Edge computing testbed for V2I applications prototyping and evaluation *

*Note: Sub-titles are not captured in Xplore and should not be used

1st Zeinab Nezami
School of Computing
University of Leeds
Leeds, United Kingdom
email address or ORCID

2nd Evangelos Pournaras
School of Computing
University of Leeds
Leeds, United Kingdom
email address or ORCID

6th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—**

***Index Terms*—Edge Computing, V2I application, Testbed**

I. Introduction

Challenges: ITS applications should deal with various data provided through various communication links. The density and the mobility of the vehicles are the major factors to take into account when developing V2X applications that are destined to run in mobile edge computing distributed environment. In a high-density scenario, the application must use the available bandwidth efficiently in order to deliver its service continuously to the subscribed nodes. The application should also deal with nodes joining/leaving the network during their mouvement and guarantee the service continuity. The MEC infrastructure is intended to offer a management system and a couple of standard services that facilitate the aforementioned properties. The MEC infrastructure introduces a whole new service model that requires an adapted development paradigm. An important aspect that should be taken into account while developing a V2X application that targets MEC as a deployment environment is to ensure the proper interaction of the application with the management entities and the system components. At the development phase of a V2X MEC service components testing in real environment increase deployment time, costs, and complexity. In the next section, we analyze the standard deployment architecture proposed by the European Telecommunications Standards Institute (ETSI) to extract the main components and their functionalities.

II. Related work

TABLE I
Table Type Styles

Table Head	**Table Column Head**		
	Table column subhead	***Subhead***	***Subhead***
copy	More table copy[a]		

[a]Sample of a Table footnote.

funded by AT

III. Testbed Design

The proposed testbed aims at helping researchers and developers to design and verify distributed V2I applications and algorithms destined to run in an edge computing environment. The main components of the testbed are agent, network infrastructure, service distributor, connector, and system orchestrator. The following subsections outline the modules as shown in Fig.1.

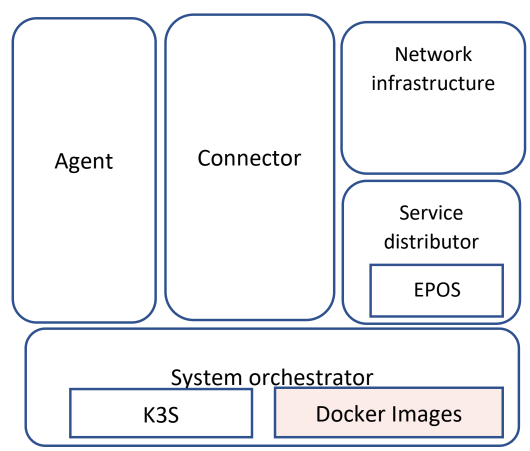

Fig. 1. Proposed test-bed model

A. Agent

An agent is an abstraction of a mobile entity such as vehicle or mobile device equipped with a set of sensors (e.g., GPS, camera), enabling the surrounding environment perception (e.g., position, speed, temperature) and a set of actuators (e.g., display, acceleration).

1) Mobility module: Each agent has a mobility profile. This module's tasks consist of initiating/updating vehicle nodes position, creating/destroying communication's links, and managing mobility The mobility support is achieved through the use of the mobile nodes' positions regarding the fixed edge nodes and the city map to create and destroy communication links in runtime.

B. Network infrastructure

The edge infrastructure offers a distributed limited storage and processing capabilities in the vicinity of the agents to cope with the Cloud issues in running V2I applications regarding real-time services ultra-low latency requirements. Edge computing offers ultra-low latency, high bandwidth, and real-time access to radio network information that can be leveraged by emerging vehicular safety and navigating applications. The "edge node" term refers to the combination of base stations (or access points) and edge servers close to the mobile radio network.

C. Connector

Cellular networks are characterized by a wide communication range, which allows a base station to maintain connectivity with a network agent (vehicle) as long as possible, which means fewer handover operations. The upcoming Fifth Generation cellular network (5G) is one of the leading technologies that promises to grant a very high network capacity that guarantees high throughput/bandwidth for demanding applications like Augmented reality on autonomous vehicles. 5G networks will natively include mobile edge computing capabilities by design that may support different vehicular communication technologies. In our testbed, connector is responsible for connecting different entities together and synchronising data pipelines. The communication of vehicles with a serving entity through a network is also managed with the connector.

D. System orchestrator

The management layer named orchestrator maintains an overall view on available computing, storage, and networking resources and services. It is responsible for:

- Scaling up and down the available resources as required by the running applications.
- Allocating and releasing the storage, networking, and compute resources offered by the service distributor.
- Storing application images for faster instantiation procedure when it is required.
- Providing support for fault and performance monitoring by collecting resources and running application data and transmitting them to evaluation entities(?).

E. Service distributor

It handles the task of the appropriate host selection for requested services deployment by taking into account the application requirements, the available resources, and the network nodes' positions.

IV. Architecture/implementation

The general architecture and the essential components of our proposed testbed are illustrated in Fig.2

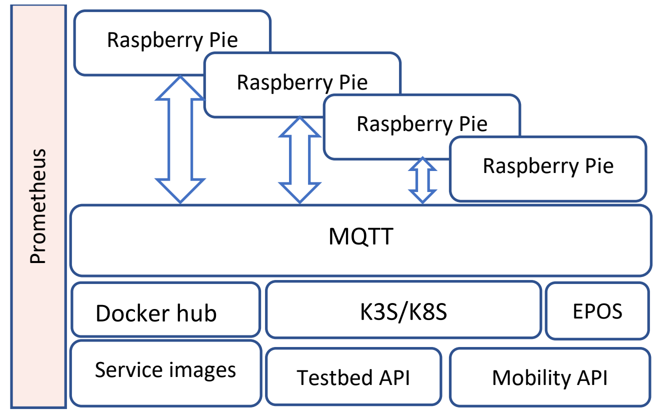

Fig. 2. Proposed test-bed architecture

A. V2I testbed API

V2I API is the core component of the testbed. It implements the required functions and interfaces to create the infrastructure environment. The testbed API implements an abstraction for edge node that are necessary to build a network environment (topology definition and links creation). Furthermore, it allows the developer to apply resource models that define the available resources for each host, and interact with the real components. The real environment is built of raspberry pies and regular servers....

The testbed API interacts with kubernetes to instantiate the cluster nodes from raspberry pies and then instantiate V2I application components from pre-created Docker container images. A container image can be instantiated more than once in a network environment. An edge node is a collection of real nodes and their respective resources model and its associated access point to emulate a group of edge nodes. This abstraction allows the management of the related set of edge server and edge access points as a single entity. The V2X services could run in one or more nodes inside an edge environment.

The developed edge computing testbed API enables adding/removing agents and edge nodes, and connecting/disconnecting containers to agents dynamically within the created network topology. These extended features allow the emulation of real-world edge infrastructures in which it is possible to start and stop services instances at any point in time. Also, it allows the adjustment of the containers and the resource limitations (processing power and memory resources) at runtime by interacting with the kubernetes engine.

B. Kubernetes

C. MQTT

D. EPOS

E. Rasbpery pies

Each edge node has a fixed position in a city that is the input provided by the developer. The nodes positions could be extracted through the interaction with a real-world map offered by the interconnection with a mobility model that calculates the successive node position following the implemented mobility model.

F. SUMO

The developed mobility API can be used to implement different mobility models. A real-world map, in our tests Munich city map, could be imported to the SUMO mobility simulator to model an accurate mobility model that simulates vehicle movements.

G. Services

Deploying Services in the form of Docker instances offers a virtualized isolated environment for service execution at each host. The use of virtual interfaces and virtual links allows different nodes and instances to communicate easily.

V. Application/scenario

The workflow of creating and evaluating atypical V2I application using our testbed is demonestrated in the Fig.3.

A. Methodology for evaluation

VI. Evaluation

A. Threats to validity

B. Conclusion

References

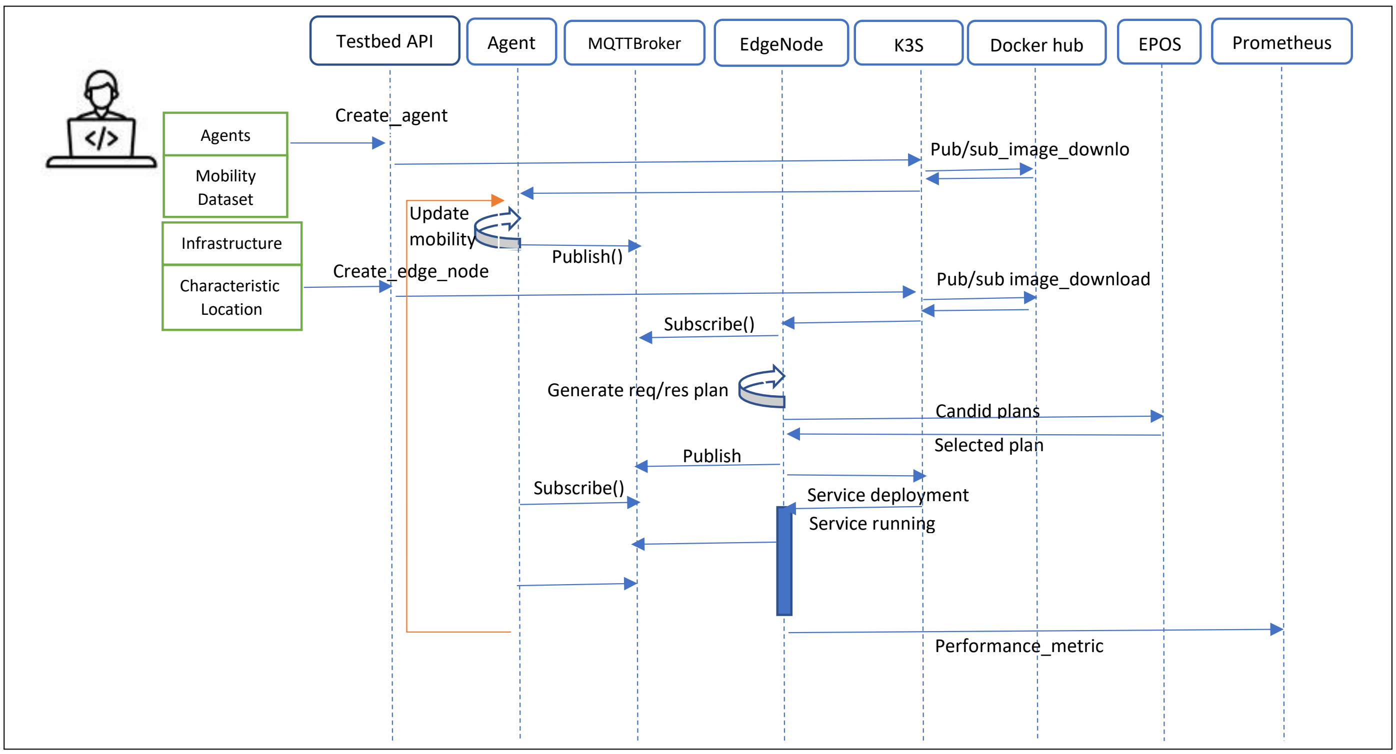

Fig. 3. V2I application workflow