- VISCUM -

A New Light Weight Software Architecture of MEC

Feng Li, Sriram Sidhar, Jae Won Chung and Jamal Hadi Salim

Outline

1. Introduction
2. Related Work
3. Viscum: process level virtualization
4. Conclusion

Mobile Edge Computing (MEC)

- Mobile Cloud Computing (MCC)
  - Pros:
    - Offloading battery consuming computation tasks from UE to cloud;
    - Supporting mobile user access complicated service in cloud;
    - Providing high volume storage resources for mobile users.
  - Cons:
    - Introducing additional network load on both radio and core networks;
    - Increasing higher latencies, data has to be sent to cloud servers.
- MEC to Address Challenges on MCC
  - Move cloud service to mobile edges;
  - Reducing latencies and allowing interactive between service providers and customers.
  - Reducing load on mobile providers' core network and/or radio networks.

Software Architecture in MEC

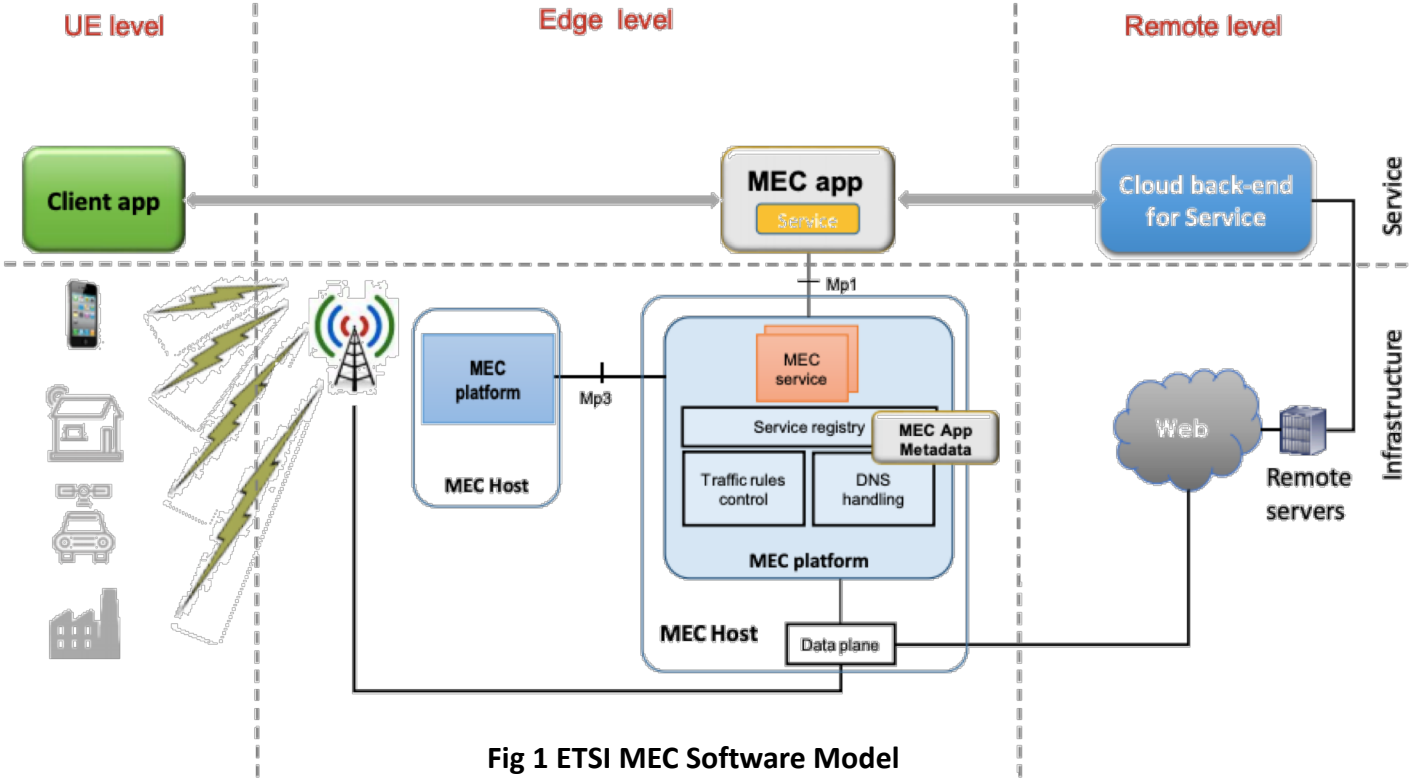


Fig 1 ETSI MEC Software Model

ETSI Software Model

- Split cloud service into edge and cloud components.
  - Extra development efforts.
  - Balancing between edge nodes and cloud nodes.
  - Different management for edge and cloud components.
- MEC Platform
  - Traffic steering
  - Horizontal scalability
  - Virtualization technology
  - Service management

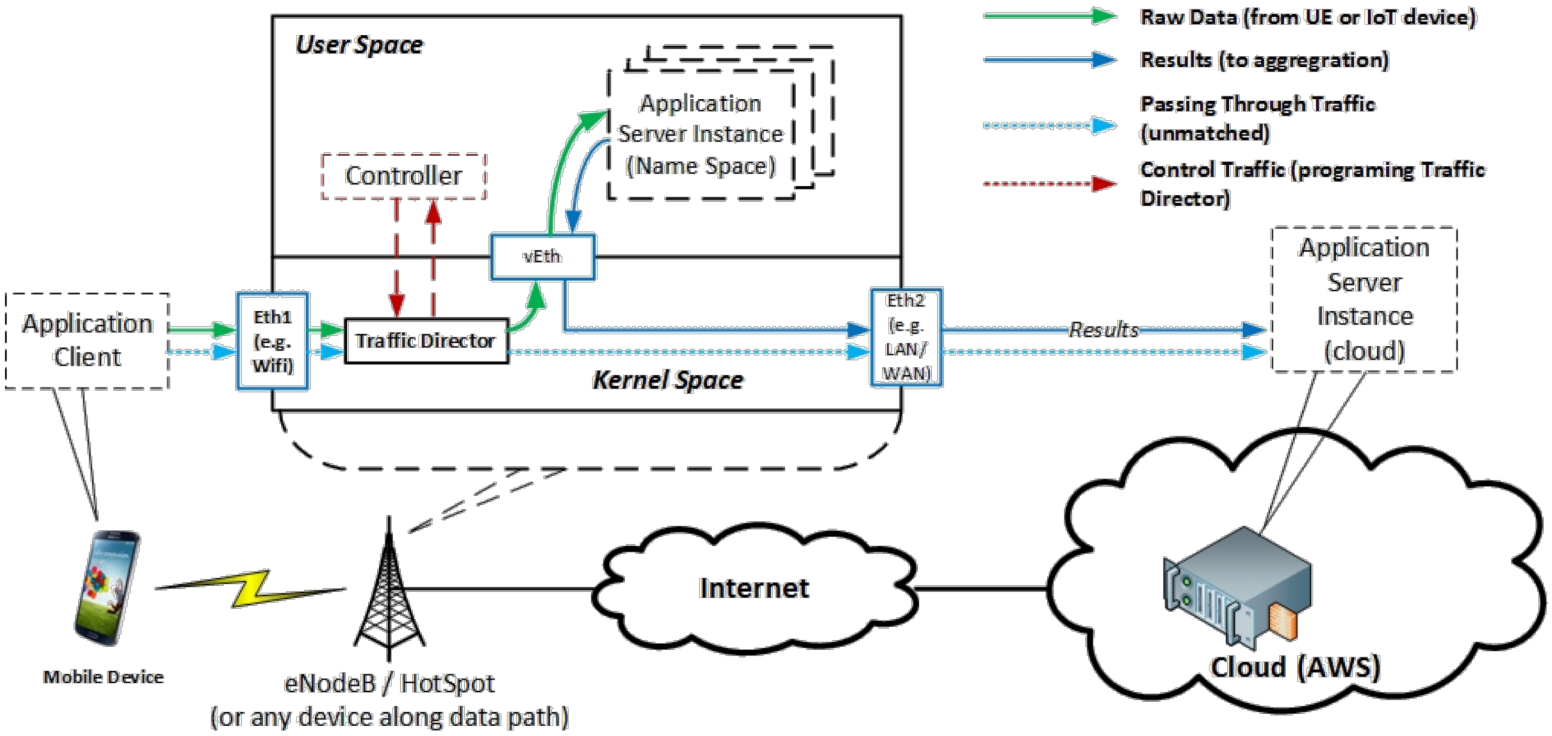
MEC Questions?

- Where is Edge?
- When will MEC be ready to host applications?
- Can we gradually deploy applications on MEC as they are ready?

Think small and be practical

Let’s start from the edge devices

Viscum Architecture

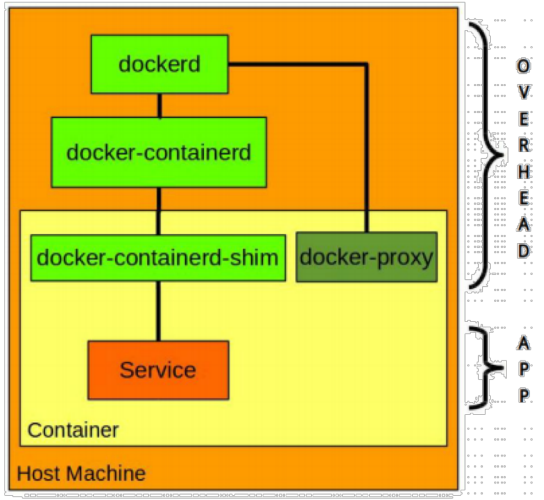


Viscum

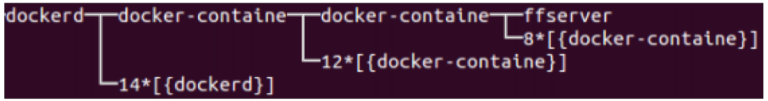
- Light weight process level virtualization
- Consists of four components
  - Linux Network Namespace contains applications;
  - Linux Traffic Control (TC) based traffic redirector;
  - A pair of Virtual Ethernet Interfaces (veth) as a pipe between two namespaces;
  - Control module manages above components.
- Named it Viscum to mimic the small MEC service deployed onto edge routers (hosts).

Why not use full containers ?

- Container overheads
  - At least four more process need to be created for one service process.
  - Not required for simple service deployment.
- Network namespace
  - Providing network resource duplication.
  - Reusing existing control plane function to manager namespaces.
  - Support embedded devices, or Linux distribution with old kernel release. (Redhat supports docker from 7.0, while namespace is supported from 2.6.32.



(a) Diagram of the Docker Processes Tree



(b) Screenshot of the output of `pstree`

Create Viscum Host

```
1   create_ns_tc() {
2         echo "============= Creating netns and tc rules ============="
3         (
4         set -x
5         sudo sysctl -w net.ipv4.ip_forward=1
6
7         # Create VETH device, configure the host side first.
8         $IP link add $H_VETH address $VETH_HOST_MAC type veth peer name $NS_VETH
9         $IP link set $H_VETH up mtu $MTU
10        $IP addr add $IP_ADDRESS_H/16 dev $H_VETH
11
12        # Create the namespace
13        $IP netns add $NAMESPACE
14        $IP -n $NAMESPACE link set dev lo up
15
16        # Create the namespace end of the veth device.
17        $IP link set dev $NS_VETH netns $NAMESPACE
18        $IP -n $NAMESPACE link set dev $NS_VETH up address $VETH_NS_MAC mtu $MTU
19        $IP -n $NAMESPACE addr add $IP_ADDRESS/16 dev $NS_VETH
20        $IP netns exec $NAMESPACE ip route add default via $IP_ADDRESS_H
21
22        # You can see the configuration from the inside of your namespace as follows:
23        sudo ip netns exec $NAMESPACE ifconfig
24
25        # Then to redirect packets to the container
26        $TC qdisc add dev $PHY_ETH ingress
27
28        $TC filter add dev $PHY_ETH parent ffff: prio 1 protocol ip \
29        handle 0x1 flower classid 1:1 dst_ip $IP_ADDRESS_T \
30        action skbmod dmac $VETH_NS_MAC index 1 \
31        action mirred egress redirect dev $H_VETH index 1
32        )
33  }
```

1. Create veth inside host namespace

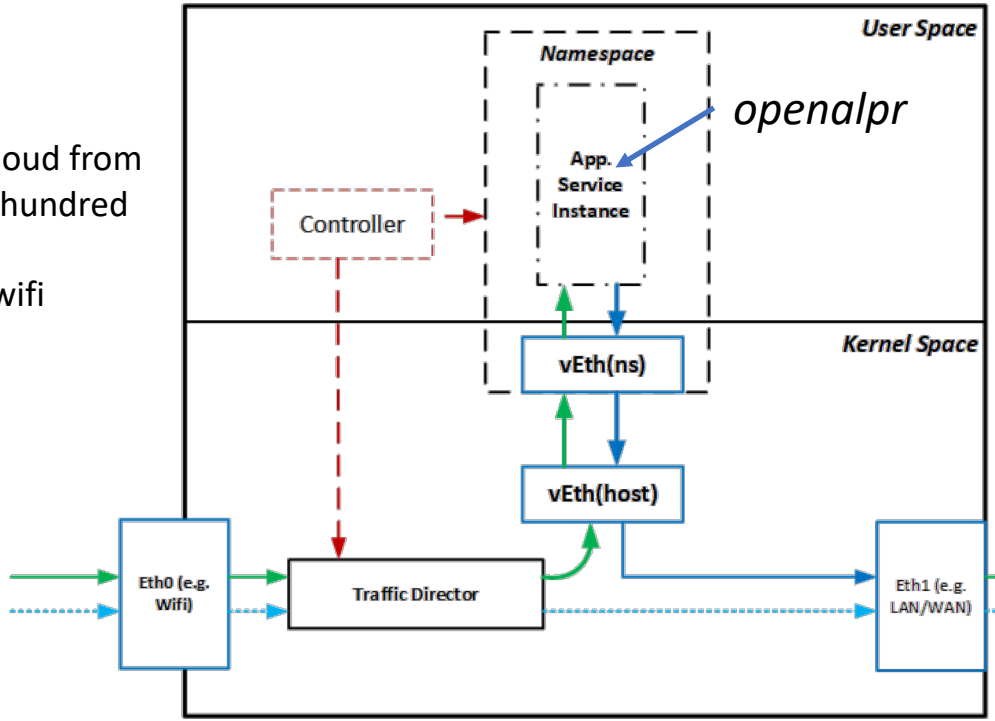
2. Create namespace to host MEC service.

3. Create veth inside guest namespace

4. Create TC based Traffic Redirector

Sample Use Case 1 : License Plate Detection

- Reducing data to cloud from hundred kbytes to hundred bytes.
- Could be done on wifi hotspot, eNodeB.



- Huge saving on bill and RTT if using expensive links, e.g. satellite links

NY, ABC 123

Running on AP

Sample Use Case 2 : SpeedTest Server on Edge

- Motivation
  - Wireless link (e.g. 5G) may not be bottleneck link anymore.
  - Current cloud based speed test framework may report inter-domain congestion while customer and mobile provider has more concern on wireless link congestion.
  - Saturate 5G wireless link from cloud may not be a trivial task.
  - Bandwidth measurement traffic is meaning less on non bottle neck links.
- Viscum Solution
  - Simply deploy iperf3 daemon inside a Viscum host AP.
  - No new hardware, no traffic on core network, and accurately measures wireless link capacity.

Conclusions

- Viscum
  - A light-weight software architecture for edge device MEC
  - Support gradually deployment model.
  - Convert any Linux based network node into MEC application servers.
  - Deploy off-the-shelf software onto mobile edge nodes (e.g. WIFI hotspot, eNodeB) without software splitting.
  - Reduce service delay and traffic volume in mobile providers' core networks and/or radio access networks.
- Future works
  - Distributed application deployment and management
  - Viscum on OpenWRT WIFI routers

Thank you.