Hyperledger under swarm

Problems and solutions

Siriwat K.

24 August 2016

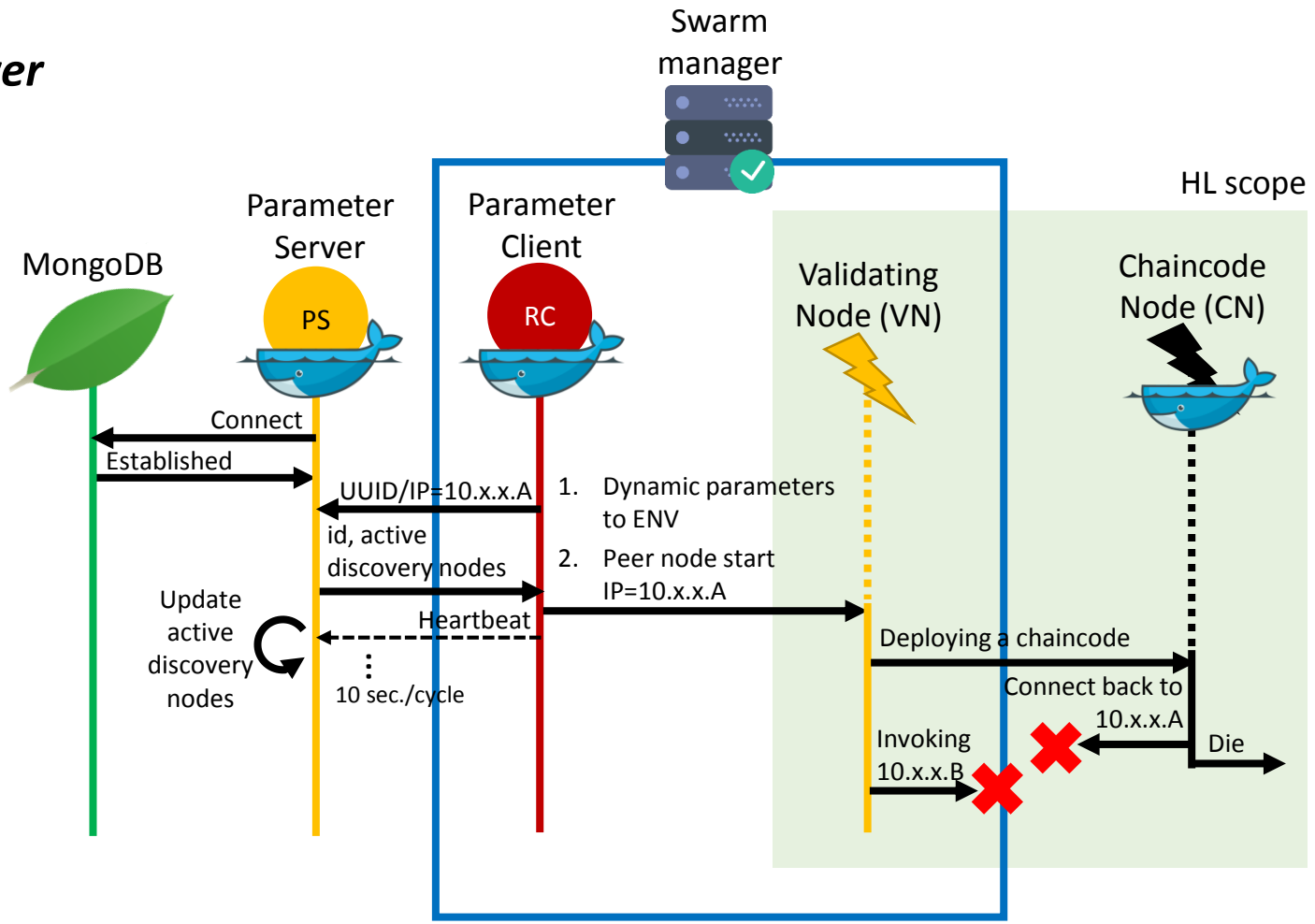
Objective

- To run Hyperledger (HL) at scale with Docker swarm.
- To derive the benefits of Docker swarm on provisioning HL nodes.
  - Cluster
  - Automatic fail-over
  - Resilience, scaling
  - Load-balancing

A way to run Hyperledger (HL) under swarm

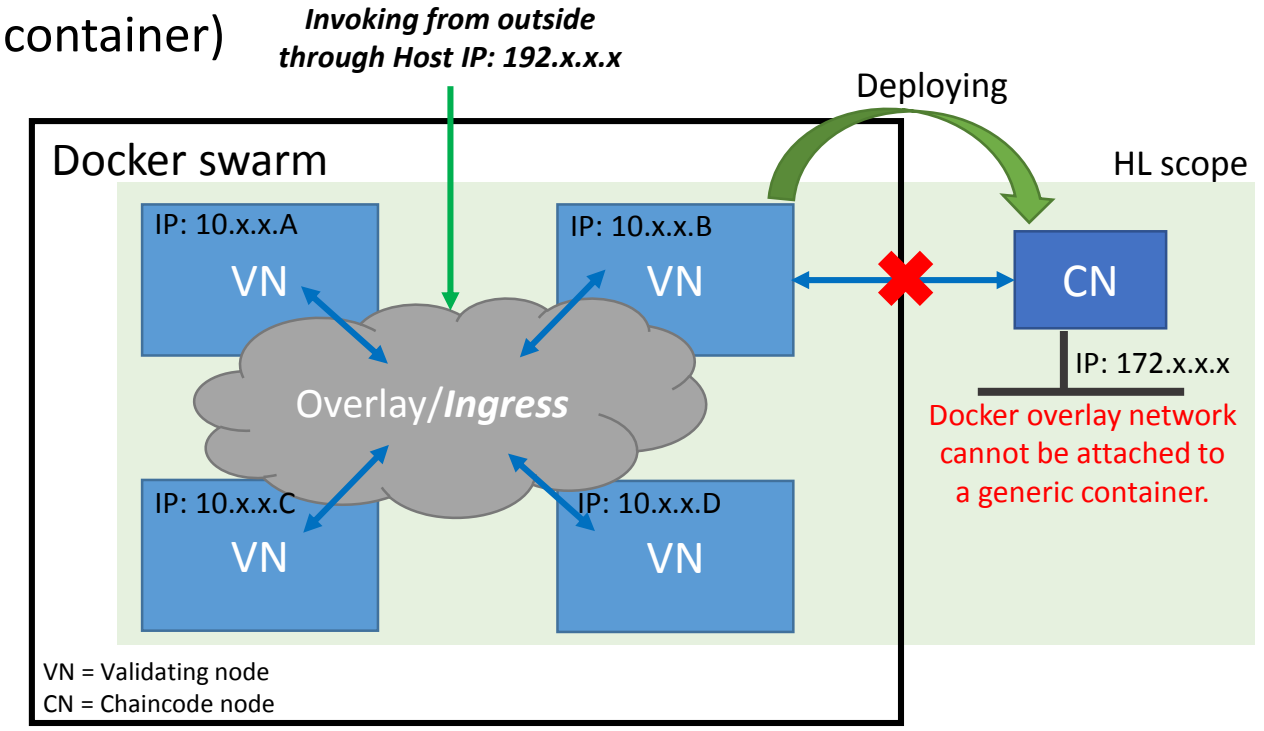
- We need to attach a ***parameter server*** to generate dynamic parameters for each of validating node.
- E.g. Environment variables
  - CORE_PEER_DISCOVERY_ROOTNODE=..
  - CORE_PEER_ADDRESS=..
  - CORE_PEER_ID=vp..
  - Etc.

- **Problem**
  - Deploying a chaincode was done outside of swarm.
  - Connection between them is ***not possible***.



Problem

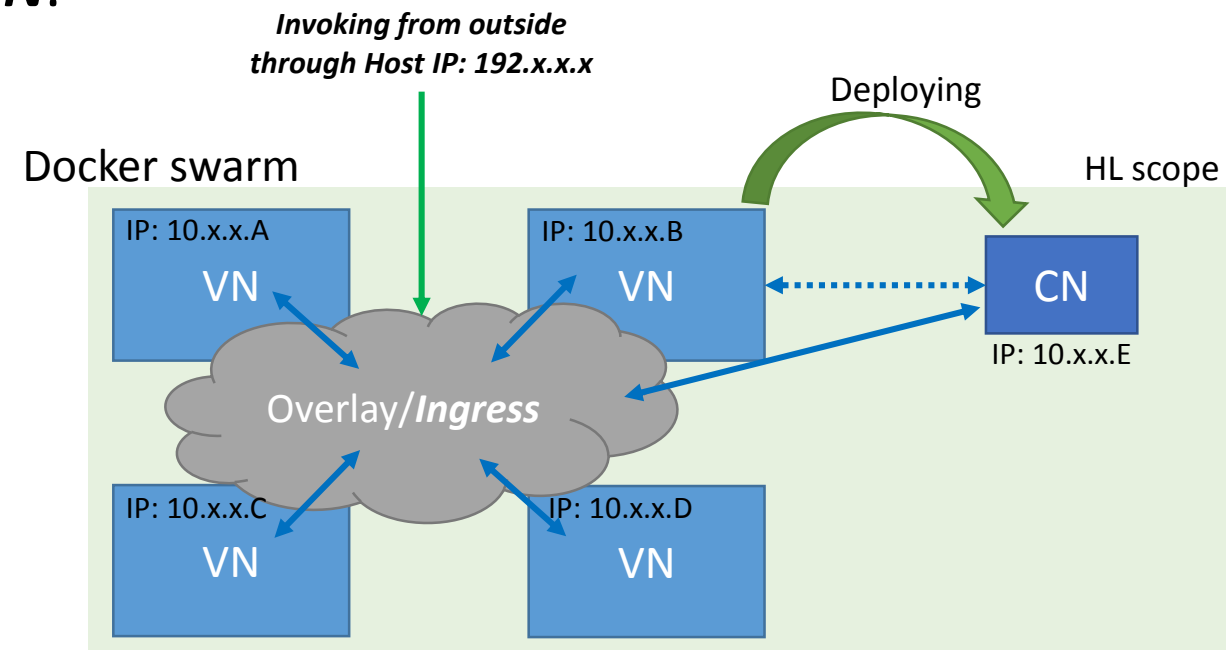
- Swarm needs only an overlay network.
  - E.g. ingress
- Accessing to the service under swarm (each container) is done only through the Swarm manager
  - Each container will be load-balanced through the ***ingress*** network.
  - This internal IP address (10.x.x.x) ***cannot*** be accessed from ***outside***.
- VN passed its IP address to CN during the image creation time.
  - CN known only about 10.x.x.B ***(and not accessible)***
  - VN ***do not know*** about a new CN IP address.



The architecture under swarm

Hyperledger connection requirements

- ***VN*** needs to communicate between ***VN***.
  - Without swarm
    - Option --net=host
    - Port mapping
  - With swarm
    - Overlay network
- ***VN*** needs to communicate to its ***CN***.
  - Without swarm
    - Option --net=host
  - With swarm
    - ***No ingress***
    - ***No Docker standard overlay***
    - ***How?***



VN = Validating node
CN = Chaincode node

The architecture under swarm

(ideal case)

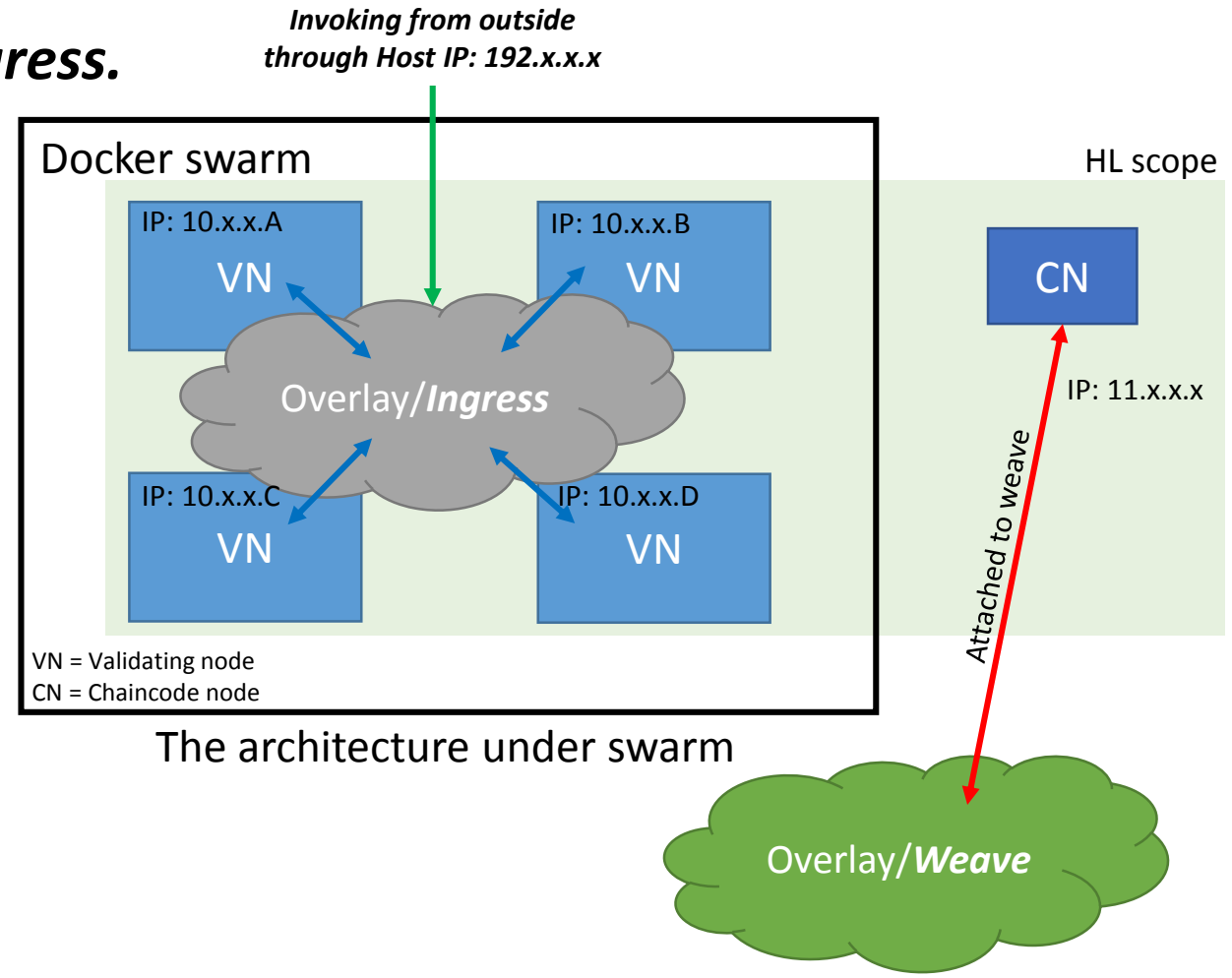
Solution is

- Weave Net
  - https://github.com/weaveworks/weave
  - A kind of mesh network that connects containers over the host.
  - A kind of an overlay network.
  - Build for a generic container.
- Attaching Weave during CN creation time is done by
  - Setting an environment variable for “peer node start”
    - ***CORE_VM_DOCKER_HOSTCONFIG_NETWORKMODE=weave***

- **Sound cool!!, But…**
  - ***Weave do not support to be attachable with Docker swarm.***

Problem of ***Weave*** under swarm

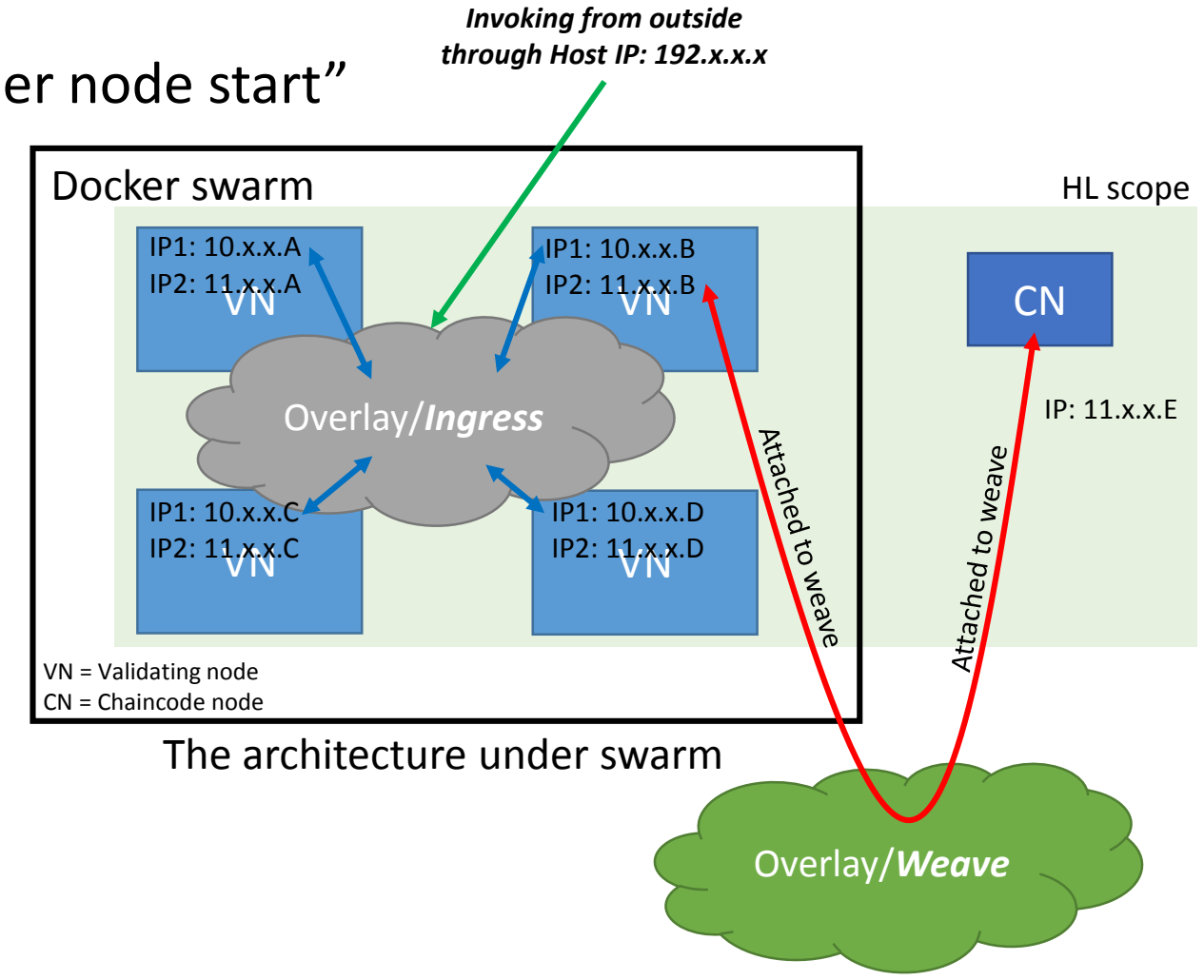
- We **cannot** attach ***Weave*** instead of ***Ingress.***
  - Upon the swarm service creation time.
- We can attach Weave to …
  - A ***generic container***.
- But!!
  - Swarm creates a ***generic container*** under its environment.
  - VN is also a ***generic container***.
  - Why not?



The architecture under swarm

Solution of the problem (of the solution of the problem)

- Before we let swarm creates **VN** by “peer node start”
  - We try to attach ***Weave*** as a second network interface.
- Then we run “peer node start”.
- That results to
  - ***External traffic*** goes through
    - ***Ingress***
  - ***Internal traffic*** goes through
    - ***Weave***
- ☺

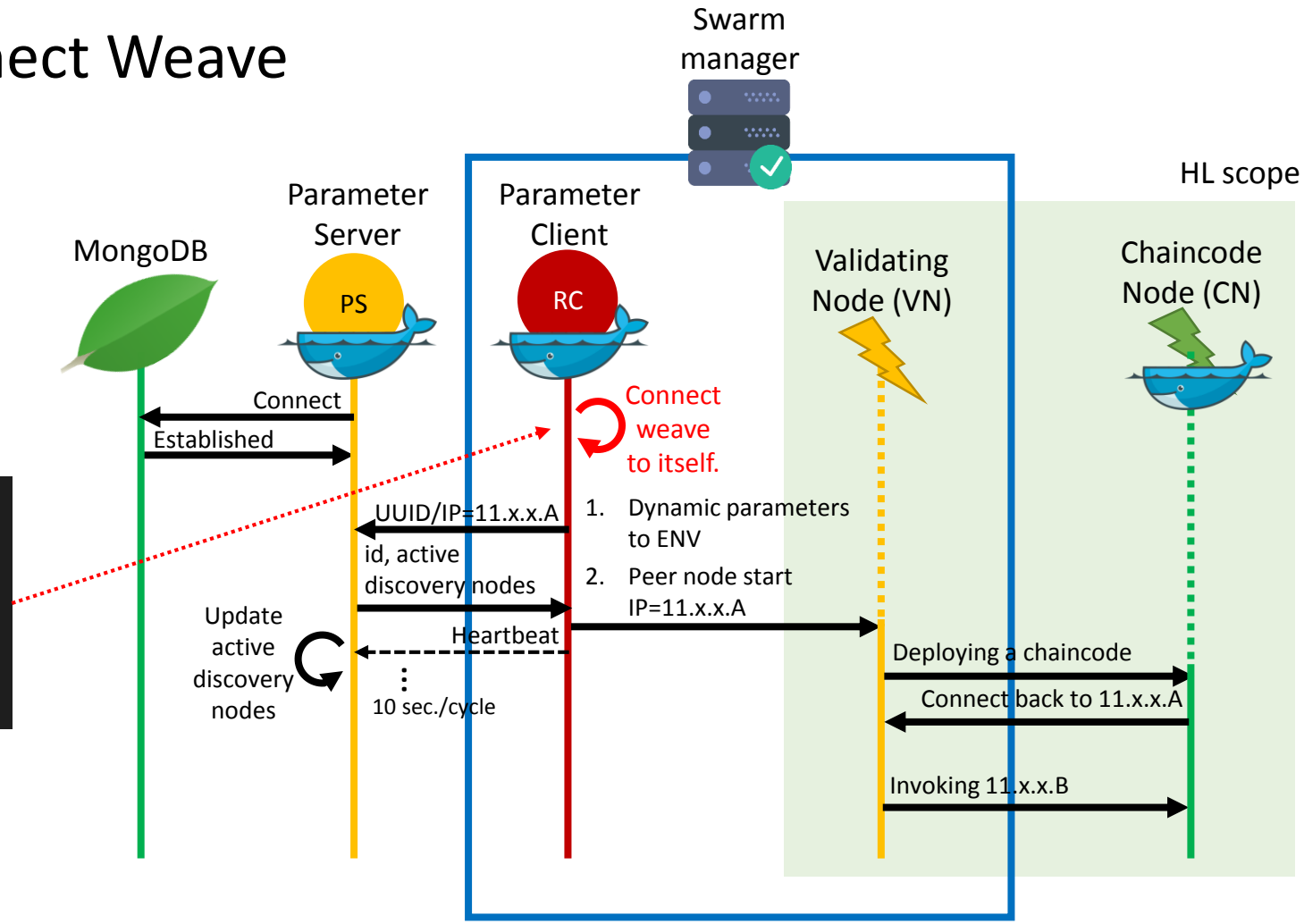


The architecture under swarm

Overall solution for Hyperledger under Swarm

- Look at how we connect Weave

```
var Docker = require('dockerode');
var docker = new Docker({ socketPath: '/var/run/docker.sock' });
var container = docker.getContainer(process.env.HOSTNAME);
var network = docker.getNetwork('weave');
network.connect({
    Container: container.id
}, function (err, data) {
    console.log("Success");
});
```



Conclusion

- Hyperledger is still be scalable by Docker swarm.
- Docker Swarm can still do load-balancing the traffic for Hyperledger.
- Fail-over is still work as its behavior.
- The connection between ***internal swarm nodes*** and the ***outside chaincode node***, is made on top of the ***Weave Net.***