

# CNI

Container Network Interface

Konstantinos Karampogias  
Software Engineer, Swisscom

## Swisscom Application Cloud

Public Offering  
[developer.swisscom.com](http://developer.swisscom.com)

Swisscom  
Enterprise Application Cloud  
Offering



# Agenda

- Container Standards
- Container Runtime
- CNI Introduction
- Demo
- Conclusions

# Container Standards @CN Landscape



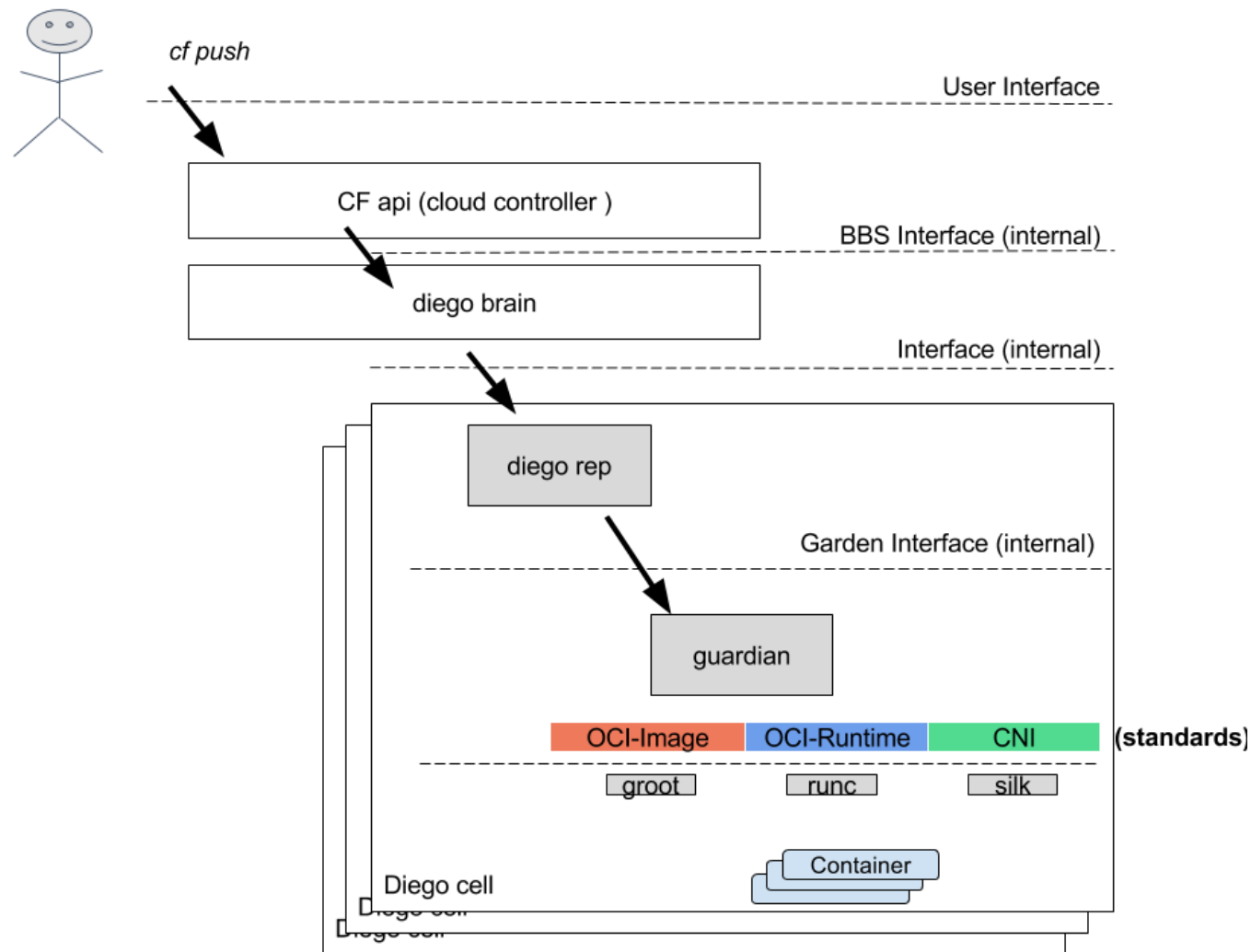
OCI-Runtime

OCI-Image

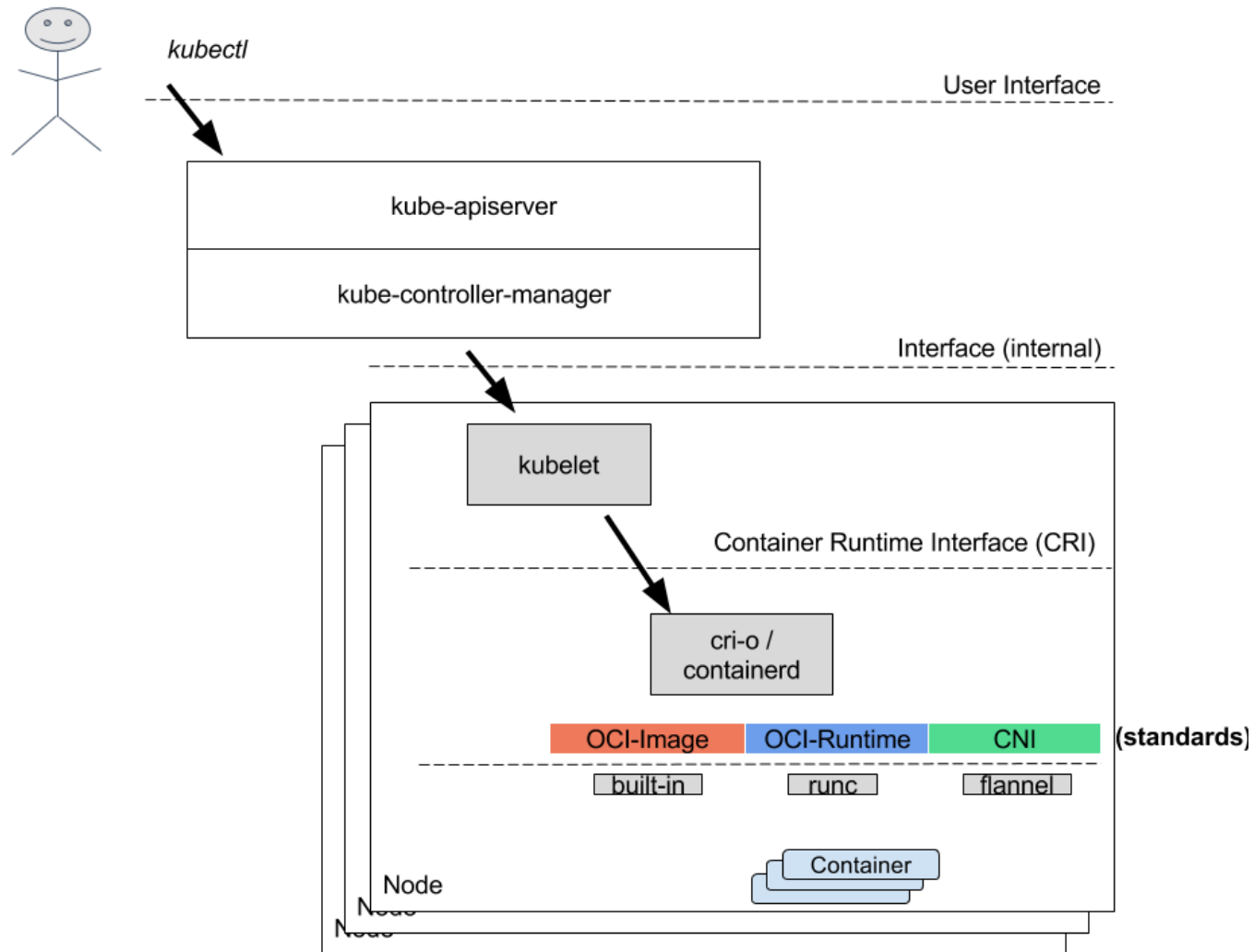


CNI

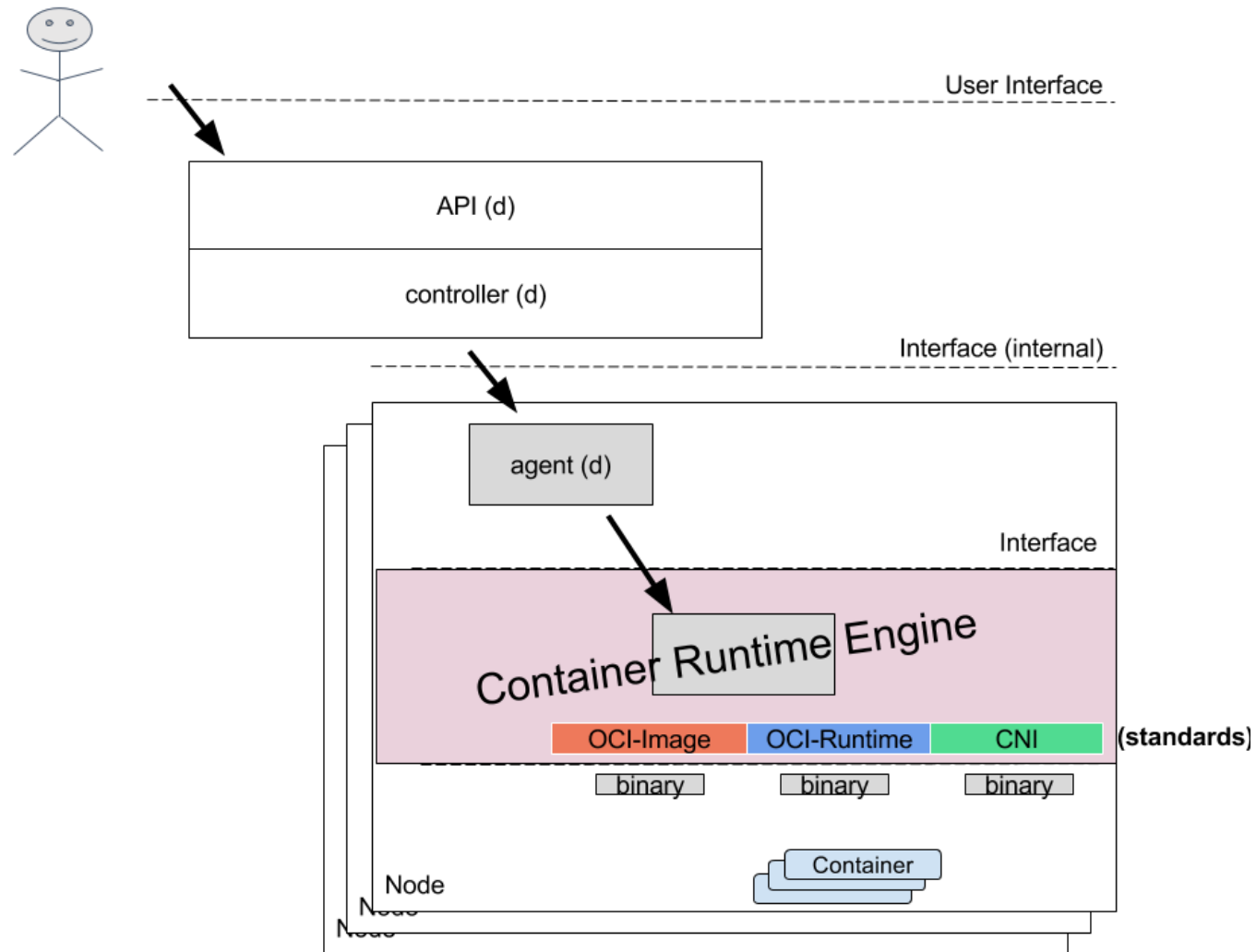
# Container Standards @Cloudfoundry



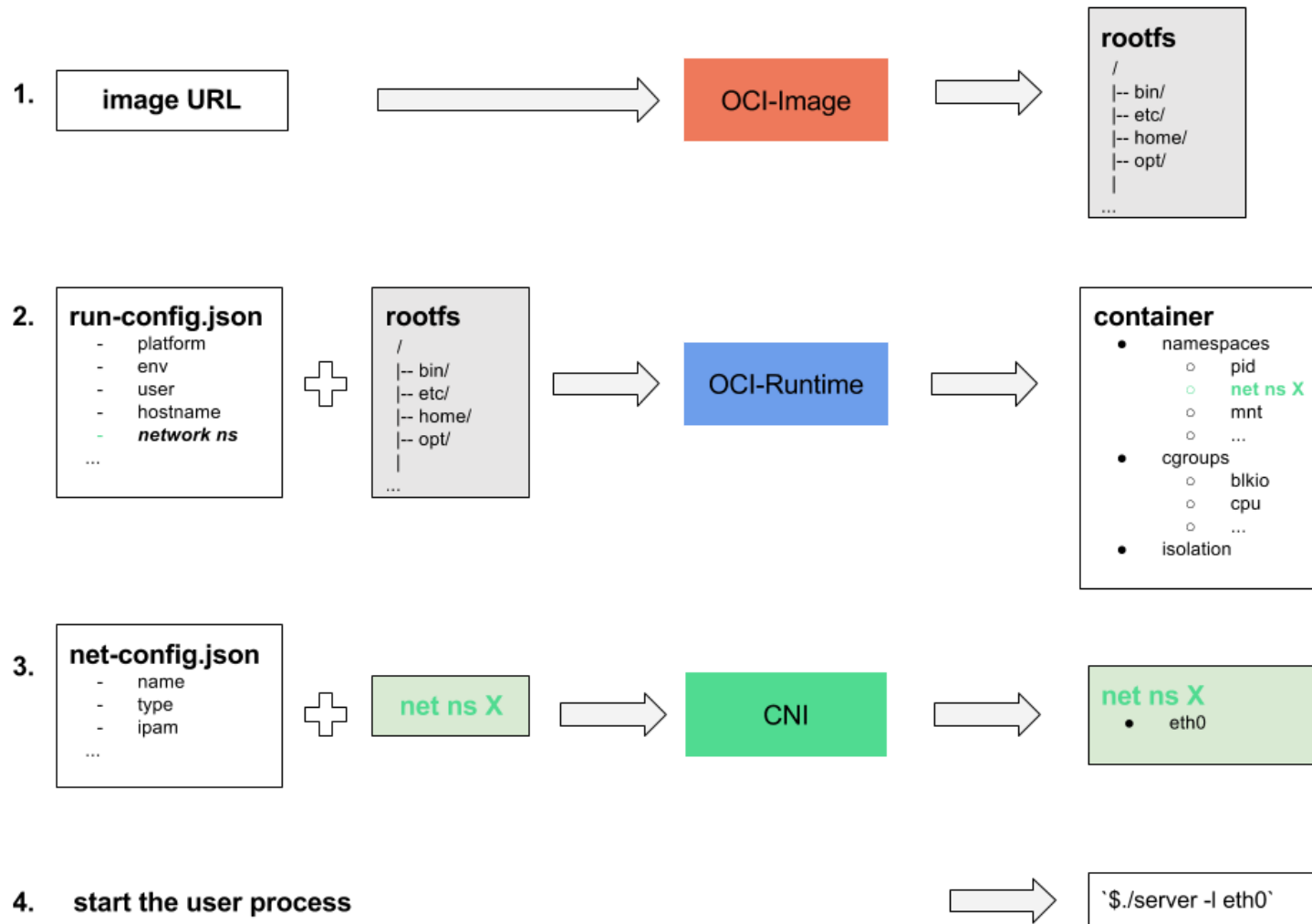
# Container Standards @Kubernetes



# Generic Architecture



# Container Runtime Engine





# What is CNI

A minimal standard way to connect a container to a network

- CNI Specification: defines an API between runtimes and network plugins
- Library: provide a Go implementation of the CNI specification
- Plugins: provide reference implementation for a variety of use-cases

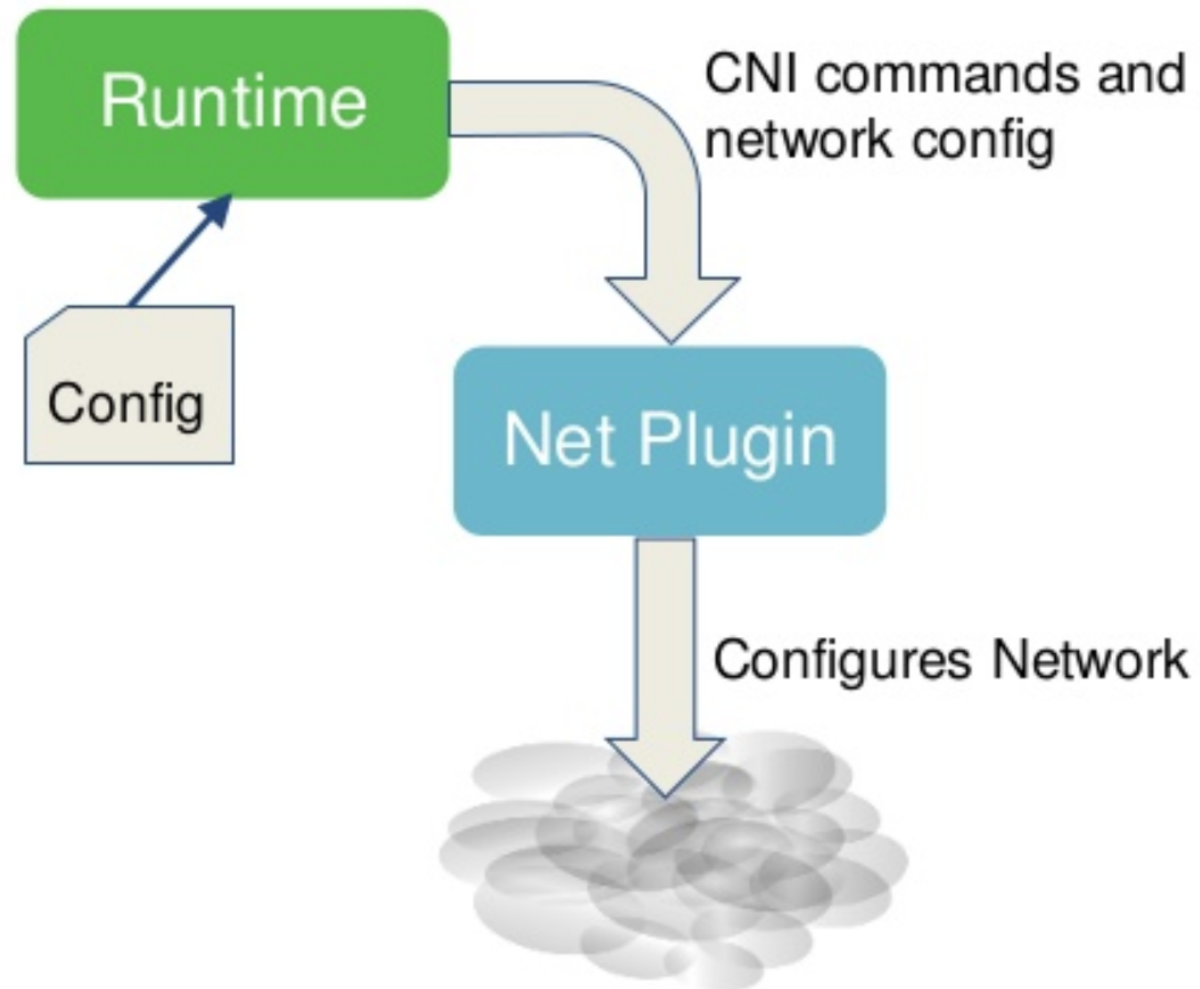
<https://github.com/containernetworking/cni>

<https://github.com/containernetworking/plugins>

# What is CNI

- 5 maintainers
- 60 contributors at 10+ companies
- 10+ plugin vendors
- 6+ runtimes (k8s, CF, Mesos, OpenShift, rkt, Amazon ECS)
- Started from rkt
- Spec separated from rkt April 2015
- CNCF project since May 2017

## Single interface



# CNI spec

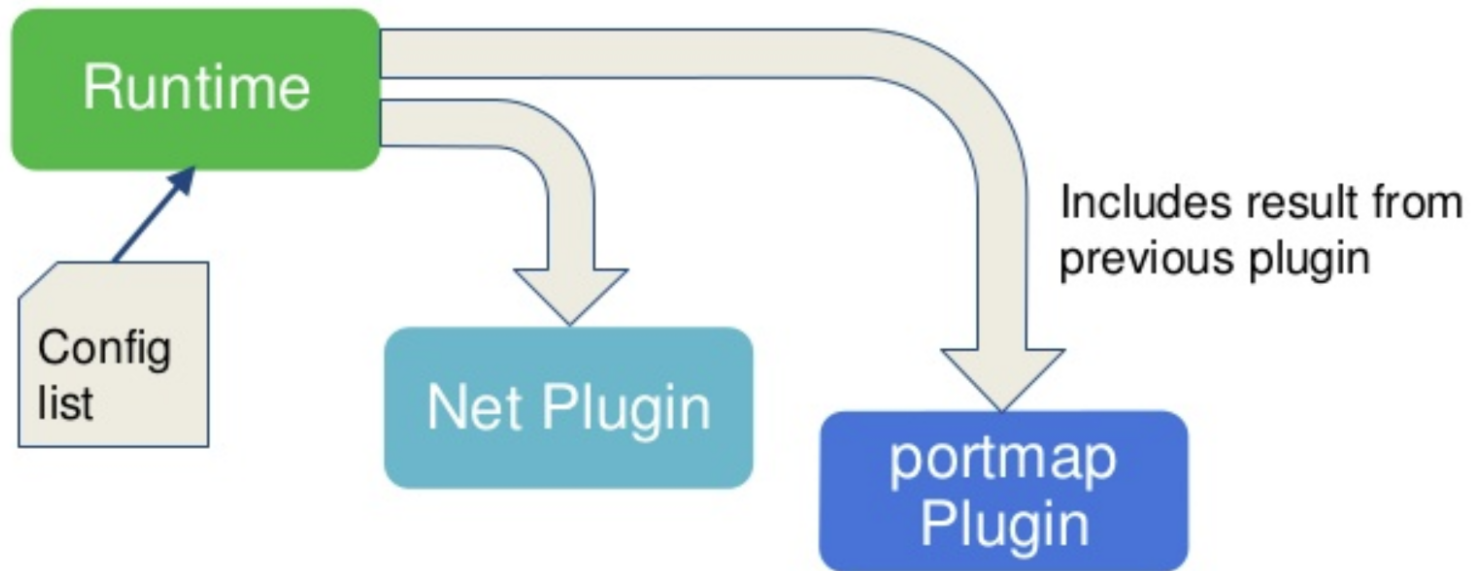
- 3 commands: ADD, DEL and VERSION
- Configuration on stdin, results on stdout
- Runtime parameters via env (CNI\_ARGS for arbitrary runtime params)

```
CNI_COMMAND=ADD \
CNI_CONTAINERID=$id \
CNI_NETNS=/proc/$pid/ns/net \
CNI_PATH=/opt/cni/bin \
CNI_IFNAME=eth0 \
my-plugin < my-config
```

Can be either ADD, DEL or VERSION

A JSON document defining the network

## Chained Plugins



# DEMO: Bridge + Port forwarding

# CNI is

## **about:**

- actions during creation/deletion
- container to internet traffic
- internet to container traffic
- hiding network complexity
- vendors to add their stuff

## **not about:**

- realtime changes (e.g. policy enforcement)
- setting up the network medium (e.g. bridge creation)
- multihost host container traffic (e.g. distribute routes)

# Open Positions



---

## Senior DevOps Engineer Container Platform

Bern oder Zürich 80% to 100%

Your focus is to develop, engineer and operate a Kubernetes-based Container-as-a-Service product within Swisscom's Application Cloud.



# Thank you

Konstantinos Karampogias  
Software Engineer, Swisscom

@karampok (<http://twitter.com/karampok>)

