Towards a Service Centric Contextualized Vehicular Cloud

Xiping Hu[1], Lei Wang[2], Zhengguo Sheng[1], Peyman TalebiFard[1],
Li Zhou[3], Jia Liu[4], Victor C.M. Leung[1]
[1] Dept. of Electrical & Computer Engineering, The University of British Columbia, Vancouver, Canada
[2] Centre for Telematics and Information Technology, University of Twente, Enschede, The Netherlands
[3] College of Electronic Science and Engineering, National University of Defense Technology, Changsha, China
[4] Dept. of Communication Engineering, Beijing Information Science & Technology University, Beijing, China
{xipingh,zhengguo,peymant,vleung}@ece.ubc.ca, wangl@ewi.utwente.nl, zhouli2035@nudt.edu.cn

ABSTRACT

This paper proposes a service-centric contextualized vehicular (SCCV) cloud platform to facilitate the deployment and delivery of cloud-based mobile applications over vehicular networks. SCCV cloud employs a multi-tier architecture that consists of the network, mobile device, and cloud tiers. Based on this architecture, we present a seamless solution for delivery of personalized mobile applications to vehicular users in an intelligent and reliable manner. We further develop and deploy prototype mobile applications on SCCV cloud, to demonstrate the desired functionality and feasibility of SCCV cloud. Our practical experiments show that SCCV cloud works efficiently with low networking overhead on popular mobile devices in real-world transportation scenarios.

Categories and Subject Descriptors

D.2.4 [**Computer Systems Organization**]: COMPUTER COMMUNICATION NETWORKS - Distributed Systems - Distributed applications.

General Terms

Design, Human Factors, Economics.

Keywords

Vehicular cloud, mobile application, service, context.

1. INTRODUCTION

Driving is an integral part of our everyday lives, and the average driving time of people globally is increasing to 84 minutes everyday [1], which is a time when people are uniquely vulnerable. Poor road condition, traffic congestion and long driving time may bring negative emotion to drivers and increase the chance of traffic accidents.

With the development of mobile computing, cloud computing and vehicular networks, mobile devices such as smart phones can work with customized cloud computing platforms to provide emerging and personalized services to vehicular users anytime and anywhere [2]. Previous research works have demonstrated that such cloud based mobile services in vehicular networks can be effectively used for many purposes, such as: (a) Safety improvements: applications that improve the safety of drivers, passengers and pedestrians by notifying them about any dangerous situations on the roads [3]; (b) Traffic management: applications that provide users with up-to-date traffic information and recommendations that enable them to make better decisions to reduce travel time, and hence improving traffic flow and driving efficiency [4]; (c) Entertainment: applications that enable the streaming, downloading, or sharing of multimedia files among travelers using vehicular networks [5].

Currently, a variety of system level solutions have been proposed for cloud based mobile applications in vehicular networks, such as V-Cloud [6], ITS-Cloud [7], and TIaaS [8]. Nevertheless, these solutions mostly only focus on the efficient dissemination of traffic information across cloud and vehicles. Not much work exists that has considered how to enable users in vehicular environments to conveniently and widely access personalized cloud services that fit their preferences. Development of such an approach is challenged by the heterogeneous service requirements and preferences of vehicular users (which may depend on their age, sex, ethnic origin, etc.) and different operating systems (OSs) (e.g., Android, iOS, Windows Phone) employed in mobile devices onboard vehicles. There is no specific mobile application that can fully meet such requirements and conveniently be used by different vehicular users [9].

Furthermore, as the availability and connections of an individual's mobile device onboard a vehicle may be unreliable, such as due to mobile OS crashing, battery exhaustion, and intermittent networking disconnection in vehicular networks, this may result in service failures and impact the quality of experience (QoE) of vehicular users [10]. However, most of the existing solutions [11, 12] for service failure handling rely on specific transport layer protocols, and could not support different cloud based mobile applications in vehicular networks widely. Consequently, a seamless and effective solution that could facilitate the widely real-world deployment and delivery of different personalized cloud based mobile applications for vehicular users in transportations still needs be investigated.

This paper fills the gaps identified above by proposing SCCV cloud, a service-centric contextualized vehicular cloud to facilitate the deployment and delivery of personalized mobile applications for vehicular users in an intelligent and reliable manner. Our major contributions are summarized as follows:

- We present the overall design and practical implementation of SCCV cloud, which is first systematic approach that can intelligently and reliably optimize the composition and

DIVANet'14, September 21--26, 2014, Montreal, QC, Canada.

http://dx.doi.org/10.1145/2656346.2656351

delivery of cloud services to mobile devices onboard vehicles. Also, SCCV cloud provides a novel multi-tier architecture and foundation supports for the deployment of different personalized cloud based mobile applications for vehicular users.

- We deploy and evaluate SCCV cloud through a set of real-world scenarios, which not only verify the feasibility of SCCV cloud, but also provide practical experience that inspires future research and deployment of cloud based mobile applications for vehicular networks.

The organization of the rest of the paper is as follows. Section 2 reviews the background of related techniques and concepts about SCCV cloud. Section 3 presents the system architecture design and key components of SCCV cloud, and discusses how they can facilitate the delivery of personalized cloud services to mobile devices onboard vehicles. Section 4 shows application examples of the proposed SCCV cloud. Section 5 evaluates the system performance of SCCV cloud. Section 6 reviews existing system level cloud based solutions for transportation applications and compares them with SCCV cloud. Section 7 concludes the paper.

2. BACKGROUND

Web services have paved the way for a new type of mobile software system like mobile cloud platforms, and the best-practice of service-oriented architecture (SOA) will help to make web services a success [13]. The Representational State Transfer (REST)-ful Web Service [14] is an approach to using REST purely as a communication technology to build SOA, where services are defined using SOA style decomposition and REST-based Web Services are leveraged as a transport. Different from conventional SOA based solutions, the RESTful Web Service is well suited for basic, ad-hoc integration scenarios with a low resource overhead (e.g., network overhead) as it is based on a light-weight design principle [15]. Thus, in this paper, we adopt the RESTful Web Service to develop the SCCV cloud for the deployment of cloud based mobile applications and services for vehicular networks.

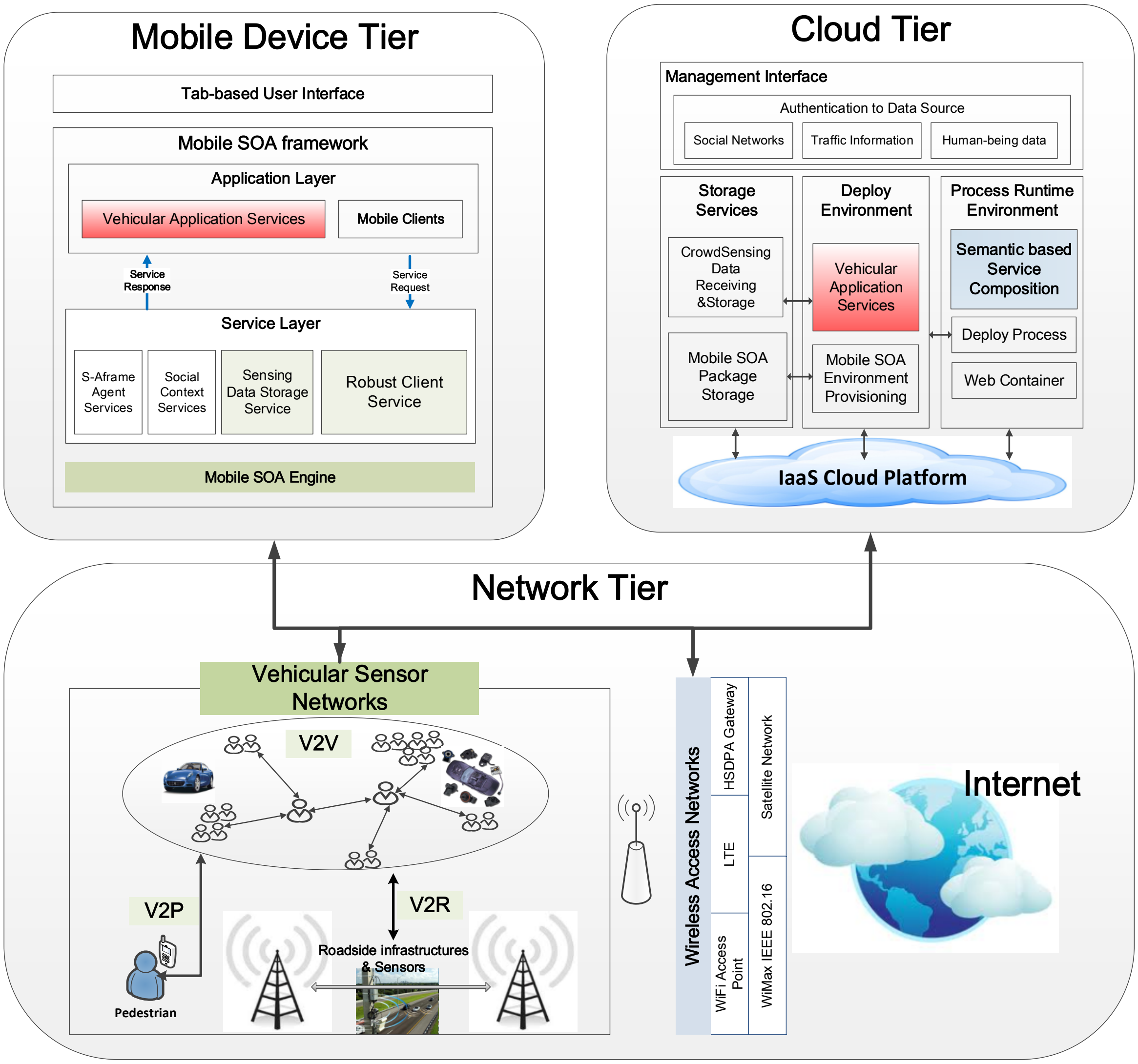


Figure 1. System architecture of SCCV cloud

3. OVERVIEW OF SCCV CLOUD

As shown in Figure 1, the system architecture of SCCV cloud consists of the network, mobile device, and cloud tiers.

Network tier: The network tier makes use of any available connectivity upon vehicular ad-hoc networks (VANETs) and Internet, such as WiFi direct between mobile devices on board vehicle, Dedicated Short Range Communications (DSRC) between vehicles and roadside infrastructures, and cellular data services for Internet access. As discussed later in this section, one of the crucial challenges on cloud based mobile applications in vehicular networks is how to optimize the deployment of sensors between vehicles and roadside infrastructures, so as to enable the efficient collection of contextual data for use by such applications..

Mobile device tier: Based on the S-Aframe [16] and mobile SOA framework [17] developed in our former work, mobile web service-based vehicular applications can be flexibly deployed on different mobile devices, and work self-adaptively across VANETs and cloud platform via Internet. Moreover, we design and incorporate a novel robust client service (RCS) that handles possible service failures with regard to system crashes and network failures, in addition to ensuring a high level of robustness of cloud based mobile applications when delivering and processing the computational tasks across the cloud and mobile devices onboard vehicles. More details about this will be presented in Section 3.2.

Cloud tier: The cloud tier works in parallel with the mobile device tier. The cloud tier could be built based on the Vita cloud platform [18] introduced in our earlier work. Also, the cloud tier works as a central coordinating platform to: (i) aggregate the sensing data, traffic data (e.g., road traffic monitoring through mobile crowdsensing [19]), and Internet services (e.g., weather and geographic information, social activities) from multiple sources; (ii) interpret the data, and compose and deliver the personalized mobile services to different vehicular users dynamically. For this, we design a novel semantic based service composition (SSC) model which can intelligently compose and deliver different personalized mobile services to vehicular users in real-time, which will be introduced in Section 3.3.

3.1 Optimization of sensor deployment in vehicular networks

In SCCV cloud, we define sensors as a low-level context that is directly referred to as raw data. Also, for cloud based vehicular applications, a sensor describes not only a physical device, but also a data source that could be useful for context representation. In SCCV cloud, a wide range of contextual data may be collected in terms of representations of real-world phenomena as cyber-world entities. Thus, in the network tier of SCCV cloud, we propose a heterogeneous network infrastructure to support real-time sensing data fusion. Table 1 classifies the usage and deployment of multidimensional sensors in the network tier.

The integration of sensing and advanced computing capabilities in network-enabled mobile devices to produce sensing data and exchange information among local or system-wide resources in a social scale. This concept of Internet-of-Things (IoT) [20] and wireless sensor networks [21] will form a collection of autonomous, ambient intelligent and self-operated network nodes (e.g., independently acting smartphones) that are well aware of surrounding context, circumstances and environments.

Similar to our former work [22], the sensor deployment in SCCV cloud uses various devices to capture events or monitor status of different things, such as traffic patterns or accident, which are relayed through gateways to upper layers via wireless, wired, or hybrid networks. With optimizations of sensor deployment in vehicular networks [22], the new network architecture could enhance the credibility, quality, and privacy of data and ease the sharing of data among vehicular application and services.

3.2 Semantic based service composition for delivery of cloud service

Efficient delivery of cloud services demands leveraging the contextual information to identify the most relevant service for the consumers. In the SOA based design principle, such consumers of services can be user devices that need to adapt to the network connectivity or type of service, or other auxiliary services that need to perform dynamic functional composition or change the mode of service.

In our former work [23], we have demonstrated that semantically rich applications can benefit from a higher-level view of topology that can manifest the connectivity of objects such as services, content items and users to enhance the control and data planes. Thus, similar to our proposals in [23, 24], we design the SSC model in the cloud tier of SCCV cloud by bringing a new level of semantic meaning towards networking of information, so that the underlying transport can benefit from the Information Centric Networking (ICN) paradigm to enrich the vehicular cloud services with semantics. The SSC model also considers the connectivity of information objects and services that mimics the social connectivity of objects, using the social context service module shown in Figure 1. It is based on commonality of interests, such as contextual information, among entities as indicated by how nodes are connected to the neighboring clusters and what characteristics make them a better alternative as candidate seed nodes for a specific type of content described by the prefixes.

Table 1. Classification of Sensor Deployment in SCCV Cloud

Type of sensors	Descriptions
Physical sensors	Refer sensors which can capture almost any physical world belonging data, e.g., GPS: location, accelerometer: activity.
Virtual sensors	To have source information from software applications and/or services and imply a semantic data obtained through cognitive inference, e.g., location info by manually entered place pinpoint through social network services or computation power of devices.
Logical sensors	Define combination of physical and virtual sensors with additional information obtained through various sources by user interactions, e.g., databases, log files.

Furthermore, we integrate an ICN-based edge cloud framework and a context-aware service composition methodology [25] in the SSC model. Then, the network functions could be virtualized on a generic ICN-based service platform. Also, configurable and context-aware service orchestration can be located in the vicinity of the operator's Central Office (CO) or Points-of-Presence (PoP) that enables local instantiation of ICN based services to consumers while providing global service delivery through a unified service infrastructure.

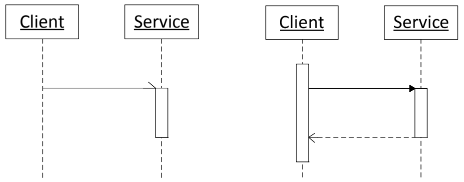
In addition, as shown in Figure 1, the social context service module can perform contextual information management that is in charge of the following tasks [26]: *context collection and aggregation* that involve collection of information from various sources (i.e., service context, user context or network context) and monitoring the dynamics of a service for context changes; *context processing* that involves modeling different types of context to perform filtering and transformation or inferring higher level context information for application specific interpretation; and *dispatch and distribution* of various contextual information.

3.3 Robustness enhancement between mobile devices on board vehicles and cloud

Normally in vehicular networks, the interactions between mobile devices and cloud services are based on messages exchange [27]. If a mobile client application changes its state, e.g., the values of its variables or its embedded databases, it sends messages to cloud services to inform them about this change. However, system crashes and network failures may result in loss of messages. In this case, only one side changes its state, which may result in global state/behavior inconsistencies and possible deadlocks. In this part, we propose a novel RCS to enable robust interactions between mobile devices onboard vehicles and cloud services in the presence of system crashes and network failures.

3.3.1 Interaction Patterns

Interaction failures are specific to interaction patterns. Service interaction patterns are discussed in [27], in which 13 interaction patterns are identified. In this paper, we focus on *send*, *receive* and *send-receive* patterns. However, since the more complex interactions patterns can be decomposed into simple patterns, it is therefore sufficient to look at the interaction failures of these basic interaction patterns.

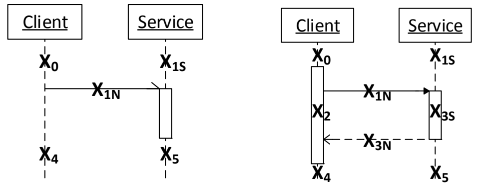


(a) *receive* (b) *send-receive*

Figure 2. Interaction patterns

Figure 2a shows a mobile client that sends a one-way message to a cloud service. The mobile client corresponds to the *send* pattern while cloud service behavior corresponds to the *receive* pattern. In pattern *send-receive* in Figure 2b, the mobile client starts a synchronous interaction by sending a request and getting a response.

3.3.2 Interaction Failure Analysis



(a) *send* and *receive* (b) *send-receive*

Figure 3. Interaction failures

The possible interaction failures with regard to system crashes and network failures are marked as X_0~$X5$ in Figure 3. X_0, X_4 and X_5 are system crashes. These failure points are irrelevant as they have no impact on the interactions. Failure points X_1~X_3 represent *service unavailable*, *pending request failure* and *pending response failure*, respectively.

X_1 *service unavailable* represents the situation that the cloud service is not available. Failure X_{1S} is caused by a system crash of the cloud service. This is a rare event that can, however, happens sometimes; e.g., the Amazon EC2 cloud services suffered a crash in April, 2011 [28]. X_{1N} is caused by a failure of the network to deliver a request message. In both cases, the mobile client is not able to establish a connection with the cloud service. At the implementation level, the mobile client is aware of the failure through a catchable exception of the client implementation language. X_2 *pending request failure* represents the situation that the mobile client fails after sending a request message. The client is informed about the failure after restart, e.g., through catchable exceptions. Our experiments indicate that the cloud service is not aware of such a failure, such that it replies with the response message and continues execution. X_3 *pending response failure* represents the situation when the response message gets lost. X_{3S} is caused by a responder system crash. X_{3N} is caused by a network failure in response message delivery. In both cases, after restart the cloud service replies with the response message and continues execution. However, the connection gets lost and the mobile client cannot receive the response message. The client is aware of this failure after a timeout.

Failure Assumptions

Due to the heterogeneous infrastructure, e.g., different mobile OS implementations or network environments, we have to make the following assumptions concerning the failure behavior of the infrastructure:

1) *Persistent Execution States*. The states of a mobile client (e.g., it embedded database) are kept persistent and can survive system crashes.

2) *Network Failures*. Service interactions are supported by exchanging HTTP messages (RESTful messages in our scenario) over TCP connections. HTTP normally uses the same TCP connection for the synchronous request and response messages (Figure 2b). Therefore network failures interrupt the established network connections, so that all the messages that are in transit get lost.

3.3.3 Our Recovery Method

Our idea is to transform mobile apps into their robust counterpart, as shown in Figure 4. The robust apps is deployed on the Mobile SOA platform and is recoverable from the above mentioned interaction failures.

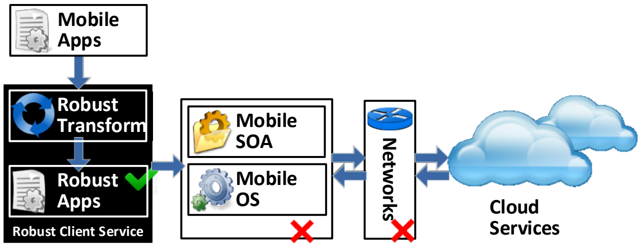


Figure 4. Overview of our recovery method

The transformation principle is to make the mobile apps resend the request messages when interaction failures happen. The design of the cloud services incorporates the capabilities of caching response messages and using them as replies if the mobile clients resend request messages due to interaction failures.

Mobile Client Transformation

The transformation is to make the mobile client resend request messages after a failure. The transformation works at the client implementation language level. Given a mobile client implementation, where the cloud service is invoked, shown as the upper part of Figure 5, we transform it into the robust counterpart, shown as the lower part of Figure 5, in a JAVA like pseudo code.

original client,
a cloud service invocation:

```
result = cloudService.invoke(parms);
```

transformed client,
robust cloud service invocation:

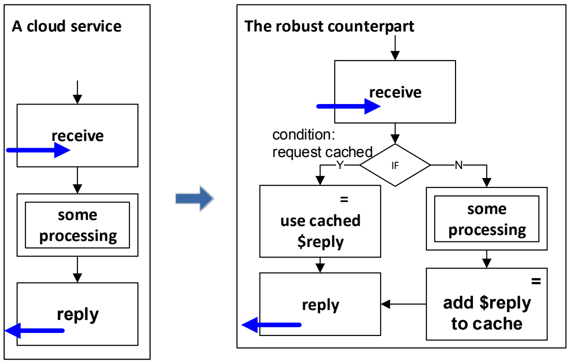
```
success = false;
while(!success) {
 try {
   result = cloudService.invoke(params);
   success = true;
 } cache (InteractionException e) {
   Thread.sleep(t);
 }
}
```

Figure 5. Mobile client transformation

The cloud service invocation is nested in a try-cache block such that if an interaction failure happens, the failure will be caught and the outside "while" iteration will cause the service request message to be sent again. If no failure happens, the flag variable *success* will be assigned to *true* to end the "while" iteration.

Robust Cloud Service Design

If a request message can be safely repeated, it is called idempotent, e.g., a request for weather information. However, not all request messages are idempotent, e.g., an order submission request. If a request is not idempotent, the cloud service should use the cached response message as a reply when it receives a request message that is resent due to a failure. The criteria of whether a request is idempotent are discussed in [29]. As shown in Figure 6a, the cloud service receives a request message, does some processing and then replies. Our corresponding robust design is to replace the processing and reply by an *if* branch, where the condition of the *if* checks whether the request message is cached. If it is cached, the cloud service uses the cached response as the reply. If the message is not cached, which implies that the message is sent for the first time, the message is processed. The response message is cached and replied.



(a) **(b)**

Figure 6. Our robust cloud service design

3.3.4 Implementation and Validation

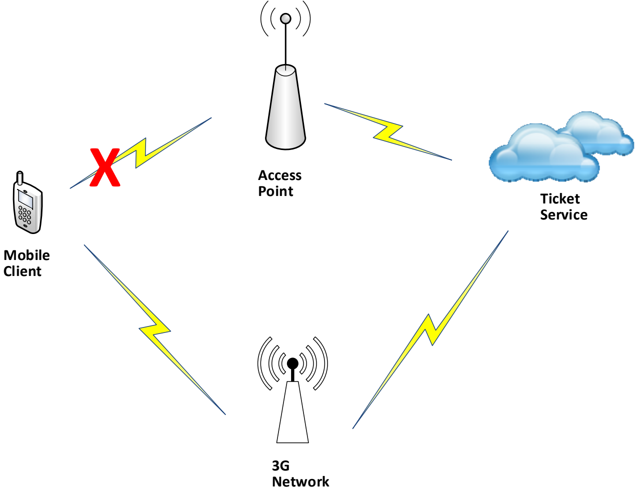


Figure 7. Setup of the test environment

The setup of the test scenario is shown as Figure 7. A ticket service is deployed on the Amazon EC2 compute cloud. The mobile client sends a request message to subscribe a ticket via a WiFi connection to the Internet. To simulate a pending response failure, we shut down the access point after the ticket service has received the request. Then we switch on the 3G cellular radio in the mobile client. After a network reconnection, the client resends the request message. The ticket service sends the previous cached subscription result, without reprocessing the duplicate request message. The failure recovery is transparent to the user.

4. APPLICATION EXAMPLE

Based on the SCCV cloud, we have developed and deployed a prototype of SAfeDJ [30] to demonstrate the functionalities of

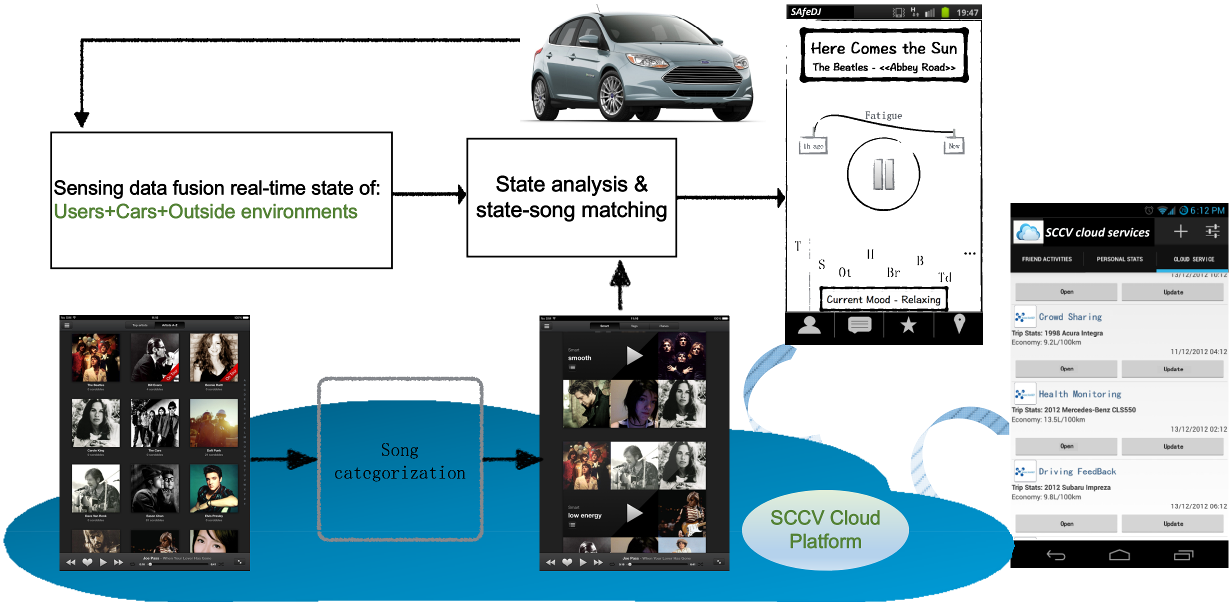


Figure 8. Application example of SCCV cloud

SCCV cloud for cloud service delivery. As shown in Figure 8, SAfeDJ consists of three major components: (i) sensing data fusion service, (ii) music library and matching service, and (iii) music sharing and social networks. The sensing data fusion service provides real-time state analysis of the current driving situation by integrating the driver's health data, the car sensors, and the road information. The health data (e.g., heart rate) of drivers can be collected by wearable sensors such as wristbands, while the car data (e.g., speed, temperature, fuel consumption) can be read by advanced onboard diagnostic (OBD) port scanners. The outside environment can be deduced from data provided by roadside and in-car sensors, news summary, and updates from other drivers. All the input data are fed into the real-time state analysis module in the cloud to infer the mood of the driver and the current driving condition, based on which the state analysis module performs music matching. For example, if the driver is under stress as he is late for work in a traffic jam, then some relaxing music will be played to relieve his stress. The music library is updated from time to time in the cloud, and connected to the music sharing service via social networks. This enables drivers to share enjoyable music with their friends who have similar music preferences or experience similar traffic conditions.

At the same time, further leveraging the advantages of cloud computing, the SSC model in cloud can extract and aggregate contextual data from multiple sources (see details in Section 3.1). The personalized mobile services uploaded by developers with driver feedbacks can be automatically and intelligently delivered to the drivers' mobile devices. For instance, the screen shot in the right corner of Figure 8 shows the extended *crowd sharing service*, *health monitoring service* and *driving feedback service* which could be intelligently delivered to the drivers' mobile devices from the cloud during driving.

5. EVALUATIONS

We evaluate the system performances of SCCV cloud in terms of two parameters: time efficiency and networking overhead of mobile devices onboard vehicles, as these parameters are of particular concern for safe driving vehicular applications in the real-world. The communications between the cloud tier and the mobile device tier of SCCV cloud use the standard web service format based on HTTP and XML. The experimental environment involves: ***Hardware*** **-** Amazon EC2 M1 Medium Instance with 3.75 GB memory, 2 EC2 compute units, 410 GB storage, 32-bit or 64-bit platform, moderate I/O performance, and no EBS-optimization; ***Software*** **-** Ubuntu 14.04 OS, Apache Tomcat 8.08 servers, and Apache ODE1.3.4 BPEL engine; ***Mobile Client*** - 2005 Toyota Sienna vehicle with ELM327 Bluetooth OBD-II module and Google Nexus 4 smartphones with Android 4.4.2 OS.

Two Nexus 4 smartphones (with LTE module) were used onboard the 2005 Toyota Sienna in driving tests. A total of 5 tests were run, each lasting 30 minutes, and the average results were calculated. Using the □£-GALEN ontology [31] as a benchmark, computation tasks were performed to index and calculate the similarities of concepts on this ontology under four different size assertions (1000, 1500, 2000, 36000), from which we obtained four corresponding sets of data. For each data set, we tested the time efficiency of the task in two situations: performing the task on Nexus 4, and performing the task by uploading it to the cloud platform of Vita. Tasks were initiated according to a Poisson distribution with a rate of E=5/min. In addition, we recorded the network overhead on the Nexus 4 for uploading the tasks to Vita.

The experimental results are summarized in Table 2. The time delay when performing a task in the cloud consists of: (i) the response and communication time between the Vita cloud platform and the mobile device; and (ii) the processing time of the task in the cloud. We find that the response times of the four sets of data are similar, with all averaging about 4.5s, while the processing time mostly depends on the size of the data set. From Table 2, we can see that the Nexus 4 gets a better time efficiency when the sizes of the data sets are 1000 and 1500, while the cloud performs much better when the sizes of data sets are larger than 1500. Since SCCV cloud enables the computation tasks to be performed dynamically across mobile devices and the cloud platform, thus the maximum time delay is lower than 7s even in

Table 2. Overall System Performance

Parameters	Data set 1 (1000)	Data set 2 (1500)	Data set 3 (2000)	Data set 4 (36000)
Time delay via cloud - response time + processing time	Average: 4683+40=4723 msec	Average: 4475+461=4936 msec	Average: 4626+702=5328 msec	Average: 4395+2483=6878 msec
Network overhead	1.67MB/150 requests	1.69MB/152 requests	1.64MB/147 requests	1.59MB/143 requests
Time delay for local computation	Average: 2234 msec	Average: 4736 msec	Average: 7445 msec	Average: 136073 msec

situations with intensive computing, e.g., when the size of the data set is 36000. Thus, mobile applications deployed on SCCV cloud could be delivered to mobile devices onboard vehicles in an efficient manner and widely used in transportation scenarios.

6. RELATED WORKS

There has been much research works done on cloud based mobile applications for vehicular networks. For example, V-Cloud [6] is a cloud computing enabled cyber-physical system for vehicles, which provides a three-layered architecture to offer cloud computing based real-time services in order to improve driver's safety and comfort. A cloud computing model for intelligent transportation system is presented in [7], which consists of two sub-models: conventional cloud and vehicular cloud. Leveraging advantages of VANETs, a new service layer called Traffic Information as a Service (TIaaS) is proposed in [8] to securely provide vehicular nodes with fine-grained traffic information.

Partly inspired by but different from these works, SCCV cloud aims to provide a general model at the architecture design level by leveraging the advantages of a number of techniques, which salient parts are integrated and orchestrated into a flexible and efficient architecture to facilitate the deployment and delivery of personalized cloud-based mobile applications for vehicular users. Also, SCCV cloud integrates RCS to enable such applications to work robustly and seamlessly with augmented cloud computing platforms in vehicular environments.

Moreover, security and privacy is always a concern for cloud based mobile applications. A number of solutions exist in the literature that could be adopted to address such concerns. For instance, a solution that uses an authorization paradigm for privacy protection in vehicular networks is proposed in [32]. This solution incorporates the Mandatory Access Control mechanism, which is inspired by e-health systems, to define access levels for accessing different parts of personal data, and Geolocation-based Trust Propagation, which is used to obtain trust information concerning certification and attribute authorities valid for a specific location in a vehicular network. The vehicular network trust model in [33] integrates cryptography-based entity trust, which provides security protections (i.e., origin and data integrity), and email-based social trust, which provides a level of belief in the transmitted data. The model adopts identity based cryptography to integrate entity trust and social trust so as to deploy a unique identity to each entity, and use their attributes to develop secure group communications.

7. CONCLUSIONS

In this paper, we have proposed SCCV cloud, a service centric contextualized vehicular cloud that employs a multi-tier architecture to facilitate the deployment and delivery of cloud based mobile applications and services for vehicular users. Based on this architecture, we have provided a seamless solution in terms of optimization of sensor deployment, semantic based cloud service composition and delivery, and robust connection between mobile devices on board vehicles and cloud platform. Our proposed SCCV cloud enables the deployment and delivery of personalized mobile applications for vehicular users in an intelligent and reliable manner. In addition, we have presented novel prototype mobile applications developed and deployed on SCCV cloud to demonstrate its feasibility and usefulness in transportation scenarios. Furthermore, results of practical experiments have been presented to demonstrate the effectiveness and feasibility of SCCV cloud for real-world deployment.

8. ACKNOWLEDGMENTS

This work is supported in part by the Canadian Natural Sciences and Engineering Research Council through the NSERC DIVA Strategic Research Network, by TELUS and other industry partners.

9. REFERENCES

[1] Ding, D., Gebel, K., Phongsavan, P., Bauman, A. E., and Merom, D. 2014. Driving: a road to unhealthy lifestyles and poor health outcomes. *PloS one*. *9*, 6.

[2] Whaiduzzaman, M., Sookhak, M., Gani, A., and Buyya, R. 2014. A survey on vehicular cloud computing. *Journal of Network and Computer Applications*. *40*, 325-344.

[3] Hu, X., Leung, V. C. M., Li, K. G., Kong, E., Zhang, H., Surendrakumar, N. S., and TalebiFard, P. 2013. Social drive: a crowdsourcing-based vehicular social networking system for green transportation. In *Proc. ACM DIVANet*, 85-92.

[4] Lee, E., Lee, E. K., Gerla, M. and Oh, S.Y. 2014. Vehicular cloud networking: architecture and design principles. *IEEE Communications Magazine*. 52, 2, 148-155.

[5] Hossain, E., Chow, G., Leung, V. C. M., McLeod, R. D., Mišić, J., Wong, V. W., and Yang, O. 2010. Vehicular telematics over heterogeneous wireless networks: A survey. *Computer Communications*. 33, 7, 775-793.

[6] Abid, H., Phuong, L. T. T., Wang, J., Lee, S., and Qaisar, S. 2011. V-Cloud: vehicular cyber-physical systems and cloud computing. In *Proc. ISABEL*.165.

[7] Bitam, S. and Mellouk, A. 2012. ITS-cloud: Cloud computing for intelligent transportation system. In *Proc. IEEE GLOBECOM*. 2054–2059.

[8] Hussain, R., Abbas, F., Son, J., and Oh, H. 2013. TIaaS: Secure cloud-assisted traffic information dissemination in vehicular ad hoc networks. In *Proc. IEEE/ACM CCGrid*. 178–179.

[9] Takeda, K., Miyajima, C., Suzuki, T., Angkititrakul, P., Kurumida, K., Kuroyanagi, Y., Ishikawa, H., Terashima, R., Wakita, T., Oikawa, M., et al. 2012. Self-coaching system based on recorded driving data: Learning from one's experiences. *IEEE Transactions on Intelligent Transportation Systems*, 13, 4, 1821–1831.

[10] Lee, E.A. 2008. Cyber physical systems: Design challenges. In *Proc. IEEE ISORC*. 363–369.

[11] Moser, O., Rosenberg, F., and Dustdar, S. 2008. Non-intrusive monitoring and service adaptation for ws-bpel. In *Proc. WWW*, 815–824.

[12] OASIS. Web Services Atomic Transaction Version. Available: http://docs.oasis-open.org/ws-tx/wsat/2006/06.

[13] Erl, T. 2005. Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall Englewood Cliffs.

[14] Richardson, L. and Ruby, S. 2007. RESTful web services. Farnham: O'Reilly.

[15] Pautasso, C., Zimmermann, O., and Leymann, F. 2008. Restful web services vs. big'web services: making the right architectural decision. In *Proc. WWW*. 805–814.

[16] Hu, X., Zhao, J., Zhou, D., and Leung, V.C.M. 2011. A semantics-based multi-agent framework for vehicular social network development. In *Proc. ACM DIVANet*. 87–96.

[17] Hu, X., Leung, V.C.M., Du W., Seet B., and Nasiopoulos P. 2013. A Service-oriented Mobile Social Networking Platform for Disaster Situations. In *Proc. HICSS*. 136-145.

[18] Hu, X., Chu, T.H., Chan, H.C., and Leung, V.C.M. 2013. Vita: A Crowdsensing-oriented Mobile Cyber Physical System. *IEEE Trans. Emerging Topics in Computing*. 1, 1, 148-165.

[19] Hu, X., Li, X., Ngai, E.H., Leung, V.C.M., and Kruchten, P. 2014. Multidimensional context-aware social network architecture for mobile crowdsensing. *IEEE Communications Magazine*. 52, 6, 78–87.

[20] Sheng, Z., Yang, S., Yu, Y., Vasilakos, A.V., McCann, J.A., and Leung, K.K. 2013. A survey on the ietf protocol suite for the internet of things: Standards, challenges, and opportunities. *IEEE Wireless Communications*. 20, 6, 91–98.

[21] Sheng, Z., Leung, K.K., and Ding, Z. 2011. Cooperative wireless networks: from radio to network protocol designs. *IEEE Communications Magazine*. 49, 5, 64–69.

[22] Sheng, Z., Fan, J., Liu, C., Leung, V.C.M., Liu, X., and Leung, K. 2014. Energy efficient relay selection for cooperative relaying in wireless multimedia networks. *IEEE Trans. Vehicular Technology*.

[23] TalebiFard, P. and Leung, V.C.M. 2012. A content centric approach to dissemination of information in vehicular networks. In *Proc. ACM DIVANet*. 17–24.

[24] TalebiFard, P., Nicanfar, H., Hu, X., and Leung, V.C.M. 2013. Semantic based networking of information in vehicular clouds based on dimensionality reduction. In *Proc. ACM DIVANet*. 69–76.

[25] TalebiFard, P., Ravindran, R., Chakraborti, A., Wang, G., and Leung, V. 2014. Towards a context adaptive ICN based service centric framework. To appear in *QShine-Q-ICN*.

[26] TalebiFard, P. and Leung, V.C.M. 2011. A dynamic context-aware access network selection for handover in heterogeneous network environments. In *Proc. IEEE INFOCOM-WKSHPS*. 385–390.

[27] Barros, A., Dumas, M., and Ter Hofstede, A.H. 2005. Service interaction patterns. In *Business Process Management*. 302–318.

[28] Hesseldahl, A. 2011. Amazon's cloud crashed overnight, and brought several other companies down Too. Available: http://allthingsd.com/20110421/amazons-cloud-crashed-overnight-and-brought-several-other-companies-down-too/.

[29] Wang, L.; Wombacher, A; Pires, L.F.; van Sinderen, M.J.; Chi, C. 2013. Robust Client/Server Shared State Interactions of Collaborative Process with System Crash and Network Failures. In *Proc. IEEE SCC*. 192-199.

[30] Hu, X., Deng, J. Hu, W., Fotopoulos, G., Ngai , E. C.-H., Sheng, Z., Liang, M., Li, X., Leung, V. C. M., and Fels, S. 2014. SAfeDJ Community: Situation-Aware In-Car Music Delivery for Safe Driving. To appear in *ACM MobiCom'14*.

[31] Rector, A., Rogers, J., Zanstra, P., and Van Der Haring, E. 2003. Opengalen: open source medical terminology and tools. In *Proc. AMIA Annual Symposium*. 982.

[32] Serna, J., Luna, J., and Medina, M. 2008. Geolocation-based trust for vanet's privacy. In *Proc. ISIAS*. 287–290.

[33] Huang, D., Zhou, Z., Hong, X., and Gerla, M. 2010. Establishing email-based social network trust for vehicular networks. In *Proc. IEEE CCNC*. 1–5.