **Watches** for important keys

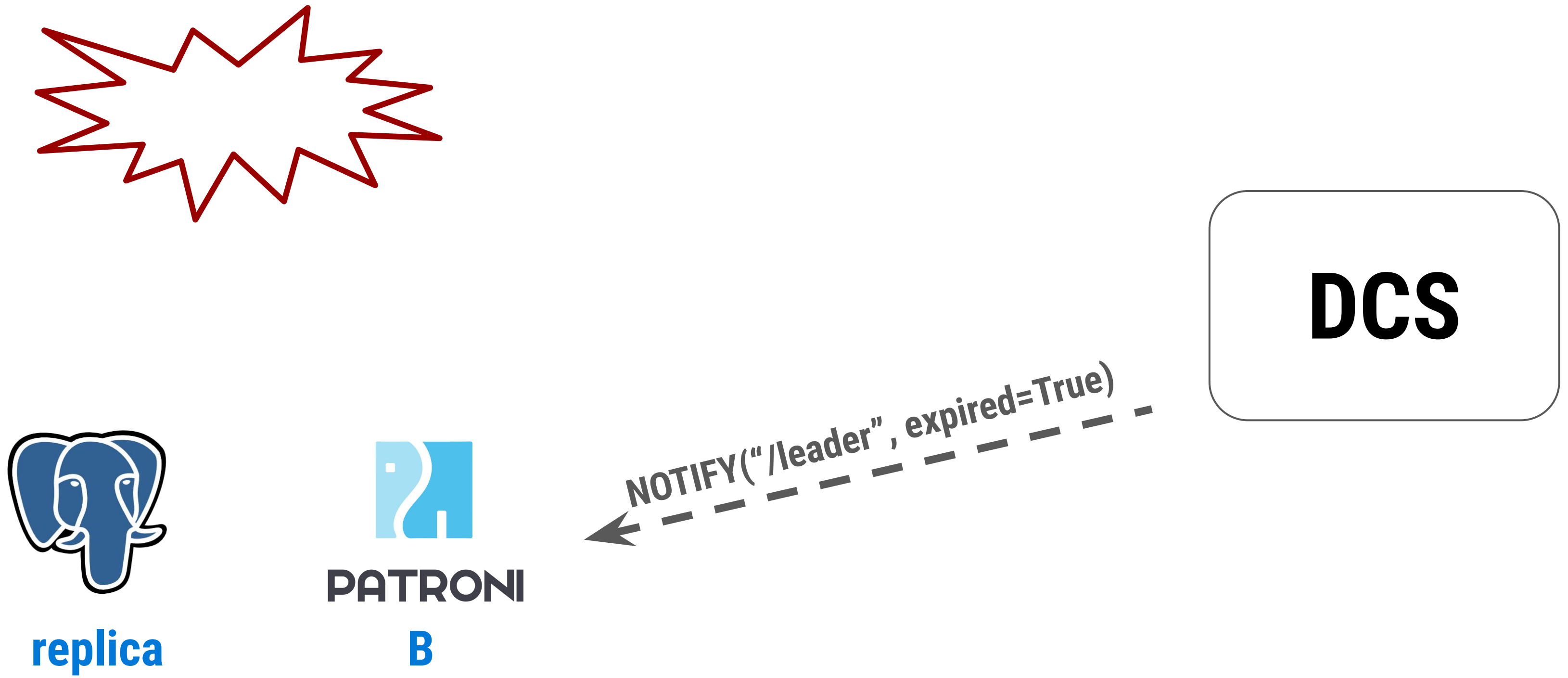
Leader race


promote
PATRONI
A
CREATE "/leader", "A", ttl=30, prevExists=False
Success
DCS
PATRONI
B
CREATE "/leader", "B", ttl=30, prevExists=False
Fail

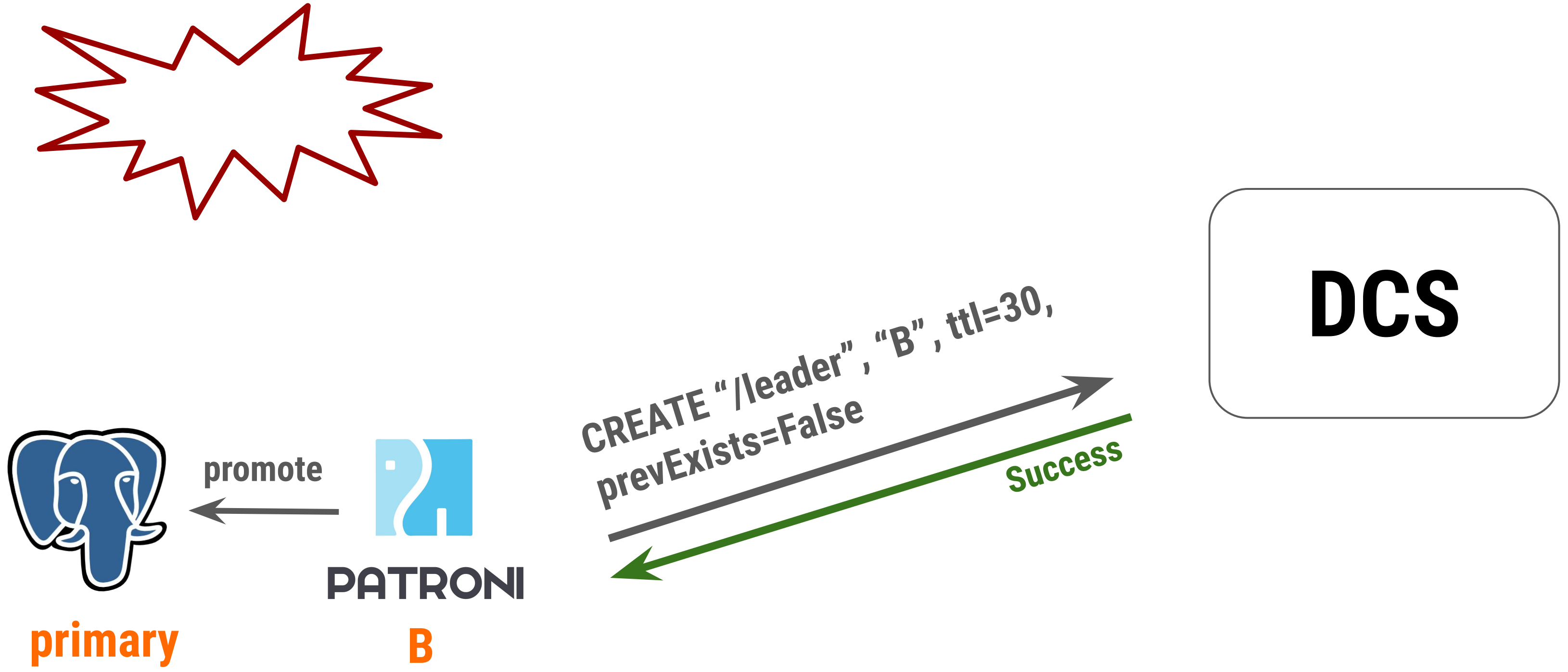
Normal operational mode



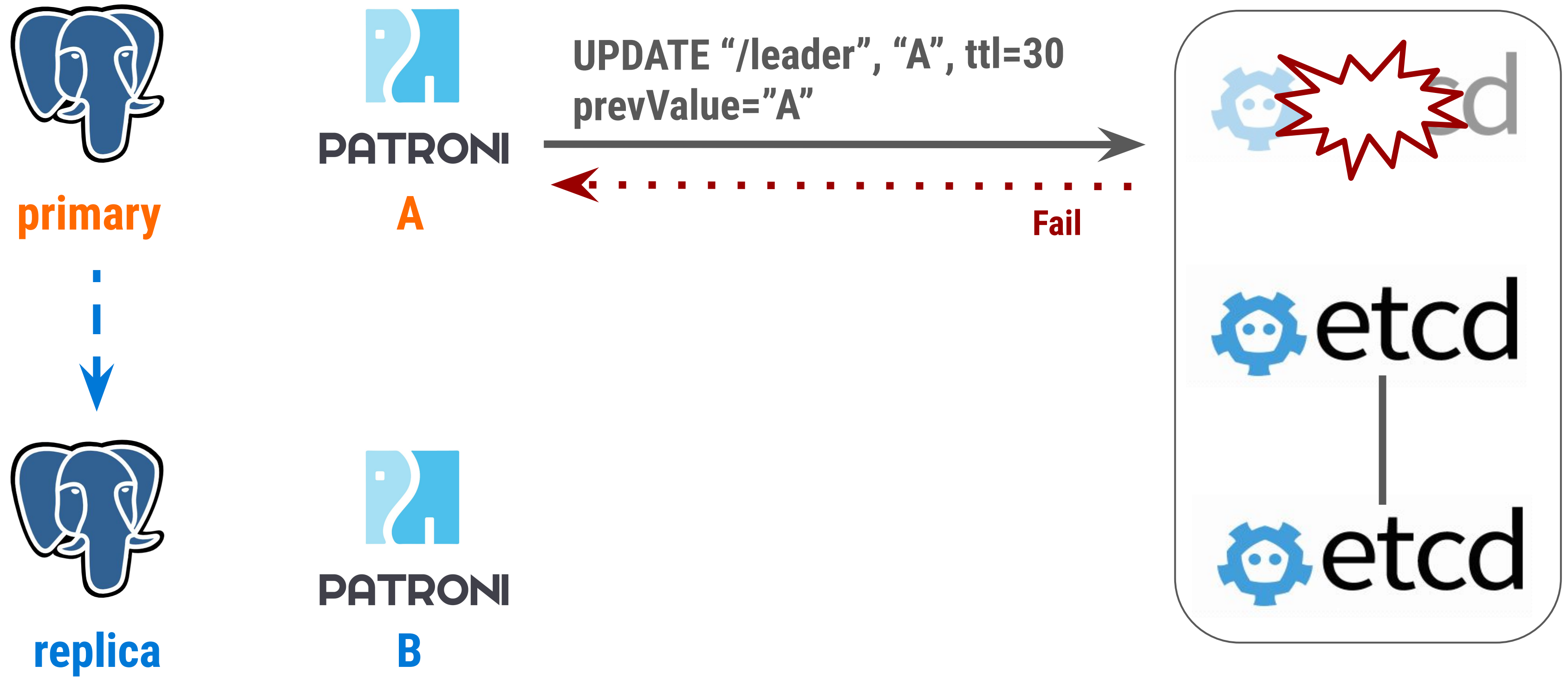
Normal operational mode



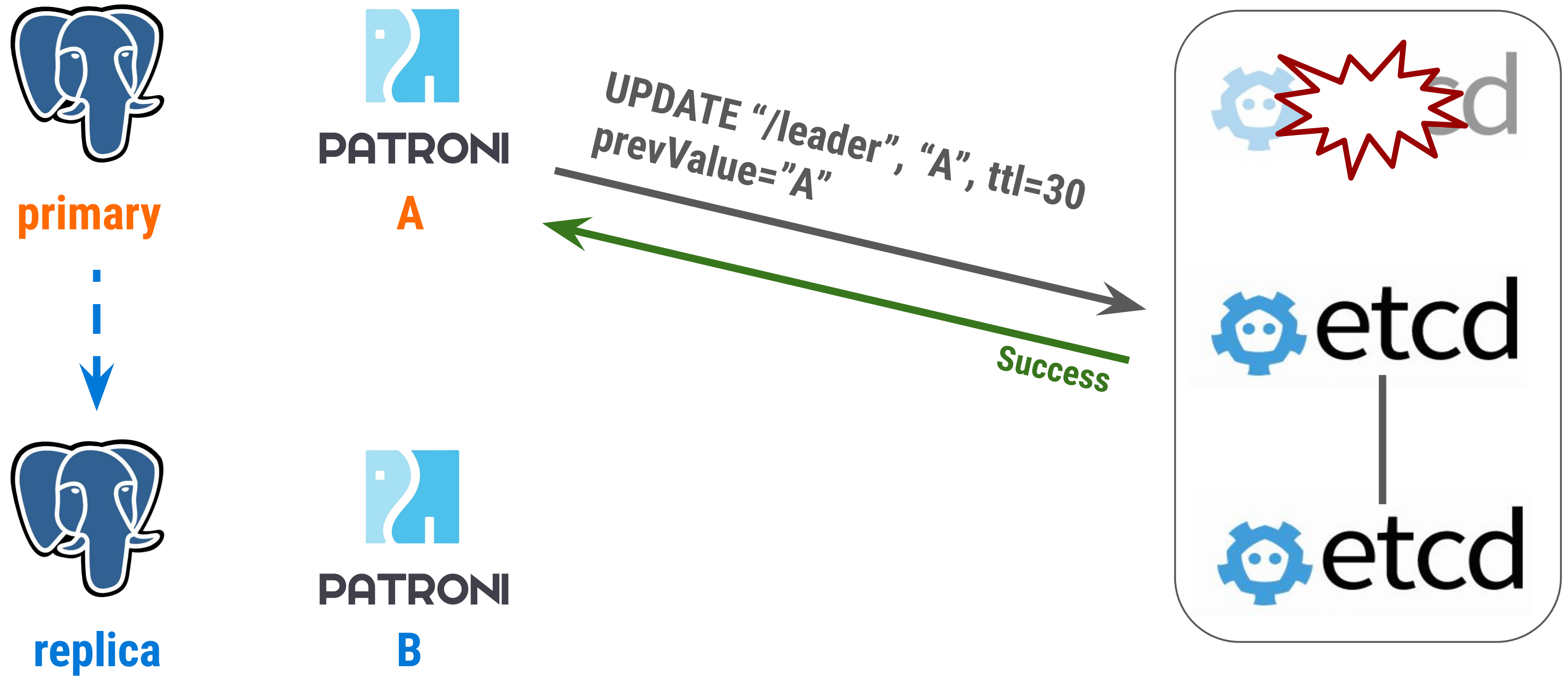
Normal operational mode



DCS can't be accessed

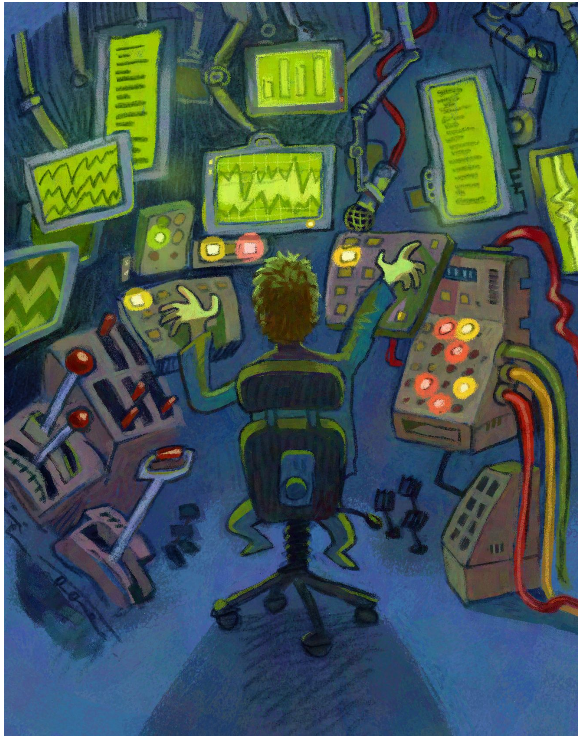


DCS can't be accessed

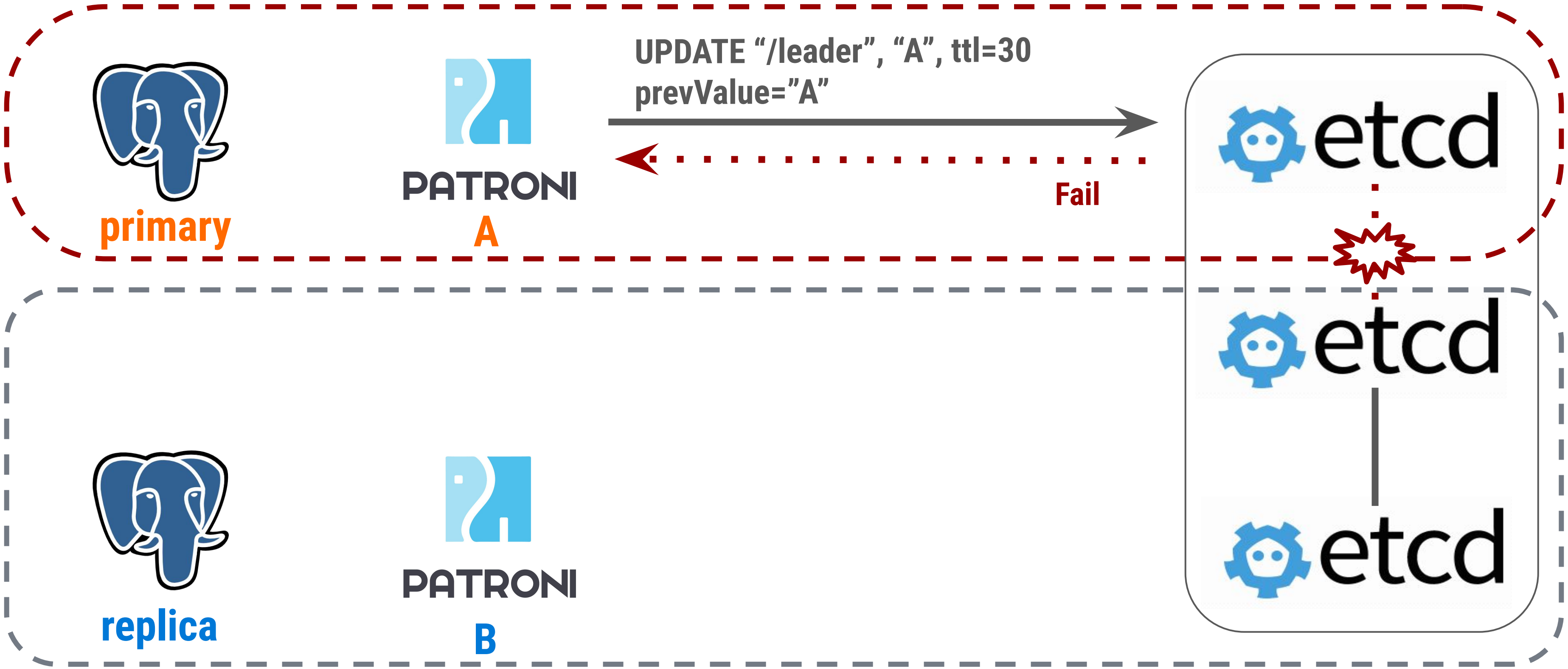


Why did update fail?

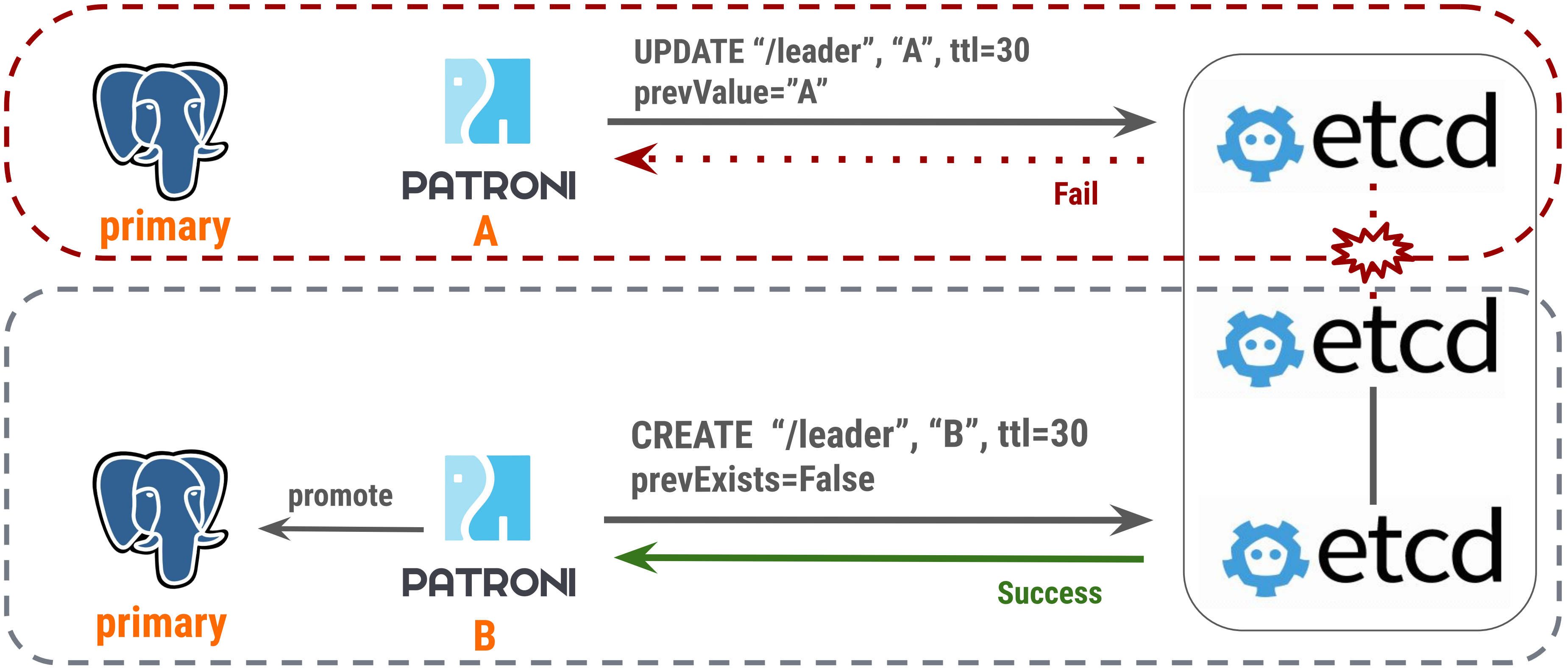
- DCS is down?
- Network issues?



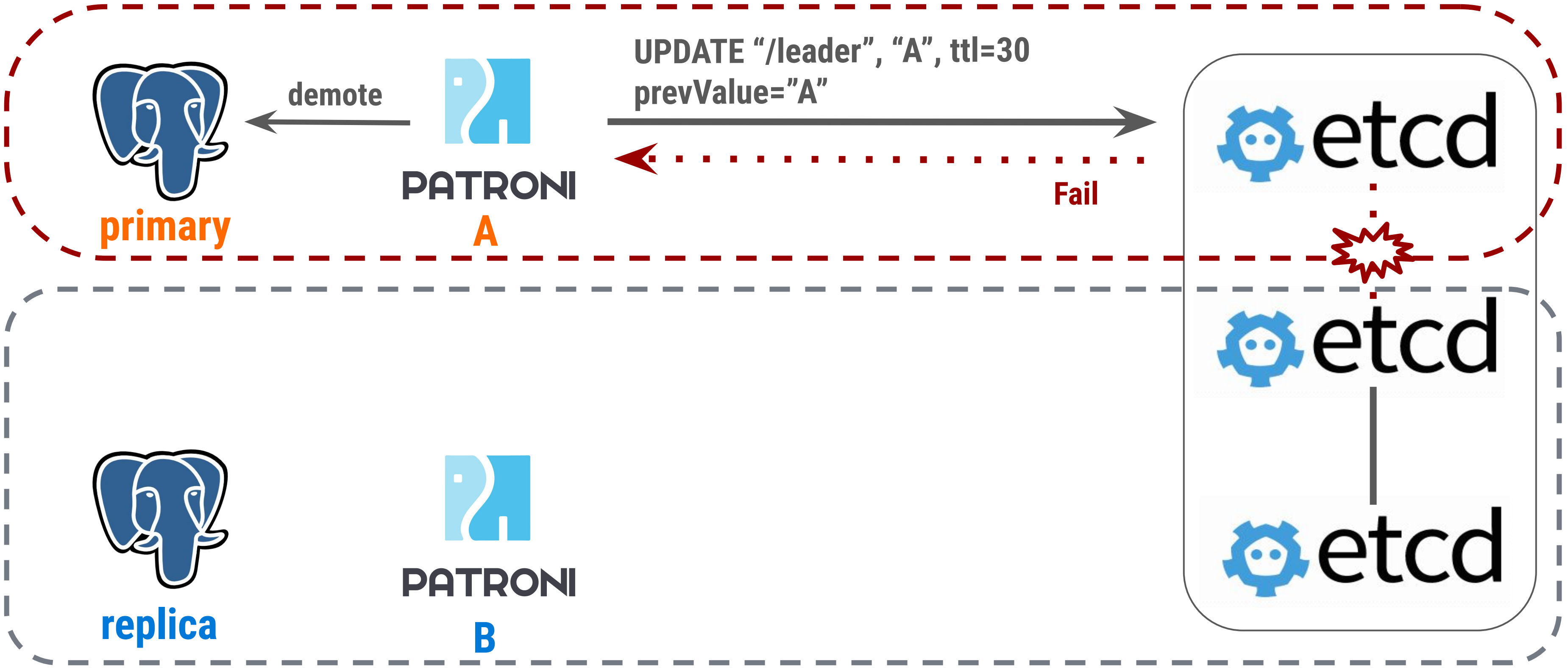
Network partition



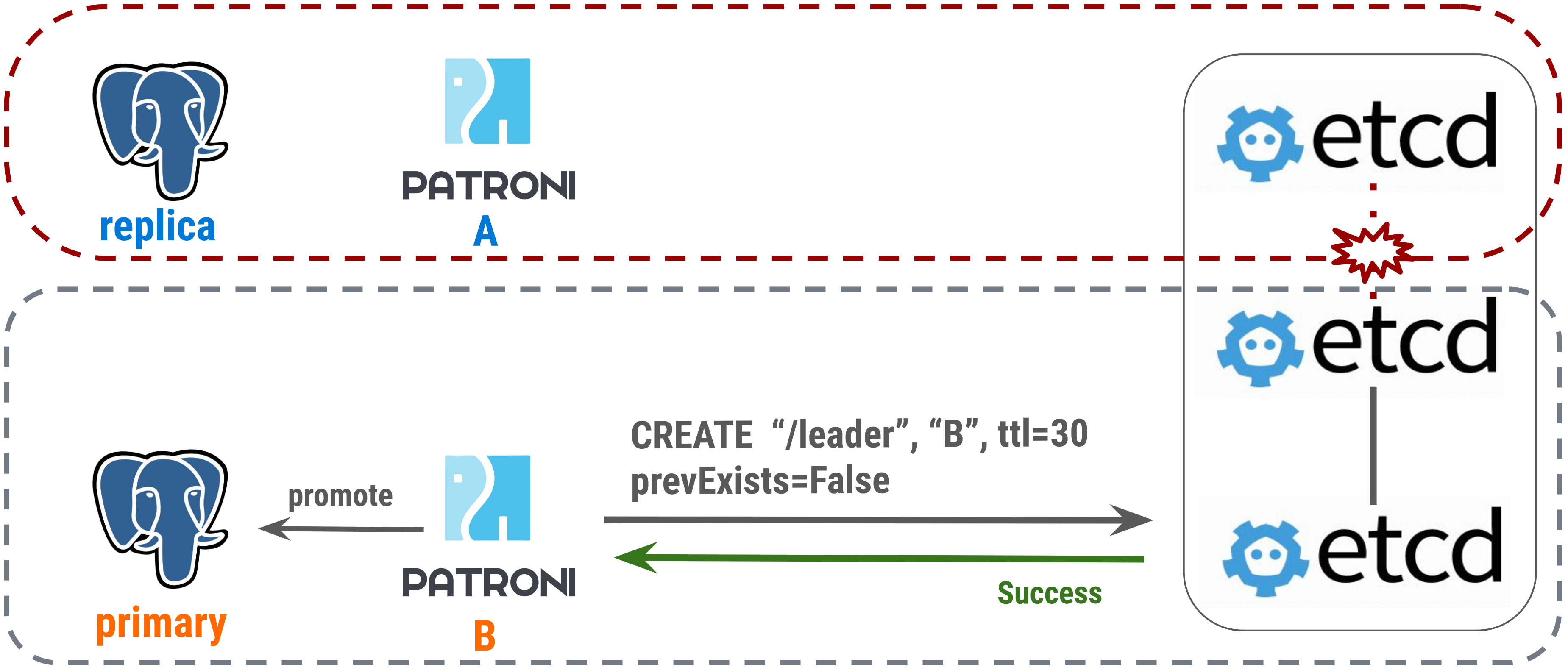
Leader key expired



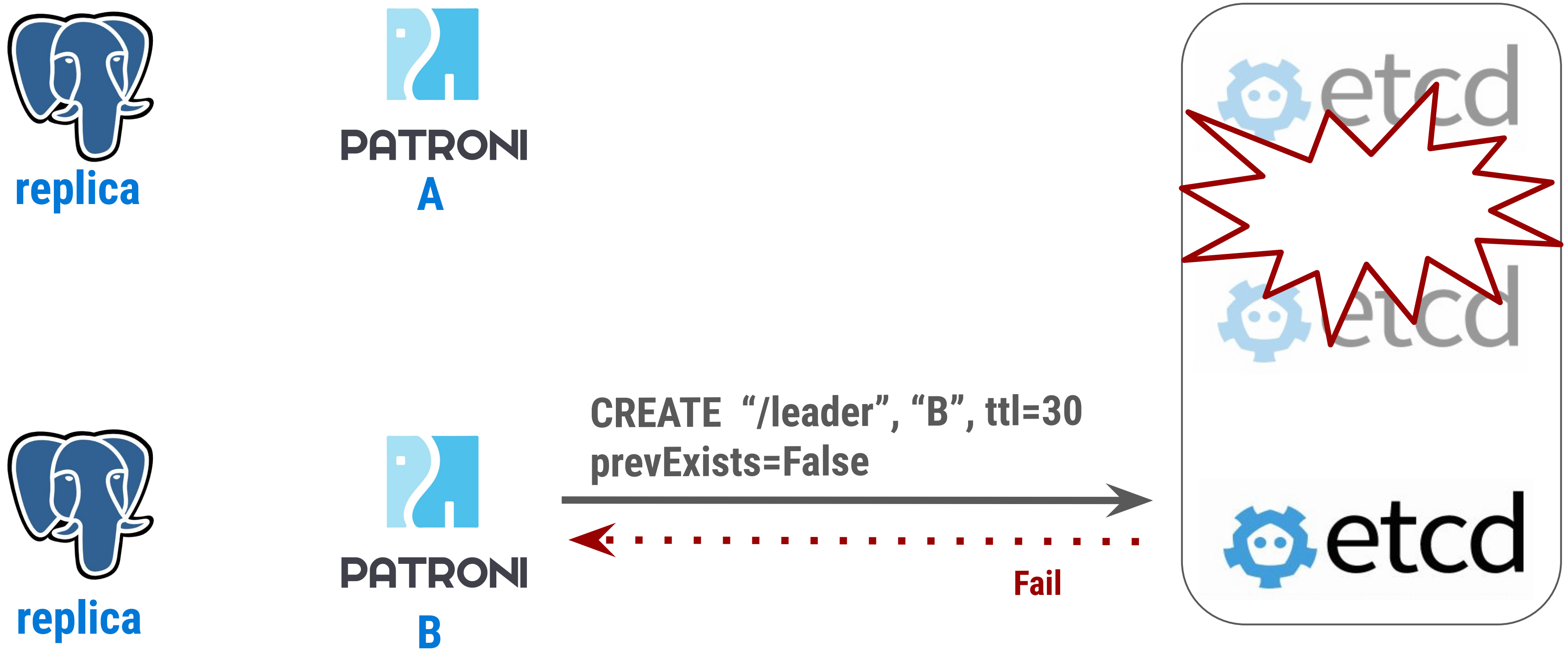
So, to be on the safe side…



So, to be on the safe side…



Still not perfect

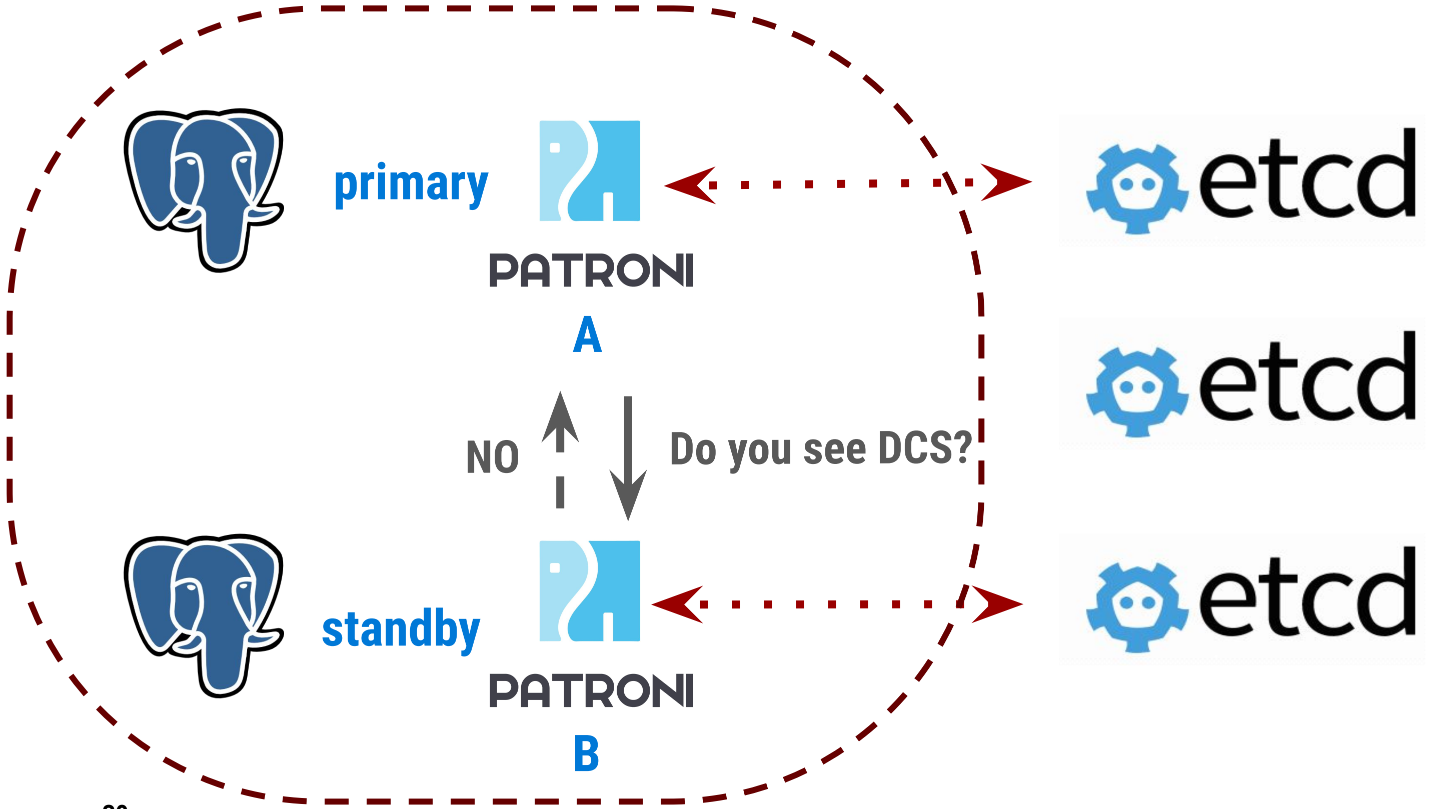


DEMO

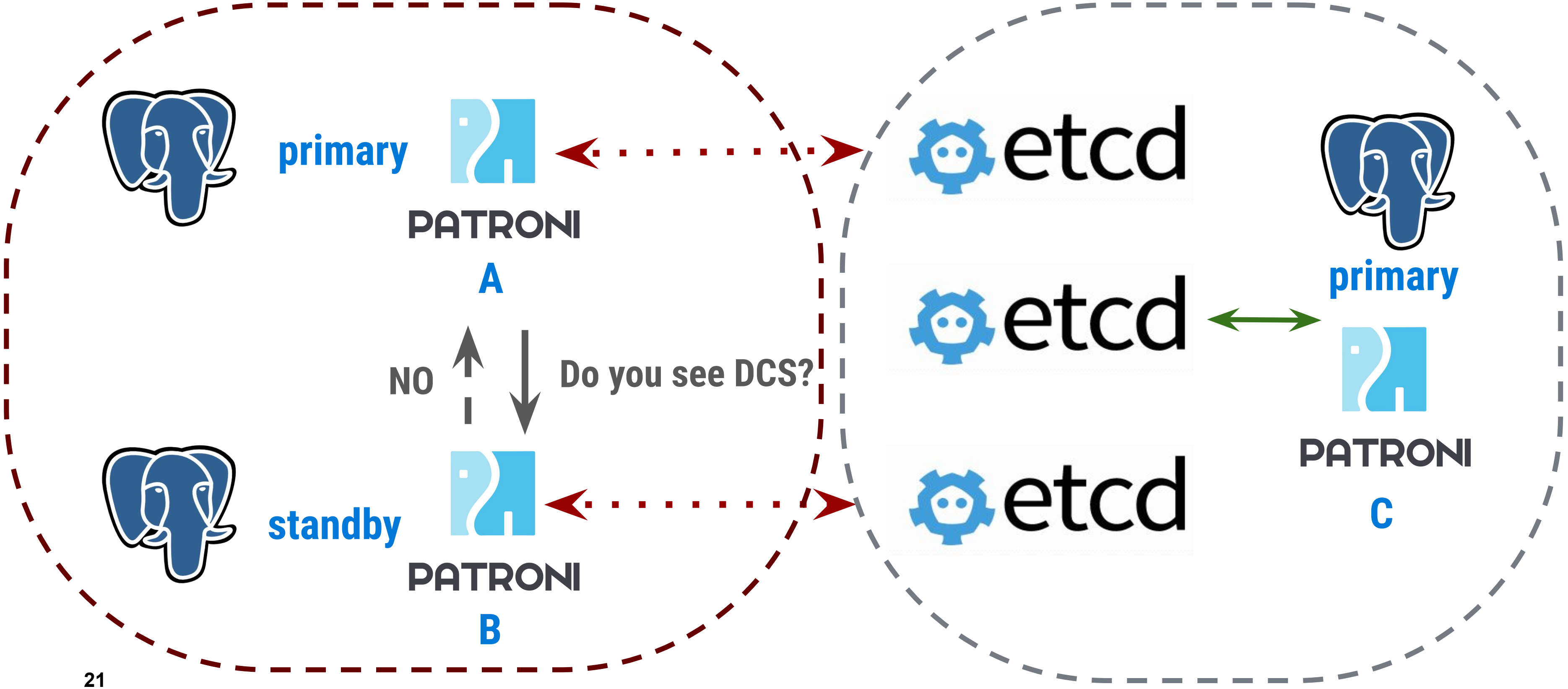
DCS down

- **Etcd, Zookeeper** - very unlikely (if configured correctly)
- **Consul** - local agent is a SPoF!
- **Kubernetes control-plane** - typical SLA for managed services is 99.95% (4h22m per year)

What if…


primary
PATRONI
A
NO
Do you see DCS?
standby
PATRONI
B
etcd
etcd
etcd

Split-brain!


primary
PATRONI
A
NO
Do you see DCS?
standby
PATRONI
B
etcd
etcd
etcd
primary
PATRONI
C

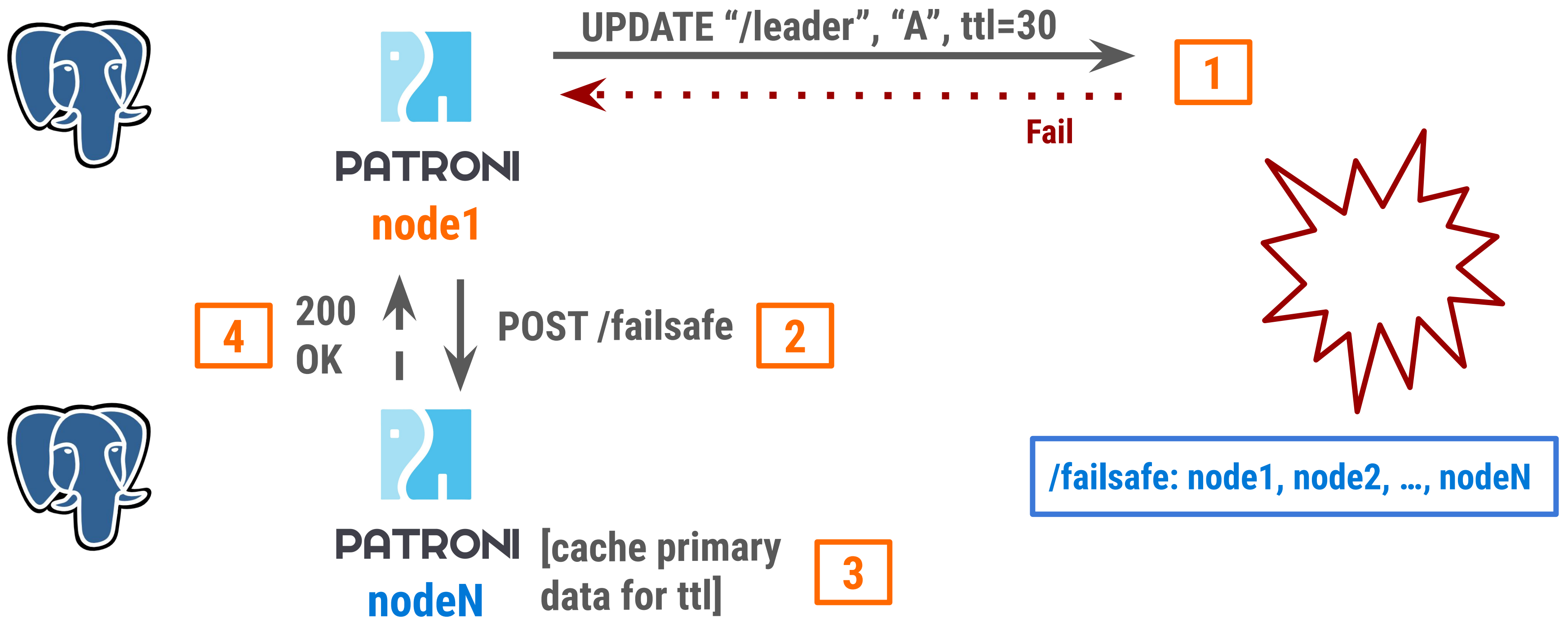
Idea

- Continue to run as primary if can see ALL Patroni nodes
- Don’t allow “unknown” nodes to become primary!

“Unknown” node?

- Patroni clusters are mainly “static”, but nodes can join and leave
- If topology changes - write list of of Patroni nodes names to DCS
- Nodes outside of this list are “**unknown**” and not allowed to become primary

DCS failsafe mode



Implementation details

- Introduce **/failsafe** key - list of **currently presented members** in the cluster
  - Maintained by the leader
  - Cache its value in Patroni (on all nodes)
- Introduce **POST /failsafe** REST API endpoint
  - Payload contains information about primary and permanent logical slots
  - Primary checks response code and demotes if not 200

Implementation details (continue)

- Replica disqualifies itself from the leader race if not listed in the DCS **/failsafe** key
- Primary executes the failsafe check only with nodes from the **failsafe** list
  - Continue as primary if ALL nodes are accessible
  - Otherwise demote
- Replicas call **pg_advance_replication_slot()** if necessary.

How to enable failsafe mode

```
$ patronictl edit-config
---
+++
@@ -4,3 +4,4 @@
  use_pg_rewind: true
retry_timeout: 10
ttl: 30
+failsafe_mode: on

Apply these changes? [y/N]: y
Configuration changed
```

```
$ etcdctl get /service/batman/failsafe
{
 "postgresql0": "http://127.0.0.1:8008/patroni",
 "postgresql1": "http://127.0.0.1:8009/patroni"
}
```

```
$ curl http://127.0.0.1:8008/failsafe
{
 "postgresql0": "http://127.0.0.1:8008/patroni",
 "postgresql1": "http://127.0.0.1:8009/patroni"
}
```

Monitoring

```
$ curl -s http://127.0.0.1:8008/patroni | jq .

{
 "state": "running",
 "postmaster_start_time": "2023-01-26 16:11:04.848424+00:00",
 "role": "master",
 "server_version": 150001,
 "xlog": {"location": 67419584},
 "timeline": 2,
 "replication": [
   {"usename": "replicator", "application_name": "postgresql1", "client_addr": "127.0.0.1",
     "state": "streaming", "sync_state": "async", "sync_priority": 0}],
 "cluster_unlocked": true,
 "failsafe_mode_is_active": true,
 "dcs_last_seen": 1674749503,
 "database_system_identifier": "7192993973708324892",
 "patroni": {"version": "3.0.0", "scope": "demo"}
}
```

DEMO

When not to use it

- When nodes could change their names after “restart” (with old storage)
    - If ALL nodes are restarted at the same time cluster will not recover automatically

Example:

- K8s deployment without StatefulSet
    - Crunchy Postgres Operator (PGO)

Thank you!

Questions?