Industrial OSGi Discovery

Open-sourced by

Industrial-tsi.com

Ahmed Aadel
ahmed.aadel@remainsoftware.com

Foundation

- Industrial OSGi Discovery is based on Apache ZooKeeper

- ZooKeeper is an Apache Hadoop project:
  - "ZooKeeper is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services"

- Industrial OSGi Discovery nicely embeds ZooKeeper so that the whole runs smoothly as an OSGi service

Ahmed Aadel
ahmed.aadel@remainsoftware.com

Industrial OSGi Discovery

- Compliant with RFC 119, namely:
  - Publishes services wrapped as ServicePublication by any Distribution software
  - Delivers (nearly realtime) DiscoveredServiceNotification(*s)* to any DiscoveredServiceTracker being registered as such.

Ahmed Aadel
ahmed.aadel@remainsoftware.com

Configuration- One property

- Industrial Discovery needs just one property to be set per whole VM.
- This property is set only once by any service at registration time indicating one or more IP's to connect to. It'll trigger the discovery.

For example:

```
…
// Any service may set this configuration property
props.put("industrial.flavor.standalone","192.0.32.10, 192.0.32.11");
//Some  service  registration
context.registerService(IFoo.class.getName(), new Foo(), props);
…
```

Ahmed Aadel
ahmed.aadel@remainsoftware.com

Discovery 3 Flavors

- Industrial Discovery can run in one of three flavors:
  - Standalone:
    - Property to set: industrial.flavor.standalone
  - Centralized
    - Property to set: industrial.flavor.centralized
  - replicated
    - Property to set: industrial.flavor.replicated

Ahmed Aadel
ahmed.aadel@remainsoftware.com

Cenralized

- Centralized:
- for exmaple:

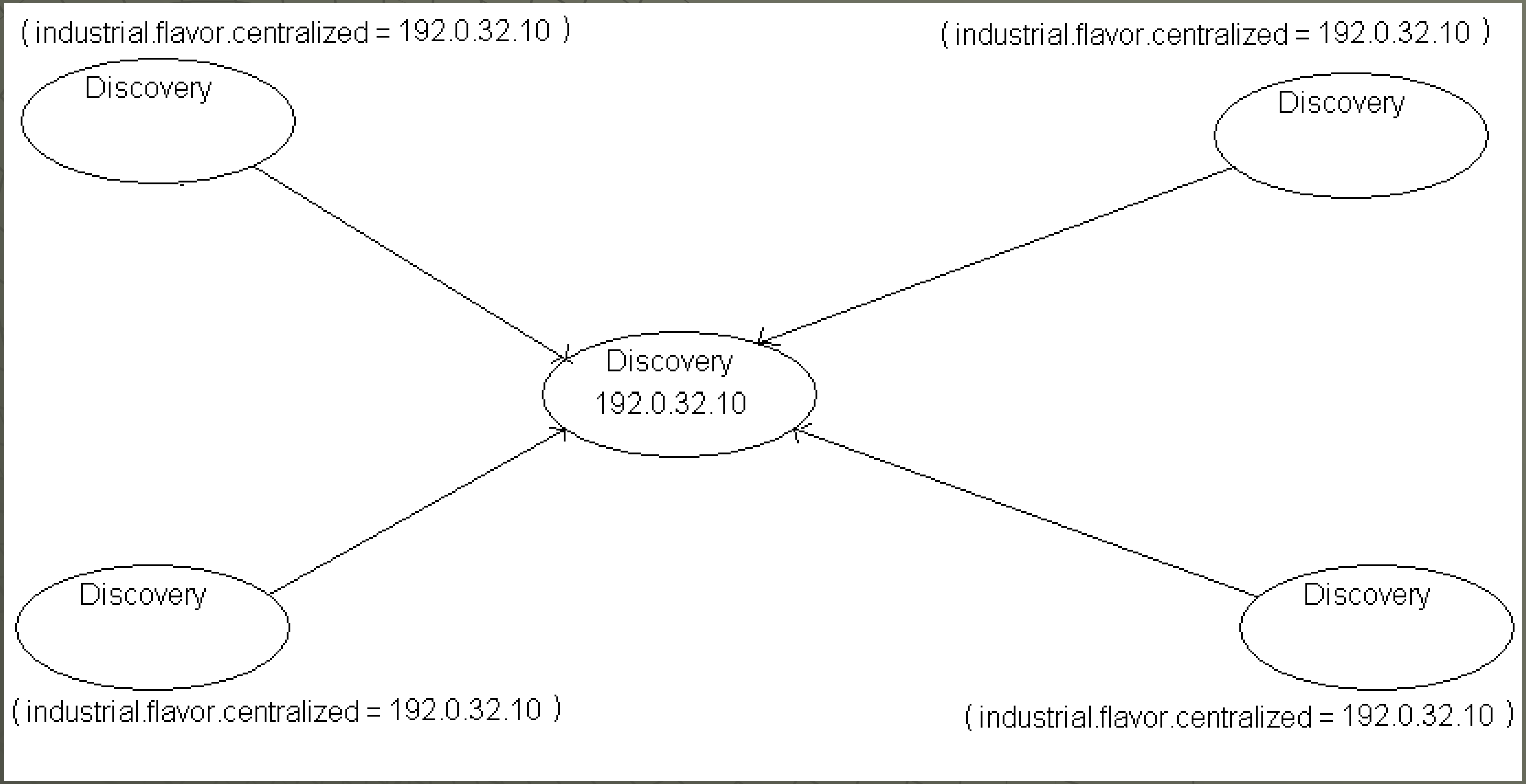
```
props.put("industrial.flavor.centralized","192.0.32.10");
...
```

All Discovery instances pubish to (192.0.32.10) one central point and read all services at that central point published by all others. While at 192.0.32.10 runs a discovery instance as Server.

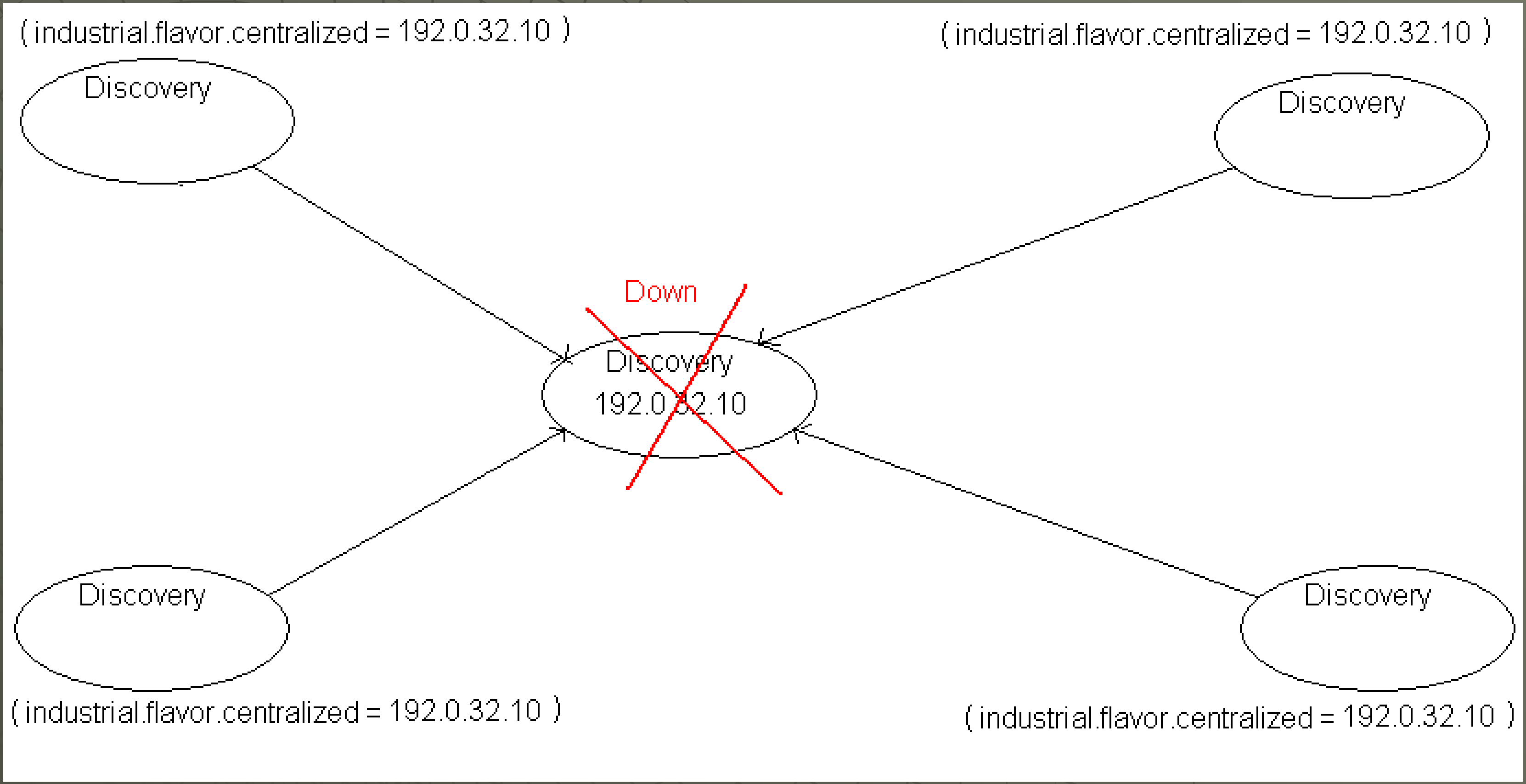
- The Discovery underlying zooKeeper at the central point runs as server, the rest as clients.

- If the Discovery running as server is down all services are lost, and all client loose their discoverd services

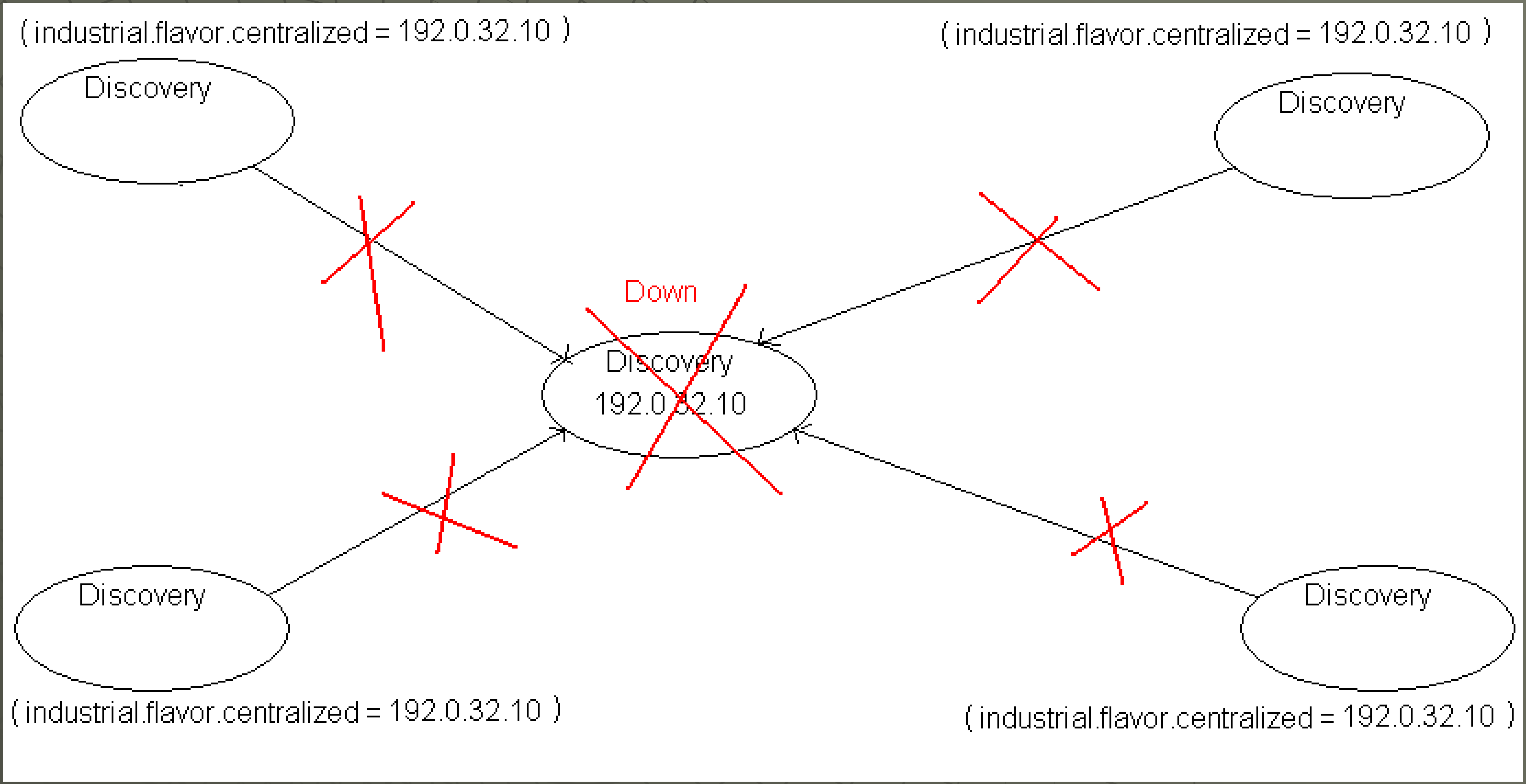
Centralized


(industrial.flavor.centralized = 192.0.32.10)
Discovery
(industrial.flavor.centralized = 192.0.32.10)
Discovery
Discovery
192.0.32.10
Discovery
(industrial.flavor.centralized = 192.0.32.10)
Discovery
(industrial.flavor.centralized = 192.0.32.10)

Ahmed Aadel
ahmed.aadel@remainsoftware.com


(industrial.flavor.centralized = 192.0.32.10)
Discovery
(industrial.flavor.centralized = 192.0.32.10)
Discovery
Down
Discovery
192.0.32.10
Discovery
Discovery
(industrial.flavor.centralized = 192.0.32.10)
(industrial.flavor.centralized = 192.0.32.10)

Cenralized 3


(industrial.flavor.centralized = 192.0.32.10)
Discovery
(industrial.flavor.centralized = 192.0.32.10)
Discovery
Down
Discovery
192.0.32.10
Discovery
Discovery
(industrial.flavor.centralized = 192.0.32.10)
(industrial.flavor.centralized = 192.0.32.10)

Ahmed Aadel
ahmed.aadel@remainsoftware.com

Standalone

- Standalone:

for exmaple:

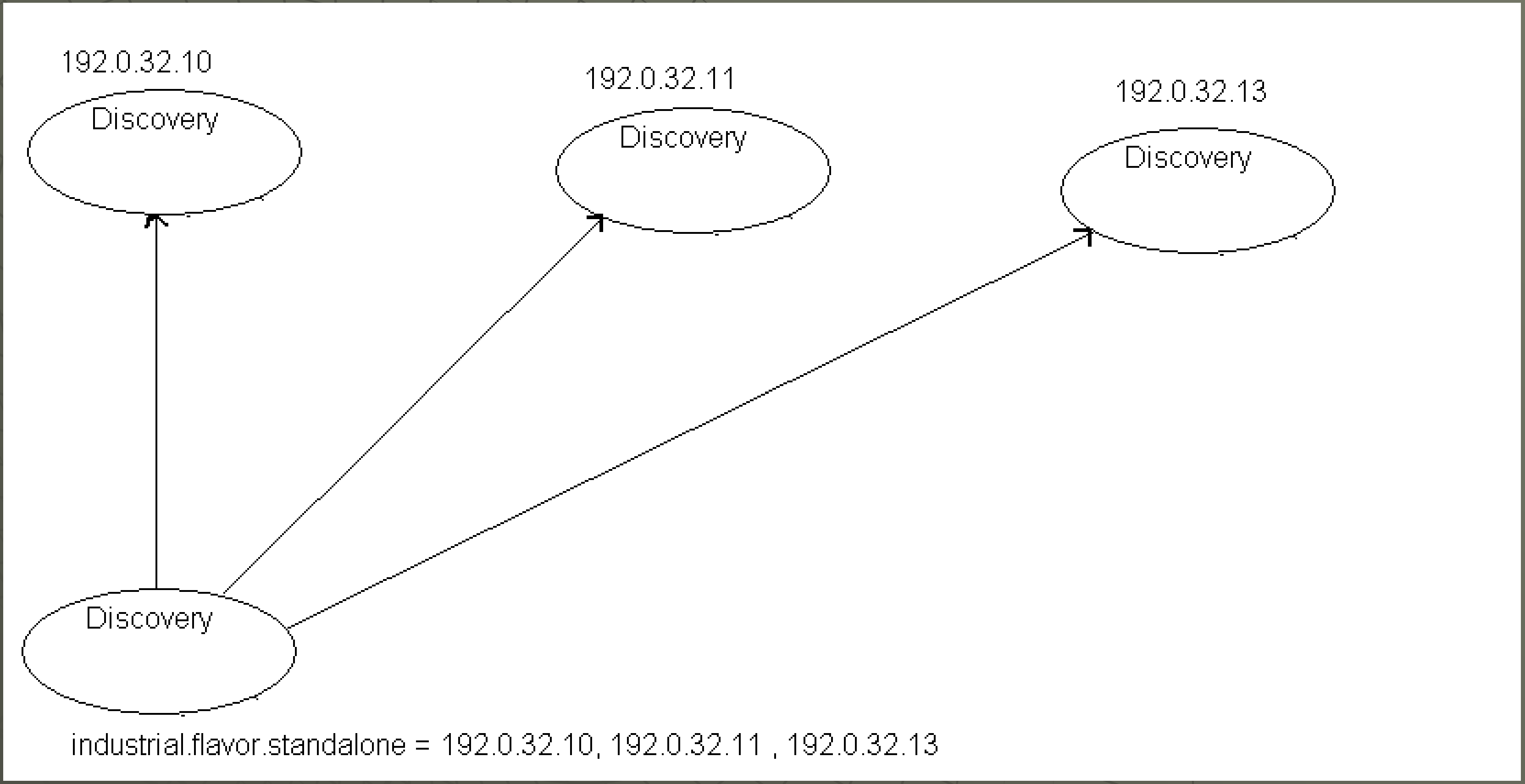
```
…
props.put("industrial.flavor.standalone","192.0.32.10, 192.0.32.11 ,
    192.0.32.13 ");
…
```

This Discovery instance connects to and discovers services at 192.0.32.10, 192.0.32.11 and 192.0.32.13.

- If one Discovery of those is down, only its services are lost.

Ahmed Aadel
ahmed.aadel@remainsoftware.com

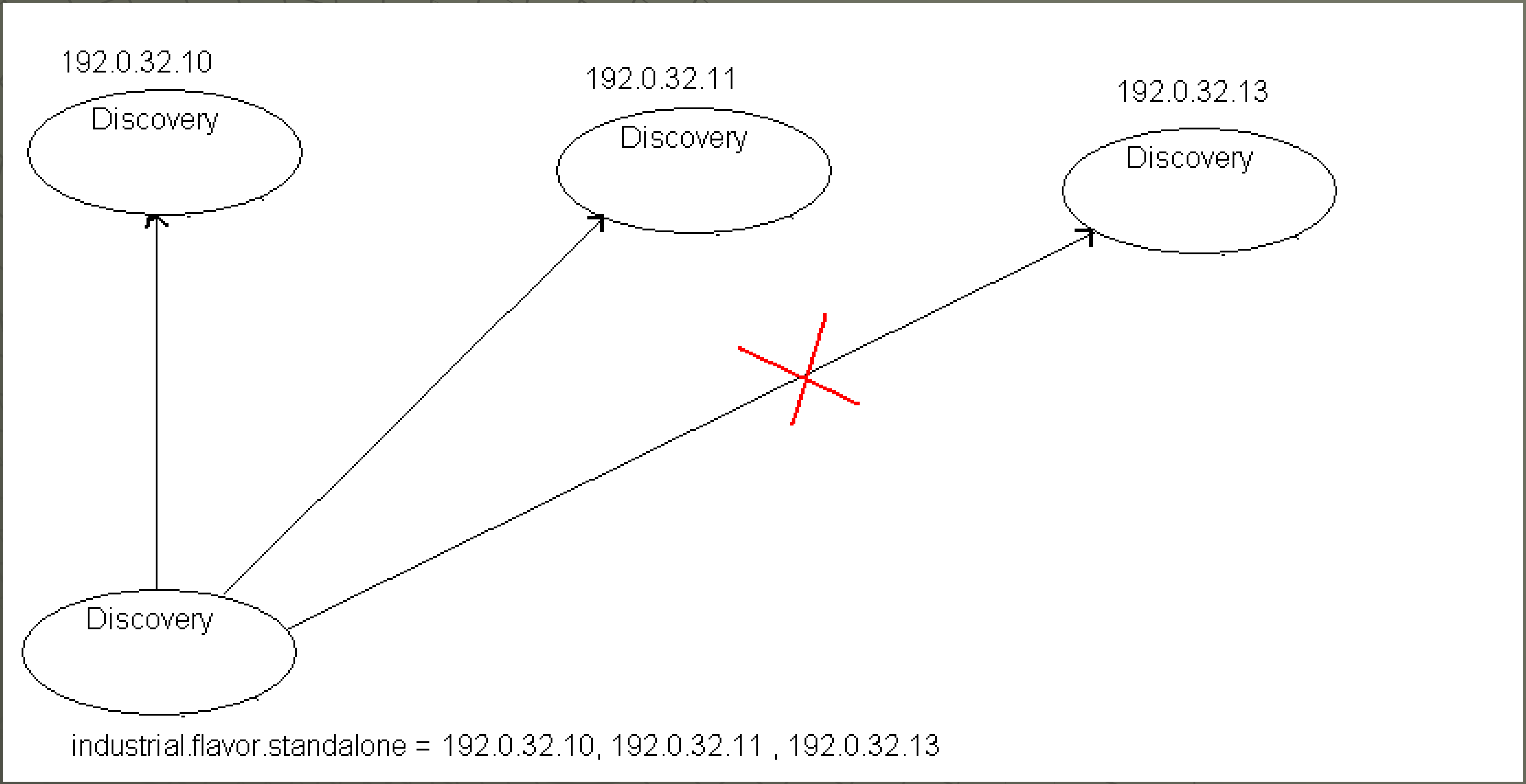
Standalone



192.0.32.10
Discovery
192.0.32.11
Discovery
192.0.32.13
Discovery
Discovery
industrial.flavor.standalone = 192.0.32.10, 192.0.32.11 , 192.0.32.13

Ahmed Aadel
ahmed.aadel@remainsoftware.com

Standalone


192.0.32.10
Discovery
192.0.32.11
Discovery
192.0.32.13
Discovery
Discovery
industrial.flavor.standalone = 192.0.32.10, 192.0.32.11 , 192.0.32.13

Replicated

- Replicated:

for exmaple:

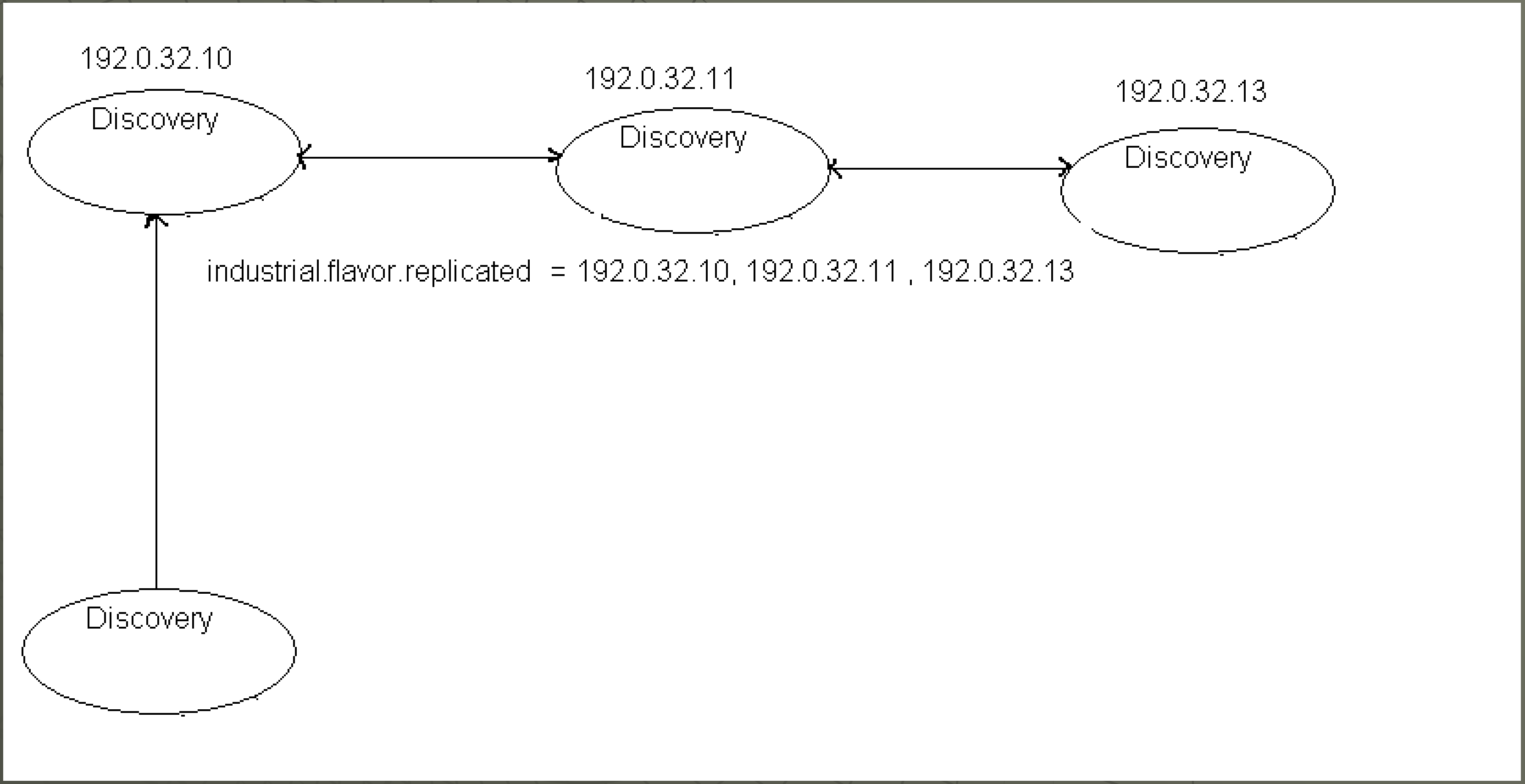
```
 ...
 props.put("industrial.flavor.replicated","192.0.32.10, 192.0.32.11 ,
      192.0.32.13 ");
 ...
```

- All these Discovery instances act as quorum, replicating and synchronizing services.
- Services known by one discovery are the same for all.

- So connect to 192.0.32.10 and get services published 192.0.32.11 and 192.0.32.13 as well.

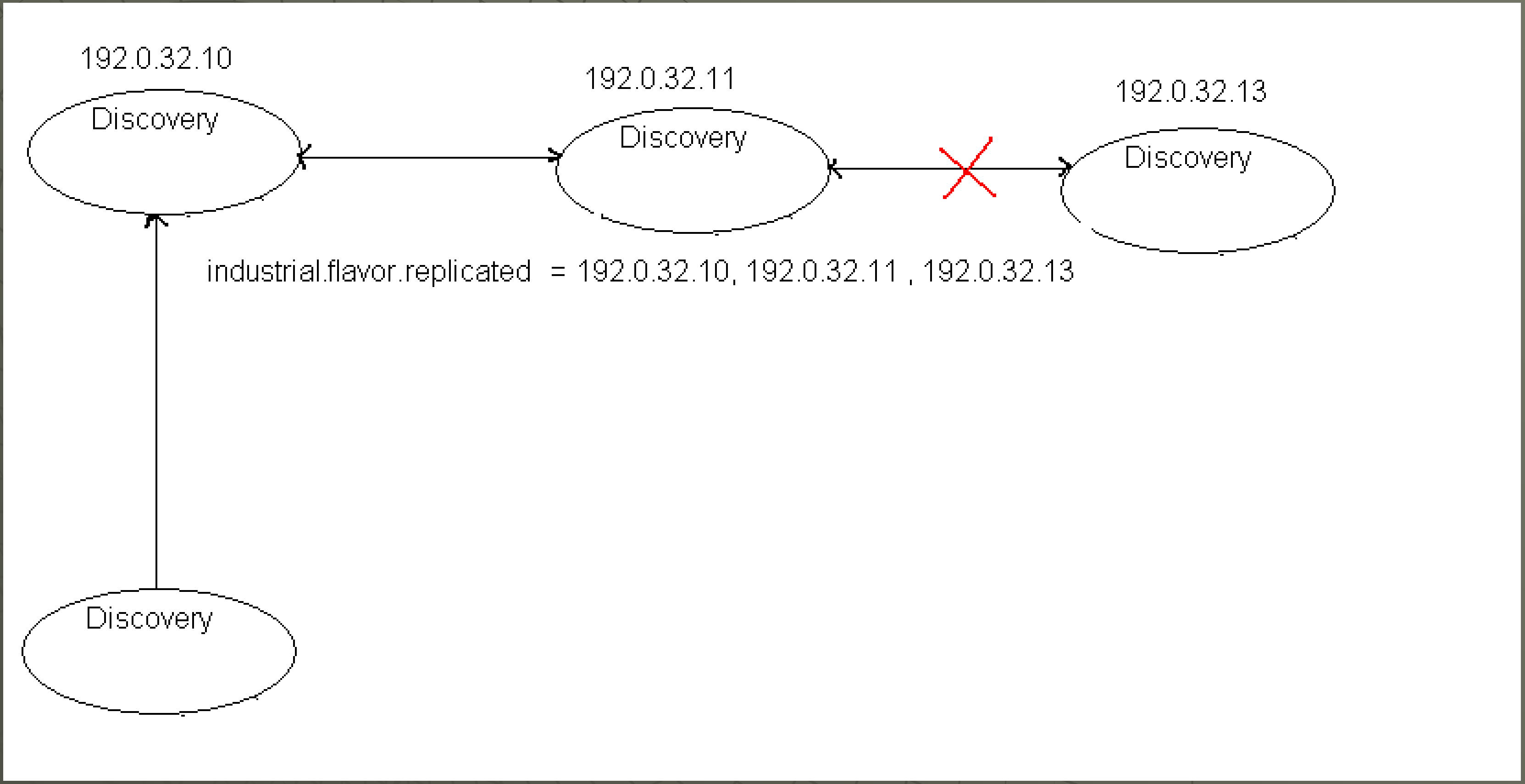
Ahmed Aadel
ahmed.aadel@remainsoftware.com

Replicated 1


192.0.32.10
Discovery
192.0.32.11
Discovery
192.0.32.13
Discovery
industrial.flavor.replicated = 192.0.32.10, 192.0.32.11 , 192.0.32.13
Discovery

Ahmed Aadel
ahmed.aadel@remainsoftware.com

Replicated 2


192.0.32.10
Discovery
192.0.32.11
Discovery
192.0.32.13
Discovery
industrial.flavor.replicated = 192.0.32.10, 192.0.32.11 , 192.0.32.13
Discovery

Ahmed Aadel
ahmed.aadel@remainsoftware.com

Yet 3!

- Industrial Discovery does its ‘discovering’ job regardless of property choice!
- 3 properties = 3 choices
- Those flavors/modes are made available to reflect the whole ZooKeeper power even embeded and internally configured by Industrial Discovery

Ahmed Aadel
ahmed.aadel@remainsoftware.com

TODO

- Filter service notifications
- Configure Industiral discovery instances running in the same LAN automaically using some kind of zeroconf
- Port to new OSGi spec.