Designing a Cloud-native Weigh-In-Motion

Sivaramalingam Kirushanth
[1]*Department of IT*
Cape Peninsula University of Technology
Cape Town, South Africa
[2]*Department of Physical Science*
Vavuniya Campus of the University of Jaffna
Vavuniya, Sri Lanka
SivaramalingamK@vau.jfn.ac.lk

Boniface Kabaso
Department of IT
Cape Peninsula University of Technology
Cape Town, South Africa
KabasoB@cput.ac.za

***Abstract*— This paper discusses the system architecture of a cloud-native Weigh-In-Motion (WIM) system prototype. The prototype WIM system was developed to ingest Vehicular Telematics (VT) data from several Internet of Vehicles (IoV). It is a necessity to build a cloud-native system to serve VT data from IoV devices. There are some design issues to be considered before designing. This paper discusses how the system prototype was designed by considering the design aspects as a reactive system. Four major design solutions, 1) Kubernetes, 2) GlusterFS, 3) Kafka Cluster, 4) Akka Streams are used in the design. The explanations for choosing the solutions were also discussed.**

Keywords*— *Cloud-native, Internet of Vehicles, Weigh-In-Motion, Vehicular Telematics, Reactive Systems, Kubernetes Cluster

I. INTRODUCTION

Vehicular Telematics (VT) data is becoming more abundant and more accurate due to the technological improvements in sensors and connectivity. This enables the rise of the Internet of Vehicles (IoV) [1]. Determining the weight of a vehicle, also known as Weigh-In-Motion (WIM), has been done in various ways. Traditional WIM uses stationary scales such as static-weighbridges, which are commonly used to measure industrial and commercial vehicles such as Trucks and Lorries [2]. They, however, take a quite long time to weigh each vehicle. Moreover, their costs of system installation and maintenance are expensive [3], Static-weighbridges are more accurate than non-static weighbridges, only when the vehicle is stationary. Non-static WIMs (Low-speed/High-speed WIMs) are deployed in roads to detect load violations. Non-static WIMs use various parameters to detect a vehicle's weight [4]. Some of the parameters of non-static WIMs are pavement vibration [5], and magnetic signal based on single micro-electro-mechanical system (MEMS) magnetic sensor [6]. Even various kinds of WIM solutions are being used in practice, each of which has its advantages and disadvantages. Machine Learning (ML) approaches are widely used in identifying driving behaviour, road anomaly detection [7], and also in other WIM solutions [8]. Building a new (WIM) system to infer the payload of a vehicle on any road segment using VT data and ML would help the transport industry [9]. A prototype WIM system was developed to validate the idea of the new WIM system approach. The system comprises three components,1) an Onboard Diagnostics II (OBD-II) Bluetooth/WiFi module, 2) an android mobile device, 3) a WIM ML inference engine.

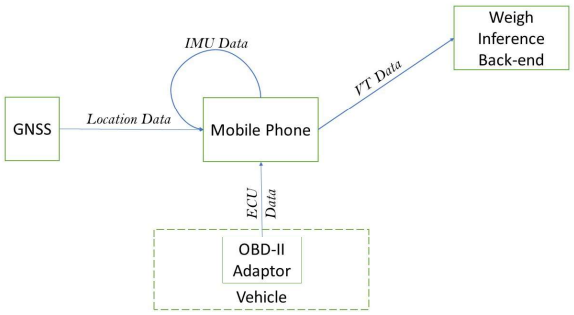

Fig. 1. Representation of the developed prototype WIM system with its components

The OBD-II module was used to collect the Controller Area Network (CAN) bus data. The Android mobile device was used to fetch CAN bus data from the OBD-II module via Bluetooth/WiFi. Android mobile device collected Engine Control Unit (ECU) data for each second and stored it along with the data from the built-in Global Navigation Satellite System (GNSS) position data. The collected/stored data was then sent to WIM Application Programming Interface (API) server using the REST clients through the service API endpoints. Fig. 1 shows the schematic diagram of the developed prototype. An Android phone collected the data from the OBD-II module via Bluetooth, its internal Inertial Measurement Unit (IMU), and GNSS. The collected data was then transferred from the phone to the back-end server WIM application. The system was built to collect weather data and VT data. The collected data is used to train the ML inference models. The trained models are then used to infer the payload of vehicles based on the incoming VT data. The whole system was built as a cloud-native application as it ingests big VT data from various IoV devices. Next section discusses the design challenges of the WIM system, introducing the reactive manifesto [10]. The following sections explain the solution built to overcome the challenges.

II. WIM SYSTEM DESIGN CHALLENGES

In addition to connectivity and security concerns, the cloud-native system needs to be able to handle huge data from IoV devices (i.e. VT data collection modules) in the real world. The

978-1-7281-3464-2/19/$31.00 ©2019 IEEE