Akraino MEC Hackathon

Here is our current plan for the µMEC setup in the Akraino MEC Hackathon on November 18, 2019. The event happens at the same week as Kubecon, has an ETSI label, and is short, so the constraints are

- Use Kubernetes
- Provide ETSI MEC interfaces
- Make something easily accessible for developers

The challenge

The grand purpose of the Hackathon is to demonstrate the power of edge computing. There are three benefits for doing computing at the edge, and we hope to demonstrate these:

1. Edge computing can provide better latencies, especially for AR/VR applications or interactive gaming
2. Processing at the edge can reduce the amount of data that has to transmitted to a central site
3. Edge can host different sensors at the edge

The setup is that there is a base station, like a 5G gNB, and a small computer (µMEC) attached to it, and some of the traffic is routed through the µMEC. The end user can

- interact with the µMEC directly by a specialized app or a browser, using a FQDN (Fully Qualified Domain Name). The FQDN can of course come from a 2D bar code at a a location
- get information from a web site where different µMECs can provided information that has been collected and analyzed on the site

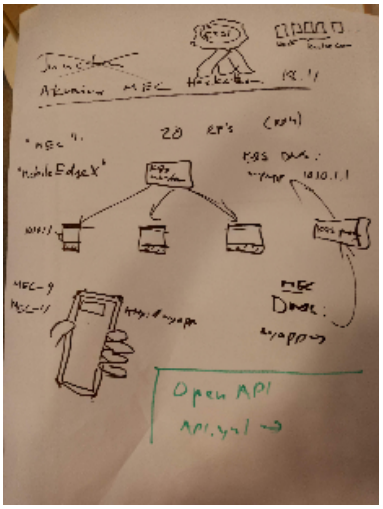
The exact scenario will be revealed at the event. As an imaginary scene, we can imagine providing services for visitors and employees in a zoo. We could have apps that provide extra information about an animal, shows what the animals have been up to, or tell the zookeepers how active the animal has been and what it has been eating.

The ETSI MEC standard interfaces in the Hackathon include:

- Registering a new service
- Searching for services based on name or category
- Creating new DNS services so that eg. a Web app be found with a name only
- Providing information about how to access an app especially if it does not use https with FQDN
- Standard methods of accessing services in the µMEC platform, using OpenAPI that can be compiled into stubs in different programming languages. An example of this is accessing the µMEC camera

The developers can

- create applications that run on the µMEC in containers. If the app is a browser based one, it can of course host html5 code that is then executed in the browser
- create applications that run on a centralized server that use data provided by µMECs
- create applications that run on a phone. This obviously is the most challenging alternative



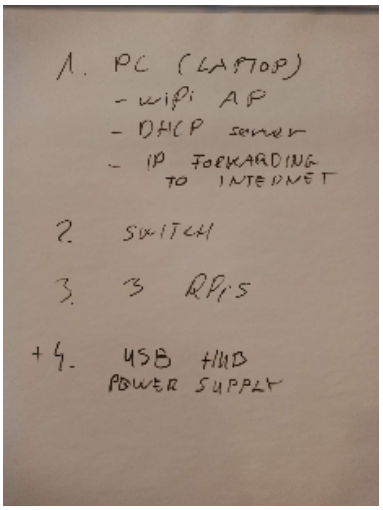
Setup

In order to make the setup of the event as simple as possible, we will use

- A common WLAN network that could be provided by us with a standard WLAN base station. This needs to be connected to Internet also

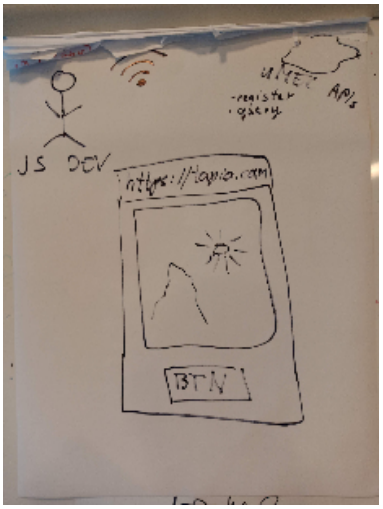
- A few Raspberry Pis that act as µMECs and are preconfigured
- A PC that can also be configured ahead of time

The Raspberrys will all be connected as worker nodes in a single Kubernetes cluster. The Kubernetes master will either run in one of the Raspberrys or in the PC.

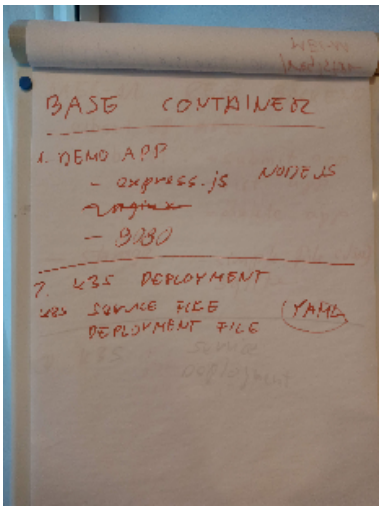


Example material

There will be example applications that can be used as a basis for development, to speed things up. The first example app will be based on Javascript code that implements a Web server, using the standard JS web server. The web server will server html5 code that takes a screenshot or video with the users camera, and sends the result to the server.



The second example app will be a simple container that accesses the camera on the µMEC and sends temperature to a database on the PC.



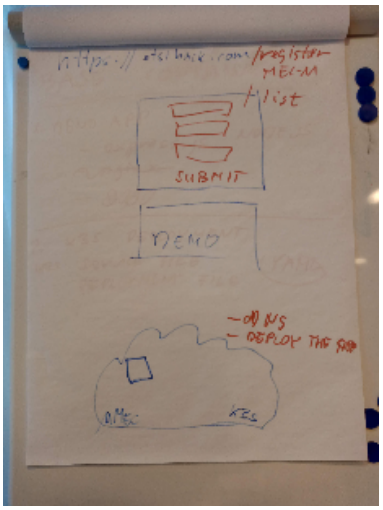
Infrastructure

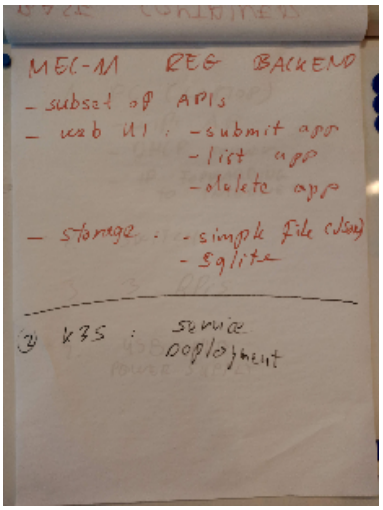
The infrastructure available to developers will include

- An ETSI MEC-9 compliant service registry that provides the server API and DNS API but will reject any requests for Traffic Management API ("no permission")
- A web front end to the ETSI MEC-9 service registry
- An ETSI MEC-11 compliant Camera API

We have also discussed about

- Location API
- ML API (Tensorflow_serving?)




MEC-M REG BACKEND
- subset of APIs
- web UI : - submit app
- list apps
- delete app
- storage : - simple file (JSON)
- Sqlite
3) K8S : service
deployment