
REACTIVE SYSTEMS
AT
MOIA
SOCIAL MOVEMENT

Managing an Akka Cluster on Kubernetes

Markus Jura

MOIA

Disclaimer

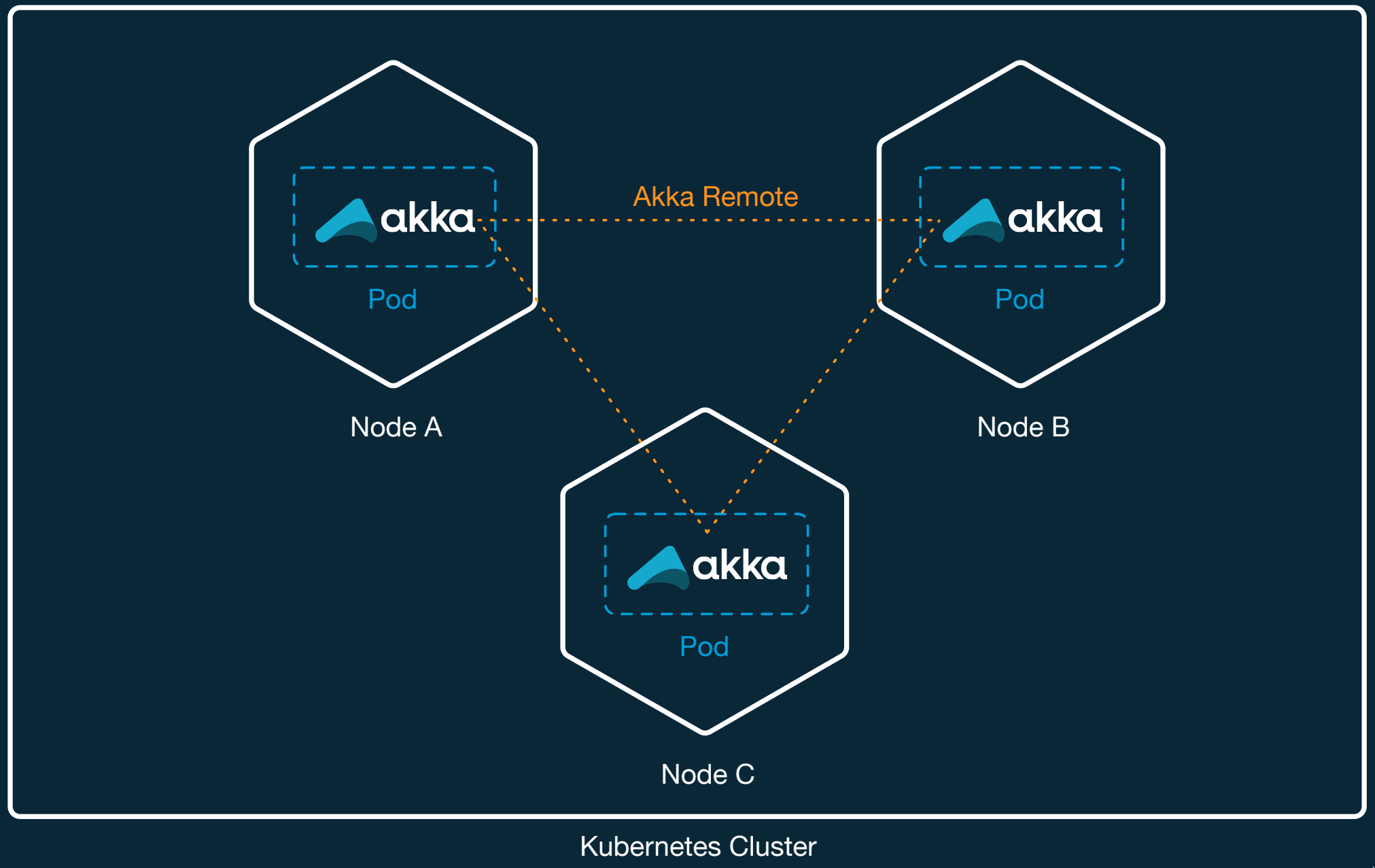
[A career in distributed systems] is both exhilarating and frustrating. When things work, it's like a symphony. When they don't, it's like an eleventh-birthday party where half of the kids are on speed.

— Jeff Darcy, HekaFS (formerly CloudFS) project lead

MOIA

ABOUT ME

- Scala Lead at MOIA
- Previously worked in Lightbend ConductR team
- Scala / Akka since 7 years


akka
Pod
Akka Remote
akka
Pod
Node A
Node B
akka
Pod
Node C
Kubernetes Cluster

MOIA

BOOTSTRAPPING AKKA CLUSTER

MOIA

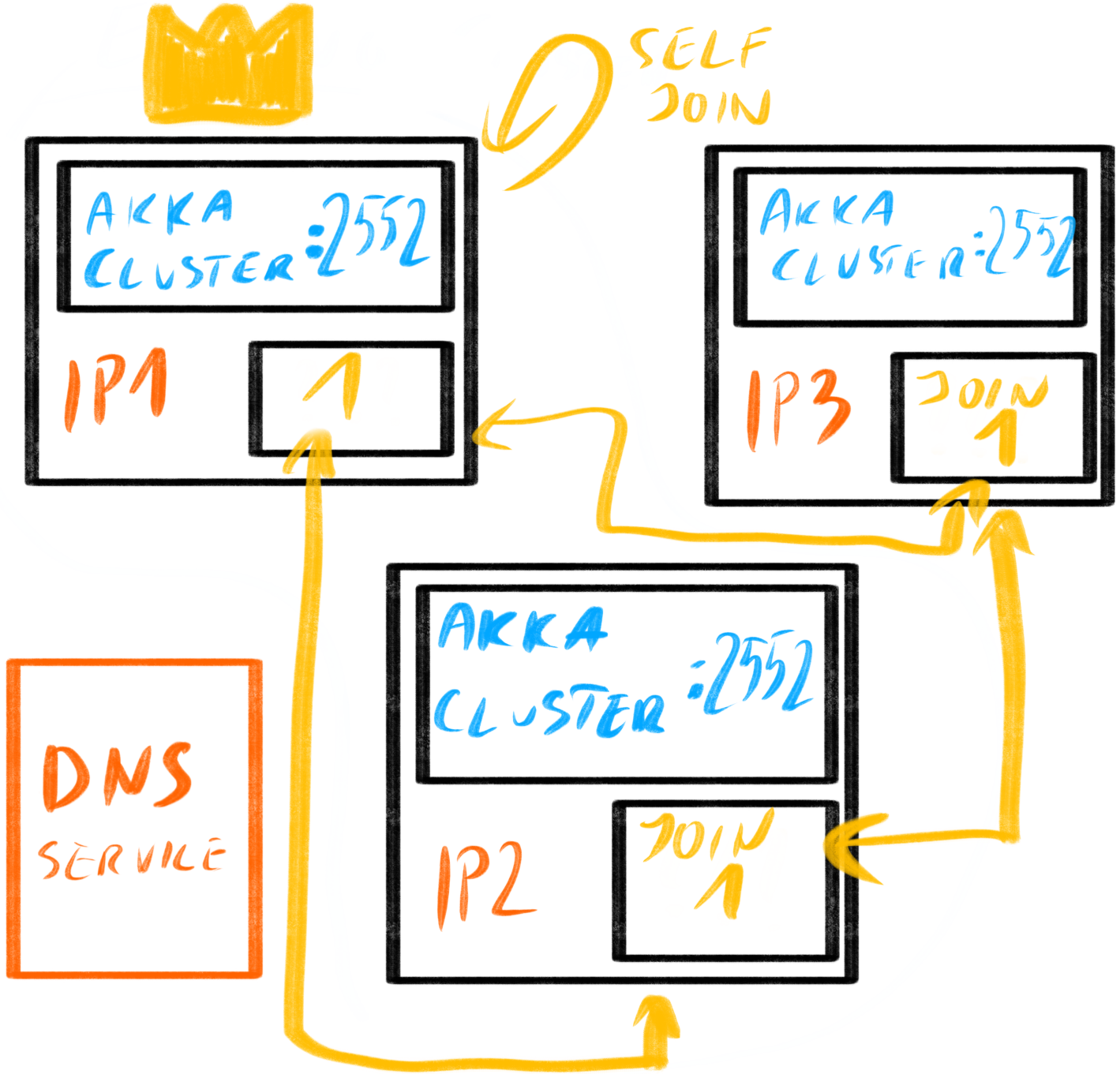
PHASES

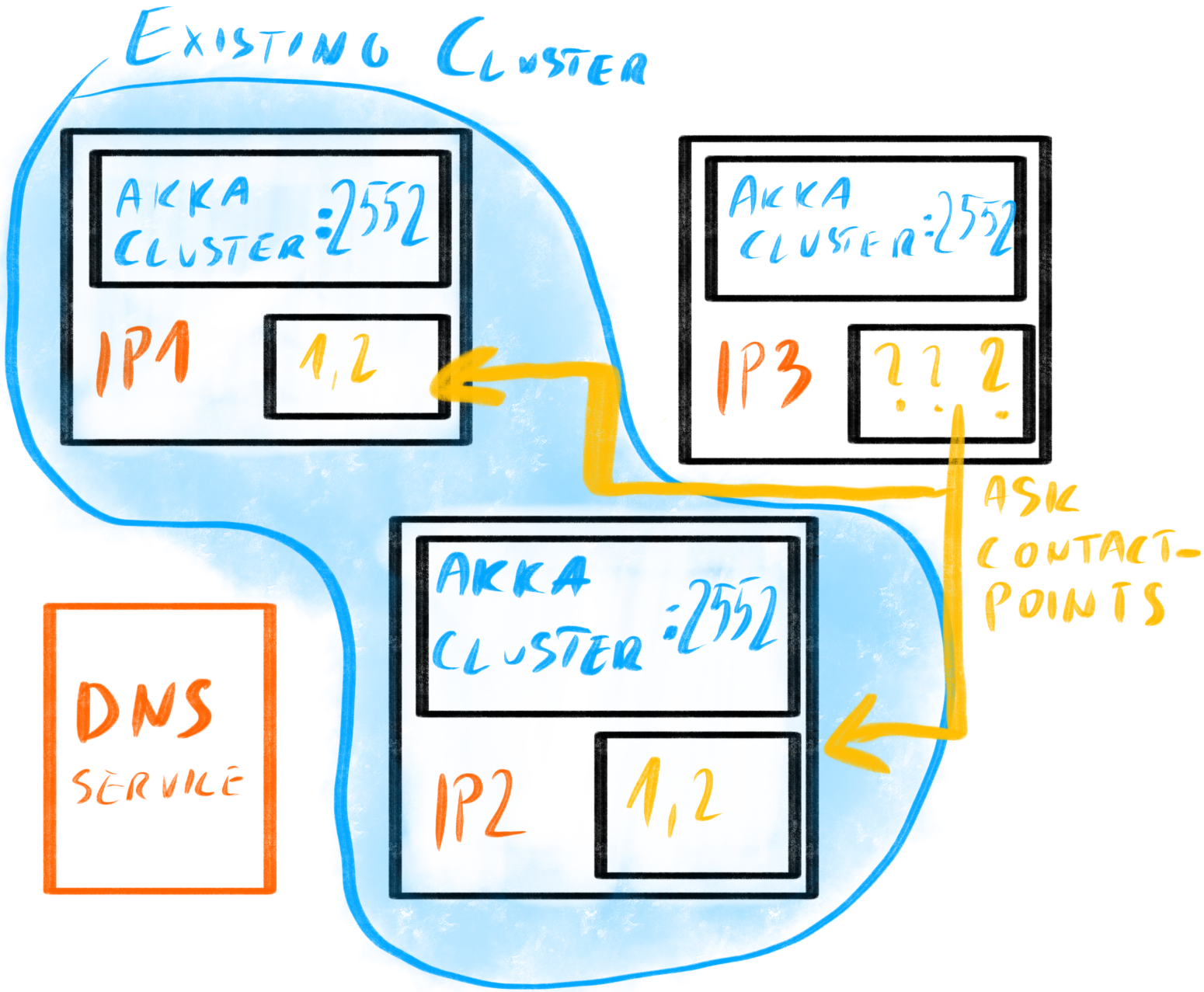
01/ CONTACT POINT PROBING

- Based on DNS (Kubernetes API)
- By actor system name
- Ask for known seed nodes

02/ EPIDEMIC JOINING

- Join cluster by seed nodes
- Advertise own contact point as seed node


SELF JOIN
AKKA CLUSTER :2552
IP1
1
AKKA CLUSTER :2552
IP3
JOIN 1
AKKA CLUSTER :2552
IP2
JOIN 1
DNS SERVICE


EXISTING CLUSTER
AKKA CLUSTER :2552
IP1
1, 2
AKKA CLUSTER :2552
IP3
? ? ?
AKKA CLUSTER :2552
IP2
1, 2
ASK CONTACT-POINTS
DNS SERVICE

DEMO

MOIA

COORDINATED SHUTDOWN

MOIA

REGISTER STOP LOGIC

```
import akka.actor.CoordinatedShutdown
import akka.actor.CoordinatedShutdown.PhaseBeforeClusterShutdown

CoordinatedShutdown(system)
  .addTask(PhaseBeforeClusterShutdown, "my.module") { () =>
    // Add graceful stop logic here
  }
```

MOIA

TRIGGER COORDINATED SHUTDOWN

```
import akka.actor.CoordinatedShutdown.Reason

private final case object MyShutdownReason extends Reason

CoordinatedShutdown(system).run(MyShutdownReason)
```

MOIA

PHASES

01/ BeforeServiceUnbind

02/ ServiceUnbind

03/ ServiceRequestsDone

04/ ServiceStop

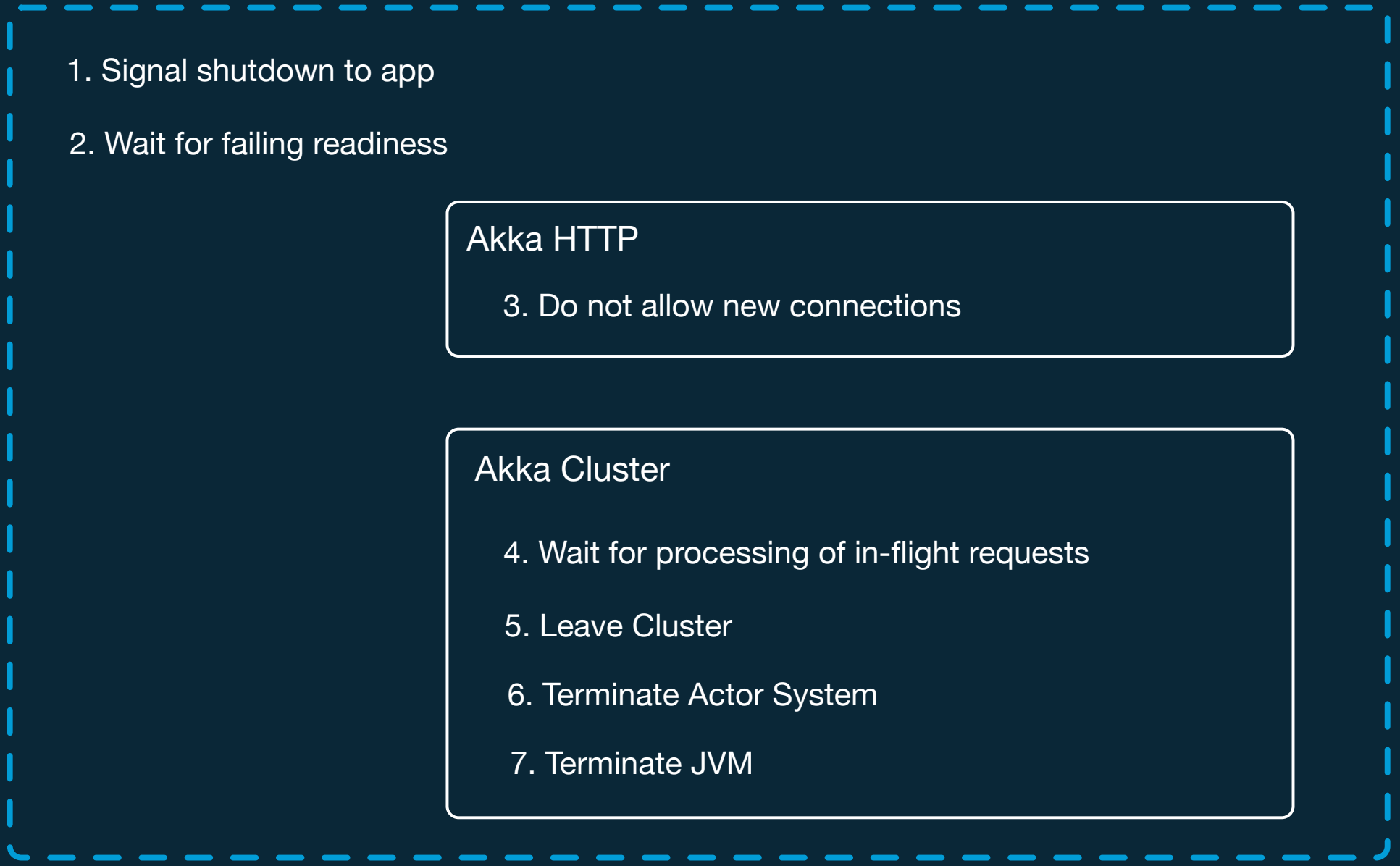
05/ BeforeClusterShutdown

06/ ClusterShardingShutdownRegion

...

12/ PhaseActorSystemTerminate

MOIA


1. Signal shutdown to app
2. Wait for failing readiness
Akka HTTP
3. Do not allow new connections
Akka Cluster
4. Wait for processing of in-flight requests
5. Leave Cluster
6. Terminate Actor System
7. Terminate JVM
Pod

CODE

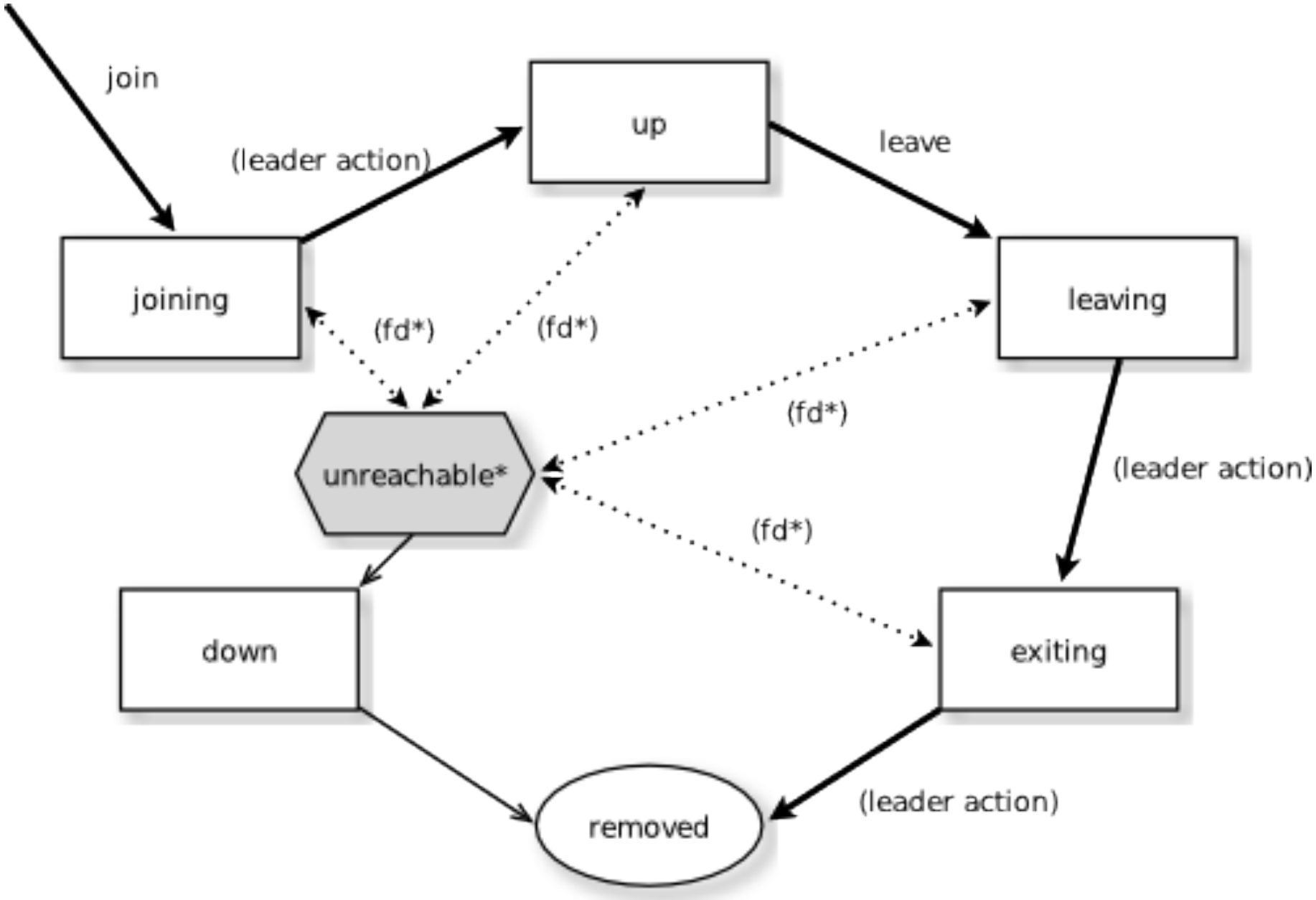
MOIA

SPLIT BRAIN RESOLUTION

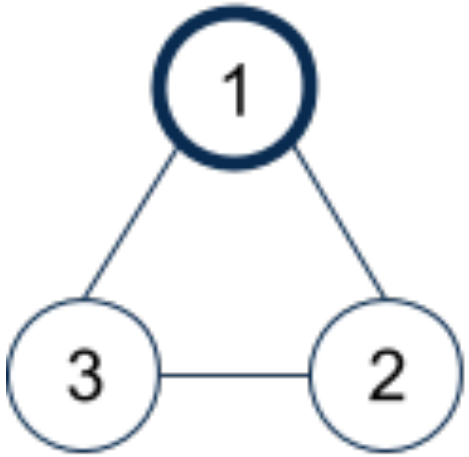
MOIA

OVERVIEW

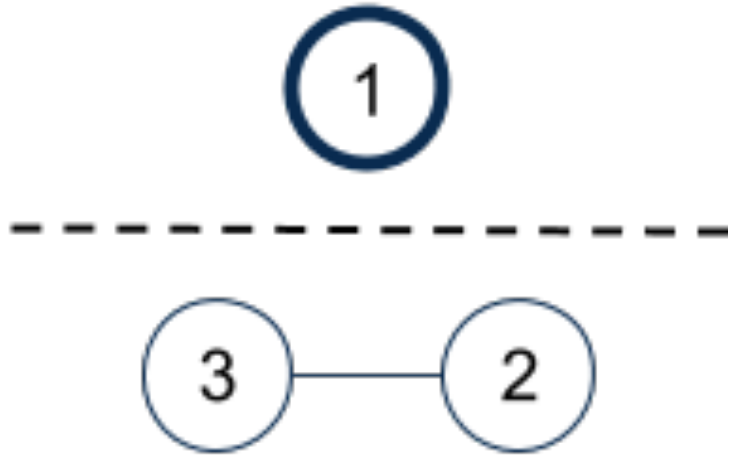
- Akka Cluster failure detector marks nodes as unreachable
  - Periodic heart-beats to check reachability
- Split Brain Resolver decides which unreachable nodes to down


join
(leader action)
up
leave
joining
(fd*)
(fd*)
leaving
(fd*)
unreachable*
(leader action)
(fd*)
down
exiting
removed
(leader action)

Initial state



Split-brain



Leader node

MOIA

STRATEGIES

- Static Quorum
- Keep Majority
- Keep Oldest
- Keep Referee

LIBRARIES

COMMERCIAL

- Lightbend SBR (recommended)

OPEN SOURCE

- Akka Cluster Custom Downing
- Akka Down Resolver
- Simple Akka Downing

Q & OPTION[A]

MOIA


REACTIVE SYSTEMS
AT
MOIA
SOCIAL MOVEMENT