



AKKA & KUBERNETES: THE HOLY GRAIL TO ELASTICITY?





Fabio Tiriticco
Scala / Akka consultant
@ticofab



Reactive
Amsterdam





@adamsand0r



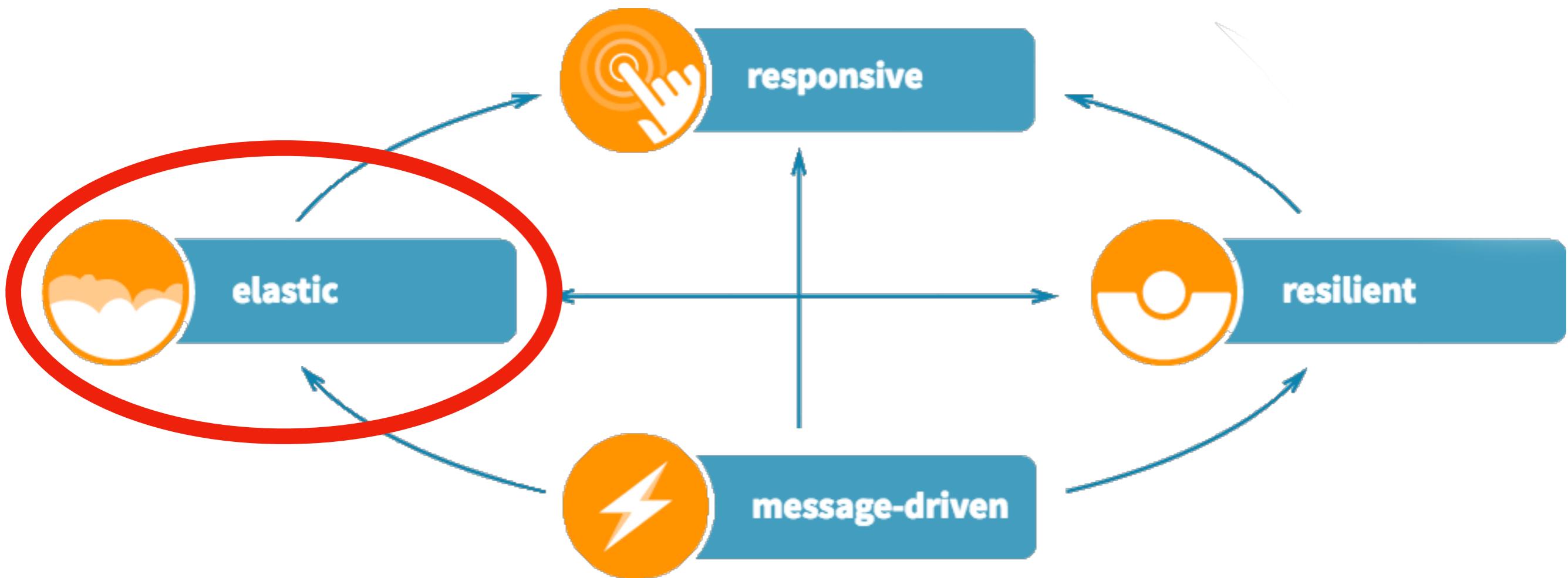
- My co-speaker is on holiday
- Only relevant concepts are introduced
- Slides use pseudo-code. Real deal on GitHub

The Reactive Manifesto

@ticofab



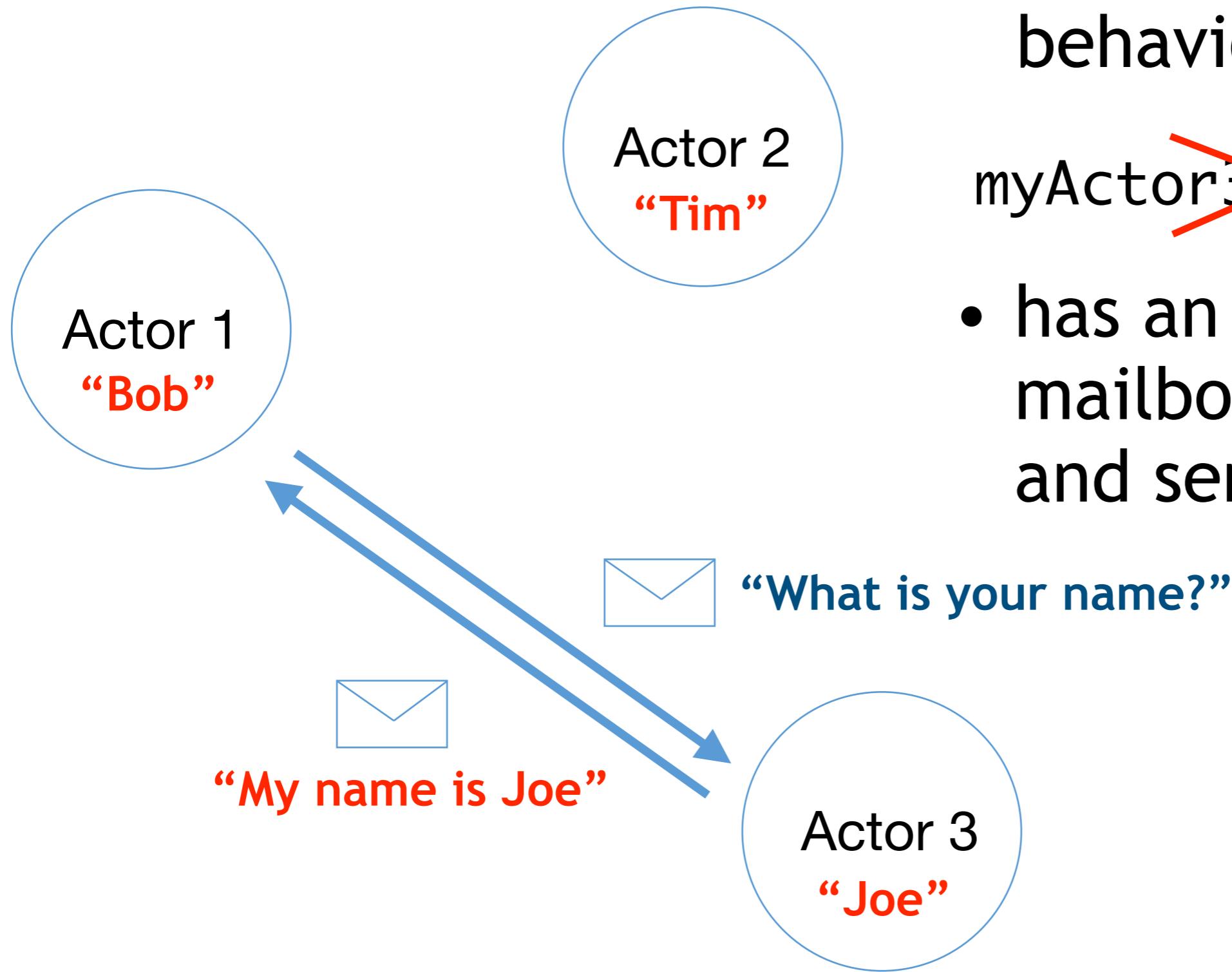
2013, J. Boner, R. Kuhn, D. Farley, M. Thompson



Elasticity: “use just the right amount”

Akka's building blocks: Actors

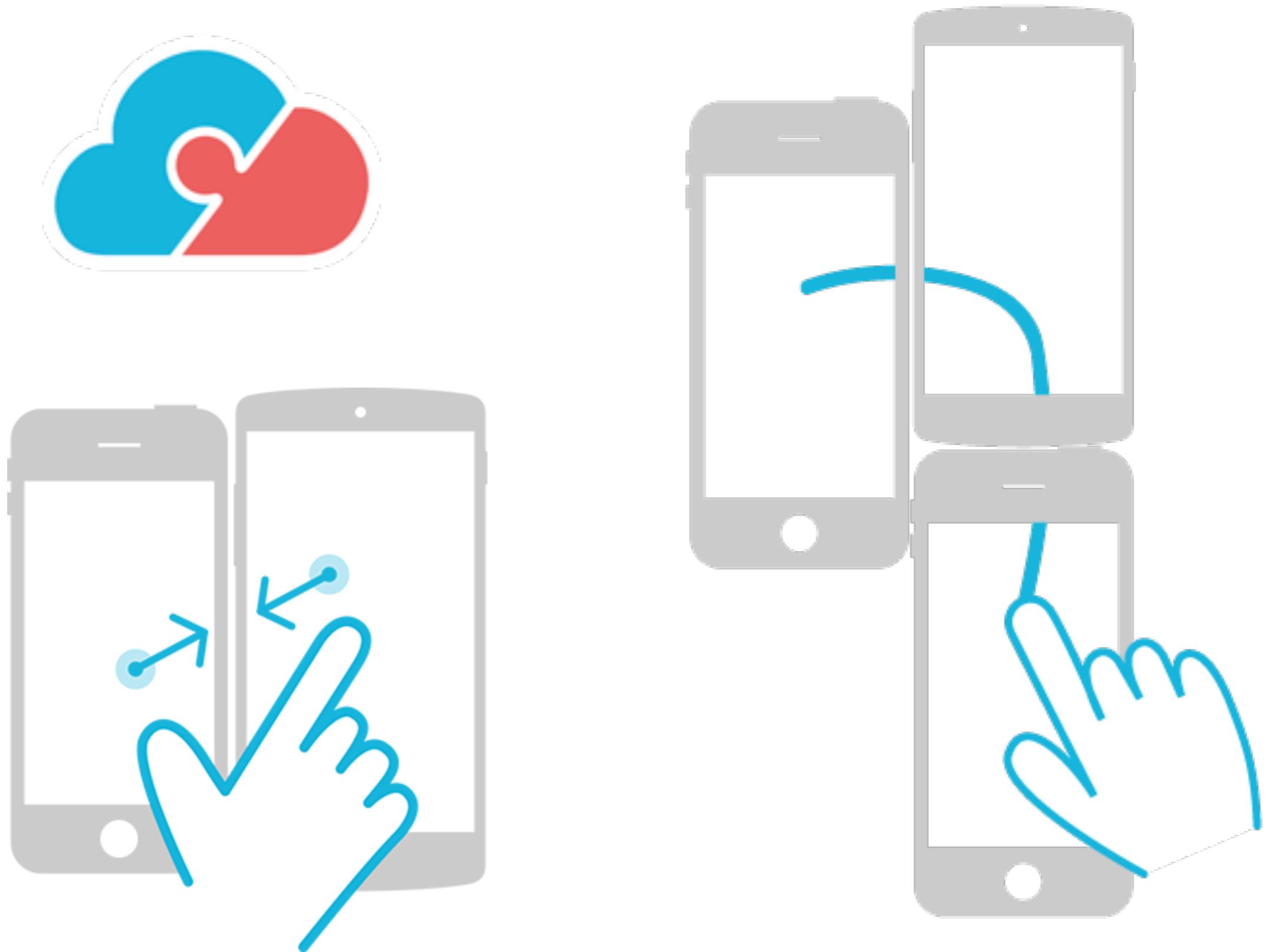
@ticofab



- contains state & behaviour logic
~~myActor3.getName()~~
- has an addressable mailbox to receive and send messages

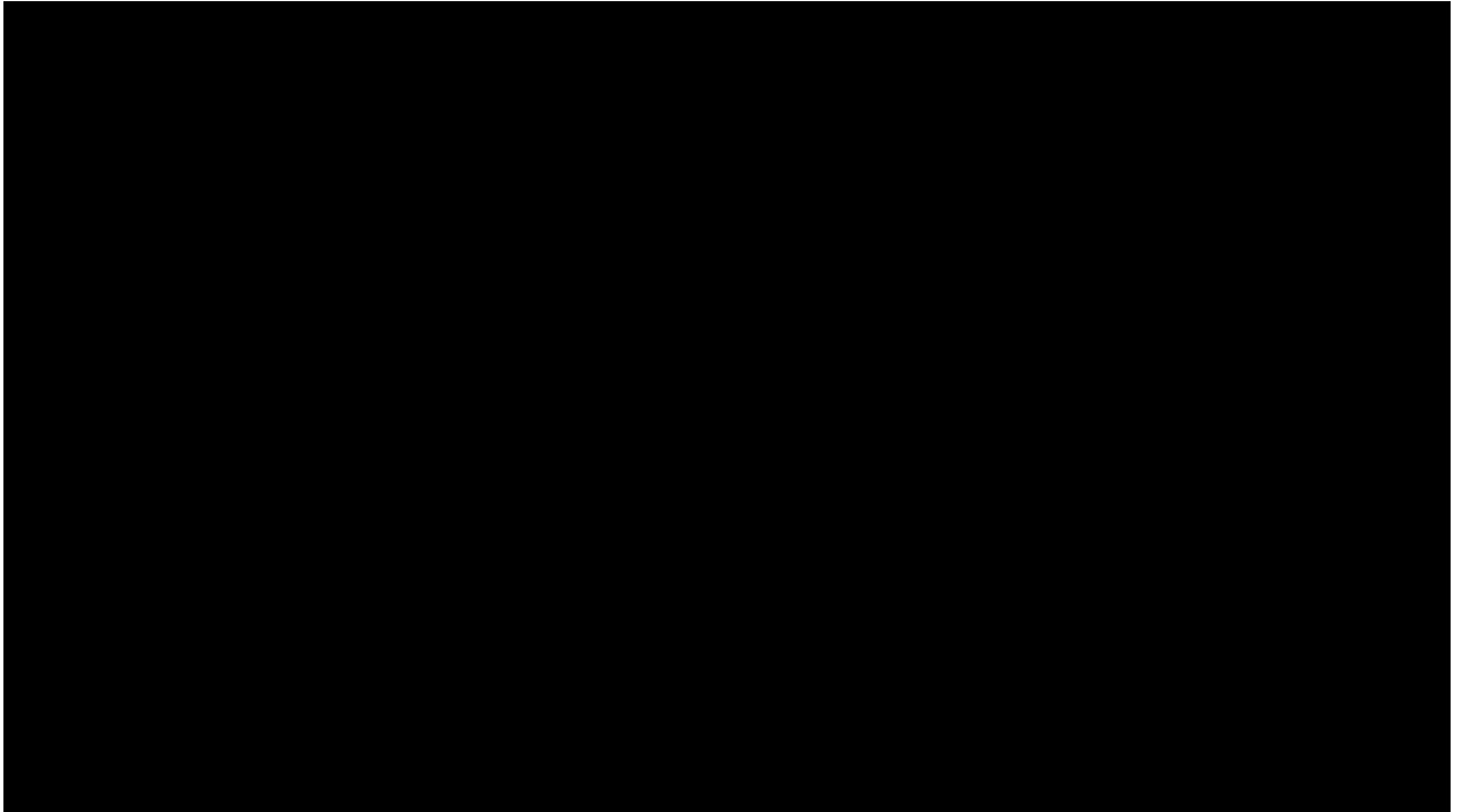
My first journey in the Reactive world

@ticofab



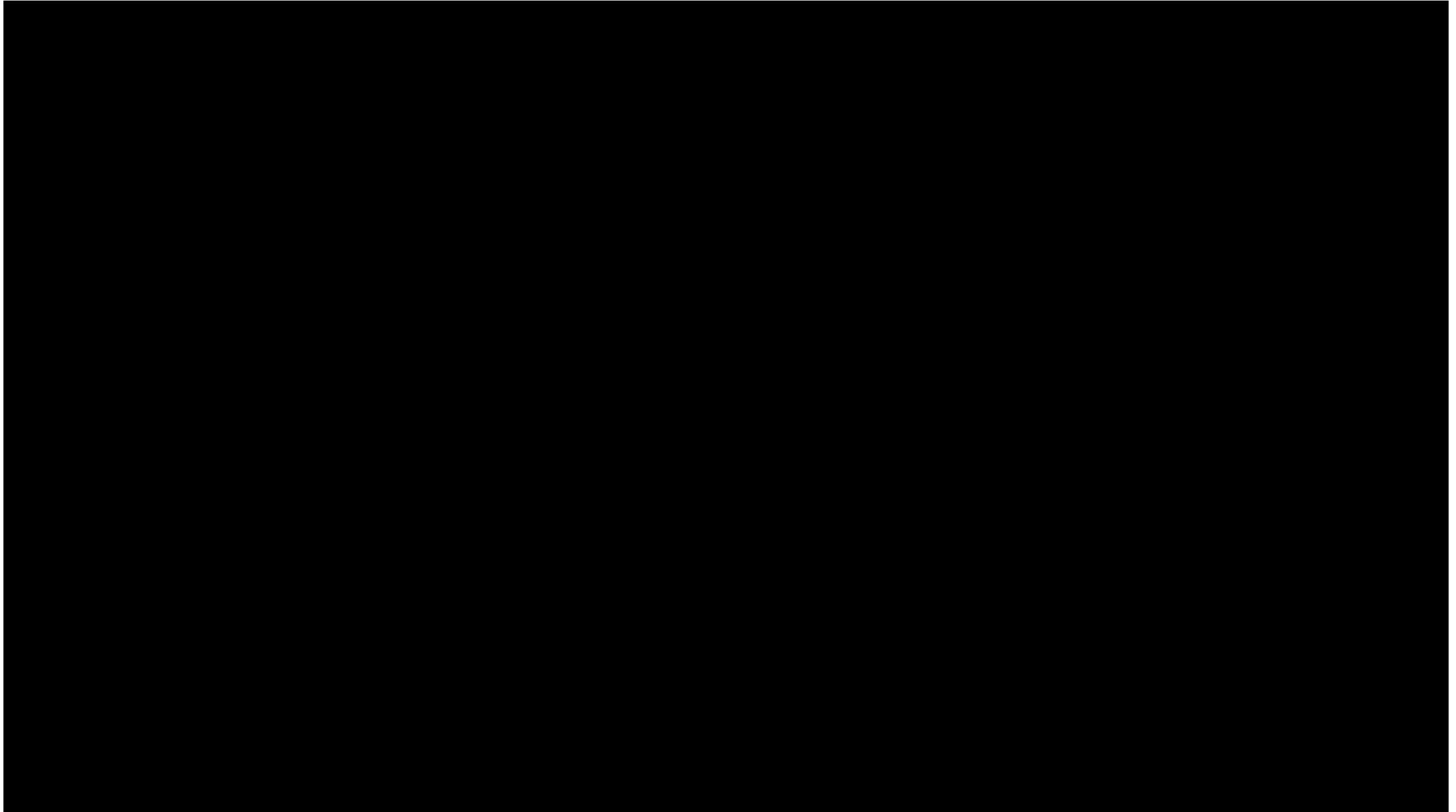
My first journey in the Reactive world

@ticofab



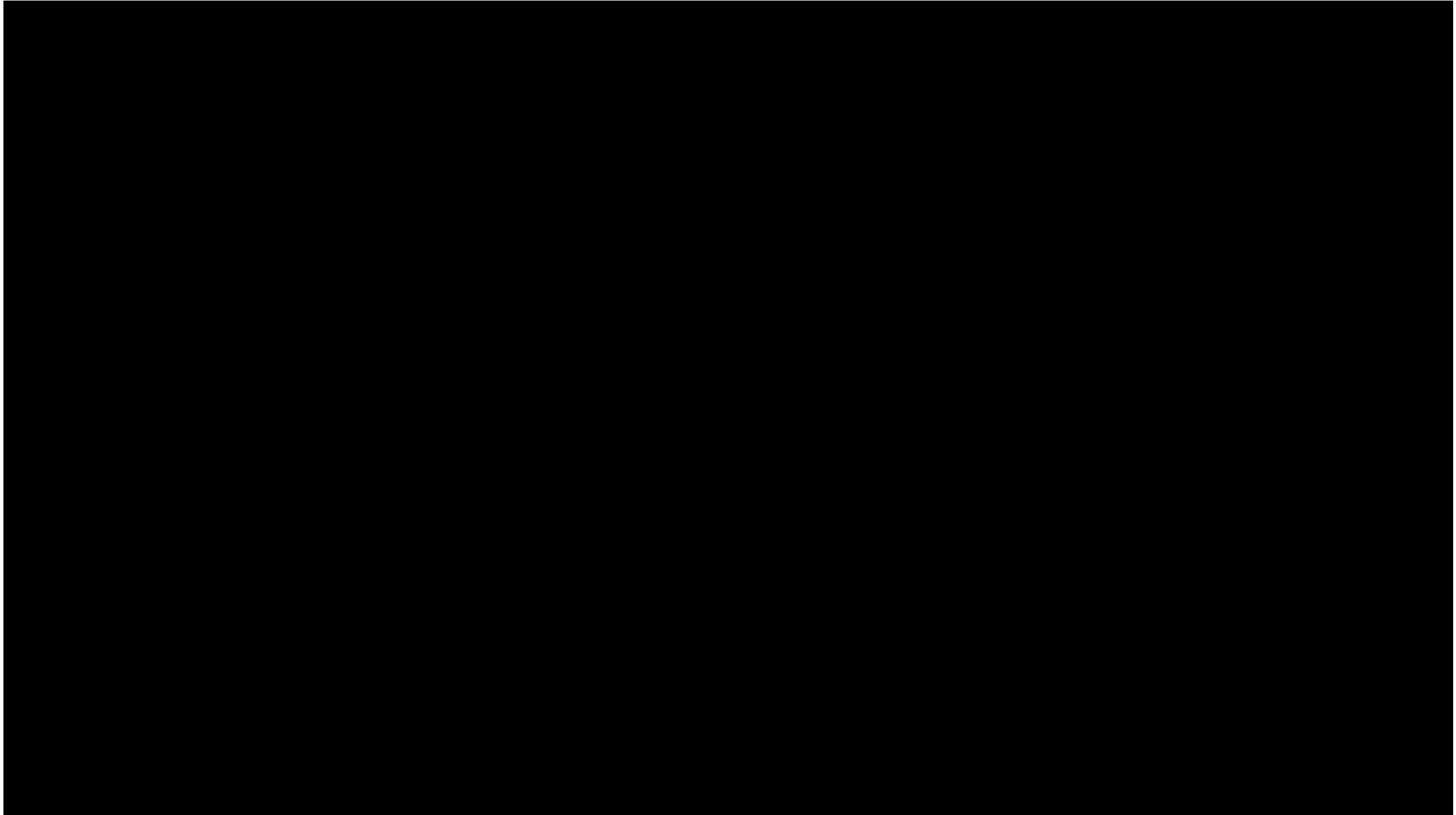
My first journey in the Reactive world

@ticofab



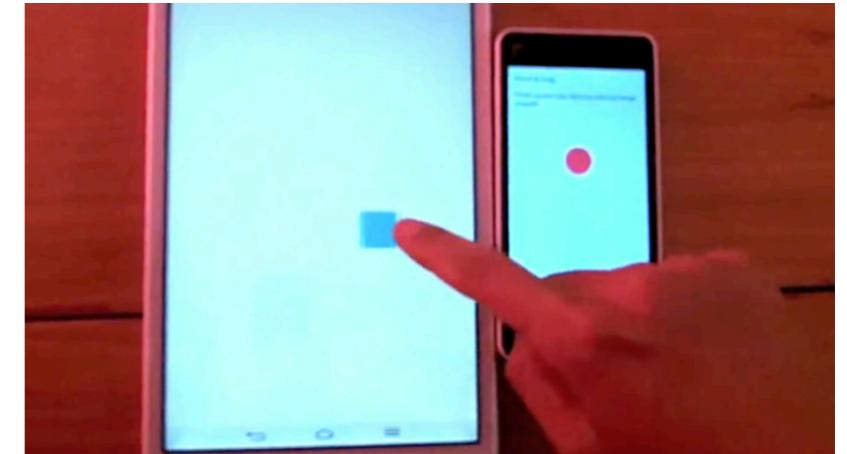
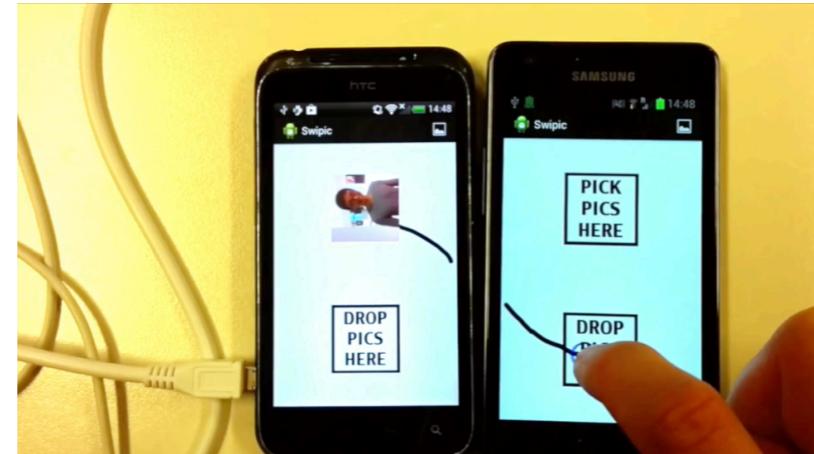
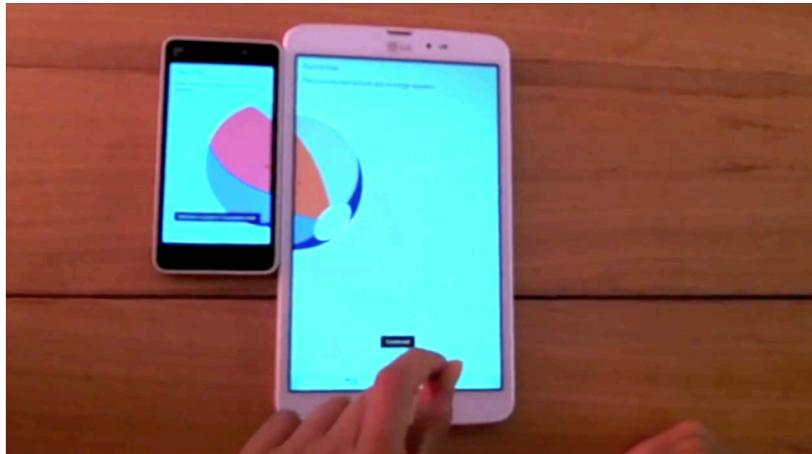
My first journey in the Reactive world

@ticofab



Engine requirements

@ticofab

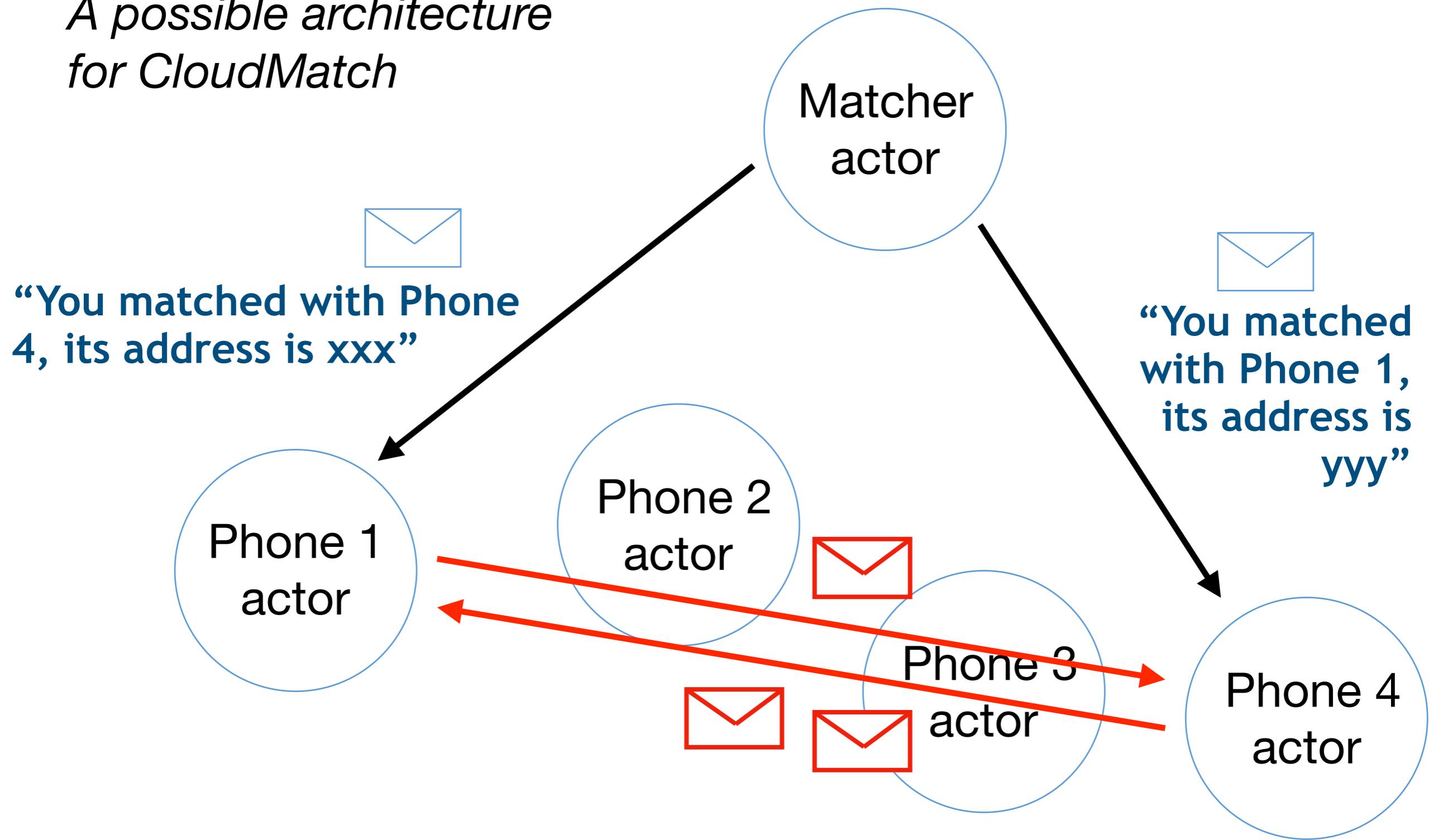


- Manage many devices concurrently
- Hold state for each device
- Deal with short device lifecycles
- FAST bi-directional channel between devices

My first journey in the Reactive world

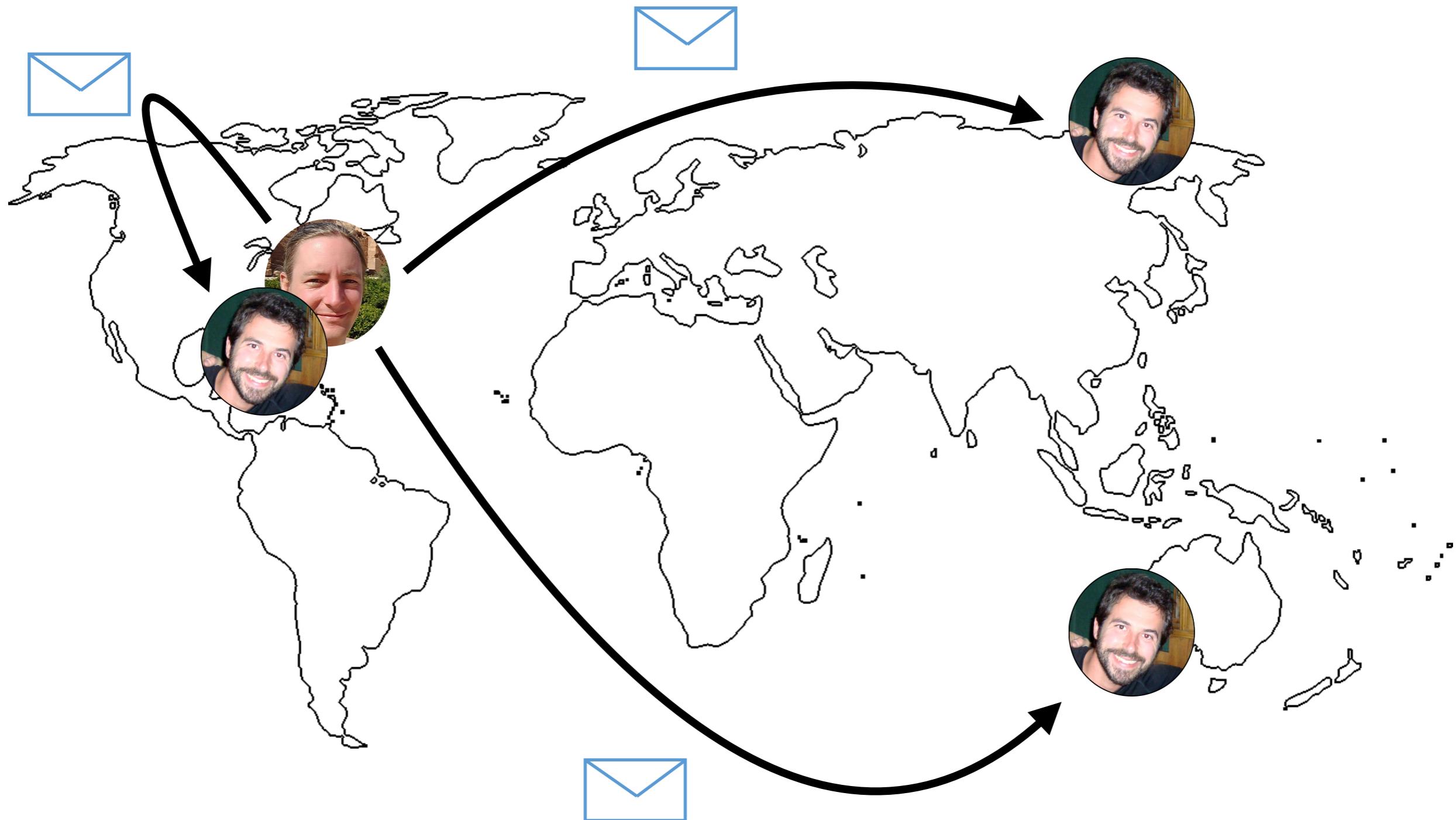
@ticofab

*A possible architecture
for CloudMatch*



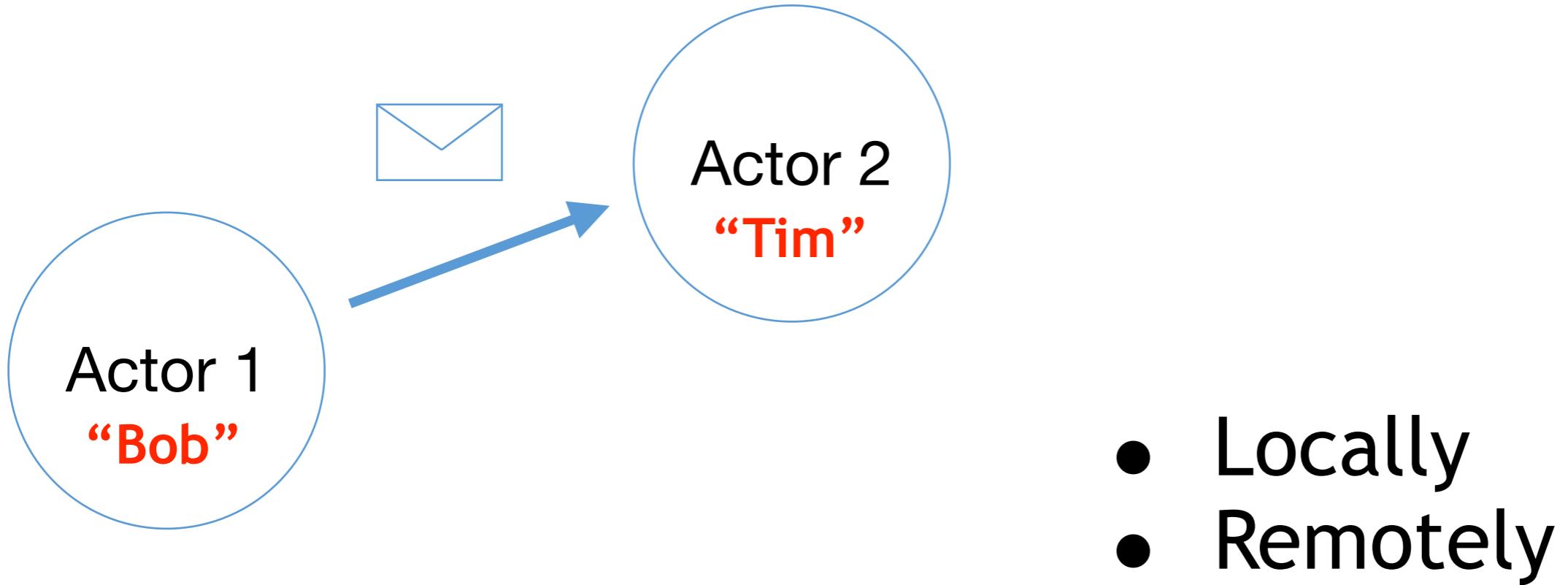
Location transparency

@ticofab



Location transparency

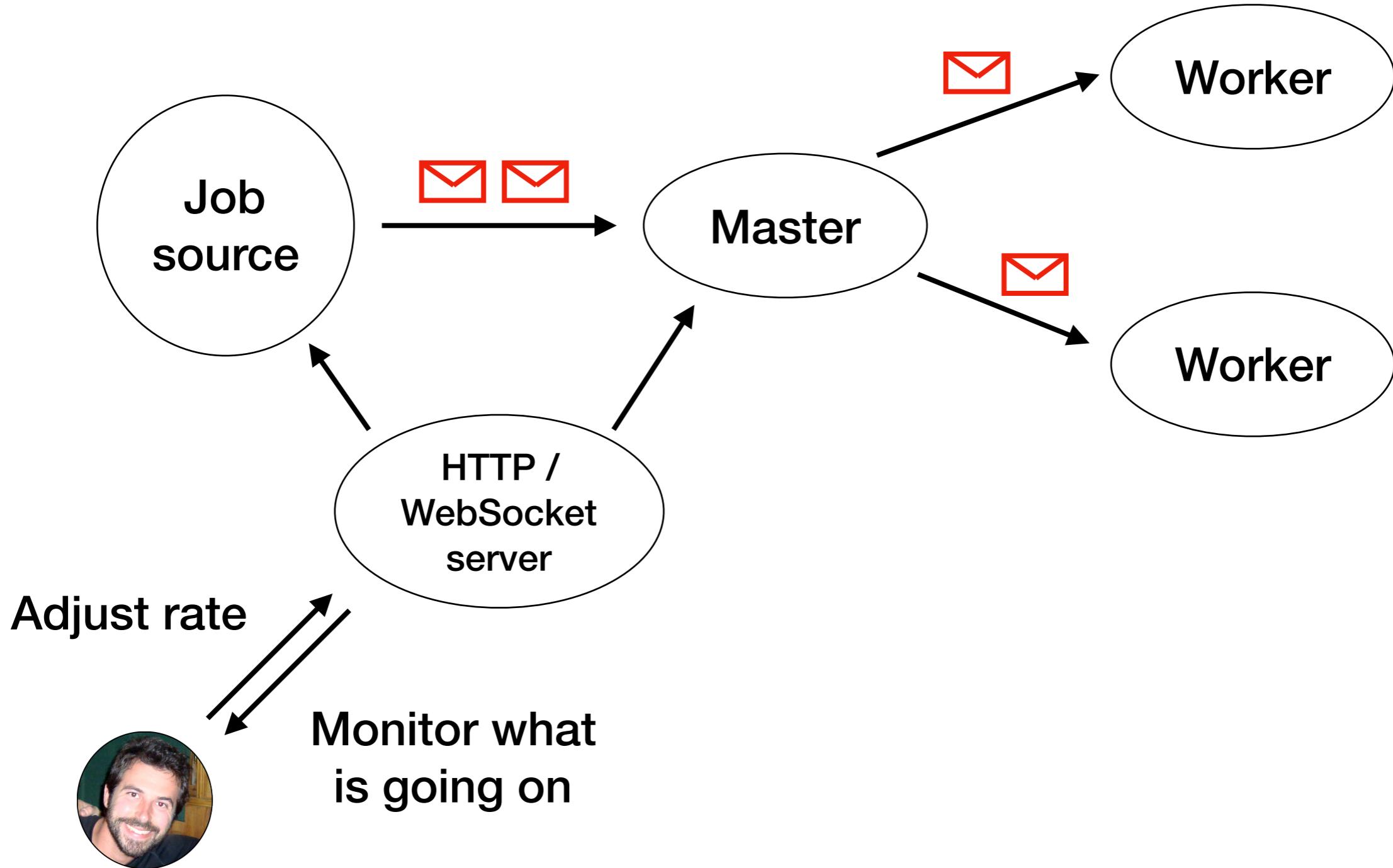
@ticofab



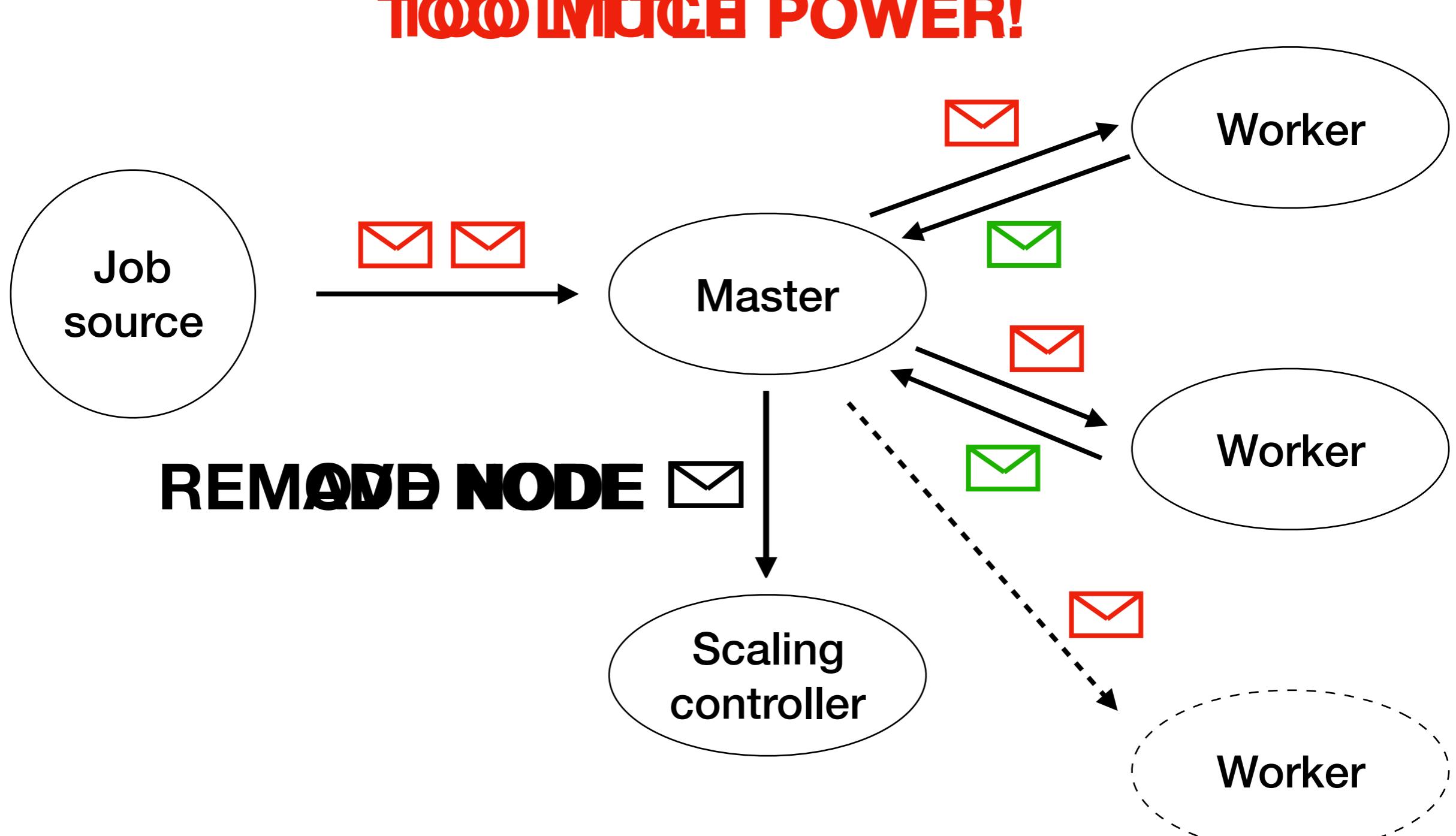
```
// in Scala + Akka  
timAddress ! Message
```

Elasticity test system

@ticofab

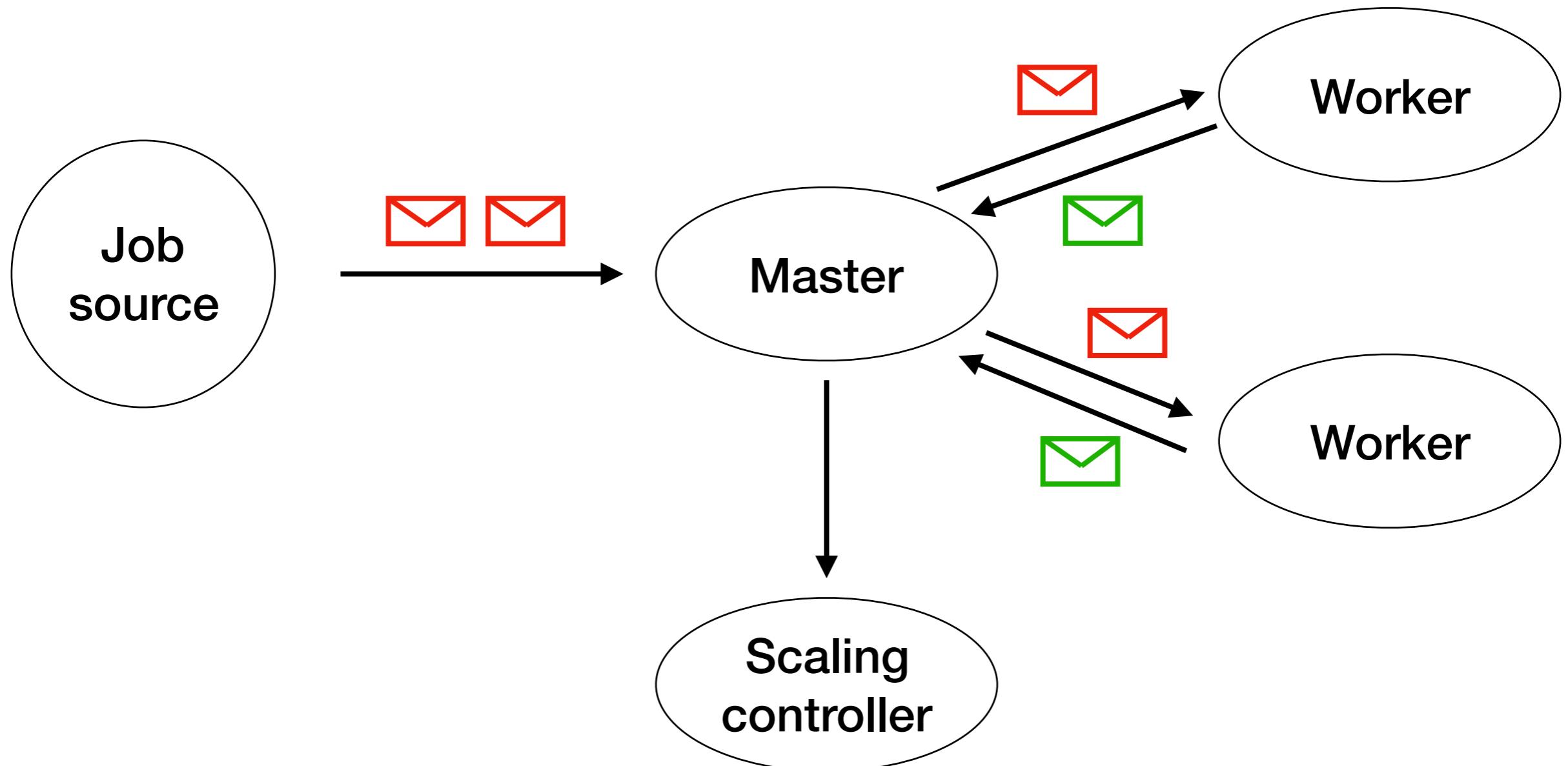


```
class Worker extends Actor {  
    // ...  
    case job: Job =>  
        // ... simulate a CPU-intensive workload  
        master ! JobCompleted(job.number)  
}
```



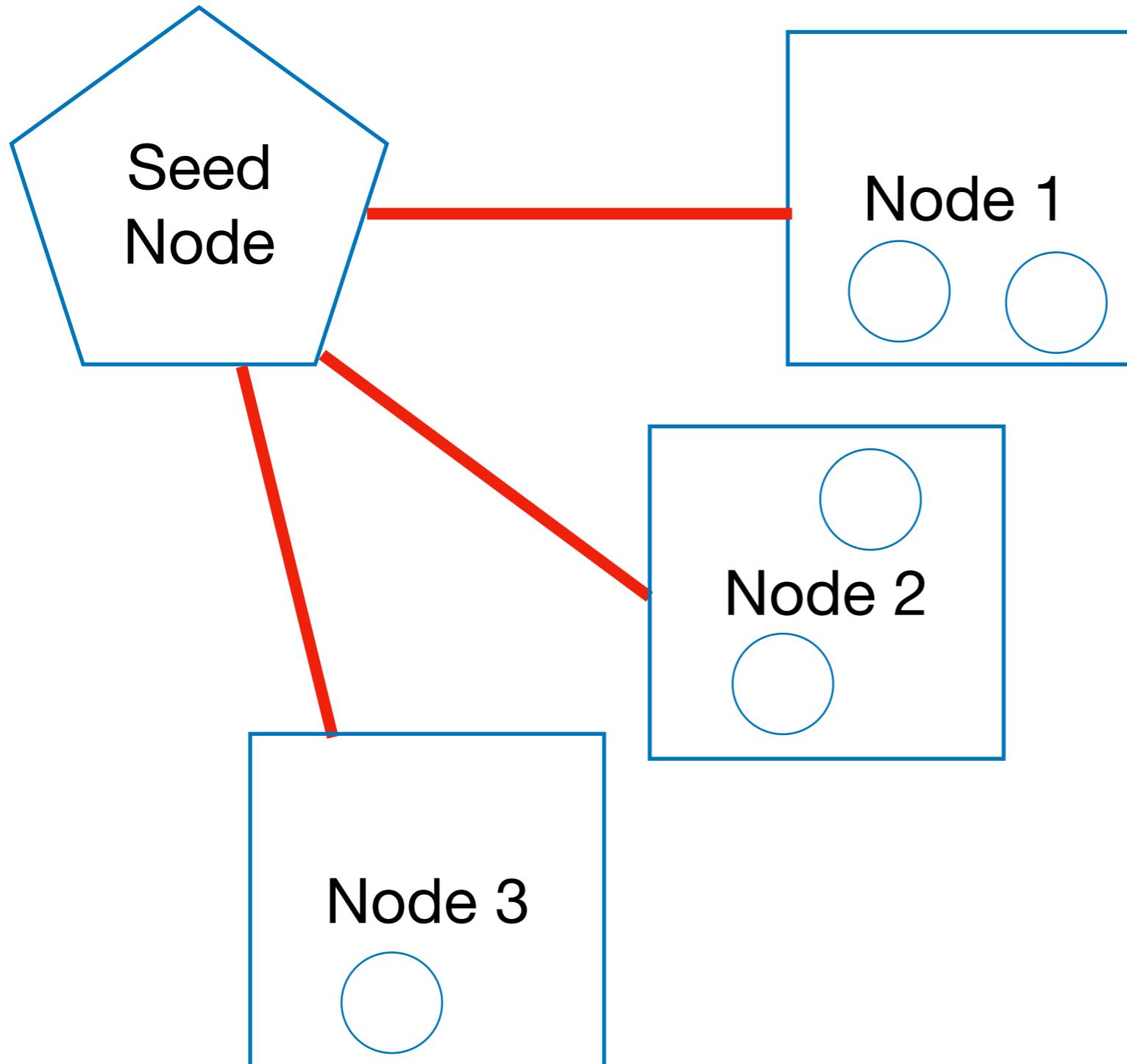
Issue #1: how to distribute this?

@ticofab



Akka Cluster to the rescue!

@ticofab

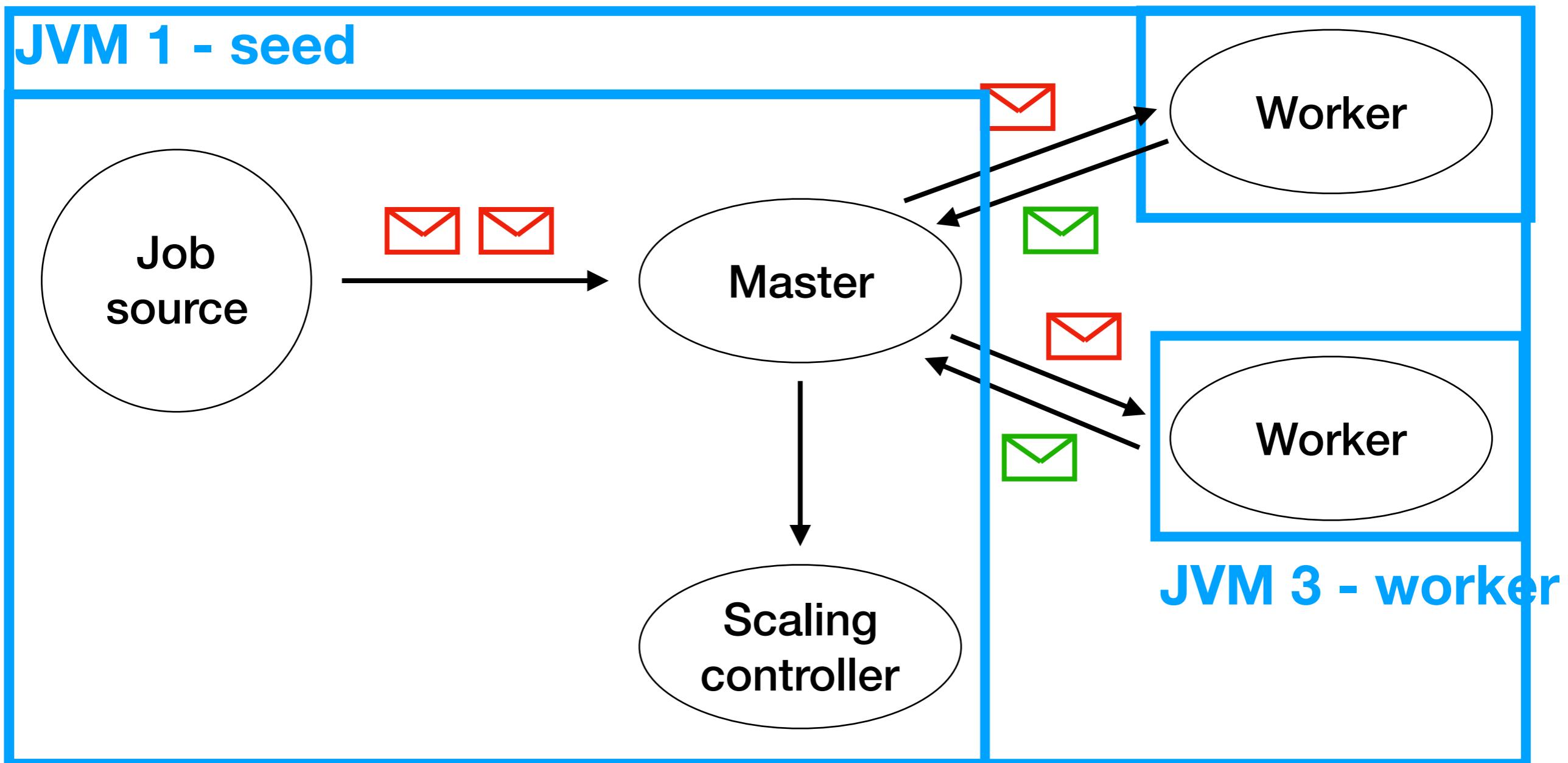


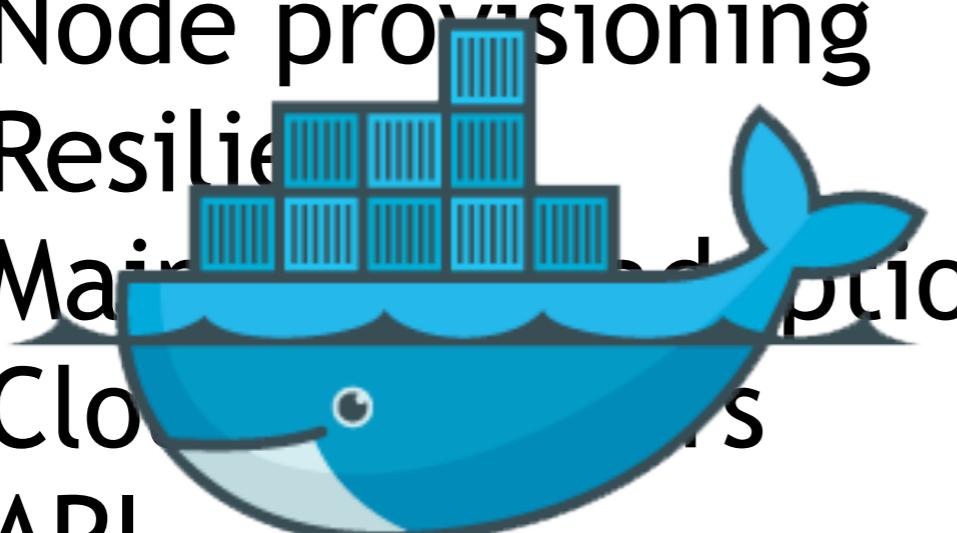
- Joining
- Up
- Leaving
- Unreachable
- ...

NOTE: nodes need to know where the seed is

Moving #2 workers to shift from and shutdown nodes?

Single JVM

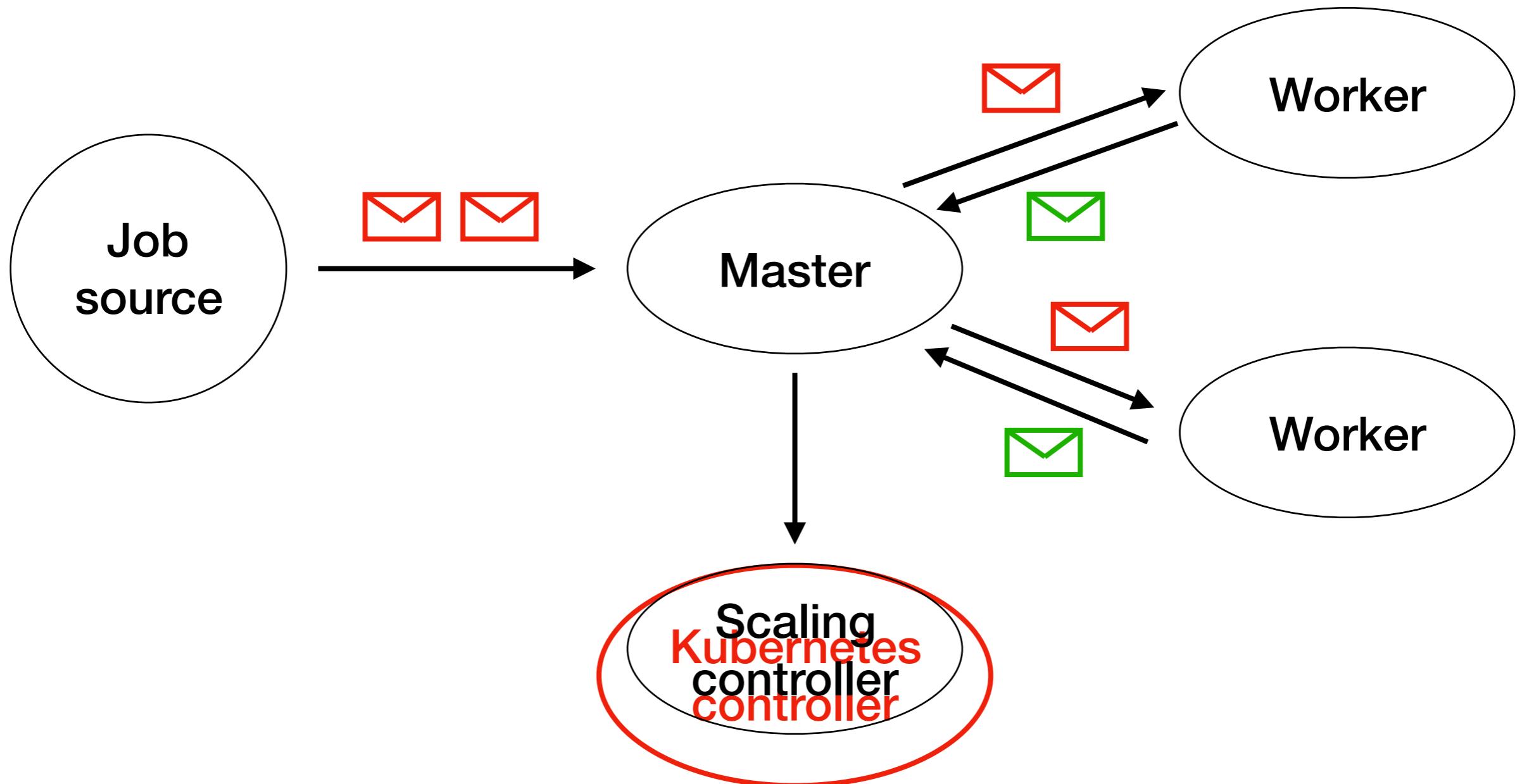


- Node provisioning
 - Resiliency
 - Maintenance and upgrade
 - Cloud portability
 - API
- 



Scaling Controller → K8S Controller

@ticofab



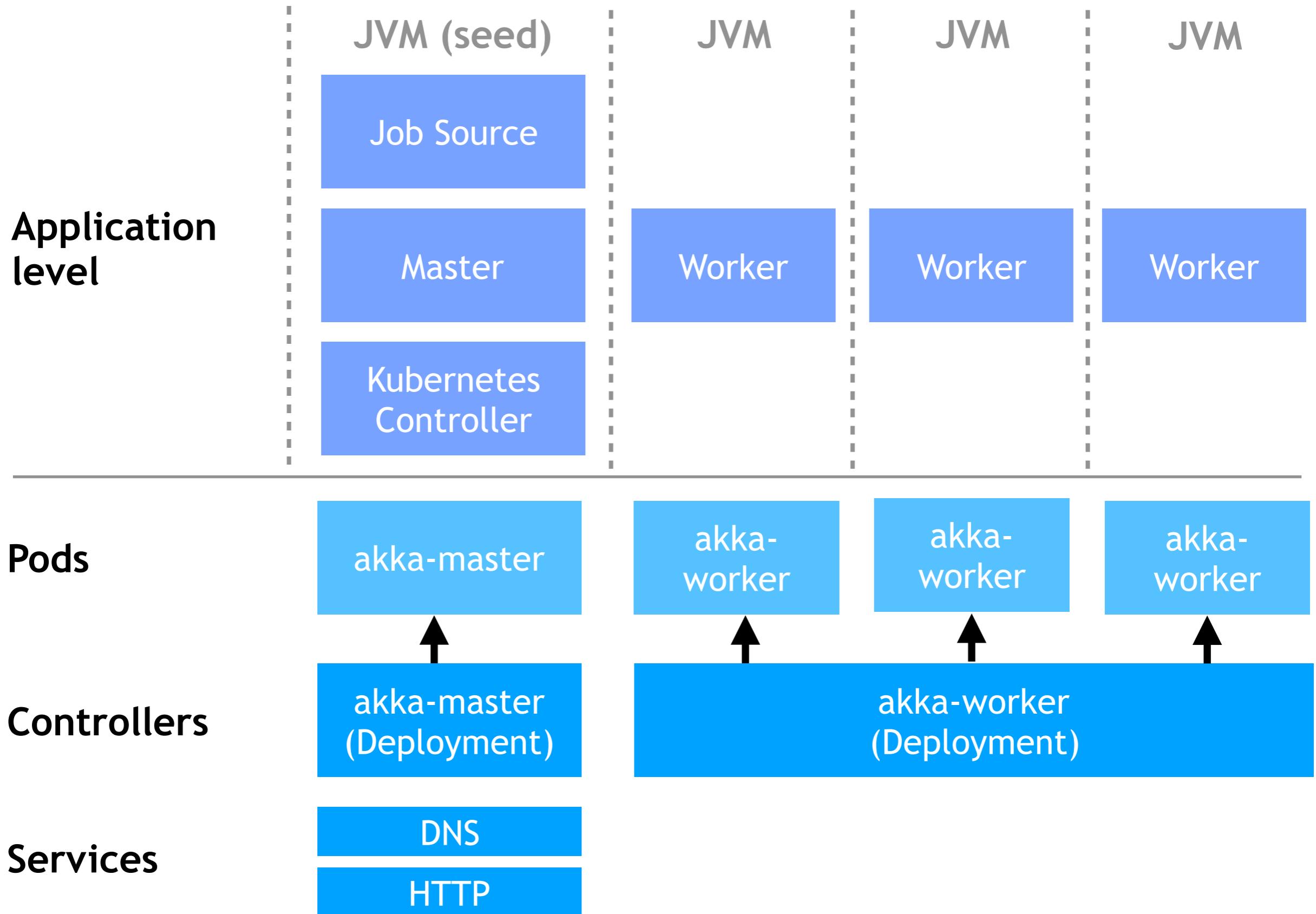
```
class KubernetesController extends Actor {  
  
    case AddNode =>  
        // use kubernetes api to scale up  
  
    case RemoveNode =>  
        // use kubernetes api to scale down  
  
}
```

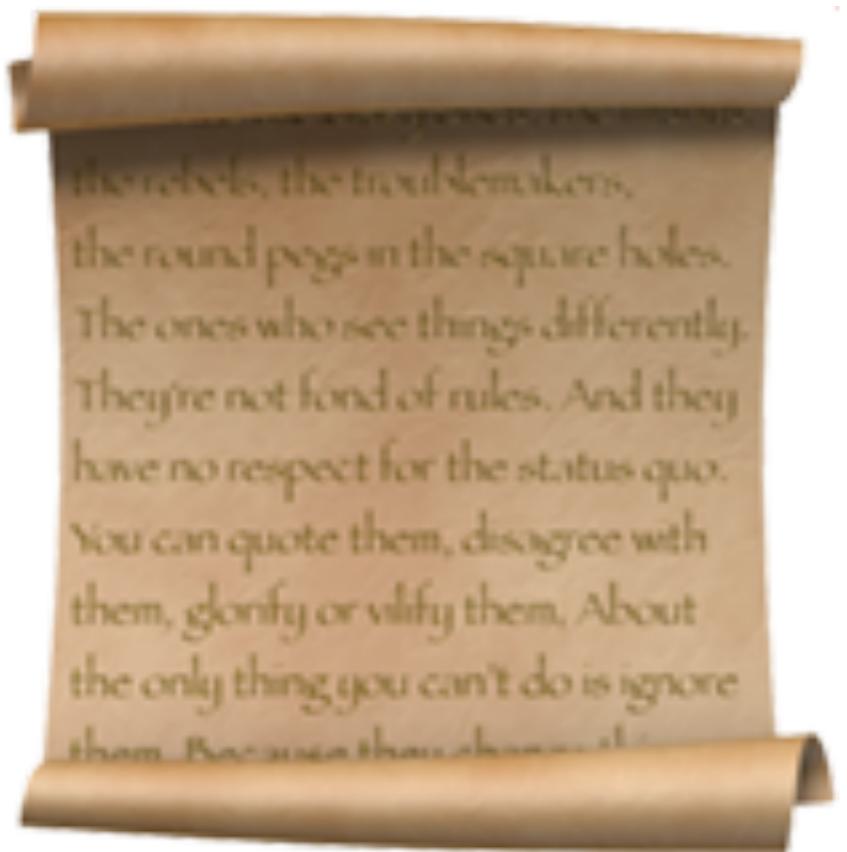
How does the master ask for changes? @ticofab

```
class Master extends Actor {  
  
    // to add a node  
    kubernetesController ! AddNode  
  
    // to remove a node  
    kubernetesController ! RemoveNode  
  
}
```

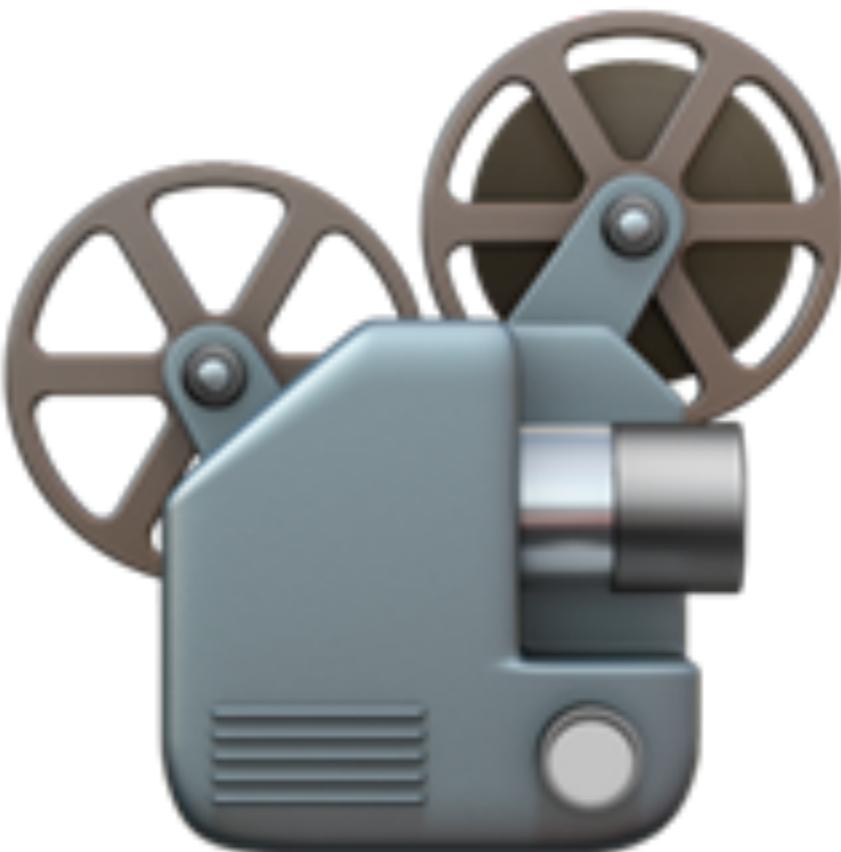
Kubernetes Architecture

@ticofab





,



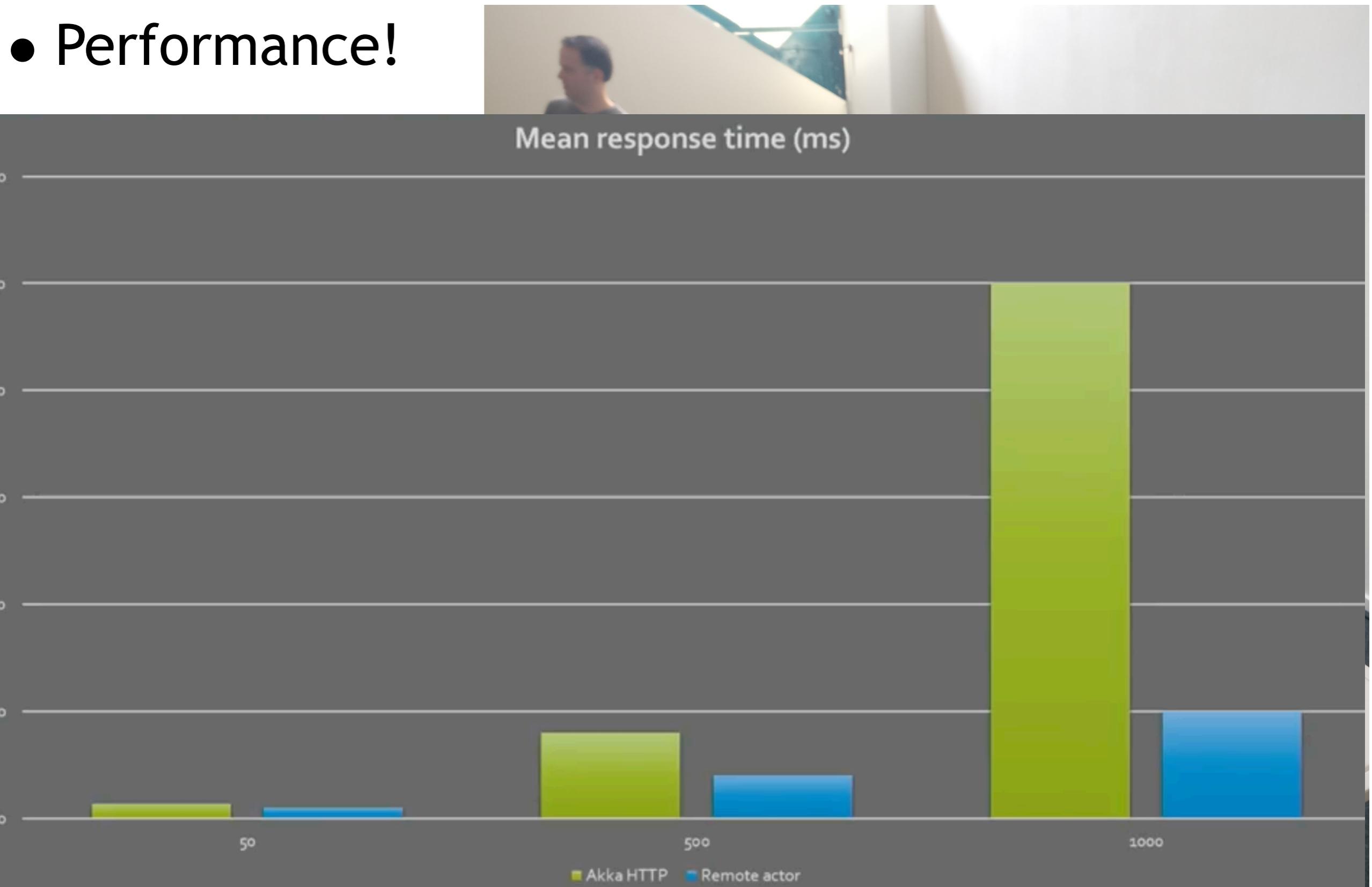
REACTIVE SYSTEM DESIGN

=

APPLICATION AND INFRA

- HTTP is ubiquitous
- Actors have built-in location transparency
- Actor messages do not necessarily need to wait for a response
- Less complexity (if you already use actors)

- Performance!



github.com/ticofab/akka-cluster-kubernetes



**Master
(Scala)**

**Worker
(Scala)**

**Common
(Scala)**

**Stats-visualizer
(html, javascript)**

And oh, about the holy grail

@ticofab



Choose your own holy grail 😊



THANKS



@ticofab
@adamsand0r

github.com/ticofab/akka-cluster-kubernetes