District Department of Transportation

Development of Prototype On Board Unit for Connected Vehicle Initiatives

Final Report

November 15, 2017

Disclaimer

This research was performed in cooperation with the District Department of Transportation (DDOT) and the Federal Highway Administration (FHWA). The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official view or policies of the FHWA or DDOT. This report does not constitute a standard, specification, or regulation.

Development of Prototype On Board Unit for Connected Vehicle Initiatives

Final Report

Mike Mollenhauer, Principal Investigator
Virginia Tech Transportation Institute

November 2017

Research Project
Final Report 2017-01

Technical Report Documentation Page

1. Report No. DDOT-RDT-2017-01	2.	3. Recipient's Catalog No.
4. Title and Subtitle Development of Prototype On Board Unit for Connected Vehicle Initiatives		5. Report Date November 15, 2017 6. Performing Organization Code
7. Author(s) Elizabeth White, Michael Mollenhauer, Kevin Heaslip, Jonathon Overton, Sondra Iverson, Dean Iverson		8. Performing Organization Report No.
9. Performing Organization Name and Address Virginia Tech Transportation Institute 3500 Transportation Research Plaza Blacksburg, VA 24061		10. Work Unit No. 11. Contract or Grant No. DCKA-2015-C-0011 Task 17
12. Sponsoring Organization Name and Address District Department of Transportation 55 M Street, SE, 5th Floor Washington, DC 20003		13. Type of Report and Period Covered Final Report, Sept. 2016 – Nov. 2017 14. Sponsoring Agency Code
15. Supplementary Notes Conducted in cooperation with the U.S. Department of Transportation, Federal Highway Administration.		

16. Abstract
This project focused on creating a platform for the development and evaluation of connected vehicle applications that can enhance safety and mobility, reduce environmental impacts, and improve operations within DDOT's operational region. First, the current status of DDOT's intelligent transportation system (ITS) infrastructure was evaluated and suggestions for improvements to support CV initiatives were made. VTTI developed an onboard unit (OBU) package for installation in five DDOT work trucks to support V2I communications. VTTI developed and integrated a number of software packages with the goal of improving DDOT's operating efficiency. Two software applications previously created by VTTI and based on the existing Virginia Connected Corridors (VCC) infrastructure, VCC Monitor and VCC Mobile, were adapted by integrating the data collected by the OBE installed in DDOT's trucks (i.e., temperature and pothole data). Furthermore, an Android smartphone application was developed from the ground up to collect road surface data, detect potholes, and send that information to VCC Monitor for easy identification and tracking. To support DDOT's data analytics goals using the real-time data transmitted by the equipped DDOT vehicles, VTTI provided DDOT with the necessary information to access and interact with the data in the VCC Cloud. This platform will position DDOT on the cutting-edge of Intelligent Transportation Systems (ITS) research in a short period of time, with efforts targeted specifically at local transportation challenges.

17. Key Words Connected Vehicles, Intelligent Transportation Systems, Connected Vehicle Infrastructure, Surface Temperature, Potholes (Pavements), Pothole detection		18. Distribution Statement No restrictions. This document is available from the Research Program upon request.	
19. Security Classification (of this report) Unclassified.	20. Security Classification (of this page) Unclassified.	21. No. of Pages 145	22. Price N/A

Acknowledgements

The authors would like to thank Jason Tao, Kelli Raboy, Stephanie Dock, Charles Tenbrook, and Victor Brown from the District Department of Transportation for their assistance during this project. The authors would also like to thank the Virginia Transportation Research Council (VTRC) of the Virginia Department of Transportation (VDOT), and specifically Cathy McGhee, for supporting the integration of DDOT systems into the VCC.

Contents

Definitions

Term	Description
Aftermarket Safety Device (ASD)	A connected device in a vehicle that operates while the vehicle is mobile, but that is not connected to the data bus of the vehicle.
Backhaul	The closed network communication links between a traffic management center (or other back offices) and field installations (such as traffic signal controllers, traffic cameras, and other sensors).
Basic Safety Message (BSM)	The core data set transmitted by the connected vehicle (vehicle size, position, speed, heading acceleration, brake system status) approximately 10 times per second. A secondary set is available depending upon events (e.g., ABS activated) and contains a variable set of data elements drawn from many optional data elements (availability by vehicle model varies). This would be transmitted less frequently. The BSM is tailored for low latency, localized broadcast required by V2V safety applications but can be used with many other types of applications.
Connected Device	Any device used to transmit to or receive messages from another device. A connected device can be sub-categorized as an OBE, ASD, vehicle awareness device, or RSE. In many cases, the connected device will be a DSRC device, but other types of communications can and are expected to be supported.
Connected Vehicle	A vehicle containing an OBU or ASD. Note that vehicles may alternatively include a vehicle awareness device, which transmits the BSM but does not receive broadcasts from other devices and cannot directly support vehicle-based applications.
Connected Vehicle Reference Implementation Architecture (CVRIA)	A set of system architecture views that describe the functions, physical and logical interfaces, enterprise and institutional relationships, and communications protocol dependencies within the connected vehicle environment. The CVRIA defines functionality and information exchanges needed to provide connected vehicle applications.
Dedicated Short-Range Communications (DSRC)	DSRC is a technology for the transmission of information between multiple vehicles (V2V) and between vehicles and the transportation infrastructure (V2I) using wireless technologies.
Intelligent Transportation Systems (ITS)	Systems that apply data processing and data communications to surface transportation to increase safety and efficiency. ITS will often integrate components and users from many domains, both public and private.

Term	Description
Interoperability	The ability of two or more systems or components to exchange information and to use the information that has been exchanged.
Latency	A measure of time delay experienced in a system, the precise definition of which depends on the system and the time being measured. For a data element in this context, latency is the time difference between the time that data value is acquired by the source and the time the message is transmitted.
National Transportation Communications for Intelligent Transportation System Protocol (NTCIP)	The NTCIP is a family of standards designed to achieve interoperability and interchangeability between computers and electronic traffic control equipment from different manufacturers.
Onboard Equipment (OBE)	This term refers to the complement of equipment located in the vehicle for the purpose of supporting the vehicle side of the applications. It is likely to include the DSRC radios, other radio equipment, message processing, driver interface, and other applications to support the use cases described herein. It is also referred to as the vehicle ITS station. When referring to the DSRC radio alone, the correct term is OBU (see below).
Onboard Unit (OBU)	A vehicle-mounted device used to transmit and receive a variety of message traffic to and from other connected devices (other OBUs and RSUs). Among the message types and applications supported by this device are vehicle safety messages used to exchange information on each vehicle's dynamic movements for coordination and safety.
Original Equipment Manufacturer (OEM)	An OEM refers to the entity that originally manufactures an item that may be branded and sold by others. In the connected vehicle environment, it is commonly used to refer to automobile manufacturers.
Roadside Equipment (RSE)	Term used to describe the complement of equipment to be located at the roadside; the RSE will prepare and transmit messages to the vehicles and receive messages from the vehicles for the purpose of supporting V2I applications. This is intended to include the DSRC radio, traffic signal controller where appropriate, interface to the backhaul communications network necessary to support the applications, and support of such functions as data security, encryption, buffering, and message processing. It may also be referred to as the roadside ITS station. When speaking of the DSRC radio alone, the correct term is RSU (see below).
Roadside Unit (RSU)	An RSU is a DSRC radio-equipped device that demarcates component between vehicles and other mobile devices and existing traffic equipment.

Term	Description
	The RSU can be a permanent roadside installation or temporary installation deployed to support operation around an incident, work zones, or special events. At the time of this writing, RSU version 4 is the latest specification, but is still considered to be a pre-production stage of specification. Equipment that is version 4 compliant can be upgraded to specification version 4+ and version 5 via firmware upgrade.
Signal Phase and Timing (SPaT)	SPaT is a message type that describes the current state of a signal system and its phases, and relates this to the specific lanes (and therefore to maneuvers and approaches) in the intersection.
Vehicle-to-Vehicle (V2V) Communications	A system designed to transmit basic safety information between vehicles to facilitate warnings to drivers concerning impending crashes.
Vehicle-to-Infrastructure (V2I) Communications	A system designed to transmit information between vehicles and the road infrastructure to enable a variety of safety, mobility, and environmental applications.
Vehicle-to-Infrastructure (V2I) Reference Implementation	An interface system that supports the collection, integration, and dissemination of data between infrastructure and vehicles to enable integrated, interoperable V2I safety, mobility, and environmental applications.

BACKGROUND

As outlined in the most recent Action Agenda (District Department of Transportation, 2010), the District Department of Transportation's (DDOT's) five core values and functions are safe passages, sustainability, maintenance and investment in capital assets, identifying prosperous places, and investing in the DDOT workforce. The development and deployment of connected vehicle infrastructure (CVI) in the District will help the DDOT serve these values and functions by leveraging technology to provide a safer and more sustainable transportation system while improving the efficiency of identifying and prioritizing maintenance of the District's capital assets.

Washington, D.C., has consistently been ranked among the most congested areas in the United States, with recurring congestion, work zones, and incidents leading to unreliable travel times and safety challenges. The ability to identify, collect, process, exchange, and transmit real-time data provides drivers with an opportunity for greater situational awareness of the events, potential threats, and imminent hazards within the vehicle environment. When combined with technologies that intuitively and clearly present alerts, advice, and warnings, drivers can make better and safer decisions. In fact, United States Department of Transportation (U.S. DOT, 2010) research suggests that connected vehicle (CV) technologies can address up to 81% of unimpaired crashes.

In addition, CV data may be used by DDOT to improve operations and efforts to maintain capital assets. Crowed-sourced data available through vehicle-to-infrastructure (V2I) communications provide a new, actionable resource. At moderate levels of deployment, DDOT could obtain the data needed to operate advanced algorithms to monitor a variety of V2I components (e.g., roadway surface quality and bridge stability) through large-data analytics utilizing vehicle sensors such as accelerometers and gyroscopes. Ultimately, a fully connected transportation infrastructure will create an environment for numerous advanced applications that may help address all aspects of DDOT's core values.

VIRGINIA CONNECTED CORRIDORS

More than 15 years ago, the Virginia Department of Transportation (VDOT) and the Virginia Tech Transportation Institute (VTTI) partnered to create one of the first active and integrated roadways featuring intelligent transportation systems: the Virginia Smart Road. Today, VTTI has teamed once again with VDOT and its research arm, the Virginia Transportation Research Council (VTRC), to enhance and complement the development and deployment of the next generation of vehicular technology with the Virginia Connected Corridors (VCC).

The VCC is an initiative that will help fulfill the ultimate goals of integrating connectivity within the transportation system: to improve mobility, enhance sustainability, and save lives. The mission of the VCC is to provide an open CV environment where concepts can be developed, tested, deployed, and evaluated in real-world operating environments.

The VCC comprises more than 60 roadside units (RSUs), which are connected to a low-latency backhaul network via Dedicated Short-Range Communications (DSRC) and cellular technology. The RSUs are positioned along two corridors: (1) the Virginia Smart Road in Blacksburg, Virginia, and (2) select freeway and arterial roadway sections in Northern Virginia that make up one of the nation's most congested corridors. Linking DDOT's new CV initiatives into the VCC will instantly provide a data, computation, and communications hub that supports a focus on application development.

INTRODUCTION

This project, Development of Prototype Onboard Units for Connected Vehicle Initiatives, focused on creating a platform for the development and evaluation of CV applications that can enhance safety and mobility, reduce environmental impacts, and improve operations within DDOT's operational region. The platform is based on the existing VCC deployment, and with a spirit of collaboration, the Virginia Department of Transportation (VDOT) has agreed to allow the use of relevant portions of their backend system during this project. This platform will position DDOT on the cutting-edge of Intelligent Transportation Systems (ITS) research in a short period of time, with efforts targeted specifically at local transportation challenges.

The project and this report were divided into five tasks:

- Task A – Evaluate DDOT Infrastructure
- Task B – Develop Onboard Equipment (OBE)
- Task C – Install Onboard Units (OBUs)
- Task D – Provide Monitoring Software
- Task E – Support Data Analytics

During Task A, VTTI conducted a full inspection of DDOT's existing infrastructure to determine the interoperability of current ITS infrastructure with the OBE equipment. The inspection included reviewing computer-aided design (CAD)/computer-aided manufacturing (CAM) files and geographic information system (GIS) data, conducting field visits, and coordinating with the Transportation Management Center (TMC), the ITS Systems Integration and Development Branch and the ITS Systems Support Branch. VTTI then drafted a report, included below in the Task A section, that assessed the ability of DDOT to deploy the following V2I functionalities: intersection messages, end-of-queue warnings, work zone warnings, and curve warning systems. The report also discusses the integration of V2I functionalities in multimodal systems, including transit systems (possibly for the integration of transit signal priority).

During Task B, VTTI developed an onboard unit (OBU) package for installation in five DDOT work trucks to support V2I communications. This task required some product research to determine the best sensors and components to fit DDOT's V2I needs. Due to the lack of a reliable pothole-detection system (as determined through a literature review; see Appendix A), VTTI developed a pothole detection application as part of this OBU package using information and recommendations for algorithm development gleaned from the literature review.

VTTI then worked with DDOT representatives to determine the optimal installation plan, focusing on hardware mounting locations. Once the plan was finalized, the OBU packages were installed in the five DDOT trucks to complete Task C.

Task D included the development and/or integration of multiple VTTI software packages with the goal of improving DDOT's operating efficiency. Two software applications previously created by VTTI, VCC

Monitor and VCC Mobile, were adapted by integrating the data collected by the OBE installed in DDOT's trucks (i.e., temperature and pothole data). Furthermore, an Android smartphone application was developed from the ground up to collect road surface data, detect potholes, and send that information to VCC Monitor for easy identification and tracking.

The goal of Task E was to support DDOT's data analytics goals using the real-time data transmitted by the equipped DDOT vehicles. As a Task E deliverable, VTTI provided DDOT with the necessary information to access and interact with the data in the VCC Cloud.

TASK A – EVALUATE DDOT INFRASTRUCTURE

This section describes the system requirements for a DDOT CVI deployment based on the current state of DDOT's ITS infrastructure and is the final deliverable for Task A of the Development of Prototype On Board Unit for Connected Vehicle Initiatives project. The section is subdivided into six sections: (1) DDOT's Current ITS State, (2) DDOT's ITS Master Plan, (3) CVI Deployment for DDOT, (4) CVI Architecture for DDOT, (5) CVI Risk Identification, and (6) CVI Security. The CVI deployment is based on current or future deployments, which are described in the U.S. DOT's *National Connected Vehicle Field Infrastructure Footprint Analysis* (Wright et al., 2014) and documents developed under the U.S. DOT's Connected Vehicle Pilot (CVP) Deployment Program.

The focus here is on V2I technologies that support CVs, such as roadside equipment (RSE), backhaul, and facilities, and items that support CVI operations. This report does not address vehicle-to-vehicle (V2V) communication devices such as OBUs and Aftermarket Safety Devices (ASDs) because there is a need to clearly define the CV applications that meet the needs of DDOT. However, a few CV applications are recommended (see Table 12) based on the U.S. DOT's Connected Vehicle Reference Implementation Architecture (CVRIA).

CURRENT STATUS OF DDOT ITS INFRASTRUCTURE

Operation and Traffic Management Centers

DDOT's TMC is located at 14th and U Street NW in the Reeves Center Municipal Office Building in Washington, D.C. The center serves as DDOT's hub for communications and software. In addition to the TMC, DDOT has three operation centers: (1) the Homeland Security and Emergency Management Agency (HSEMA); (2) the Special Operations Center (SOC); and (3) the Emergency Management Agency (EMA) and Signal System Rooms. Additional details regarding the TMC and the operations centers are listed in Table 1.

Table 1: DDOT TMC and Operation Centers

Center	Name	Location	Description
Traffic Management Center (TMC)	Reeves Center Municipal Office Building	14th and U Street NW	TMC includes the following facilities: • Video wall • Two server rooms • Snow Management Center The Snow Management Center includes: • Small video wall (4 monitors) • Ten workstations that have connections to QuicNet Server • Creston touch screen video management console • Central Ethernet network switch • Two racks of analog video switches and digital video conversion equipment
Operation Centers	Primary center: Located at HSEMA SOC EMA and Signal System Rooms	2720 Martin Luther King Jr. Ave SE 3rd Street and Indiana Ave NW 55 M Street	- Operation center in HSEMA location is staffed by representatives from the following agencies: • DDOT • Metropolitan Police Department • Schools • HSEMA - HSEMA is the primary location for TMC operations. - All communications between Special Operations and EMA centers are via T-1 links furnished by DC NET. - Communications between HSEMA, Reeves Center, and Signal System enterprise services are through a 100-MB bandwidth fiber connection.

Traffic Management Software

DDOT's Traffic Management Software, called CapTOP, was developed in-house and has been updated over time. Recently, The CapTOP system has undergone a redesign, carried out by TransCore, with new functionalities. Table 2 provides a summary of the software.

Table 2: DDOT Software

	Name	Description
Advanced Traffic Management System Software	CapTOP	CapTOP was developed in-house and recently redesigned by TransCore. CapTOP is a GIS-based platform that provides centralized traffic management including: • Update dynamic message signs • Identify incidents and notify Roadway Operations Patrol (ROP), Metropolitan Police Department (MPD), etc. • View closed circuit televisions (CCTVs) from 160+ locations • See the status of the District traffic signals • View real-time traffic conditions based on sensor and Global Positioning System (GPS) data • Quickly identify which devices need maintenance

Signal System

DDOT operates 1,611 signalized intersections (8-bit, 170 Traffic Signal Controller) within the District. In addition, 40 signal modifications are under design, and another 190 are under construction. The majority of signals operate on the McCain QuicNet system and follow the RS-232 standards (CAMI Research, 2015) for transferring information. By 2013 DDOT facilitated 400 locations with Internet Protocol communication that use Digital Subscriber Line (DSL) technology (District Department of Transportation, 2013). Table 3 summarizes the status of the current signal systems within the District.

Table 3: DDOT Signal Systems

	Item	Description
Signals/Signal System and Optimization	1,611 signalized intersections 40 signal modifications under design 190 signal modifications under construction 154 school flashers 11 detection cabinets and 208 communications termination cabinets Battery backup and uninterrupted power supplies (UPS) at 123 critical intersections	DDOT uses the McCain QuicNet system platform to computerize the signal system. The enhancement to QuicNet Pro with the conversion from RS-232 standard to IP-based communication is an ongoing process. All controllers are in process of being upgraded to support standard National Electrical Manufacturers Association (NEMA) phasing.

Reversible Lanes

DDOT operates a number of reversible lanes (with the total length of approximately 10.6 miles) that facilitate more-efficient traffic management. These lanes are controlled with manual moveable barriers, static signs, and dynamic displays. The latter is controlled from a traffic signal controller. For future enhancements, DDOT is seeking real-time control of the reversible lanes. The locations include

Theodore Roosevelt Bridge, Connecticut Avenue, 16th Street NW, Chain Bridge, Canal Road, 17th Street NW, and Waterside Drive (Dey et al., 2011). Table 4 summarizes three locations with the reversible lanes and their method of operations.

Table 4: Examples of DDOT Reversible Lanes

Locations	**Method of operation**
Roosevelt Bridge	Manual moveable barriers
Center lane in Connecticut Ave	Static signs
Clara Barton Parkway/Canal Road	Dynamic displays

Detection Locations and Permanent Count Stations

Currently, DDOT has 242 vehicle detectors within the District. Most of the detectors (59%) are Sensys wireless detectors that can collect traffic volume, speed, and occupancy data. There is a DSL modem in the cabinet, which transmits the detectors' data via a twisted pair copper network to the TMC. In addition to the detectors, DDOT purchases data from INRIX to predict travel times on I-395 and I-295. Table 5 provides a summary of the vehicle detectors within the District.

Table 6 characterizes the Dynamic Message Signs (DMSs) and Portable Dynamic Message Signs (PDMSs) operating within the District.

Table 5: Vehicle Detectors

Item	**Model**	**Count**	**Description**
SpeedInfo Detectors	Radar	47	Detectors are connected to a DSL modem at the cabinet. Data are then transmitted via DSL over twisted pair cable back to TMC. Some of the detectors are currently integrated with CapTOP traffic management software.
Traffic.com	Remote Traffic Microwave Sensor (RTMS) side fire RADAR sensors	15	
Sensys Detectors (collect traffic volume, speed, and occupancy data)	Puck	142	
Over-Height Detectors	Infrared	4	
Permanent Count Stations	Loop, Acoustic, Microwave, Infrared, Video	34	

Table 6: DMSs

	Item	Count	Communication
DMS & PDMS	PDMS	43	These portable signs are controlled through the use of commercially purchased field-to-central wireless links.
	Installed DMS (permanent)	13	No communication to these signs over the network
	Under design DMSs	11	Four of them will communicate through the combination of last-mile 900-MHz wireless and twisted pair, and the remaining through cellular communication.

DDOT operates 136 CCTVs, with 15 cameras having been upgraded from analog CCTV cameras to IP CCTV. In addition, there are several law enforcement cameras (Table 7). The analog CCTV cameras use MPEG-2 encoders to encode analog videos. The CCTV/IP cameras use MPEG-4 as an encoder, and then transmit the information via DSL over the twisted pair copper network. Afterward, TMC videos are decoded and shared with the TrafficLand.com servers for public use. The re-encoded videos can be shared between DDOT centers over the DC network (i.e., DC NET).

Table 7: CCTV Camera and Video Distribution System

	Item	Count
CCTV Camera and Video Distribution System	Metropolitan Police Department (MPD) CCTV	77
	Red light cameras	50
	Speed enforcement cameras	19
	Analog CCTVs	121
	CCTV/IP	15

Communication Infrastructure

The backbone of the DDOT communication infrastructure is a twisted pair copper network. There are 271 miles of copper cable installed in underground duct banks, dedicated conduits, and aerial structure. The twisted pair network continues to address the communication needs for DDOT's major ITS infrastructure (i.e., the computerized traffic signal system, CCTV surveillance system, and the vehicle detection system). In addition, the center-to-center communication between DDOT's TMC at Reeves Center, HSEMA, and signal system enterprise services are facilitated through the 100-MB bandwidth fiber connection supplied by the DC NET broadband network. Communication between SOC, EMA, and Signal System Rooms is addressed through T-1 links supplied by DC NET. Table 8 summarizes the current DDOT communication infrastructure.

Table 8: DDOT Communication Infrastructure

	Item	Quantity/coverage	Description
Communication	Twisted pair communication network (copper cable)	271 miles	Most of this cable is greater than 25 years old and passes through ducts that in some cases are more than 100 years old.
	"Last-mile" wireless communication	At 11 locations: 1) 4 of the 11 under-construction DMSs 2) 4 of the 34 count stations 3) DDOT uses 900-Mhz radios for 3 signalized intersections.	To provide last-mile connectivity to the twisted pair copper network, DDOT currently uses unlicensed, spread spectrum, 900-Mhz Ethernet radios. The technology is appropriate for short-haul links with lower bandwidth requirements.
	Commercially purchased field-to-central links	At 13 locations: 1) 6 existing Remote Weather Information Stations (RWISs) 2) 7 of the 11 under-construction DMSs	Commercial wireless connections, including 3G digital cellular modems
	Center-to-center broadband network	- 100-MB bandwidth fiber connection is provided between the TMC at the Reeves Center, signal system enterprise services, and HSEMA. - Communication between SOC and EMA and Signal System Rooms are addressed through T-1 links supplied by DC NET.	Center-to-center, high-bandwidth connections are made on a broadband network owned and administered by the District through DC NET.

Other ITS Infrastructure

In addition to the previously stated ITS infrastructure, parking meters, ROP, and ITS infrastructure for the Transit and Non-Motorized Modes were also listed in the DDOT ITS Master Plan (District Department of Transportation, 2013). Currently, 10 employees work for the ROP, and the team responds to traffic incidents and emergencies on District roadways. The District has 17,000 parking meters, all of which are controlled by networked single-space or multi-space meters.

Currently, DDOT has 10 RSUs installed on 7th St NW and Independence Ave SW. The District operates bicycle signals at the intersection of 16th Street and New Hampshire Avenue, the intersection of Pennsylvania Avenue and 15th Street, as well as implementations being installed throughout the District. DDOT implemented transit signal priority in six corridors (16th Street, Georgia Avenue, H Street/Benning Road, Wisconsin Avenue, Theodore Roosevelt Bridge, and 14th Street) for 93 intersections (over 200 signals) and 125 buses as part of the Transportation Investment Generating Economic Recovery (TIGER) Bus Priority Corridor Enhancement project. In addition, DDOT constructed the H Street/Benning Road streetcar line that is a fixed guideway transit line. With regard to information sharing, DDOT plays an active role in regional transportation operation. The "goDCgo" program and the Metropolitan Area Transportation Operations Coordination (MATOC) programs are examples of DDOT's role in information sharing. Furthermore, through the Transportation Planning Board (TPB), regional freeway monitoring program, and regional highway monitoring program, the District's corridors are being monitored using aerial images and floating cars.

Another important role of DDOT is to maintain safety and security within the region. To address this goal, infrastructure such as mobile CCTV cameras and RWISs have been utilized. Table 9 provides a complete list of the safety- and security-related infrastructure. As a summary, the list of DDOT ITS devices and their current and planned quantities are indicated in Table 10.

Table 9: Safety- and Security-related Infrastructure

	Item	**Quantity/Description**
Safety and Security	Street Lighting	68,000 street lights
	Mobile CCTV Cameras	12 units for special events
	Highway Advisory Radio (HAR)	10 HAR locations
	RWISs	6 RWIS locations
	Weigh in Motion (WIM) Stations	2 WIM stations
	Capital Wireless Information Net (CAPWIN)	CAPWIN provides data communication center and Web services to support public safety
	Tunnel Asset Management	85 incident management cameras
	Pop-up barriers	Used by Capitol Police on Capitol Hill

Table 10: Summary of DDOT ITS Devices

Device	**Existing**
Signalized Intersections	1,611
Traffic Controllers	1,440
CCTV	160+
Video Detection Locations	15
Pedestrian Video Detection	0
DMS	24
PDMS	43
Vehicle Detection	180
RWIS	6
HAR	10
RSU	10
Parking Occupancy Monitoring Devices	500
Battery Back-up (UPS)	123

DDOT Master Plan Suggestions for ITS Improvement

For DDOT to achieve its goals of implementing a CVI, which requires increased bandwidth, flexibility, scalability, and reliability, the DDOT ITS Master Plan provides a few suggestions for improvement of the ITS network (District Department of Transportation, 2013). First, it is recommended to divide the system equipment and operations into separate sections. Since there is a need to relocate the Reeves Center, the new communication hub should be located along the main routes close to the Reeves Center. The new facility should have adequate security, backup power, and heating, ventilation, and air conditioning

(HVAC). The data center and the TMC could be located in a number of District facilities. However, it would be ideal for the data center and TMC to be colocated in the same district facility, preferably one that is equipped with a broadband service provider link (e.g., bandwidth fiber link) to the future communications hub. If the TMC and data center are not colocated, the communication between them should be maintained.

The DDOT ITS Master Plan indicated recommendations to improve ITS communication infrastructure (District Department of Transportation, 2013). Based on the DDOT ITS Master Plan, the total amount of bandwidth for DDOT as of December 2013 is 14,621.44 Mbps (slightly more than 80% is dedicated to center-to-center applications and slightly less than 20% is committed to the traffic monitoring subsystems dominated by the video applications). The twisted pair copper network can provide high-speed communication only for short distances with a high density of repeating electronics, and the bandwidth is relatively low. Therefore, the twisted pair copper network must be replaced with an optical fiber network.

Table 11 summarizes the suggested transformation of the DDOT communications network according to the DDOT ITS Master Plan (District Department of Transportation, 2013).

Table 11: Transformation of the DDOT Communications Network

	Current Status	**Suggestions according to the DDOT ITS Master Plan**	**Comment**
Communication	Twisted pair copper network	Optic-fiber-based network	The majority (over 90%) of proposed fiber trunks can be installed within existing ducts or conduit.
	Copper pair trunk cables	Fiber optic trunk cables	
	Star topology relies on Frequency-Shift Keying (FSK) over copper twisted pair from the Reeves Center	Hybrid network with a Ring/Mesh topology	
	FSK communication infrastructure	IP/Ethernet-based network	

The DDOT ITS Master Plan recommends that DDOT implement unlicensed wireless technologies for last-mile communication between the fiber-optic and ITS devices. Based on the availability of the technology, 2.4- or 5.8-GHz point-to-point connections were suggested since they offer the least costly solution. Currently, DDOT employs leased 4G and 3G cellular services throughout the ITS network. For future developments, the DDOT ITS Master Plan suggests the use of leased services technology only for temporary demands (District Department of Transportation, 2013).

Transformation of the DDOT ITS infrastructure shall be introduced in three phases, as detailed below.

- Phase A
    - Full transition of field-to-central communications to IP/Ethernet protocol.
    - Full transition from the legacy analog video platform to digital video.
    - Deployment of TCP/IP/Ethernet protocols for all ITS devices.

- Phase B
    - Construction of a communications hub, data center, and TMC.
    - Rerouting of existing copper communications infrastructure to the new location of the communications hub.
    - Installation and integration of equipment within each of the communications hubs, data centers, and TMC.
- Phase C
    - Construction of fiber optic field network to replace the twisted pair copper network.

It should be noted that since the development of the ITS Master Plan, DDOT has shifted towards a philosophy that emphasizes a focus on better operational utilization of the existing ITS assets and data.

CVI Deployment for DDOT

The requirements for CVI deployment are based on the desired CV applications that DDOT is seeking to implement. The U.S. DOT has defined over 60 CV applications based on specific goals and needs that include V2I Safety, V2V Safety, Mobility, Environment, Road Weather, Smart Roadside, and/or Agency Data. Since the average agency has limited resources, the primary emphasis has been on V2I Safety applications.

For instance, the U.S. DOT's CVP New York City Department of Transportation (CVP - NYCDOT) project aims to help the city move toward reaching the Vision Zero goals (i.e., zero injuries and fatalities due to crashes) (Talas et al., 2016). DDOT should also consider that deployments of multi-state CV environments will need collaboration and integration with current regional transportation information systems and operations. DDOT may desire to implement some of the CV applications listed in Table 12.

Table 12: CV Applications

Group	Applications	Group	Applications
V2I Safety	Red Light Violation Warning Curve Speed Warning Stop Sign Gap Assist Spot Weather Impact Warning Reduced Speed/Work Zone Warning Pedestrian in Signalized Crosswalk Warning (Transit)	Environment	Eco-Approach and Departure at Signalized Intersections Eco-Traffic Signal Timing Eco-Traffic Signal Priority Connected Eco-Driving Wireless Inductive/Resonance Charging Eco-Lanes Management Eco-Speed Harmonization Eco-Cooperative Adaptive Cruise Control Eco-Traveler Information Eco-Ramp Metering Low Emissions Zone Management AFV Charging/Fueling Information Eco-Smart Parking Dynamic Eco-Routing (light vehicle, transit, freight) Eco-ICM Decision Support System
V2V Safety	Emergency Electronic Brake Lights (EEBL) Forward Collision Warning (FCW) Intersection Movement Assist (IMA) Left Turn Assist (LTA) Blind Spot/Lane Change Warning (BSW/LCW) Do Not Pass Warning (DNPW) Vehicle Turning Right in Front of Bus Warning (Transit)		
Mobility	Advanced Traveler Information System Intelligent Traffic Signal System (I-SIG) Signal Priority (transit, freight) Mobile Accessible Pedestrian Signal System (PED-SIG) Emergency Vehicle Preemption (PREEMPT) Dynamic Speed Harmonization (SPD-HARM) Queue Warning (Q-WARN) Cooperative Adaptive Cruise Control (CACC) Incident Scene Pre-Arrival Staging Guidance for Emergency Responders (RESP-STG) Incident Scene Work Zone Alerts for Drivers and Workers (INC-ZONE) Emergency Communications and Evacuation (EVAC) Connection Protection (T-CONNECT) Dynamic Transit Operations (T-DISP) Dynamic Ridesharing (D-RIDE) Freight-Specific Dynamic Travel Planning and Performance Drayage Optimization	Road Weather	Motorist Advisories and Warnings (MAW) Enhanced MDSS Vehicle Data Translator (VDT) Weather Response Traffic Information (WxTINFO)
		Smart RoadSide	Wireless Inspection Smart Truck Parking
		Agency Data	Probe-based Pavement Maintenance Probe-enabled Traffic Monitoring Vehicle Classification-based Traffic Studies CV-enabled Turning Movement & Intersection Analysis CV-enabled Origin-Destination Studies Work Zone Traveler Information

Several elements are required for the CVI deployment, which include (a) RSE (e.g., DSRC RSUs); (b) traffic signal controller interfaces; (c) systems and processes to ensure a secure system; (d) mapping services to provide roadway geometries, signage, and asset locations; (e) positioning services; and (f) data servers for collecting and processing data. According to the “National Connected Vehicle Field Infrastructure Footprint Analysis” (Wright, et al., 2014), for a successful deployment, each of the stakeholders should address different components of the system. In this case, DDOT should provide the fundamental infrastructure and data support detailed in the list below.

- RSE (e.g., RSUs)

- Interfaces from V2I RSE to roadside transportation equipment (e.g., traffic signal controllers) and/or local roadside networks
- Supporting roadside infrastructure (e.g., pole and mounting, power)
- Secure and reliable backhaul communication from the roadside to network information services
- Data from RSE:
    - Traveler information and alerts
    - Intersection and roadway geometric data
    - Signal Phase and Timing data (SPaT)
    - Positioning system and time corrections
- Network information services as needed by particular applications

For CVI deployment, DDOT must provide or upgrade the following infrastructure: DSRC/cellular RSUs, backhaul communication deployment (field-to-center/center-to-field communication), signal controller, interface between controller and RSU, backend system operation and management (includes software, personnel, infrastructure and outsourcing), vehicle fleet data collection, and third-party data collection (e.g., INRIX, HERE).

RSU

An RSE consists of an RSU and the software application objects that reside in the RSU, as well as cabinet equipment, including Power over Ethernet (PoE) injectors, network switches, and communications backhaul. The objective of the RSU is to enable communication between the TMC and vehicles, signal controller and other mobile devices by transferring data over DSRC. All RSUs and OBUs within the District should contain two radios. One radio channel should be dedicated to monitoring or transmitting where it can “hear” the Basic Safety Message (BSM) from all vehicles within the range of the DSRC communications. In addition, RSUs will broadcast SPaT and MAP messages. The second radio channel (New York City uses Channel 178) should be used as the control channel to inform approaching vehicles of available services (Galgano et al., 2016a). Furthermore, other channels may be used to support over-the-air software updates, application parameter management, and data collection from the OBU. For the RSU to meet U.S. DOT specifications, the system requirements (examples of which are listed in Table 13) should be addressed by DDOT (U.S. Department of Transportation, 2016). The complete list of requirements is listed in the DSRC Roadside Unit (RSU) Specifications Document, Version 4.1 (U.S. Department of Transportation, 2016).

Table 13: RSU System Requirements

System Requirements	Examples
Power	PoE: The RSU SHALL support inbound power through a single, designated Ethernet port by PoE in compliance with 802.3at.
	Operating Voltage: The RSU SHALL have a nominal operating voltage between 37 and 57 volts DC, compliant with IEEE 802.3at.
Environmental	Ambient Temperature RSU: The roadside unit SHALL function as intended within the temperature range of −34 degrees C (−30 degrees F) to +74 degrees C (+165 degrees F).
	Ambient Temperature Power Injector: Any accompanying Power Injector unit SHALL function as intended within the temperature range of −34 degrees C (−30 degrees F) to +74 degrees C (+165 degrees F).

System Requirements	Examples
Physical	Weight: The weight of the roadside unit, excluding antennas, mounting hardware, and PoE Power Injector, SHALL NOT exceed 15 pounds.
	Enclosure: The roadside unit SHALL be housed in a corrosion-resistant enclosure that is compliant with the NEMA4X (IP66) rating.
Functional	RSU Set: At installation locations that require multiple RSUs to provide the required DSRC coverage, all RSUs SHALL be configured to operate as a single functional unit.
	RSU Set Configuration: All non-Master RSUs in the RSU Set SHALL be automatically configured based on the configuration of the Set "Master" RSU.
Behavioral	The RSU SHALL update all configuration parameters upon receiving an "updateconf" command from an authorized user.
	Antenna Output Power: The RSU transmit output power SHOULD be configurable.
Performance	Mean-Time-Between-Failure (MTBF): The RSU SHALL remain operational for an average of 100,000 hours.
	Availability: The RSU SHALL meet the operational availability requirements of 99.9%. Note: This does not include scheduled maintenance.
Interface	Global Positioning System (GPS): The RSU SHALL include an integrated GPS receiver (for positioning and UTC time).
	Virtual Ethernet Interfaces: The RSU SHALL support multiple, independent IPv4 and IPv6 networks.

According to the DDOT ITS Master Plan, as of December 2013, no DSRC/cellular RSUs were installed within the District. Since then, DDOT has installed 10 RSUs on 7th St NW and Independence Ave SW, and is in the early stages of planning a pilot that would install additional RSUs along approximately 12 miles of roadway. DDOT needs to provide RSUs that employ DSRC or other potential wireless technologies.

Signal Controller

The RSU should interface with the signal controller, backhaul communication network, and DSRC messages from road users. To support CV-related applications, signal controllers within the District need to employ DSRC communication, and DDOT should upgrade their software. Signal controllers must have sufficient processing power and available memory for additional tasks that will generate consequent to CVI deployment.

The signal controller devices used in the District should be facilitated with CV capabilities to exchange DSRC messages with the DDOT CV system. They should be able to receive the SPaT and MAP messages through a dedicated radio channel and use this information to support V2I safety applications. The signal controllers should be monitored and controlled by the TMC, which should communicate with the RSUs and monitor their operational status through the backhaul communication network.

To comply with the CV technology deployment, DDOT's signal controllers should incorporate Ethernet and Internet Protocol that is well-adapted for connection to DSRC-based RSE. The District's controllers lack Ethernet support, require the expense of an environmentally hardened Ethernet adaptor to upgrade the serial communications port to the Ethernet port of the RSE (Hill and Garrett, 2011). The security requirements that need to be met are listed in Galgano et al. (2016a). A list of the signal controller requirements are shown in Table 14 below.

Table 14: Signal Controller Requirements

Advanced Traffic Signal Controller System Requirements	Advanced Traffic Signal Controllers shall use International Atomic Time (TAI) time to issue security credentials.
	Advanced Traffic Signal Controllers shall issue messages to DSRC devices with security credentials that meet this document's IEEE 1609.2 requirements.
	Advanced Traffic Signal Controllers shall maintain an authenticated Network Time Protocol (NTP) based time reference.
	Advanced Traffic Signal Controllers shall export their Coordinated Universal Time (UTC) times (Line Frequency referenced from the traffic signal system) as UTC times referenced from their authenticated NTP based time reference.

DSRC Deployment

Prior to any RSU implementation, DDOT must consider developing a comprehensive field deployment plan for the DSRC system. There are constraints with a DSRC deployment, such as spectrum management considerations, and neglecting them may impose issues on the CVI deployment. Each RSU comes with different area coverage requirements, which depend on terrain and building structures. Therefore, each RSU might need different installation design and antenna selection, while also addressing the requirements of the applications that DDOT desires to implement (Bayless et al., 2015). In addition, there are other constraints, such as power for RSUs and backhaul communication availability, right-of-way, and utility conflict.

Beyond the aforementioned constraints, DDOT must ensure that DSRC operation meets Federal Communication Commission (FCC) regulations and that the operation does not interfere with other primary users like radar and satellite.

DDOT should also coordinate its activities with neighboring or overlapping jurisdictions and other DSRC system licensees operating in the Washington, D.C., metro area. DDOT should initiate informal coordination with neighboring or overlapping jurisdictions and should conduct radio frequency (RF) surveys to assure that there is no interference risk to DSRC services/operations. In addition, RF analysis (e.g., transmission power levels and risks) is a critical task that DDOT should address. Figure 1 shows the steps that DDOT should follow prior to the implementation of the DSRC RSUs (Bayless et al., 2015).

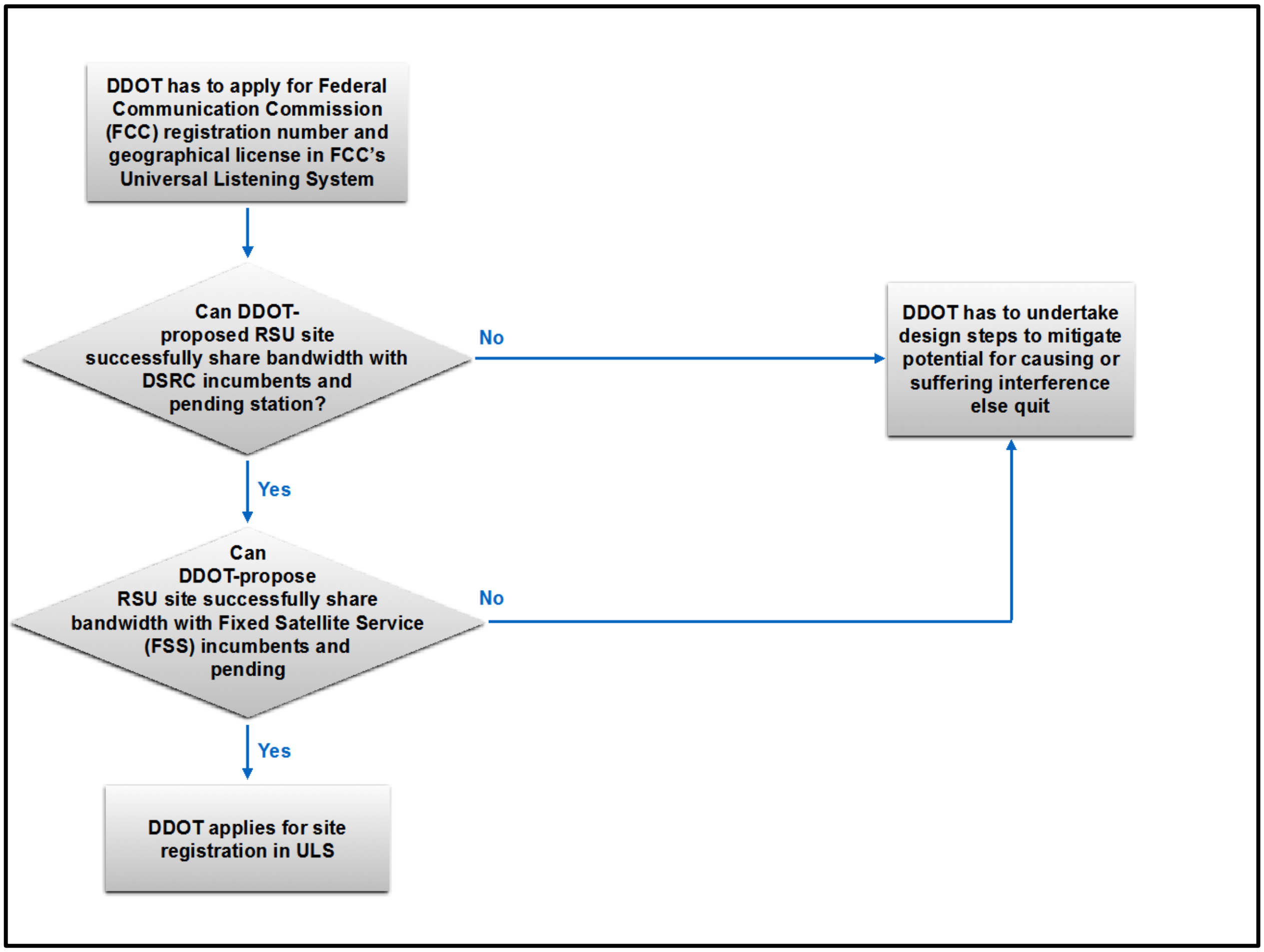


Figure 1: DSRC Deployment Plan

Backhaul Communication

The existing backhaul communication within the District is provided through the twisted pair copper network. According to the DDOT ITS Master Plan, the District desires to replace all of the twisted pair copper communication network with a fiber optic network. DDOT can use most of the existing conduit for this matter. Currently, DDOT employs unlicensed, spread spectrum, 900-Mhz Ethernet radios for last-mile communication. For the CVI deployment, DDOT can deploy DSRC technology to address communication needs with RSE. Although fiber optic network is a viable option for DDOT, there are other backhaul communication technologies that should be considered.

Other Backhaul Communication Technologies

Three states (New York, Wyoming, and Florida) participate in the U.S. DOT CV pilot deployment program. Each one provided a Concept of Operations (ConOps) document for their pilot deployment in April 2016, December 2015, and February 2016, respectively (Galgano et al., 2016b; Gopalakrishna et al., 2015; and Waggoner et al., 2016). Based on their current infrastructure, they have proposed different approaches toward CV backhaul communication networks. Among those, New York and Wyoming offered wireless alternatives, while Florida proposed a combination of cellular and terrestrial backhaul communication networks (Johnson et al., 2016b). Table 15 lists the approaches that each of the aforementioned states has selected for the CV backhaul communication network.

Table 15: Backhaul Communication Network for CV Pilot Deployment Program

Region	Case Study	Backhaul Communication
New York, NY	Three separate areas within Manhattan and Brooklyn	NYCWiN Wireless Communication Link
Tampa, FL	Tampa Hillsborough Expressway Authority (THEA)	LTE or terrestrial backhaul
WY	402 miles of the I-80 corridor	WYOLINK Radio Network, Wi-Fi hotspots, and cellular backhauls

In addition to the backhaul communication indicated in Table 15, the *National Connected Vehicle Field Infrastructure Footprint Analysis* (Wright et al., 2014) also suggested the use of other technologies (Table 16).

Table 16: Potential Backhaul Communication Network

Type	Technology	Description
Wide Area	Cellular (LTE)	Widely available, low cost, high data rates, limitations with IPv6, rural availability varies, data volumes can be limited, requires external antenna
	Cellular (GPRS)	Widely available, low cost, high data rates, limitations with IPv6, rural availability varies, may be being phased out in favor of higher bandwidth, higher priced services, requires external antenna
	WiMAX	Very good performance in terms of range and data rates, costs are determinate, no data volume costs, easily extensible by adding inexpensive client modems, highly reliable, generally limited competition, and limited downstream cost risk, requires external antenna
	Fixed Service Satellite (FSS)	Very flexible in terms of geographic locations, generally high data rates, high initial system cost, may be subject to weather-related performance issues, may require periodic maintenance
	Satellite Digital Audio Radio Service (SDARS)	Low-cost equipment and service, wide geographic capability, relatively low data rate, relatively high latency
Local Area	Ultra-wideband (UWB)	High data rate, low power, limited commercial equipment availability, uncertain regulatory environment, unstable standards environment
	Wi-Fi	May exist in RSU by default, zero added cost, no backhaul wiring installation, only suitable for local manual RSU programming using a localized device. Not a true remote backhaul, requires development of secure system access to prevent unwanted RSU tampering, may require external antenna
	DSRC	Exists in RSU by default, zero added cost, no backhaul wiring installation, only suitable for local manual RSU programming using a localized device. Not a true remote backhaul, requires development of secure system access to prevent unwanted RSU tampering, may require external antenna
	ZigBee	Low cost, low power, few commercially available, only suitable for local manual RSU programming using a localized device. Not a true remote backhaul, requires development of secure system access to prevent unwanted RSU tampering, may require external antenna
	Microwave	High data rate, low operating cost, expensive to install, difficult to splice

Backend System Operation and Management

Software

The software element of the backend system operation and management has five main functions: validation, routing, processing, distribution, and storage. DDOT's traffic management software, CapTOP, must be upgraded to address the requirements of the CVI deployment. Table 17 lists each function and describes its objective (Wright et al., 2014).

Table 17: Software Description

Function	Description
Validate Data	The first objective is to verify that the data come from a valid or trusted source. The data must be checked to ensure that they are correctly signed and, if applicable, encrypted. Once the data have gone through this initial validation, secondary data validity checks can take place.
Route Data	After validation, the software needs to route the data to the appropriate processes and recipients. For instance, a section of the data might transmit to TMC subsystems and some part of it may transmit to an electric vehicle subsystem.
Process Data	This function processes the data packets as necessary, whether they are financial data for parking or basic traffic data that need to be converted into information.
Distribute Data	The information needs to be distributed to the proper users, whether through the media, traditional Advanced Traveler Information Systems (ATIS) devices, or directly to the automobile through the DSRC or cellular network.
Store Data	This function moves the data or information into persistent storage for future analysis, processing, research, or use in real-time applications to supplement the real-time data.

Personnel

For successful day-to-day operation, personnel must address two main responsibilities: system monitoring, and system upgrades and enhancement. DDOT should consider training current personnel and recruiting experts to address the requirements of CV technology. Instructors should include both subject matter experts (SMEs) and training professionals. Similar to other practices (Ahmed et al. 2016; Galgano et al. 2016c; Hamill et al. 2016), DDOT might employ a group of DDOT staff at TMC and SMEs from different departments within the DDOT to provide training for personnel. In addition, the training for the CV pilot could be performed by third-party services.

Infrastructure and Outsourcing

The infrastructure component of the backend system includes computer hardware, physical facilities for hardware, and physical facilities for personnel. Computer hardware includes routers, servers, storage, power supplies, backup power supplies, network security, etc. The physical facility for the hardware refers to the data center building where all computer hardware is located. Traditionally, this physical facility has essentially been a personal or desktop computer. Once CV technology is in operation, significantly more data will be transferred to the TMC and the current facility will not be adequate to host the data. Thus, DDOT will have to consider an adequate room and/or building for the data center.

Physical facilities for personnel refer to facilities such as DDOT's TMC and operation centers, which should address the needs for new backhaul communication technologies (i.e., fiber optic network). As previously mentioned, DDOT's Reeves Center must be relocated. The new building will be the hub for the fiber optic network and will play a main role in the CVI deployment project.

Vehicle Fleet Data Collections

Due to issues associated with DSRC RSUs (e.g., privacy by design), DDOT must have a vehicle fleet to collect data for a variety of purposes (e.g., evaluating ride quality and road smoothness within the District and monitoring responses to snow events). The fleet should be provided with data collection capabilities, storage capacity, and a mechanism to download the stored data into DDOT's data management system. With this option in hand, DDOT can more confidently gather reliable data in addition to external sources such as the vehicle manufacturing community.

Third-Party Data Collection

Third-party providers have employed data collection equipment and mobile-based traffic data sources within the national roads. Purchasing rich traffic data from third-party service providers can help DDOT access traffic data such as speed, volume, and travel time, as well as other information such as road closures and weather information. DDOT can use this data to increase the accuracy and quantity of the data that has been collected by the agency to support CV technology requirements; DDOT currently purchases traffic data from INRIX (District Department of Transportation, 2013).

CVI Architecture for DDOT

The system architecture proposed here follows the U.S. DOT's CVRIA (CVRIA, 2016). In the District, ITS infrastructure includes the TMC, operation centers, field equipment, RSUs, traveler devices, and vehicles. Based on the reviewed documents, utilizing a data center as an additional facility within the TMC should be considered, as well as the inclusion of RSUs in the architecture. Furthermore, DSRC technology should be utilized to supply the communication between RSUs, field equipment, traveler devices, and vehicles. Field-to-center and center-to-center communication (backhaul) should be supported by an IP/Ethernet-based network. DDOT may desire to select wireless alternatives instead of an IP/Ethernet-based network. Figure 2 illustrates the physical view of the DDOT CVI deployment.

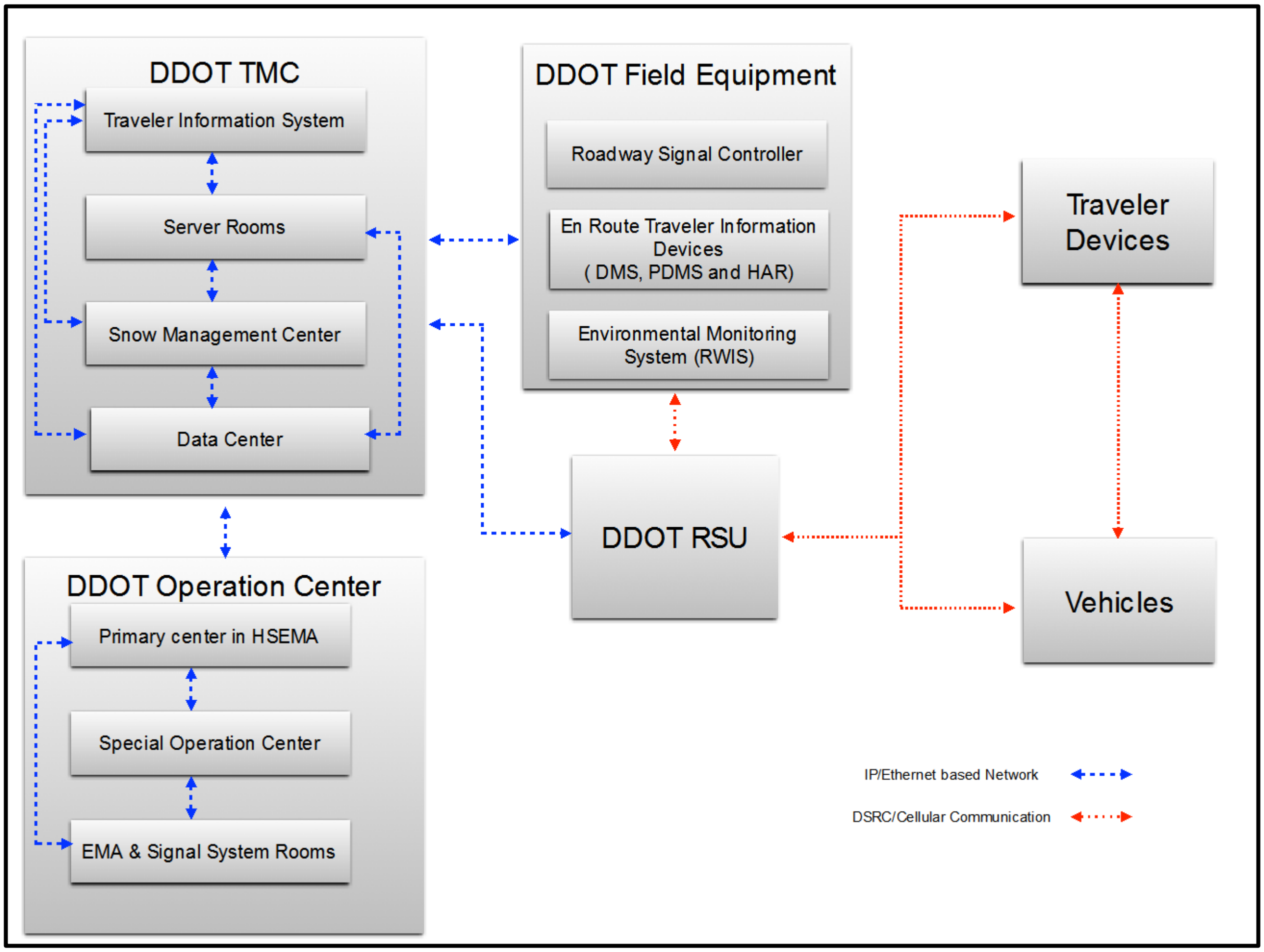


Figure 2: Physical View of the DDOT CVI Deployment

CVI Risk Identification

This section defines the main technical and institutional risks associated with the deployment of the CVI technology. Those categorized as a technical risk are mainly related to the infrastructure aspect of the DDOT CVI deployment. These risks and threats include communication, security, power outage, collective environment, and equipment risk, examples of which are detailed in Table 18 (Gopalakrishna et al., 2016; Johnson et al., 2016a; Talas et al., 2016). Communication is the backbone of the CV technology; a failure in V2I or backhaul communication can impose serious consequences to the DDOT and road users. Similar to communication, a security failure may imply a threat to different parts of the system.

Table 18: CVI Deployment Technical Risk

Risk Element	Risk Group	Examples
Technical Risks	Communication	Outage for satellite, cellular, and backhaul (DSL/Fiber/T-1)
		Wire cut or destroyed
		Router or switch configuration problem or error
		Internet Protocol Security (IPSEC) Virtual Private Network (VPN) failure
		DSRC failures due to jamming, antenna failure, or general congestion
	Security	Denial of Service attack on DSRC causing shutdown
		Man-in-the-Middle (MitM) attack aim for false messaging
		A device that is misbehaving due to hacking
		Someone steals the certificates and pretends to be an ASD, Integrated Safety Device (ISD), or Vehicle Awareness Device (VAD).
		Cyber attack of the TMC, RSU
		Someone hacks into the backhaul and pretends to be RSE
	Power Outage	Causes the Variable Speed Limit (VSL) sign to go down
		Causes malfunctioning of the traffic signal controller
		Causes disconnection for V2I communication
	Collective Environment	Disasters such as tornado can cause a disruption to a whole CV system and cause shutdown
	Equipment Risk	DDOT unable to timely integrate equipment with current standards
		DDOT unable to adequately build applications to documented system requirements
		Bluetooth range and reliability is untested
		DSRC reliability is untested
		DDOT runs into DSRC licensing problems with FCC
		Multiple RSU manufacturers contrast in interpretation of standards, creating interoperability issues
		Maintaining RSUs involves driving to each RSU to update software or to monitor performance

It is also important to identify institutional risks. Some examples of institutional risks associated with the operation and maintenance of the DDOT CV pilot program are listed in Table 19 (Gopalakrishna et al., 2016; Johnson et al., 2016a; Talas et al., 2016).

Table 19: CVI Deployment Institutional Risks

Risk Element	Examples
Institutional Risks	Change in DDOT budget that could lead to a funding shortfall
	The Security Credential Management System (SCMS) is not ready
	Federal priorities and regulatory environment
	New mayor could change the priorities
	Staffing turnover; departure of key management leaders
	Internal conflict within the DDOT
	DDOT winter operations taking priority
	Data collection process issues before and after the deployment can cause delays to the project
	Law enforcement is not a current stakeholder, but could take advantage of the CV system alerts
	Multiple layers of standards are constantly changing

CVI Security

DDOT must ensure adequate security for the physical and cyber CV environment prior to implementation. For the CV technology deployment, DDOT must install DSRC/cellular RSUs within the District and needs to upgrade the current signal controllers (equipped with DSRC/cellular technology). All ITS infrastructure must maintain non-stop, real-time communication with RSUs via DSRC or cellular technology. With backhaul communication (e.g., IP/Ethernet-based network) in place, RSUs communicate with the TMC and, in turn, the TMC communicates with the operation centers. The physical infrastructures within the District (e.g., RSUs) require adequate security and should be monitored as a prevention using detection strategies. More importantly, the communication infrastructure (e.g., DSRC) has to be secure and resilient to prevent adversaries from interrupting the system.

Since any interruption in the system can impose safety, financial, and operational issues for DDOT, it is best for DDOT to consider security measures in advance. Physical infrastructure such as RSUs and signal controllers should be monitored and protected at all times. Communication networks, both the backhaul and DSRC/cellular, must be secure and resilient for a successful implementation of the CV technology. A few examples of the CVI security measures proposed in *Connected Vehicle Pilot Deployment Program Phase 1, Concept of Operations (ConOps)-New York City* (Galgano et al., 2016b) are listed below.

- The information security manager shall investigate and monitor the data traffic usage to detect unapproved use of the IP connection.
- The information security manager shall monitor the DSRC communications performance to detect Denial-of-Service (DoS) attacks.
- All cryptographic software and firmware for the Hardware Security Module (HSM) shall be developed and installed in a form that protects the software and firmware source and executable code from unauthorized disclosure and modification.
- The device shall provide tamper evidence to detect tampering of the device.

- The RSU shall delete old certificates upon location change.
- The RSU shall implement a firewall blocking all IP access from devices to any IP address other than those approved for specific applications.

In addition to the aforementioned security measures, the U.S. DOT DSRC RSU Specifications Version 4.1 document also provides additional security requirements. DDOT must adhere to this minimum set of specifications (U.S. Department of Transportation, 2016). Examples of those requirements are indicated in the Table 20.

Table 20: U.S. DOT Security Specifications

System Requirements	Examples
Security	Physical Security: The roadside unit SHALL be compliant with Federal Information Processing Standard (FIPS) 140-2 Level 2 Physical Security Requirements.
	Authentication: The roadside unit SHALL be protected by a password compliant with either local operator security policies or a policy based on existing standards (e.g., FIPS 140- Level 3 and 4 in Section 4.3.3).
	Data Protection: The roadside unit local file system SHOULD be encrypted.

Conclusion and Recommendations

For a successful CVI deployment, DDOT will need to provide the necessary infrastructure, such as DSRC RSUs, and should upgrade the current District ITS infrastructure to meet the requirements of CVI deployment.

DDOT will require upgrades and/or modifications to signalized or unsignalized intersections. Most notably, DDOT must change the existing traffic signal controllers to incorporate SPaT capabilities, as well as to ensure that backhaul communications are available for field-to-center and center-to-center communications. DDOT must also adopt robust network security measures to improve the performance of the system against potential tampering and disruption. This process should be coordinated with signal maintenance and upgrade programs.

In addition, DDOT must collect detailed roadway geometry data to support the CV safety applications that will be deployed at intersections and curves. Data collected from the vehicle fleet and third-party service providers may be collected for specific mobility applications. By having the data integrated into advanced traffic management systems (ATMS) programs or with data from other services (e.g., RWIS), DDOT can considerably enhance the road-weather management programs.

To ensure that DDOT has adequate ITS infrastructures in place, it is recommended that DDOT replace the current backhaul communication network (twisted pair copper network) with a fiber optic network. However, DDOT should evaluate wireless alternatives as well, as the selection of the backhaul communication technology is highly dependent on DDOT resources and constraints. The RSUs may employ DSRC as a source of communication. However, dependent on the deployment limitations, (e.g., backhaul communication availability), DDOT may consider cellular technologies as well.

Another important step that DDOT should undertake is to relocate the TMC building to the new facility. Within the new building, DDOT should allocate sufficient space for the data center. The data center is one of the main elements of the CVI deployment and must exchange data with the TMC through the center-to-center communication network. DDOT should also upgrade both RSE and traffic management software to address requirements of the CVI deployment by improving the performance of the CapTOP software (currently used by DDOT), or by replacing it altogether to meet CVI deployment requirements.

Finally, prior to any implementation, DDOT must provide security for both physical and cyber environments. Future infrastructure will need to be designed with adequate security in mind from inception. DDOT must have intrusion detection primitives to prevent hacking of the CV system. In addition, DDOT should assign trained personnel to control and monitor cyber-attacks to mitigate risks associated with CV system vulnerabilities.

Task B – Develop Onboard Equipment

During Task B, VTTI developed the OBE package for installation in five DDOT work trucks to support V2I communications. Each of the OBE components is described in detail below. VTTI has extensive experience with the majority of the components and manufacturers listed below, except for the pothole detector and weather sensor system, which were unique to this project. Thus, some product research was required to identify the most suitable equipment for the pothole detection and weather-sensing functions desired by DDOT. This process is also discussed below.

OBU

At the heart of the deployed system is the Cohda Wireless MK5 OBU. The MK5 OBU is a mature product that is ready to be used in large-scale field trials and aftermarket deployments or serve as a reference design for automotive production. It is a small, low-cost module that incorporates dual IEEE 802.11 radios, a powerful processor running vehicle-to-X (V2X) software stacks and applications, a high-accuracy Global Navigation Satellite System (GNSS) positioning system, and V2X security with hardware acceleration and tamper-proof key storage. The MK5 is based upon the automotive-grade RoadLink™ chipset developed by NXP/Cohda and has unmatched radio performance in harsh outdoor, mobile environments, particularly in critical safety use-case scenarios. Each system includes a Mobilemark 3 wire antenna that integrates both DSRC and GPS antennas. The MK5 includes a software development toolkit to expand functional capabilities. The MK5 unit and antenna is used to communicate CV messages (e.g., BSMs) via cellular or DSRC in V2I and V2V scenarios.

Cellular MiFi

A cellular MiFi unit was installed in conjunction with the OBU to provide access to cellular data networks. Cellular connectivity is necessary for the OBE to communicate with the VCC Cloud since DDOT does has a limited number of RSUs to allow DSRC communication. The MiFi also provides a local Wi-Fi capability to communicate with Android phones to support user interface requirements. DDOT procured five Verizon MiFis (4G LTE Global USB Modem U620L) with unlimited data plans and shipped them to VTTI for installation in the five vehicles.

OBDII Interface

The OBE includes a direct Onboard Diagnostics II (OBD II) interface that is connected to the Cohda OBU. This connection allows the transmittance of vehicle speed data from the vehicle's Controller Area Network (CAN) to the OBU.

Power Control Module

A power control module was supplied to manage the control of power to the Android smart phones and OBUs. The module is connected to vehicle power sources, senses ignition state, and provides proper power to each subsystem. When the vehicle is turned on, this power management board supplies 12 volts of power to the equipment.

ANDROID CELL PHONE

A Samsung Galaxy S5 Android smart phone was chosen as the platform to run the VCC Mobile and pothole detection applications. The cell phone's own cellular capability was not enabled in this platform to avoid duplicate data plans, so the phone will act as a Wi-Fi-enabled display device that uses the OBE and cell module as a communications proxy. When the vehicle is turned on, the cell phone is powered on, and the VCC Mobile application will be turned on and automatically started. As the vehicle moves, it transmits mobile BSMs to the VCC Cloud that can be viewed in VCC Monitor.

POTHOLE DETECTOR

VTTI developed a pothole detection system utilizing the smart phone's integrated sensors (i.e., accelerometer and GPS). VTTI evaluated expanding the OBU's capabilities by adding acceleration and orientation sensors, but concluded that the most effective pothole detection method was to make use of the smartphone already utilized in the installation package. The smartphone has relatively high quality acceleration and orientation sensors that operate at sampling rates appropriate for this task and has the necessary computing resources to perform the algorithm processing. In addition, the smartphone application could potentially be deployed in larger numbers by inviting the general public or other user groups to download an application to their own phones.

ROADWATCH WEATHER SENSOR SYSTEM

The team's initial evaluation of weather sensors included a review of what was currently available in an off-the-shelf mobile package, focusing on roadway ice and freezing conditions. Many of these systems perform a variety of weather sensing, processing, and communication. The team felt that the best option for this OBE package would be a small sensor package that mounts on the vehicle's exterior and looks downwards to ascertain the temperature of the roadway's surface.

After surveying the available technologies that met those criteria, the team selected the RoadWatch Weather Sensor System Kit by M.S. Foster & Associates, shown in Figure 3. This system utilizes infrared measuring to accurately measure and display the temperature of the road surface with an accuracy of ±2 degrees Fahrenheit (15–100 degrees Fahrenheit under ambient stable). The system can accurately detect a 1-degree change in road temperature in 0.1 seconds. The system also displays the ambient air temperature with an accuracy of ±2 degrees Fahrenheit (−40 to 131 degrees Fahrenheit while moving greater than 5 mph).

Figure 3: Road Watch SS Road Temperature Measuring System (M.S. Foster)

https://msfoster.com/wp-content/uploads/2014/09/RoadwatchBullet8_10_14.pdf

The team also explored using a system created by WeatherCloud, which includes a license plate sensor unit as well as a unit mounted to the vehicle windshield. This system is capable of detecting raindrops, mist, fog, and ground moisture, but is at least twice as expensive as the M.S. Foster system. However, despite VTTI's repeated attempts to work with WeatherCloud, the company became unavailable to work with at the time and was dropped from consideration.

The weather system is integrated with the OBE, and data collected by the sensor are included in the generated BSMs, which are sent 10 times per second to the VCC Cloud (via DSRC) or once every 10 seconds (via cellular). This data can be viewed in real-time using VCC Monitor.

Task C – Install OBUs

VTTI worked with DDOT representatives during Task C to determine the optimal installation plan, focusing on hardware mounting locations within the trucks. Once the plan was finalized, the OBU packages were installed in the five 2016 Dodge Ram 3500 Series DDOT trucks at the DDOT warehouse.

The OBE equipment was wired according to the wiring diagram shown in Figure 4 below.

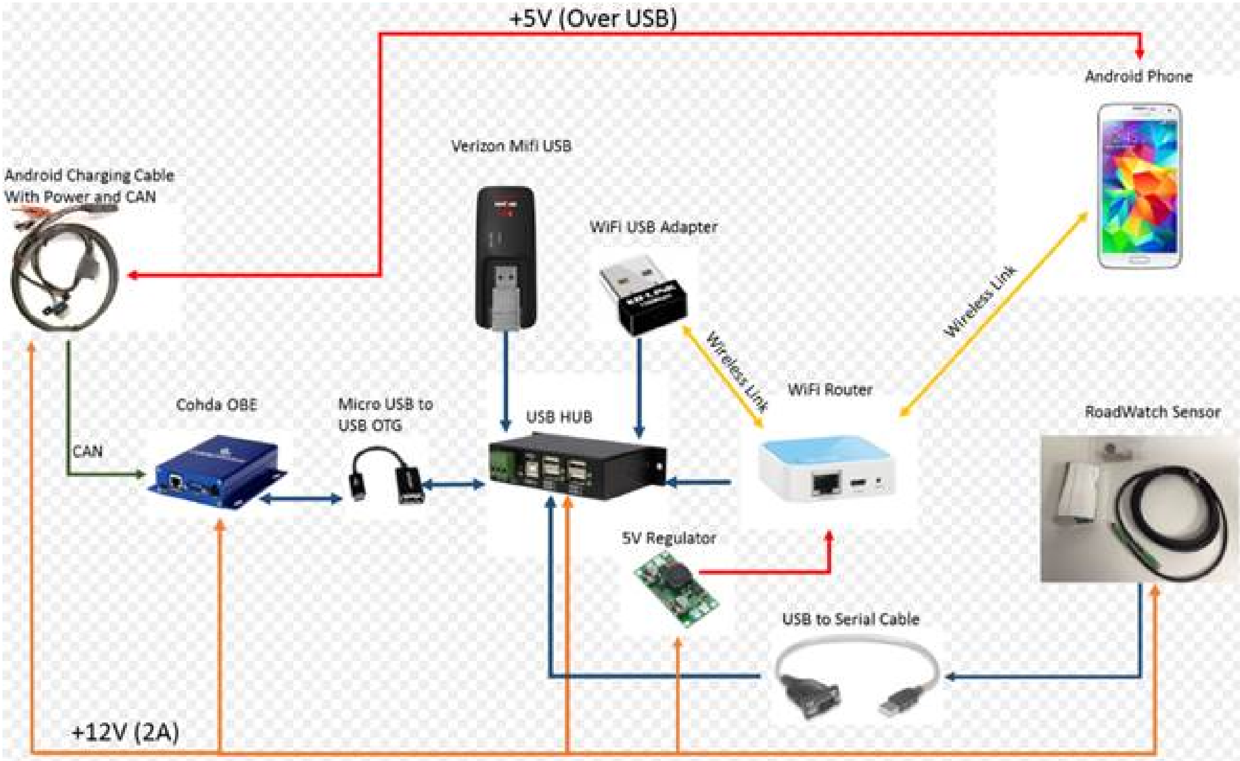


Figure 4: OBE Equipment Wiring Diagram

The majority of the OBE components were installed in a wire cage in the back of the truck cab behind the driver's seat, as shown in Figure 5 below. This cage allowed the equipment and wires to be secured while still allowing ample air flow to and from the equipment.

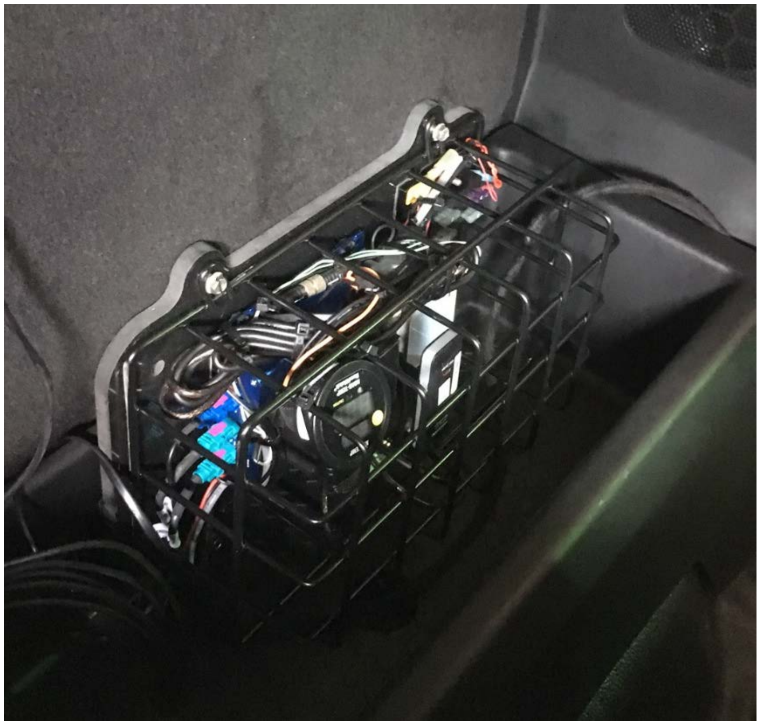


Figure 5: Equipment Cage

The Android phone was also installed on the rear wall of the truck cab behind the seats, as shown in Figure 6 below. This mounting location was chosen to minimize distractions, view obstructions, and to provide a stable mounting location for the phone. While DDOT chose to minimize driver interactions with the smart phone at this time, the phone cord is attached to allow the phone to be relocated to the dash if desired for demonstration purposes or to use as a driver interface in the future.

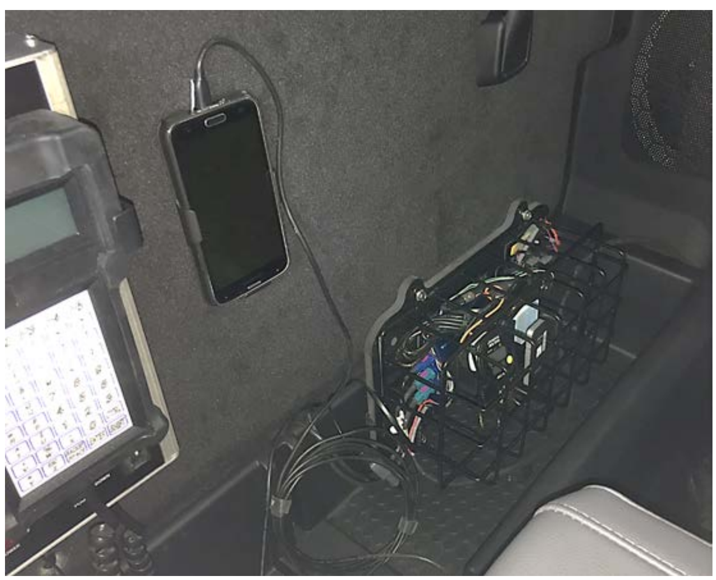


Figure 6: Phone Mounting Location

The RoadWatch Weather Sensor System was mounted to the external surface of the vehicle angled downward to ascertain surface temperature (shown in Figure 7). VTTI integrated this sensor with the OBE through a supported RS-232 interface. VTTI also integrated the weather sensor output with VCC Monitor to provide DDOT with a real-time display of the road surface and ambient temperatures corresponding to the locations of the equipped trucks. The Weather Sensor display was secured within the equipment cage.

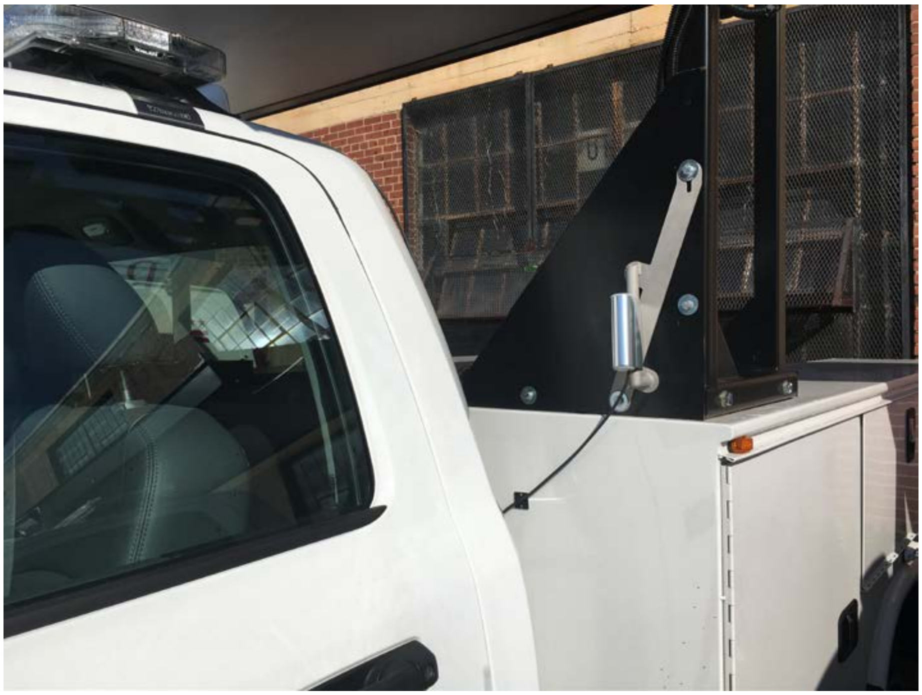


Figure 7: Installed Road Watch Weather Sensor

TASK D – PROVIDE APPLICATION SOFTWARE

Throughout Task D, VTTI focused on the application software needed to support DDOT's V2I goals. Two software applications previously created by VTTI, VCC Monitor and VCC Mobile, were adapted to DDOT's needs by integrating the additional data collected by the OBE installed in DDOT's trucks (i.e., temperature and pothole data). Furthermore, an Android smartphone application was developed to detect potholes and send that information to VCC Monitor for easy identification and tracking.

VCC MONITOR

VCC Monitor is a Web-based client created by VTTI that provides situational awareness for monitoring CV and infrastructure activity and events within the VCC environment. As previously stated, VTTI has integrated additional features within VCC Monitor to assist DDOT with real-time monitoring of their equipped vehicles and desired data (i.e., pothole locations and weather information).

The application provides a map display with detailed overlays including RSU locations and communication status, active traveler information message (TIM) postings, CV and RSE DSRC interaction, vehicle locations, speed and brake status, general traffic speeds (from Google), dynamic message sign locations, pothole cluster locations, and weather information (see Figure 8 below).

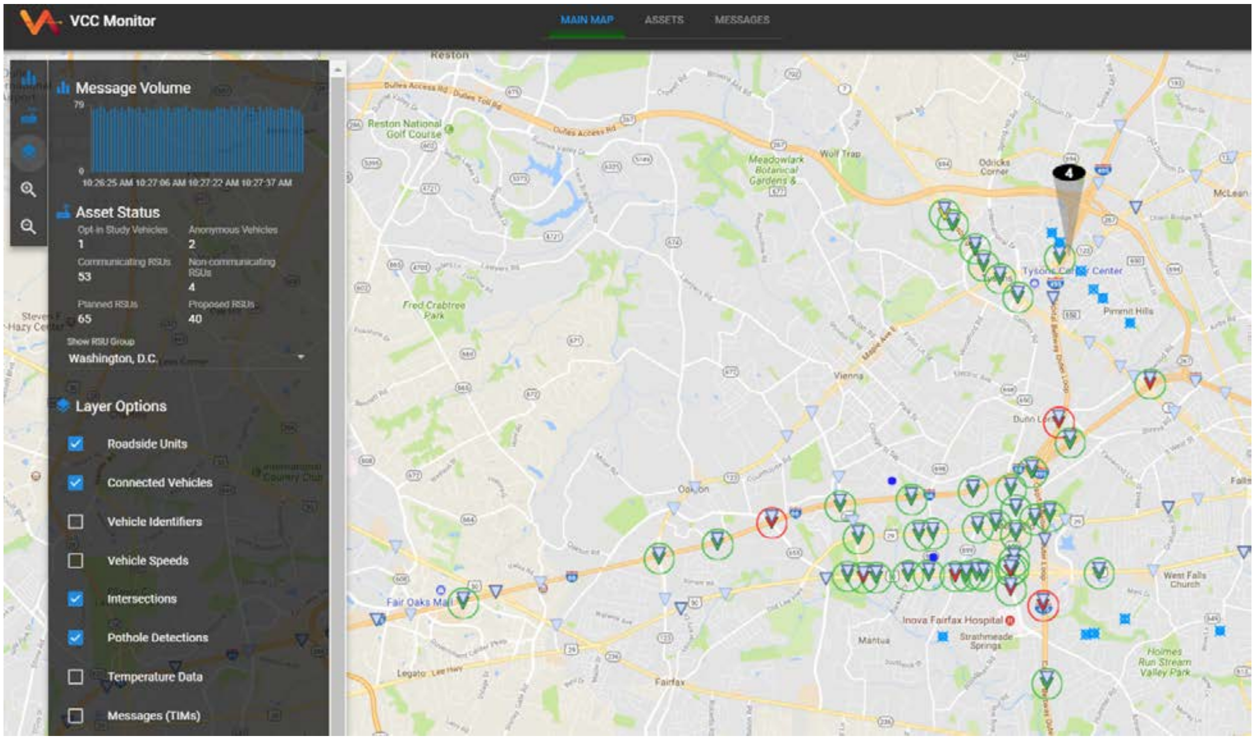


Figure 8: VCC Monitor Detailed Overlays

In addition to the real-time map display, an assets inventory page provides a dashboard of statistics describing activity on the VCC, including RSU heartbeat and uptime status, volume of messages received, and number of active TIMs. A table view can also be selected that shows detailed information for each individual OBU, including the time of last communication, number of messages received in the last 24 hours, and vehicle identification. These table views provide a way to present monitoring information to system operators. These displays were created based on VDOT and VTTI requirements and have been used to monitor the health of both the RSUs and OBUs in the field for over a year.

VTTI updated the preexisting content displayed in the VCC Monitor Web Client to present statistics relevant to DDOT, as discussed in the following sections.

Weather Sensor Integration

VTTI created a Temperature Data overlay on the main map to display the real-time road surface and ambient temperature collected by the Road Watch Weather Sensor System installed on DDOT's trucks. This temperature overlay is shown in Figure 9 below. When the user activates the Temperature Data overlay within VCC Monitor and the equipped trucks have collected temperature data within the last 30 minutes, "bread crumbs" will be displayed within VCC Monitor. "Bread crumbs" are displayed whenever the temperature changes by 1 degree Celsius, when the truck has traveled a quarter of a mile, or every 5 minutes while the truck engine is on. These bread crumbs remain visible on VCC Monitor for 30 minutes. The top number represents the ambient air temperature collected by the sensor and the bottom number represents the road surface temperature. The individual temperature readings are color coded as follows: white shading represents a temperature below 32 degrees Fahrenheit, blue represents a temperature between 32 and 37.4 degrees Fahrenheit, green represents a temperature between 37.4 and 91.4 degrees Fahrenheit, and orange represents a temperature above 91.4 degrees Fahrenheit.

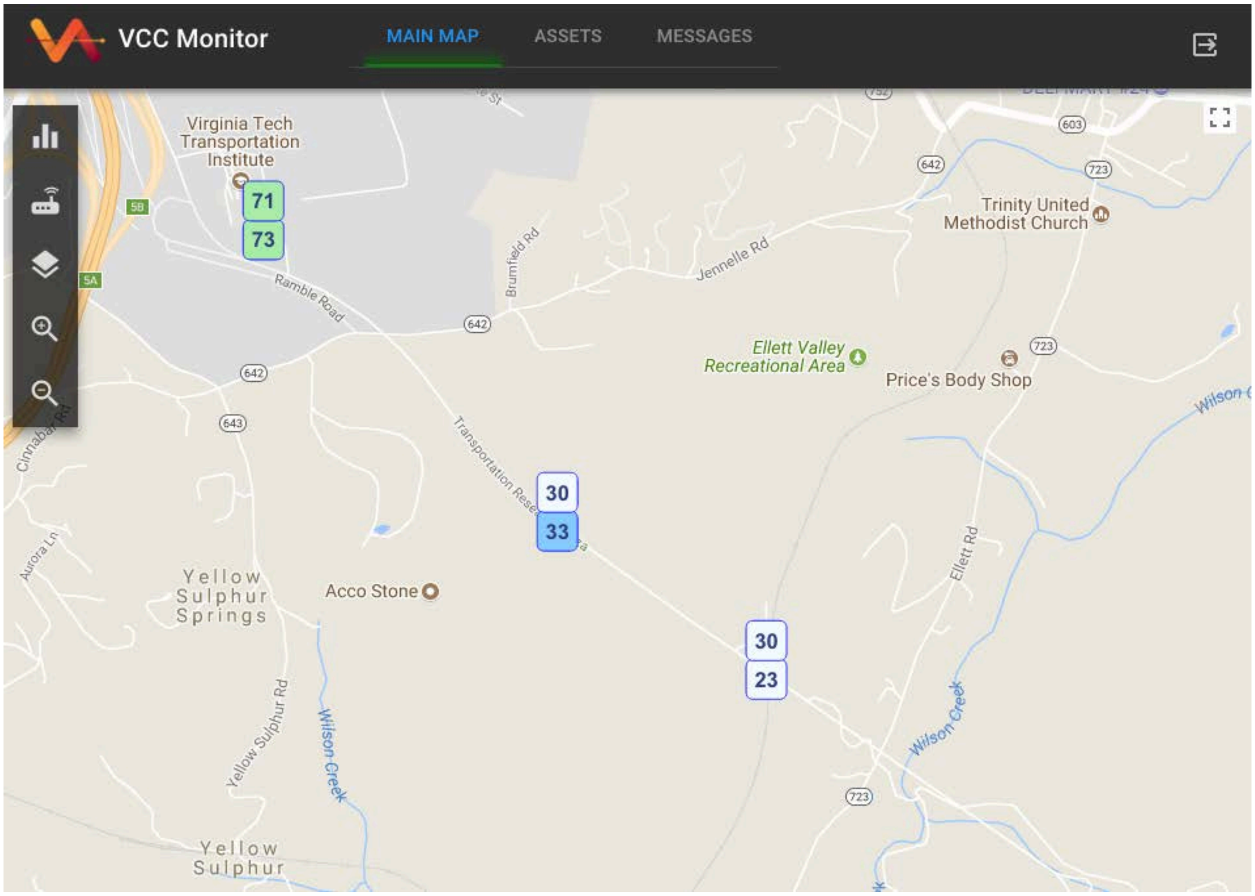


Figure 9: VCC Monitor Temperature Data Overlay

Pothole Identification

VTTI also created a Pothole Detections overlay within VCC Monitor to display the potholes identified by the pothole detector (discussed in detail later in this section). As potholes are detected, they are displayed within a matter of seconds within VCC Monitor, as shown by the black and gray icons in Figure 10 below. These pothole detection icons can represent a single pothole, or a cluster of potholes, as indicated by the pothole counter in the black ovals. Zooming in on a pothole cluster will allow the user to ascertain more specific information regarding each pothole cluster, such as whether the cluster is actually multiple clusters located close to each other. The pothole icons remain active until manually cleared by a user within VCC Monitor.

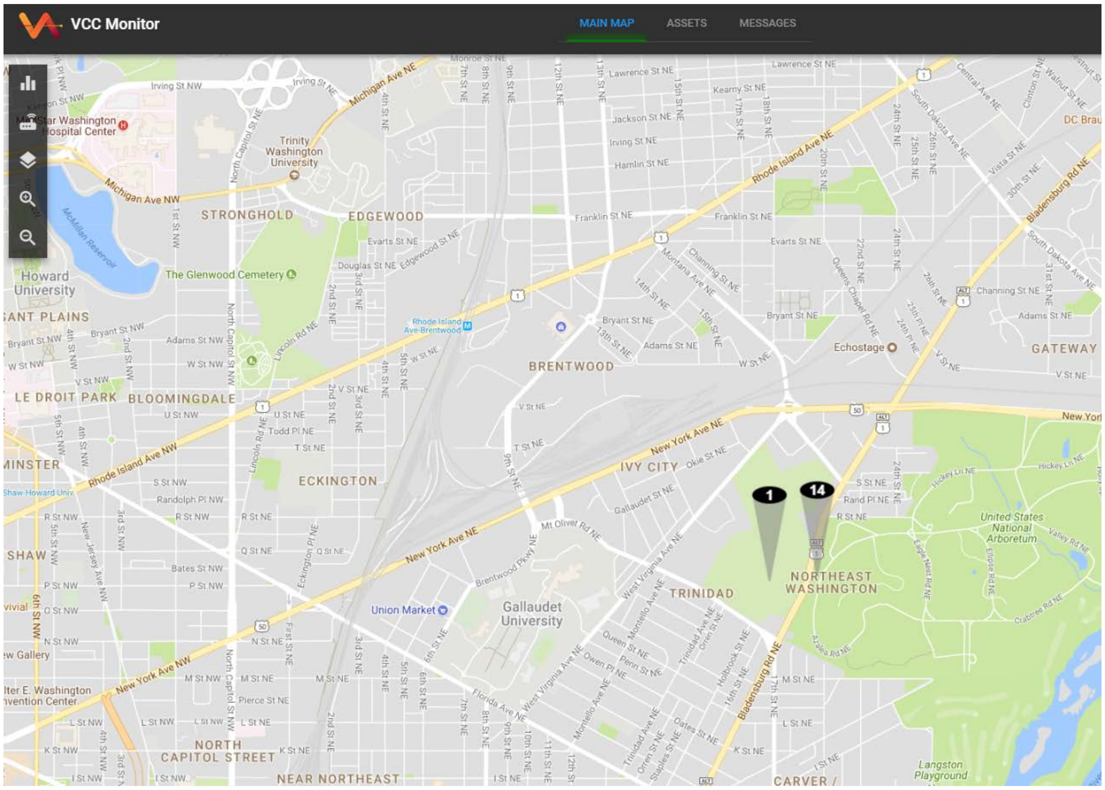


Figure 10: VCC Monitor Pothole Detections Overlay

VTTI created a VCC Monitor Training document (embedded within Appendix C) to facilitate DDOT's learning process with VCC Monitor. In addition, VTTI provided a more comprehensive in-house presentation at the DDOT facility on October 3, 2017.

TIMBuilder Application

A TIMBuilder application was previously developed by VTTI for the VCC to allow users to manually create and schedule TIMs that will be sent to the OBUs and the VCC Mobile application via cellular or DSRC communications. By selecting an area on the map, a manual TIM can be created including the specification of short text, long text, category, directionality, start/end time, RSE deployment set, and scheduled for display during a specified duration. Once the user selects the option to schedule the TIM, the TIMBuilder application interfaces with the VCC Cloud to push the TIM message to the appropriate RSUs (based on the geo-location of the TIM) and to the cellular data interface. Both channels are used to push TIMs to the OBUs and the VCC Mobile App through the best possible channel. An example of the TIMBuilder application interface is shown in Figure 11.

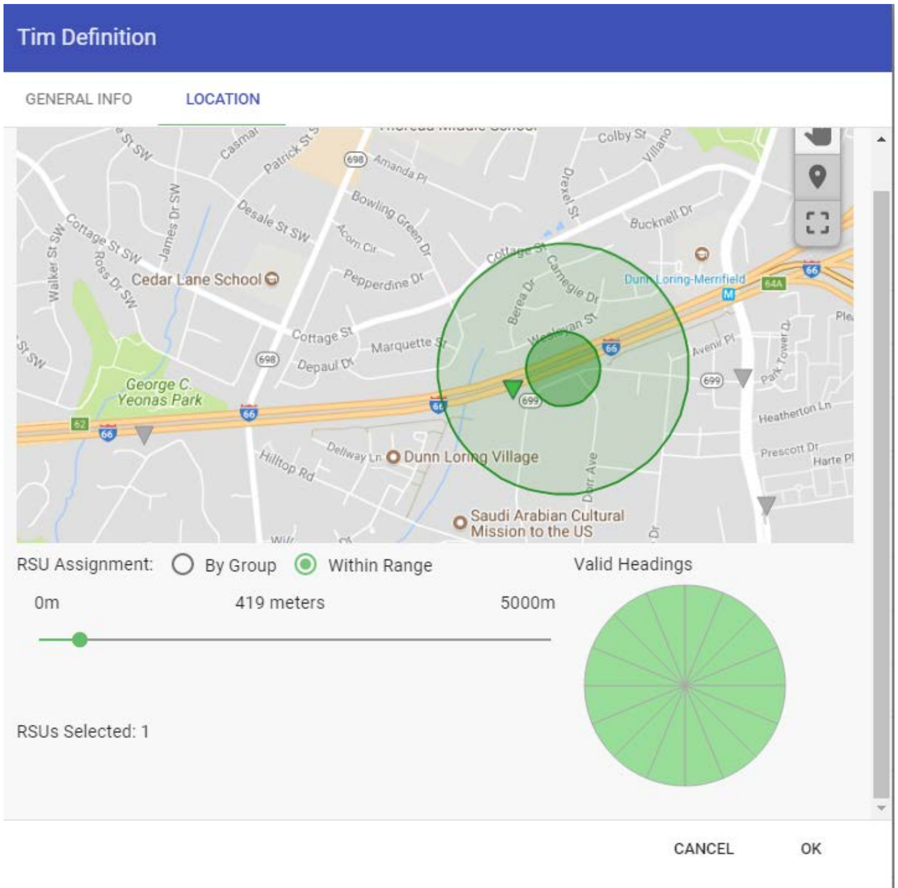


Figure 11: TIMBuilder Application Example Display

VCC Mobile

VTTI has also created an Android smartphone-based application called VCC Mobile to provide a platform extension that interfaces with the OBE and provides a means for the driver to interact with the system. The application provides real-time messaging capabilities to drivers in a format that limits attentional demand and is appropriate for use while driving. With that being said, DDOT chose to have the smartphone installed on the back wall of the truck cab to minimize distraction and driver interaction with the VCC Mobile system. Thus, with the current setup, VCC Mobile is typically used as a data collection tool running in the background whenever the truck's ignition is on.

If DDOT were to choose to utilize VCC Mobile for additional demonstration purposes, the application has additional interactive capabilities as described below. (A full VCC Mobile training document is also embedded within Appendix C and was presented to DDOT in person on October 3, 2017.)

The VCC Mobile app acts as an interface to the OBE via a Wi-Fi connection (utilizing the MiFi). VCC Mobile utilizes the OBE's GPS information and communication channels to exchange messages with the VCC Cloud, a centralized system that supports the management of CV message traffic between entities interacting on the VCC. VCC Mobile presents real-time, location-based TIMs including:

- Significant weather conditions

- Traffic conditions (e.g., slowing or clearing of traffic)
- Roadway incidents (e.g., stalled vehicles or accidents)
- DMS content (e.g., active traffic management [ATM]/high occupancy vehicle [HOV] status)
- Emergency vehicle presence
- Work zone alerts
- Other driving related messages

The driver may also make verbal reports of road conditions and call for help if stalled or stranded. TIMs and other customized messages are presented through a user interface that presents visual and/or audio speech messages when relevant, an example of which is shown in Figure 12.

Figure 12: VCC Mobile Application

Messages posted regarding locations ahead of the vehicle's path of travel are listed in a queue, and the distance to the message area is counted down as the driver approaches. Once the vehicle enters the active geo-area for the message, the message displays to the driver with a detail panel, and a text-to-speech voice interface reads the message content to the driver. Drivers may also tap on an upcoming TIM in the queue that they are approaching to get a preview of the information it contains. This same interface has been expanded to support an ATM display, traffic signal assistance display, and current speed limit display function. Aspects of the display strategy are context-relevant, meaning that certain display features are only posted at times when they are relevant to the current driving task. For example, the ATM display is only present when driving on an ATM-controlled road, and a signal display is only present when approaching signals that support SPaT output. The current SPaT display is used to validate the receipt of the data and is not yet refined into a display formation that is intended for a general public driver to use in everyday operation.

VCC Mobile Worker Application

The VCC Mobile Worker Application is another application created by VTTI for the VCC that facilitates the communication of relevant work zone and maintenance information between work zone personnel and drivers. Work zone personnel select their duty status (i.e., on/off duty) and their current activity (e.g., snow plow, mowing, static work area, etc.). This information is sent to the VCC Cloud via cellular communications, where it is processed and advisories and alerts are created for drivers. The VCC Cloud builds dynamic traveler messages and pushes them to drivers on the VCC Mobile app via DSRC or cellular depending on their position, speed, direction, etc. (Figure 13).

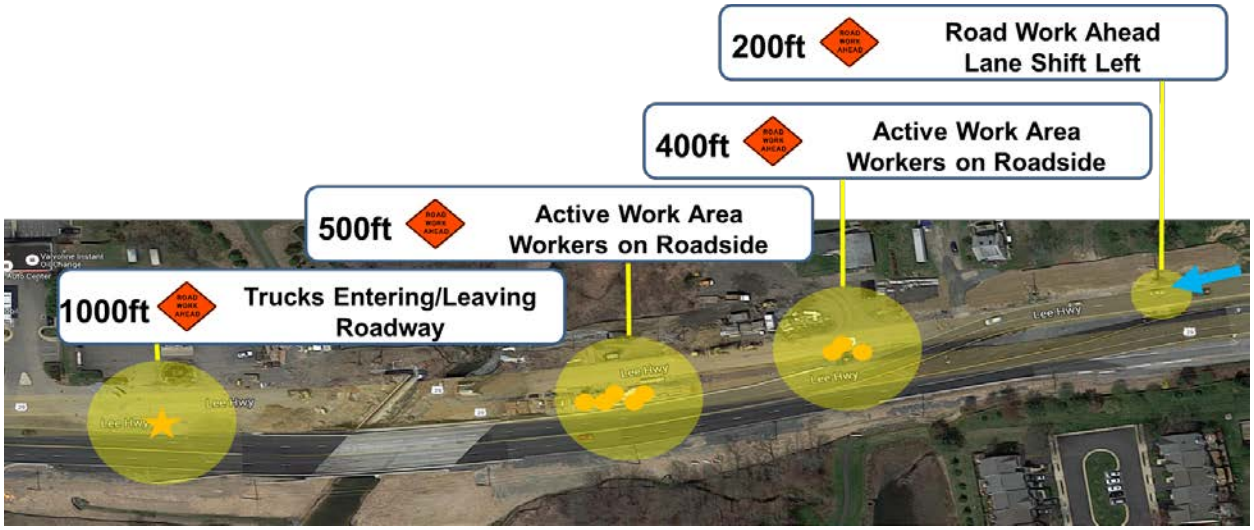


Figure 13: VCC Mobile Worker App Dynamic Traveler Messaging

Pothole Detector

As the result of a literature review regarding the state-of-the-art in smartphone-based pothole detection, VTTI decided to develop an in-house pothole detection algorithm and application utilizing the Samsung Galaxy S5's local sensors. The literature review provided a starting point for VTTI to develop Android code that runs the algorithms identified in the literature review.

Development Process

The applicable literature focused on two different solutions to smartphone-based pothole detection: applying predetermined functions to the accelerometer input and passing the accelerometer input to a neural network looking for features associated with potholes. For the purpose of a smartphone-based pothole detection application, the VTTI team decided against using a neural network, as presented by Kulkarni et al. (2014) for two reasons: (1) to keep the detection algorithm lightweight so that it could run on an end user's phone without significant battery drainage, and (2) so the accelerometer functions would be less processor-intensive.

The initial focus during development of the pothole detector was to implement algorithms from various research papers, namely the four accelerometer input functions discussed by Mednis et al. (2011). These functions identify a pothole when there is a large z-acceleration, a large change in z-acceleration over a short time, a period of accelerometer readings with a high standard deviation, and a near-zero accelerometer reading (produced while falling into the pothole), respectively. The team discovered that the algorithms did not generalize well since the test data tended to be from a single vehicle driving at a consistent speed on a fixed track with a high-end inertial measurement unit (IMU). The focus shifted to finding a more general solution that would work for a variety of speeds, vehicles, and less precise accelerometer data from a lower-end IMU such as the type present in the typical Android phones.

Significant work was done to filter noise out of the accelerometer data. Noise comes from a variety of sources, such as road noise and engine noise. Removing the noise leaves a cleaner accelerometer signal to utilize in the rest of the algorithm to detect potholes. Several versions of Fourier transforms were implemented to both learn a noise profile and to filter out the noise. Since different vehicles have different physical characteristics such as suspension and wheel base, it is important to create a custom model for each vehicle rather than using a pre-generated generic model. It was also discovered that the noise profile changes considerably for a given vehicle at different speeds, so it was more effective to create multiple speed-based noise profiles. Both exponential moving averages and linear moving averages, as well as a cumulative standard deviation model, were utilized, and several iterations of tuning the coefficients for the various models were conducted. Additionally, multiple methods of applying the learned speed-based noise profiles to remove the noise from the sample were tried, including subtractive noise filtration and multiplicative noise filtration. Weighting was applied to the multiple speed profiles to generate a profile as accurate as possible for each speed. These experiments with filtering noise allowed a custom noise profile to be created for each vehicle at various speeds and to continually update the model to reflect the noise from the latest road surface.

Some tests were done with a more complex pothole signature. These tests defined a pothole as a series of events in a quick sequence. The events, in order, were a drop (off the edge of a pothole), an impact (with the bottom of the pothole), a rise (up the far edge of the pothole), and finally a levelling off. Although it seemed like this definition should be more accurate in identifying potholes, it was discovered that depending on how a pothole was struck and the size of the pothole, the resulting pothole signature did not always follow this scheme. While experimenting with this pothole signature, additional calculations on the width and depth of the pothole derived from this data and the vehicle speed were used to determine the feasibility of a pothole with the calculated dimensions.

Experiments in detecting a cornering maneuver based on the transverse acceleration values and suspending the pothole detection algorithm were tried. The lateral acceleration values caused by driving around a corner change the effectiveness of the pothole detection algorithm. Additionally, a corner in an intersection is frequently accompanied by a bump from the crown of the intersection, which can look similar to a pothole.

Various methods of grouping acceleration values were tried. It was quickly determined that using a single buffer of acceleration values had poor performance, so using multiple buffers for the various speeds was selected. A separate buffer is used for approximately each 5-mph range. Additionally, experiments were done on weighting and not weighting the values as they were placed in the buffer, as well as incorporating a delta-speed threshold at which acceleration values were discarded. Weighting was also applied to combine multiple speed buffers to create a custom speed buffer at any speed.

Multiple iterations of manual tuning were done with various threshold formulas and threshold values to increase the number of potholes correctly identified and to decrease the number of non-potholes identified as potholes.

Using the Pothole Detector

The phone can be positioned in any orientation, but it must be placed in either a holder securely mounted to the vehicle body (as shown in Figure 6) or on a firm surface. If the phone is held in the user's hands, the accelerometer values will be dampened, preventing the potholes from being identified.

The vertical component of the orientation of the phone is determined using the accelerometer values. The direction of the largest acceleration is normally due to gravity and therefore is identified as the downward direction. Maneuvers that cause large accelerations in a sideways direction, such as high-speed cornering, can affect this calculation of the vertical component and result in an apparent dampening of the vertical accelerometer values.

The GPS is used to obtain the speed, latitude, and longitude of the phone and corresponding vehicle. The speed is used in the algorithm. The latitude and longitude are used to report the location of a detected pothole. This program requires a reliable GPS input to function optimally. If GPS becomes unavailable during a drive, the previous GPS reading is utilized until a new reading is received. This affects both the detection and reporting of potholes.

While the pothole detection application is running, the program receives a continuous stream of accelerometer readings and periodic GPS readings. The vertical component of the acceleration is extracted from the accelerometer input. The pothole detection algorithm looks for instances when these accelerometer values are outside of normal. Normal is determined by analyzing the previous accelerometer values obtained when the vehicle was driving at a similar speed. During testing, it was determined that what is normal for one vehicle can be very different than normal for a second vehicle, so it is important to dynamically determine a baseline. Similarly, it is important to have a different definition of normal at various speeds to compensate for the variation in vehicle suspension at different speeds, as well as the increased jolt a small bump creates at higher speeds. Additionally, the calculations of expected normal accelerometer values are continually being updated using the newly received data. This is particularly useful as the vehicle transitions between roadway surfaces.

The signature of a pothole tends to include two separate acceleration spikes. The pothole detection algorithm requires these two spikes to occur reasonably close together. The thresholds currently being used to determine when an accelerometer value is significantly outside of normal varies by speed. Both spikes must be quite large, but different thresholds are used for both. At low speeds, one spike must be approximately four sigma (99.994%) and the second spike must be approximately three sigma (99.7%) or greater to be considered a pothole. At higher speeds, the threshold values increase at a linear rate, but always require two spikes close together.

Data Recorder Application

In addition to the pothole detector application, a second Android application was developed to allow the collection and annotation of pothole data. This application stores the accelerometer and GPS data to a file, along with any manually flagged annotations indicating a pothole of various sizes or a non-pothole jolt. This application allowed the collection of real-world data from the actual trucks in which the pothole detector algorithm will be used.

Data Viewer/Algorithm Analysis

To facilitate the experimentation and analysis of various algorithms, thresholds, and coefficients within the algorithms, a stand-alone application was developed. This application loads in the data files generated by the data recorder application and applies the same algorithm used by the pothole detection application to search for potholes. This application provides an overview of the potholes detected and the annotations manually added to the drive. In addition, it provides detailed charts of the speed and accelerometer data to facilitate analysis of the algorithms (see Figure 14 for example charts).

Figure 14: Pothole Detection Speed and Accelerometer Charts

Testing and Results

VTTI worked with DDOT to identify a set of reference potholes in Washington, D.C., that were representative of potholes that would be prioritized for maintenance action. After some target potholes were identified, a VTTI researcher rode along with a DDOT driver in one of the OBE-equipped trucks and collected data while driving across the target pothole sites at various speeds and angles. The resulting data allowed the algorithm to be tuned for optimal performance in that situation.

The initial results from several of these short data collection runs resulted in an approximate 60% accuracy. A larger, more formal assessment of pothole detection performance was outside the scope of this initial development effort. VTTI recommends adjusting the algorithm strategy towards the use of machine learning techniques to improve overall detection performance. The system, as developed, provides a reasonable platform for extending the detection capability towards the application of the machine learning processes.

Future Suggestions

The team has identified a number of suggested ways to improve upon the accuracy of the pothole detection application. First, it may be beneficial to assume that the phone is securely mounted and the orientation will not change during a drive. This would allow the orientation to be calculated while the vehicle is at rest and then used throughout the drive. Second, an automated "scoring" method could be developed to evaluate changes to the algorithm. This could be used, in conjunction with the gradient descent from the test drives, to optimize the threshold formula. Additionally, the noise filtering functionality and test performance could be reincorporated into the recorded data. This has promise in providing additional generalization to various vehicle types. Another suggestion is to process acceleration values in blocks rather than using rolling calculations. This might make it easier to detect a pothole near the end of a section of bad road or to detect the second of successive potholes. Lastly, it may be prudent to discontinue pothole detection functionality if the GPS signal is not present to prevent combining values from potentially different speeds into the wrong buffer, thereby corrupting the speed-based calculations.

Task E – Support Data Analytics

The VCC Cloud manages the collection, storage, and distribution of data collected by the OBE installed in DDOT's trucks. The VCC Cloud provides a means to subscribe to real-time data feeds and access archived data. The VCC Monitor web client, discussed in detail above, utilizes this real-time data interface for all of its application data needs, providing a variety of data visualization and quantification functions.

DDOT has been provided with several methods to access data from the VCC Cloud. In its present form, the VCC Cloud receives real-time incoming data such as BSMs, basic mobility messages (BMMs) (not currently supported by this project), and TIMs and places them into a message queue system. The message queues may be accessed via direct access from server applications running on the cloud server or through a REST application programming interface (API) interface for external applications.

The public API provides access to the following:

- BSMs received from RSUs (also cellular)
    - Information is a subset of all BSM data including location, speed, temperature, surface temperature, and vehicle state.
- TIMs broadcast by RSUs
- SPaT for specific intersections
- Map data describing the lane configuration of specific intersections
- DMS location and message info
- Pothole report data from VCC Mobile pothole application
- RSU location info

A copy of the API documentation has been provided in Appendix B. The documentation includes an example of a working Python script that connects to the WebSocket and receives BSM data. DDOT users have been provided with credentials to allow access to real-time data through the REST API interface.

The VCC Cloud also includes a data archive system. Real-time data flowing through the cloud and message queue system are monitored by the data archive module. The archive module pushes all BSM, BMM, and TIM messages to an archive data server located in VTTI's data center. The archive data server receives the messages and formats them for storage in a DB2 database system for longer-term storage and non-real-time transactional data access. DDOT users were provided with a means to access the existing data archive storage system to retrieve data per their requirements.

REFERENCES

Ahmed, Mohamed, et al. (2016). *Connected Vehicle Pilot Deployment Program Phase 1, Participant Training and Education Plan–ICF/Wyoming*. No. FHWA-JPO-16-294. Washington, DC: U.S. Department of Transportation.

Allouch, A., Koubaa, A., Abbes, T., & Ammar, A. (2017). RoadSense: Smartphone Application to Estimate Road Conditions Using Accelerometer and Gyroscope. *IEEE Sensors Journal,* vol. 17, no. 13, Jan. 2017, pp. 4231-4238. doi:10.1109/jsen.2017.2702739.

Bayless, Steven, et al. (2015). *Recommended Practices for DSRC Licensing and Spectrum Management: A Guide for Management, Regulation, Deployment, and Administration for a Connected Vehicle Environment*. No. FHWA-JPO-16-267. Washington, DC: U.S. Department of Transportation.

Brisimi, T. S., Cassandras, C. G., Osgood, C., Paschalidis, I. C., & Zhang, Y. (2016). Sensing and Classifying Roadway Obstacles in Smart Cities: The Street Bump System. *IEEE Access*, vol. 4, pp. 1301-1312. doi:10.1109/access.2016.2529562

CAMI Research. (2015). *The RS-232 Standard*. http://www.camiresearch.com/Data_Com_Basics/RS232_standard.html

CVRIA. (2016, May 5). Office of the Assistant Secretary of Transportation for Research and Technology. *Connected Vehicle Reference Implementation Architecture Website*. http://www.iteris.com/cvria/

Dey, Soumya, & Ma, Jianming. (2007). Reversible Lane Operation for Arterial Roadways–the Washington, DC Experience. *ITE 2007 Annual Meeting and Exhibit*.

District Department of Transportation. (2010). *2010 Action Agenda.* Washington, D.C.

District Department of Transportation. (2013). *District Department of Transportation ITS and Communications Master Plan*. https://app.box.com/s/1keg4ac0zump5uo62em6ey1qyle9kwom

Du, Y., Liu, C., Wu, D., & Li, S. (2016). Application of Vehicle Mounted Accelerometers to Measure Pavement Roughness. *International Journal of Distributed Sensor Networks,* vol. 12, no. 6. doi:10.1155/2016/8413146

Eriksson, J., Girod, L., Hull, Bret, Newton, Ryan, Madden, Samuel, Balakrishnan, H. (2008). The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring. *Proceeding of the 6th International Conference on Mobile Systems, Applications, and Services - MobiSys '08*. doi:10.1145/1378600.1378605

Galgano, Steve, et al. (2016a). *Connected Vehicle Pilot Deployment Program Phase 1, System Requirements Specification (SyRS)–New York City*. No. FHWA-JPO-16-303. Washington, DC: U.S. Department of Transportation.

Galgano, Steve, et al. (2016b). *Connected Vehicle Pilot Deployment Program Phase 1, Concept of Operations (ConOps)-New York City.* No. FHWA-JPO-16-299. Washington, DC: U.S. Department of Transportation. https://ntl.bts.gov/lib/59000/59300/59360/FHWA-JPO-16-299.pdf

Galgano, Steve, Talas, Mohamad, & Bradley, Peg. (2016c). *Connected Vehicle Pilot Deployment Program Phase 1, Participant Training and Stakeholder Education Plan–New York City*. No. FHWA-JPO-16-306. Washington, DC: U.S. Department of Transportation.

Ghose, A., Biswas, P., Bhaumik, C., Sharma, M., Pal, A., & Jha, A. (2012). Road condition monitoring and alert application: Using in-vehicle Smartphone as Internet-connected sensor. *2012 IEEE International Conference on Pervasive Computing and Communications Workshops*. doi:10.1109/percomw.2012.6197543

Gopalakrishna, Deepak, et al. (2015). *Connected Vehicle Pilot Deployment Program Phase 1, Concept of Operations (ConOps), ICF/Wyoming*. No. FHWA-JPO-16-287. Washington, DC: US Department of Transportation. https://ntl.bts.gov/lib/56000/56900/56997/FHWA-JPO-16-287.pdf

Gopalakrishna, Deepak, et al. (2016). *Connected Vehicle Pilot Deployment Program Phase 1, Safety Management Plan – ICF/Wyoming*. No. FHWA-JPO-16-289. Washington, DC: U.S. Department of Transportation. https://ntl.bts.gov/lib/59000/59100/59130/FHWA-JPO-16-289.pdf

Grimmer Software. (n.d.). Retrieved September 20, 2017, from http://www.grimmersoftware.com/home.html

Hamill, Mary, Wacht, Mike, & Brown, Jeff. (2016). *Connected Vehicle Pilot Deployment Program Phase 1, Participant Training and Stakeholder Education Plan–Tampa (THEA)*. No. FHWA-JPO-16-318. Washington, DC: U.S. Department of Transportation.

Hill, Christopher J., & Garrett, J. Kyle. (2011). *AASHTO Connected Vehicle Infrastructure Deployment Analysis*. No. FHWA-JPO-11-090. Washington, DC: U.S. Department of Transportation.

Jo, Y., & Ryu, S. (2015). Pothole Detection System Using a Black-box Camera. *Sensors,* vol. 15, no. 12, pp. 29316-29331. doi:10.3390/s151129316

Johnson, Steve, et al. (2016a). *Connected Vehicle Pilot Deployment Program Phase 1, Deployment Readiness Summary–Tampa (THEA)*. No. FHWA-JPO-16-322. Washington, DC: U.S. Department of Transportation. https://ntl.bts.gov/lib/60000/60600/60699/FHWA-JPO-16-322.pdf

Johnson, Steven, et al. (2016b). *Connected Vehicle Pilot Deployment Program Phase 1, Comprehensive Pilot Deployment Plan – Tampa (THEA)*. No. FHWA-JPO-16-321. Washington, DC: U.S. Department of Transportation. https://ntl.bts.gov/lib/60000/60600/60698/FHWA-JPO-16-321.pdf

Joubert, D., Tyatyantsi, A., Mphahlehle, J., & Manchidi, V. (2011). Pothole Tagging System. Retrieved from https://researchspace.csir.co.za/dspace/bitstream/handle/10204/5384/Joubert_2011.pdf?sequence=1&isAllowed=y

Kulkarni, A., Mhalgi, N., Gurnani, S., & Giri, N. (2014). Pothole Detection System Using Machine Learning on Android. *International Journal of Emerging Technology and Advanced Engineering,* vol. 4, no. 7. Retrieved March 30, 2017. Retrieved from http://www.ijetae.com/files/Volume4Issue7/IJETAE_0714_55.pdf

Mednis, A., Strazdins, G., Zviedris, R., Kanonirs, G., & Selavo, L. (2011). Real Time Pothole Detection using Android Smartphones with Accelerometers. *2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*. doi:10.1109/dcoss.2011.5982206

Mohan, P., Padmanabhan, V. N., & Ramjee, R. (2008). Nericell: Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones. *Proceedings of the 6th ACM conference on Embedded network sensor systems - SenSys '08*. doi:10.1145/1460412.1460444

National Operations Center of Excellence. (2015). [Diagram showing the relationship of V2I standards, both current and needed, and how they relate to the vehicles, RSEs, and infrastructure.] *V2I Deployment Coalition.* Retrieved from http://www.transportationops.org/twg-5-deployment-standards

Orhan, F., & Eren, P. (2013). Road Hazard Detection and Sharing with Multimodal Sensor Analysis on Smartphones. *2013 Seventh International Conference on Next Generation Mobile Apps, Services and Technologies, 2013*. doi: 10.1109/ngmast.2013.19

Radopoulou, S. C., & Brilakis, I. (2015). Patch Detection for Pavement Assessment. *Automation in Construction*, vol. *53*, pp. 95-104. doi:10.1016/j.autcon.2015.03.010

Radopoulou, S. C., & Brilakis, I. (2016). Improving Road Asset Condition Monitoring. *Transportation Research Procedia,* vol. 14, pp. 3004-3012. doi:10.1016/j.trpro.2016.05.436

Rode, S. S., Vijay, S., Goyal, P., Kulkarni, P., & Arya, K. (2009). Pothole Detection and Warning System: Infrastructure Support and System Design. *2009 International Conference on Electronic Computer Technology*. doi:10.1109/icect.2009.152

Singh, G., Bansal, D., Sofat, S., & Aggarwal, N. (2017). Smart Patrolling: An Efficient Road Surface Monitoring Using Smartphone Sensors and Crowdsourcing. *Pervasive and Mobile Computing,* vol. 40, pp. 71-88. doi:10.1016/j.pmcj.2017.06.002

Talas, Mohammad, et al. (2016). *Connected Vehicle Pilot Deployment Program Phase 1, Deployment Readiness Summary – New York City*. No. FHWA-JPO-16-310. Washington, DC: U.S. Department of Transportation. https://ntl.bts.gov/lib/60000/60700/60710/FHWA-JPO-16-310.pdf

U.S. Department of Transportation (2010). *Frequency of Target Crashes for IntelliDrive Safety Systems.*

U.S. Department of Transportation. (2016, October 31). *DSRC Roadside Unit (RSU) Specifications Document*, Version 4.1. No. DTFH61-12-D-00020. Washington, DC: U.S. Department of Transportation.

Waggoner, Joe, et al. (2016). *Connected Vehicle Pilot Deployment Program Phase 1, Concept of Operations (ConOps)–Tampa (THEA)*. No. FHWA-JPO-16-311. Washington, DC: U.S. Department of Transportation. https://ntl.bts.gov/lib/57000/57000/57032/FHWA-JPO-16-311.pdf

Wang, H., Chen, C., Cheng, D., Lin, C., & Lo, C. (2015). A Real-Time Pothole Detection Approach for Intelligent Transportation System. *Mathematical Problems in Engineering,* vol. 2015. doi:10.1155/2015/869627

Wright, James, et al. (2014). *National Connected Vehicle Field Infrastructure Footprint Analysis*. No. FHWA-JPO-14-125. Washington, DC: U.S. Department of Transportation. https://ntl.bts.gov/lib/52000/52600/52602/FHWA-JPO-14-125_v2.pdf

APPENDIX A: POTHOLE DETECTION LITERATURE REVIEW

Introduction

Pavement roughness significantly affects riding quality, driving safety, and fuel consumption. Therefore, many municipalities and roadway operators are trying to find efficient and reliable ways to make roads safer and smoother.

Road surface monitoring methods are typically divided into two categories: imaging and mobile sensing, both of which are examined in this literature review. Image recognition uses snap images of the road to determine road depth through 3-D laser scanners, gray scales, and contour textures. Mobile sensing relies on smartphone sensors that provide real-time data.

This literature review provides a survey of road surface monitoring methods, including those that are commercially available for purchase as well as prototype systems that have only been used in research studies thus far. Lastly, interpretations and conclusions are discussed as related to the pothole detection application developed by the team.

Road Surface Monitoring Methods and Applications

This section provides a discussion of existing road surface monitoring data collection methods, both imaging and mobile sensing. Table 21 provides a summary of these methods.

In addition to the objective data collection methods discussed in the following sections, subjective pothole detection applications that utilize crowdsourcing are currently available for download. These applications include the nationwide Pothole Alert 311, My LA 311 (for Los Angeles, California), FresGo (for Fresno, California), and GoModesto! (for Modesto, California). These crowdsourcing apps allow users to report various conditions in their areas to relevant authorities, including potholes.

Table 21: Summary of Road Surface Monitoring Methods

	Components	Experimentation Location	Sample Size	Devices used for Data Collection	Test Vehicles	Distance Covered	Mount Points	Detection for Road Anomalies	Threshold Used	Accuracy Results
Radopoulou and Brilakis	iPhone 5s 8 MPCamera	Cambridge, UK	140	Microsoft Visual Studio.NET, Windows Presentation Foundation, Intel Core i7 CPU, iPhone 5s 8 MP, Canon VIXIA HF 5100	N/A	N/A	Upper bumper or above license plate	Road Patches	N/A	84% Precison and 96% Recall
Joubert	Kinect Technology	South Africa	N/A	Kinect	2007 Volkswagen Polo Classic	N/A	Dashboard	Potholes	60km/h	N/A
Orhan & Eren	GPS, accelerometer, camera, and microphone	Small University Roads	25	Android Smartphone	N/A	Distance unknown; duration 20	Windshield	Potholes	N/A	82% Recall and 93% Precision
Black-Box Camera	Black-box Camera	Seoul City	20	Cortex-A8 and 32 GB of memory	N/A	N/A	Windshield	Potholes and Lane Markings	N/A	71% True Positives and 88% Precision
Street Bump	Smartphone, Gyroscope, Accelerometer, and GPS	Boston	N/A	StreetBump application on Smartphone	N/A	N/A	Windshield	Potholes and Bumps	N/A	N/A
Road Bump	Smartphone, Accelerometer and GPS	Arkansas	N/A	RoadBump application on Smartphone	N/A	19 miles	N/A	IRI	N/A	98%
RoadSense	Samsung Galaxy Alpha, 3-axis accelerometer, and Gyroscope	N/A	2000	Samsung Galaxy Alpha	N/A	25km	Dashboard	Potholes, Vehicle Speed, Distance, and Road Type	N/A	98.5% Recall and Precision
PSD using Zigbee	Accelerometer, Microprocessor Control Unit (MCU), and Zigbee	Huzhou, China	N/A	GPS, Angular Velocity Sensor, Geomagnetic Sensor, ZigBee, Acorn RISC Machine (ARM), 3G, Touch Screen	N/A	N/A	Above Left and Right Rear Tires	Pavement Roughness	N/A	Riding Quality Index (RQI) Relative Error below 10%
P2	Soekris 4801 embedded with computer running Linux, WiFi Card, Sprint EVDO Rev A Network Card, External GPS mounted to the roof, 3-axis accelerometer	Boston	380	Soekris 4801 embedded with computer running Linux	7 Taxis (Toyota Prius)	9730 kilometers over 10 days	Dashboard	Potholes	N/A	Less 0.2% False-Positives
Nericell	HP iPAQ hw6965, WiTilt accelerometer, HTC Typhoon for cellular localization	Bangalore and Seattle	310	HP iPAQ hw6965, WiTilt accelerometer, HTC Typhoon for cellular localization	Toyota Qualis	622km/h	Dashboard	Bumps, Potholes, Braking, and Honking	N/A	10% False-Positives
DTW	Smartphones: Nexus 5, Samsung S5, Samsung Note3, Moto E, Samsung S4 Mini	Chandigarh City, India	N/A	Smartphones: Nexus 5, Samsung S5, Samsung Note3, Moto E, Samsung S4 Mini	Toyota Etios	4hrs/day x 8 days	Dashboard	Potholes and Bumps	N/A	88.66% Pothole Detection and 88.89% Bump Detection
Road Condition Monitoring and Alert App	Backend Server, Dashboard Client, and Smartphone	N/A	N/A	Backend Server using 52North	N/A	N/A	N/A	Rough Roads, Bumps, and Potholes	N/A	N/A
Mednis	Z-THRES, Z-DIFF, STDEV, G-ZERO	Latvia	100	Samsung i5700, Samsung Galaxy S, HTC Desire, HTC HD2	BMW 323 Touring	N/A	Dashboard	Potholes	Z-THRESH=0.4g, Z-DIFF=0.2g, STDEV=0.2g, G-ZERO=0.8g	90% True Positives
Wang	Z-THRESH, Z-DIFF, STDEV(Z), G-ZERO, Z-THRESH AND G-ZERO	N/A	N/A	Mobile Device	N/A	N/A	N/A	Potholes	N/A	100% Accuracy and No False-Positives

Image Recognition

As previously explained, image recognition methods rely on cameras to monitor the road surface. According to Joubert, Tyatyantsi, Mphahlehle, and Manchidi (2011), cameras provide a relatively inexpensive way to do pavement inspection, but require the use of complicated algorithms to produce 3-D data. Radopoulou and Brilakis (2016) believe that while image recognition is accurate, it is costly and does not provide real-time data. Furthermore, the imaging method has proven to be highly accurate but it requires high computation power, an abundance of resources, and is not suitable for mobile devices (Wang, Chen, Cheng, Lin, & Lo, 2015). On the other hand, Rode, Vijay, Goyal, Kulkarmi, and Arya (2009) believe that the effectiveness of imaging methods is limited by factors like bad weather and image processing time.

Radopoulou and Brilakis (2015) collected video data consisting of approximately 4,000 frames in Cambridge, UK, using an imaging method that detected road patches. The team utilized an iPhone 5s with an 8-MP camera mounted above the bumper of the vehicle to collect 140 images around Cambridge. Using Microsoft Visual Studio.NET, Windows Presentation Foundation, and Intel Core i7 CPU, Radopoulou and Brilakis (2015) were able to process the collected images and provide road patch detection results with a precision of 84% and 96% recall. In this study, precision is related to the detection exactness (calculated by dividing true positives by the sum of true positives and false positives) and recall refers to the detection completeness (calculated by dividing true positives by the sum of true positives and false negatives).

Joubert et al. (2011) created a low-cost sensor system which could be mounted to a vehicle and detect and analyze potholes. This system heavily relies on Microsoft's Kinect Technology for the Xbox 360, an inexpensive 3-D sensing device that features a color camera, an infrared projector, and an infrared receiver. Tested on roads in South Africa, Joubert's system was mounted on the dashboard of a 2007 Volkswagen Polo Classic that traveled at an average of 60 km/h. Its accuracy results are yet to be published.

Orhan and Eren (2013) presented a multimodal sensor analysis system for pothole detection that used the advanced capabilities of smartphones, a mobile application, a plug-in based multimodal analysis interface for signal and image processing applications, and a toolset to connect to other users or servers for sharing the results. The system provided real-time sensing, multimedia processing, and communication capabilities, and utilized an Android smartphone mounted to the vehicle windshield. The GPS, accelerometer, camera, and microphone components worked together to collect images and to detect road anomalies such as potholes and bumps, which were then stored in the central repository. Once stored, the application was able to record, analyze, and share the data. Information on incoming potholes and the location of potholes was available to users of the application through social networks, push messaging, and voice notifications based on their proximity to the incoming pothole location. The multimodal sensor system provided results with an 82% recall value and a 93% precision rate for pothole detection.

Jo and Ryu (2015) collaborated with road authorities in Seoul, Korea, to implement their pothole-maintenance system that used a method requiring a black-box camera (i.e., a forward-facing camera engulfed by a black box). The camera was mounted on the front of the windshield and is embedded with software that captures and collects a variety of attributes stored in the pothole-management server.

After collecting 20 video clips the black-box camera detected potholes and lane markings with an accuracy of 71% true positives (also known as recall) and 88% precision.

Mobile Sensing

As stated earlier, mobile sensing pothole detection methods rely on smartphone sensors that provide real-time data. Furthermore, the widespread use of smartphones allows large quantities of data to be collected using mobile sensing applications, covering vast geographical areas. There are two main advantages to mobile sensing methods: they are more cost-efficient than image recognition and they provide real-time data (Radopolou & Brilakis, 2016; Wang et al., 2015). The real-time data provided by mobile sensing methods has led them to be more widely used than imaging recognition (Radopolou & Brilakis, 2016). According to Orhan and Eren (2013), "in-car mobile sensor applications generally make use of accelerometer, magnetometer, and GPS. However, most studies rely on a single sensor… [because of] the challenges related to the development of multimodal sensor analysis applications." Despite the clear advantages, however, Wang et al.'s (2015) mobile sensing method had a fairly high rate of false positives.

Commercial Applications

Currently, there are several mobile sensing applications available that utilize the accelerometer and GPS features found in smartphones. Current applications and commercialized products include Street Bump and RoadBump. These applications have been tested and operated in metropolitan areas such as Boston, San Francisco, and Oakland.

Brisimi, Cassandras, Osgood, Paschalidis, and Zhang (2016) have developed an infrastructure-free iOS smartphone-based application called "Street Bump" that is currently being utilized in Boston. Street Bump utilizes smartphone-collected data crowdsourced from Boston citizens to sense and classify roadway obstacles throughout the city. "Bump" is a broad term to describe various road obstacles such as potholes, sunk casting, utility patches, catch basins, train tracks, and speed bumps (Brisimi et al., 2016). Their application collects data on two types of bumps and classifies those bumps as either actionable or non-actionable. Brisimi et al. (2016) define actionable bumps as potholes and sunk castings that are caused by nature or accident and require prompt repair. Non-actionable bumps are defined as those caused by expected, known, or benign obstacles (e.g., train tracks, speed bumps, and relatively flat castings) that do not require immediate attention. This classification system helps the city allocate their limited resources to the highest priority bumps. Street Bump utilizes a smartphone-based machine-learning algorithm, AdaBoost. Implemented within Street Bump's application, it allows Street Bump to collect and record gyroscope and accelerometer data.

RoadBump is an Android application currently available in the Google Play Store that utilizes a smartphone's GPS function and accelerometers to measure the roughness of a road, similar to Boston's Street Bump. The app allows the driver to see the bumps, dips, and waves in the road as well as graphs and maps that depict where those road anomalies are. The application's key features are estimated International Roughness Index (IRI), graphs, low cost, high-end analysis, compatibility with ArcGIS, two hours of recording, no network connectivity needed, and street and satellite views. RoadBump was tested in Arkansas in 2015 along a 19-mile stretch of highway, and the results were compared to the Arkansas Highway Department's data for the same roadway. Tight correlation was shown in the IRI graphs, with an overall IRI rating within 2% of true and no section being off more than 7.5% (Grimmer Software).

Prototype Applications

Allouch, Koubaa, Abbes, and Ammar (2017) created an Android smartphone application called RoadSense to detect road conditions and quality. RoadSense users can see real-time data and analysis on road quality and conditions, visualized within the application as a road trace on a geographic map. RoadSense was tested using a Samsung Galaxy Alpha smartphone mounted to the windshield of a volunteer driver's vehicle over a three-week period. The authors tested three different machine-learning algorithms when designing the Roadsense application: C4.5 (designed by the authors), support vector machines (SVM) and Naïve Bayes. SVM are supervised learning methods for classification and regression. The Naïve Bayes algorithm is a probabilistic classifier that calculates a set of probabilities by counting the frequency and combinations of values in a given data set. The results of Allouch et al.'s (2017) study found the C4.5 algorithm to be superior, with a 98.6% accuracy, and therefore this method was utilized in their application.

Du, Liu, Wu, and Li (2016) created an integrated and wireless transfer-based measuring system that can calculate the IRI by power spectral density analysis. Comprised of three components: accelerometer, microprocessor control unit (MCU), and Zigbee, Wu et al. (2016) created a system to identify pavement roughness. With two accelerometers placed above the rear left and right tires, the device is capable of detecting pavement roughness at a high accuracy, providing a Riding Quality Index (RQI) below 10% relative error.

Eriksson et al.'s (2008) study, which created a low false-positive method (less than 0.2%) for detecting potholes, is considered the first complete and valid study for mobile sensing using smartphones. Their method utilized a signal processing and machine-learning approach using data from vibration and GPS sensors. The "Pothole Patrol" (P^2) system was deployed on seven Toyota Prius taxis in Boston to collect data. The taxis were installed with Wi-Fi cards, Sprint network cards, external GPS mounted to the roof, and three-axis accelerometers. According to Mednis et al. (2011), this study used the z-peak algorithm, which thresholds the acceleration amplitude at the z-axis. The main function of P^2 was to differentiate potholes from other road anomalies (e.g., cracks, railroad crossings, expansion joints, bumps). This experiment achieved small success, covering 2,492 kilometers of Boston's roads over 10 days and 9,739 kilometers overall (Eriksson et al., 2008). By doing so, high spatial coverage and data were provided of Boston's busiest roads.

According to Wang, Chen, Cheng, Lin, and Lo (2015), the P^2 system analyzed the x-axis and z-axis accelerometer data using five data filters, including speed, high-pass, z-peak, xz-ratio, and speed versus z ratio. Wang et al. (2015) agree that these data filters can detect potholes, but believe that only the z-peak filter can obtain precise pothole information.

Nericell is another sensor system that utilizes smartphones based on the Windows Mobile operating system and an array of sensors (Mohan, Padmanabhan, & Ramjee, 2008). The proposed algorithm for this method is designed to "virtually reorient" a disoriented accelerometer automatically within the smartphone (Mohan et al., 2008). Nericell was used as an orchestrator for smartphones to perform sensing and report data to the server. Nericell was tested in Bangalore for pothole, bump, braking, and honking detection, and found to have quite a high false negative rate for bump detection of 20%–30%.

According to Wang et al. (2015), the Nericell project only considered analyzing the z-axis accelerometer, which led to their high false negative rate. According to Mednis et al. (2011), this study used the z-peak algorithm, which thresholds the acceleration amplitude at the z-axis.

Dynamic Time Warping (DTW) is a highly accurate technique developed by Singh, Bansal, Sofat, and Aggarwal (2017) to enhance crowdsourced road surface monitoring using Android smartphones. DTW provides multiple attributes that help improve road anomaly classification and detection accuracy. DTW can compare two given time-independent sequences that vary in speed, it can adjust to time deformation and different speeds associated with time dependent data, it is suitable for smartphones, and it is simple in training compared to other techniques (Singh et al., 2017). DTW was tested in Chandigarh City, India, where an Android 5.1 SDK 22 was mounted to the dashboard of a Toyota Etios. The Etios was driven for about four hours each day over an eight-day period to collect pothole and bump data. Singh et al.'s (2017) DTW system resulted in 88.66% and 88.89% detection accuracy for potholes and bumps, respectively, outdoing most present techniques.

The Road Condition Monitoring and Alert Application is a crowdsourced sensing application that utilizes smartphone GPS, accelerometer, magnetometer, compass, and connectivity functions. The application operates as a real-time alert system for drivers on upcoming road conditions and potholes. The application is intended to serve as a useful tool for cities to identify road surface issues and improve conditions. To date, Ghose et al. (2012) have provided a demo system of the Road Condition Monitoring and Alert application that has three components: a smartphone, a backend server, and a dashboard client. The dashboard client collects and displays real-time pothole data, the backend server will be used as the cloud, and the smartphone will serve as the communicator and GPS locator. A number of challenges were identified with the demo system, including network usage, eliminating false positives, keeping the driver location anonymous, and maintaining simplicity.

Mednis et al. (2011) utilized the Z-THRESH, Z-DIFF, STDEV, and G-ZERO algorithms to analyze accelerometer data on Android smartphones, including the Samsung i5700, Samsung Galaxy S, HTC Desire, and HTC HD2. The smartphones were mounted to the dashboard of a BMW 323 Touring for data collection in Latvia. According to Mednis et al. (2011), "the evaluation tests resulted in optimal setup for each selected algorithm and the performance analysis in context of different road irregularity classes show true positive rates as high as 90%."

Wang, Chen, Cheng, Lin, and Lo (2015) proposed a pothole detection method based on mobile sensing utilizing accelerometer and GPS data. Wang et al. (2015) incorporated methods and algorithms designed by Mednis et al. (2011) to propose a system design that eliminated false-positives within real-time pothole detection. Using the Z-THRESH and G-ZERO algorithm, Wang, et al. (2015) created a real-time pothole detection method with 100% accuracy, according to their experiments.

Conclusions

There are two primary methods used for road surface monitoring. Of these two methods, mobile sensing is by far the more practical solution for use in an intelligent transportation system, and is the only type of monitoring utilized by municipalities today. Mobile sensing requires fewer resources, and more importantly, provides real-time data, which is an important feature made possible through CV technologies. For these reasons, VTTI and DDOT pursued the development of a mobile sensing, smartphone-based, pothole-detection application to be used in conjunction with DDOT's deployed OBE and VTTI's existing software applications.

While the literature review discussed a number of successful preexisting smartphone-based mobile sensing applications, the algorithms in these applications did not generalize well since the test data tended to be from a single vehicle driving at a consistent speed on a fixed track with a high-end IMU. Therefore, VTTI focused on finding a more general solution that would work for a variety of speeds, vehicles, and less precise accelerometer data from a lower-end IMU such as the type present in the typical Android phones.

APPENDIX B: PUBLIC API DOCUMENTATION

VCC External Communication System

Virginia Tech Transportation Institute

Version 1.3, October 2017

Table of Contents

Chapter 1. VCC External Communication System

1.1. Overview

The Virginia Connected Corridor (VCC) project provides a REST interface which allows external applications to interact with the system. It is available here:

```
https://vcc-api.vtti.vt.edu
```

1.2. Running the Examples

There are several complete, working examples included in this documentation that show how to access the REST interface. To provide variety, some of these examples are written in Groovy and other examples are written in Python. Any language that supports HTTP requests can be used.

1.2.1. Groovy

To execute a Groovy example named test1.groovy, use the following command line:

```
groovy test1
```

Note that the .groovy extension is not included on the command line. The examples will automatically download needed libraries, if they have not yet been downloaded. This download will take some time, but will only happen once.

The Groovy examples use HTTPBuilder. More information about this library is available here: https://github.com/jgritman/httpbuilder

1.2.2. Python

The Python examples have been tested with Python version 2.7.10. They use a library, called Requests, that may not be installed by default, but can be easily installed. More information about the Requests library, including installation instructions, is available here: http://docs.python-requests.org/en/latest/

To execute a Python example named test1.py, use the following command line:

```
python test1.py
```

1.3. Authorization

Since the REST interface is stateless, it does not use login for authorization purposes. Instead, a special token is generated for each application/organization that is granted access to use the API. This token needs to be kept secret, similar to any other password. This token is sent in the Authorization header for each API request.

When using a client that allows the username and password to be specified, the application token is used for the username. The value provided for password will be ignored. Here is a partial Python example showing the use of the application token as a username for authorization purposes.

```
r = requests.get('https://vcc-api.vtti.vt.edu/api/rse/1',
                 auth=('MY_APP_TOKEN', 'password_is_ignored'))
```

When using a client that allows a header to be specified directly, the application token needs to be Base64 encoded. This Base64 encoding happens automatically when specifying the username and password, such as in the example above, and will produce an Authorization header similar the one produced in this example. The format for this header is:

```
Authorization: Basic THE_BASE64_ENCODED_APPLICATION_TOKEN
```

This partial Groovy example illustrates the Base64 encoding to produce a header of the appropriate format.

```
def http = new HTTPBuilder( 'https://vcc-api.vtti.vt.edu/' )
http.request(GET,JSON) { req ->
    uri.path = '/api/rse/2'
    headers.'Authorization' = 'Basic ' + 'MY_APP_TOKEN'.bytes.encodeBase64().toString
()
    ...
}
```

1.4. API

1.4.1. Basic Safety Messages (BSM)

Any Basic Safety Message (BSM) received by an RSE will be forwarded to the VCC system. These BSM are parsed and the information is made available through this API.

Table 1. The portion of the API dealing with Basic Safety Messages.

URI	Method	Description	ContentType
api/bsm	GET	List of current Basic Safety Messages (BSM). Each BSM contains a subset of commonly-used data such as location and speed. This list contains only one BSM per vehicle. It is always the most-recent BSM per vehicle. When a vehicle stops sending BSM, the last one will be considered current for about one second.	JSON
api/bsm/key	GET	Obtain a key for use with the BSM WebSocket. Each key can only be used once and will expire shortly after creation.	TEXT
ws/bsm?key=KEY&rse_id=RSEID[,RSEID,…]	WebSocket	A WebSocket periodically streaming the list of new Basic Safety Messages (BSM). Each BSM contains a subset of commonly-used data such as location and speed. The list of BSMs can optionally be limited to only the BSMs from one or more RSEs by providing a comma-separated list of RSEIDs. BSMS sent from a cell phone rather than through an RSE can be filtered using the value of "cell" in the list of RSEIDs. The key parameter is required and the value for it is obtained from the 'api/bsm/key' request.	Stream of JSON

Example 1. Retrieve information for current BSM.

This Groovy example retrieves the information for all current BSM at this moment.

getCurrentBsm.groovy

```
@Grab(group='org.codehaus.groovy.modules.http-builder',
      module='http-builder', version='0.5.0-RC2' )
import groovyx.net.http.HTTPBuilder
import static groovyx.net.http.Method.GET
import static groovyx.net.http.ContentType.TEXT
import static groovyx.net.http.ContentType.JSON

// initialze a new builder and give a default URL
def http = new HTTPBuilder( 'https://vcc-api.vtti.vt.edu/' )
def appToken = 'MY_APP_TOKEN'

http.request(GET,TEXT) { req ->
  uri.path = '/api/bsm'
  headers.'Authorization' = 'Basic ' + appToken.bytes.encodeBase64().toString()

  response.success = { resp, reader ->
    assert resp.status == 200
    println "Response status: ${resp.statusLine}"
    System.out << reader // print response reader
  }

  // called only for a 401 (not authorized) status code:
  response.'401' = { resp ->
    println 'Not authorized'
  }
}
```

The output contains the information about all current BSM in JSON format, as shown below.

```
Response status: HTTP/1.1 200 OK
[ {
  "rseID" : 11,
  "tempVehicleID" : "81",
  "timestamp" : 1439001653784,
  "latitude" : 38.123456,
  "longitude" : -77.123456,
  "speed" : 0.02,
  "heading" : 0.0,
  "elevation" : 74.0,
  "airTempDegC": null,
  "roadTempDegC": null,
  "brake" : 0
} ]
```

Example 2. Retrieve BSM continuously over WebSocket.

This Python example opens a WebSocket to continuously receive the information about BSM as they are generated. The VCC WebSocket currently broadcasts BSM information at 2 Hz.

Although the WebSocket specification has been updated to support headers, such as an Authorization header, most WebSocket libraries have not yet implemented that feature. To allow authorization of a WebSocket, this example first requests a key from 'api/bsm/key' and then uses that key as a parameter when creating the WebSocket. This example also includes the rse_id parameter to limit the returned BSMs to only those from RSEs 11 and 12, plus the BSMs from a cell phone. If the rse_id parameter is not included, all BSMs will be sent over the WebSocket.

getBsmWebSocket.py

```
import requests
from websocket import create_connection

r = requests.get('https://vcc-api.vtti.vt.edu/api/bsm/key',
                 auth=('MY_APP_TOKEN', 'password_is_ignored'))

print r.status_code
bsmKey = r.content

ws = create_connection("wss://vcc-api.vtti.vt.edu/ws/bsm?rse_id=cell,11,12&key=" +
bsmKey)
print "Receiving BSM from WebSocket..."
for i in range(0,10):
  result = ws.recv()
  print "%d: Received '%s'" % (i, result)
ws.close()
```

If this example is run at a time when no BSM are being generated the output will look like this. Since the BSM are broadcast at 2 Hz, there will be a 0.5 second delay between messages received from the WebSocket.

```
200
Receiving BSM from WebSocket...
0: Received '{ "bsm": [] }'
1: Received '{ "bsm": [] }'
2: Received '{ "bsm": [] }'
3: Received '{ "bsm": [] }'
4: Received '{ "bsm": [] }'
5: Received '{ "bsm": [] }'
6: Received '{ "bsm": [] }'
7: Received '{ "bsm": [] }'
8: Received '{ "bsm": [] }'
9: Received '{ "bsm": [] }'
```

When BSM are being broadcast, the output is like the following.

```
200
Receiving BSM from WebSocket...
0: Received '{ "bsm": [{
  "entityID" : "11",
  "latitude" : 37.1234567,
  "longitude" : -80.1234567,
  "heading" : 0.0,
  "speed" : 0.0,
  "airTempDegC": null,
  "roadTempDegC": null,
  "rseID" : 11,
  "brake" : 0,
  "timestamp" : 1438988681543
}, {
  "entityID" : "10",
  "latitude" : 37.1234567,
  "longitude" : -80.1234567,
  "heading" : 0.0,
  "speed" : 0.0,
  "airTempDegC": null,
  "roadTempDegC": null,
  "rseID" : 11,
  "brake" : 0,
  "timestamp" : 1438988681627
}] }'
1: Received '{ "bsm": [{
  "entityID" : "10000000",
  "latitude" : 37.1234567,
  "longitude" : -80.1234567,
  "heading" : 127.3,
  "speed" : 5.1,
  "airTempDegC": 10,
  "roadTempDegC": 11,
  "rseID" : "cell",
  "brake" : 0,
  "timestamp" : 1438988682149
}] }'

   ... output for iterations 2 - 9 has been removed.
```

1.4.2. Dynamic Message Signs (DMS)

There are many Dynamic Message Signs in Virginia. Information about each of these signs, including the location of the sign and the message being displayed is available through this API.

The messages sent to a DMS often contain formating information. The various formating codes are enclosed in square brackets, such as [n1].

Table 2. The portion of the API dealing with Dynamic Message Signs.

URI	Method	Description	ContentType
api/dms	GET	List of all Dynamic Message Signs (DMS).	JSON
api/dms/:id	GET	List of one or more Dynamic Message Signs matching the specified :id, where :id is either the integer ID of the desired DMS or a partial name, such as NOVA.	JSON
api/dms/count	GET	Count of all Dynamic Message Signs (DMS).	INT
api/dms/count/:name	GET	Count of all Dynamic Message Signs with a name that exactly matches or contains the provided :name, such as NOVA.	INT

Example 3. Retrieve information for a single DMS.

This Groovy example retrieves the information for a single DMS.

getOneDms.groovy

```
@Grab(group='org.codehaus.groovy.modules.http-builder',
      module='http-builder', version='0.5.0-RC2' )
import groovyx.net.http.HTTPBuilder
import static groovyx.net.http.Method.GET
import static groovyx.net.http.ContentType.TEXT
import static groovyx.net.http.ContentType.JSON

// initialze a new builder and give a default URL
def http = new HTTPBuilder( 'https://vcc-api.vtti.vt.edu/' )
def appToken = 'MY_APP_TOKEN'

http.request(GET,TEXT) { req ->
  uri.path = '/api/dms/1429'
  headers.'Authorization' = 'Basic ' + appToken.bytes.encodeBase64().toString()

  response.success = { resp, reader ->
    assert resp.status == 200
    println "Response status: ${resp.statusLine}"
    System.out << reader // print response reader
  }

  // called only for a 401 (not authorized) status code:
  response.'401' = { resp ->
    println 'Not authorized'
  }

  // called only for a 404 (not found) status code:
  response.'404' = { resp ->
    println 'Not found'
  }
}
```

The output contains the information about the requested DMS in JSON format, as shown below.

```
Response status: HTTP/1.1 200 OK
[ {
  "id" : 1429,
  "name" : "HamptonRoads-DMS-64-I10",
  "uniqueID" : "RE1TAAAAAAAC",
  "latitude" : 36.12345,
  "longitude" : -76.1234,
  "direction" : "in",
  "geoSection" : "HamptonRoads",
  "link" : "http://www.vdotdatasharing.org/xmldb/DmsSignStatus/HamptonRoads-DMS-64-I10",
  "location" : "EB I 64 AT INDIAN RIVER EB I 64 EXIT",
  "timRegionData" : null,
  "mileMarker" : 286.5,
  "signState" : "on",
  "message" : "[jl2]HOV-2[nl]ONLY[nl]"
} ]
```

1.4.3. MapData (MAP) Message

MapData messages contain information about an intersection, such as the location of various lanes within the intersection and which signal group controls the movement of each lane.

Table 3. The portion of the API dealing with MapData (MAP).

URI	Method	Description	ContentType
api/v2/mapData	GET	Get a list of MapData information for all supported intersections. The data for each intersection includes the id, revision, and the MapData message. The MapData message is an UPER-encoded MessageFrame message that contains a MapData message following the 2016 standard. This binary message is then base64 encoded and sent as a string.	JSON list
api/v2/mapData/:id	GET	Get MapData information for the one specified intersection. The information includes the id, revision, and the MapData message. The MapData message is an UPER-encoded MessageFrame message that contains a MapData message following the 2016 standard. This binary message is then base64 encoded and sent as a string.	JSON

1.4.4. Pothole Reports

Information on potholes is available through this API both for individual potholes and for clusters of potholes. Multiple pothole reports that are close to each other are grouped in a cluster.

Table 4. The portion of the API dealing with MapData (MAP).

URI	Method	Description	ContentType
api/pothole?status=[open|closed|all]&start_date=mm/dd/yyyy&end_date=mm/dd/yyyy	GET	Get a list of individual pothole reports. All of the parameters are optional and will default to returning all open potholes.	JSON list
api/pothole/cluster?status=[open|closed|all]&start_date=mm/dd/yyyy&end_date=mm/dd/yyyy	GET	Get a list of pothole cluster reports. All of the parameters are optional and will default to returning all open pothole clusters.	JSON list
api/pothole/:id	GET	Get information on the single pothole specified by id.	JSON
api/pothole/cluster/:id	GET	Get information on the single pothole cluster specified by id.	JSON

1.4.5. Roadside Equipment (RSE)

There is a variety of information about the Roadside Equipment (RSE) describing its location and other details. An RSE group is a collection of RSEs.

Table 5. The portion of the API dealing with Roadside Equipment.

URI	Method	Description	ContentType
api/rse?group_id=GROUPID	GET	List of all Roadside Equipment (RSE), optionally limiting the list to only the units within the specified group. If including the group_id parameter, GROUPID will be the integer ID of the desired group.	JSON list
api/rse/:id	GET	Info for one Roadside Equipment (RSE). The ':id' needs to be replaced by the integer ID of the desired RSE.	JSON
api/rse/group	GET	List of RSE groups.	JSON list

Example 4. Retrieve information for a single RSE.

This Python example retrieves the information for a single RSE. The results are returned as JSON.

getOneRse.py

```
import requests

r = requests.get('https://vcc-api.vtti.vt.edu/api/rse/11',
                 auth=('MY_APP_TOKEN', 'password_is_ignored'))

print r.status_code
print r.content
```

If this example is executed exactly as listed, including leaving the token as 'MY_APP_TOKEN', the authorization will fail because 'MY_APP_TOKEN' is not a valid token. The output is as follows:

```
401
Unauthorized.
```

After replacing 'MY_APP_TOKEN' with a valid token, the output will be similar to the following:

```
200
{
  "id" : 11,
  "name" : "Cab533Nutley",
  "roadway" : "I-66",
  "mileMarker" : 62.5,
  "latitude" : 38.123456,
  "longitude" : -77.123456,
  "rseGroupID" : 2,
}
```

Example 5. Retrieve information for a group of RSE.

This Groovy example retrieves the information for all active RSEs in group 2 (the Northern Virginia group).

getActiveRses.groovy

```
@Grab(group='org.codehaus.groovy.modules.http-builder',
      module='http-builder', version='0.5.0-RC2' )
import groovyx.net.http.HTTPBuilder
import static groovyx.net.http.Method.GET
import static groovyx.net.http.ContentType.TEXT
import static groovyx.net.http.ContentType.JSON

// initialze a new builder and give a default URL
def http = new HTTPBuilder( 'https://vcc-api.vtti.vt.edu/' )
def appToken = 'MY_APP_TOKEN'

http.request(GET,TEXT) { req ->
  uri.path = '/api/rse'
  uri.query = [ group_id: '2' ]
  headers.'Authorization' = 'Basic ' + appToken.bytes.encodeBase64().toString()

  response.success = { resp, reader ->
    assert resp.status == 200
    println "Response status: ${resp.statusLine}"
    System.out << reader // print response reader
  }

  // called only for a 401 (not authorized) status code:
  response.'401' = { resp ->
    println 'Not authorized'
  }

  // called only for a 404 (not found) status code:
  response.'404' = { resp ->
    println 'Not found'
  }
}
```

Using a ContentType of TEXT in the above http.request() method produces output that is nice for humans to read. Switching to a ContentType of JSON produces output that is nice for processsing.

The output is a long list of all of the active NoVA RSEs. A portion of the output is shown below:

```
Response status: HTTP/1.1 200 OK
[ {
  "id" : 8,
  "name" : "MM601ChainBridgeRd",
  "roadway" : "I-66",
  "mileMarker" : 60.1,
  "latitude" : 38.123456,
  "longitude" : -77.123456,
  "rseGroupID" : 2,
}, {
  "id": 9,
  "name": "MM609Blake-ChBr",
  "roadway": "I-66",
  "mileMarker": 60.9,
  "latitude": 38.123456,
  "longitude": -77.123456,
  "rseGroupID": 2
}, {

   ... many entries removed...

}]
```

1.4.6. Signal Phase And Timing (SPAT) Message

Signal Phase And Timing (SPAT) messages from a few intersections are being forwarded to the VCC system and are available through this API.

Table 6. The portion of the API dealing with Signal Phase And Timing (SPAT).

URI	Method	Description	ContentType
api/v2/spat	GET	Get a list containing the latest SPAT for each intersection. Each SPAT is an UPER-encoded MessageFrame message that contains a SPAT message following the 2016 standard. This binary message is then base64 encoded and sent as a string.	JSON list
api/v2/spat/all or api/v2/spat/:id	GET	Obtain a URL with an embedded key that can be used to open a WebSocket to receive SPAT change messages. Specifying an integer intersectionID will limit the SPAT sent over the WebSocket to only the SPAT for that one intersection. Using the keyword 'all' will request that the SPAT for all intersections be sent over the WebSocket.	JSON

URI	Method	Description	ContentType
ws/spat?key=KEY	WebSocket	This is the URL returned by the above request. It is not necessary to hardcode this URL since it will be returned in a ready-to-use format. Each returned url can only be used once and the key will expire shortly after creation. This opens a WebSocket that will stream SPAT messages as they are received from RSUs, but only if the new SPAT message is different from the previous SPAT message for a particular intersection. The SPAT that is sent over the WebSocket will, optionally, be limited to the requested intersection. Each SPAT message is an UPER-encoded MessageFrame message that contains a SPAT following the 2016 standard. This binary message is then base64 encoded and sent as a string.	Stream of Strings

Example 6. Receive SPAT continuously over WebSocket.

This Python example opens a WebSocket to receive SPAT messages for all intersections. Only SPAT messages that indicate a state change, i.e. a SPAT that contains an updated revision ID are sent on the WebSocket.

Although the WebSocket specification has been updated to support headers, such as an Authorization header, most WebSocket libraries have not yet implemented that feature. To allow authorization of a WebSocket, this example first requests a url that contains a key and then uses that url when creating the WebSocket. This example waits for 5 SPAT messages and then stops.

getSpatWebSocket.py

```
import json
import requests
from websocket import create_connection

r = requests.get('https://vcc-api.vtti.vt.edu/api/v2/spat/all',
                 auth=('MY_APP_TOKEN', 'password_is_ignored'))

print(r.status_code)
spatUrlMap = r.content
spatJson = json.loads(spatUrlMap)
url = spatJson["url"]

ws = create_connection("wss://vcc-api.vtti.vt.edu" + url)
print("Receiving Spat from WebSocket...")
for i in range(0,5):
  result = ws.recv()
  print("%d: Received '%s'" % (i, result))
ws.close()
```

Assuming SPAT are being received and are therefore available for broadcast, the output will be similar to the following.

```
200
Receiving Spat from WebSocket...
0: Received
'ABNFAAAHfnoAAAcAEEND2MvYyAECKh63XrdADBDQ90r3SgCAhofDF8MQBQR0PVo9WgAwIaHrJeskAcEND
42vjaAQCGh7pXul'
1: Received
'ABNFAAAHfnsAAAcAEEND2MvYyAECKh63XrdADBDQ90r3SgCAhofDF8MQBQQ0Pny+fIAwIaHrJeskAcEND
42vjaAQCGh7pXul'
2: Received
'ABNFAAAHfnwAAAcAEEND2MvYyAECKh63XrdADBDQ90r3SgCAhofDF8MQBQQ0Pny+fIAwIqHtBe2kAcEND
42vjaAQCGh7pXul'
3: Received
'ABNFAAAHfn0AAAcAEEND2MvYyAECOh7BHsEADBDQ90r3SgCAhofDF8MQBQQ0Pny+fIAwIqHtBe2kAcEND
42vjaAQCGh7pXul'
4: Received
'ABNFAAAHfn4AAAcAEEND2MvYyAECGh8+Xz5ADBDQ90r3SgCAhofDF8MQBQQ0Pny+fIAwIqHtBe2kAcEND
42vjaAQCGh7pXul'
```

1.4.7. Traveler Information Messages

The VCC system provides access to the Traveler Information Messages (TIM). Most of the TIMs in the system are generated for the messages being displayed on the Dynamic Message Signs. A TIM will always have a start date and may have an optional stop date. The timData for a TIM uses the SAE standardized format.

Table 7. The portion of the API dealing with Traveler Information Messages.

URI	Method	Description	ContentType
api/tim	GET	List of Traveler Information Messages (TIM).	JSON
api/tim/:id	GET	Info for one Traveler Information Message (TIM).	JSON
api/tim/:id/data?format=[base64 | hex | json]	GET	The TIM data for the specified TIM in the requested format, which defaults to JSON if not explicitly provided.	requested format
api/tim/:id/uploads?rse_id=RSEID& start_date=STARTDATE& end_date=ENDDATE	GET	List of TIM uploads for the specified TIM, optionally limited to uploads to the specified RSEID. If a date range is specified, all activity within that date range will be returned. Without a date range, only the latest upload for each requested TIM/RSE combination will be returned. A date is specified in month/day/year format, such as 8/13/15 or 08/13/2015.	JSON
api/tim/uploads?rse_id=RSEID& start_date=STARTDATE& end_date=ENDDATE	GET	List of TIM uploads, optionally limited to uploads to the specified RSEID. If a date range is specified, all activity within that date range will be returned. Without a date range, only the latest upload for each requested TIM/RSE combination will be returned. A date is specified in month/day/year format, such as 8/13/15 or 08/13/2015.	JSON

Example 7. Retrieve information for a single TIM.

This Python example retrieves the information for a single TIM.

getOneTimPlusData.py

```
import requests

r = requests.get('https://vcc-api.vtti.vt.edu/api/tim/26',
                 auth=('MY_APP_TOKEN', 'password_is_ignored'))

print "Status code for get tim 26 = %d" % r.status_code
print r.content

r = requests.get('https://vcc-api.vtti.vt.edu/api/tim/26/data',
                 auth=('MY_APP_TOKEN', 'password_is_ignored'))

print "Status code for get tim 26 data = %d" % r.status_code
print r.content
```

The output for this program provides the information about the TIM with ID = 26, followed by a JSON representation of the timData for that TIM.


```
Status code for get tim 26 = 200
[ {
  "id" : 26,
  "uniqueID" : "RE1TAAAAAADG",
  "timData" :
"MIGAgAEQgQlETVMAAAAAAMaDAQGkaTBngAEBoQSAAgAAggIH34MDA7byhAJ9AIUBAqklMCOAAjwAgQF/o
hqhGKAQgAQXL3n+gQTR/L8EggIAAKEEgAIfQKoloCMwIaAfgR1bcHQyMG8wXVtqcDNdW2psM11GUk9OVCB
ST1lBTIUCO3c=",
  "message" : "[pt20o0][jp3][jl3]FRONT ROYAL",
  "startDate" : 1434698520000,
  "stopDate" : null,
  "crc" : "O3c="
} ]
Status code for get tim 26 data = 200
{
   "msgID": "travelerInformation",
   "packetID": "444D530000000000C6",
   "dataFrameCount": 1,
   "dataFrames": [
      {
         "frameType": "advisory",
         "msgId": {
            "furtherInfoID": "0000"
         },
         "startYear": 2015,
         "startTime": 243442,
         "duratonTime": 32000,
         "priority": 2,
         "regions": [
            {
               "direction": "3C00",
```

```
                "extent": "forever",
                "area": {
                   "circle": {
                      "center": {
                         "lat": 381234567,
                         "long": -771234567,
                         "elevation": "0000"
                      },
                      "r": {
                         "radiusSteps": 8000
                      }
                   }
                }
             }
          ],
          "content": {
             "advisory": [
                {
                   "item": {
                      "text": "[pt20o0][jp3][jl3]FRONT ROYAL"
                   }
                }
             ]
          }
       }
    ],
    "crc": "3B77"
}
```

Example 8. Retrieve information on TIM uploads for a particular RSE.

This Groovy example determines which TIMs have been uploaded to RSE 24.

getTimsOnRse24.groovy

```
@Grab(group='org.codehaus.groovy.modules.http-builder',
      module='http-builder', version='0.5.0-RC2' )
import groovyx.net.http.HTTPBuilder
import static groovyx.net.http.Method.GET
import static groovyx.net.http.ContentType.TEXT
import static groovyx.net.http.ContentType.JSON

// initialze a new builder and give a default URL
def http = new HTTPBuilder( 'https://vcc-api.vtti.vt.edu/' )
def appToken = 'MY_APP_TOKEN'

http.request(GET,TEXT) { req ->
  uri.path = '/api/tim/uploads'
  uri.query = [ rse_id: '24' ]
  headers.'Authorization' = 'Basic ' + appToken.bytes.encodeBase64().toString()

  response.success = { resp, reader ->
    assert resp.status == 200
    println "Response status: ${resp.statusLine}"
    System.out << reader // print response reader
  }

  // called only for a 401 (not authorized) status code:
  response.'401' = { resp ->
    println 'Not authorized'
  }

  // called only for a 404 (not found) status code:
  response.'404' = { resp ->
    println 'Not found'
  }
}
```

The output from this example shows two TIM upload entries for RSE 24.

```
Response status: HTTP/1.1 200 OK
[ {
  "id" : 300,
  "timID" : 86,
  "rseID" : 24,
  "loadDate" : 1431601525000,
  "status" : 1,
  "crc" : null
}, {
  "id" : 178,
  "timID" : 62,
  "rseID" : 24,
  "loadDate" : 1439216073000,
  "status" : 5,
  "crc" : "X2w="
} ]
```

Example 9. Retrieve information on TIM upload activity for a particular RSE and date range.

This Python example retrieves information on all TIM uploads to RSE 24 between 8/10/15 and 8/11/15.

getTimsOnRse24.py

```
import requests

r = requests.get('https://vcc-api.vtti.vt.edu/api/tim/uploads?' +
                 'rse_id=24&start_date=8/10/15&end_date=8/11/15',
                 auth=('MY_APP_TOKEN', 'password_is_ignored'))

print r.status_code
print r.content
```

The output from this program shows that three TIMs were uploaded to RSE 24 during that period. The timID was the same for each of the uploads, but the message was different. This is a common situation that happens when the message on a Dynamic Message Sign changes. The newest entry is listed on the bottom. A message about a disabled vehicle was initially uploaded and then later deleted. A second message about roadwork on I-66 W was then uploaded. The third message upload about the accident replaced the second message. That third message was later deleted.

An activity of 1 indicates Upload and an activity of 2 indicates Delete. A status of 0 indicates a successful upload. A status of 5 indicates a successful delete. Other values indicate failure.

```
200
[ {
  "id" : 137652,
```

```
  "timID" : 62,
  "rseID" : 24,
  "activity" : 2,
  "activityDate" : 1439216073000,
  "timData" : null,
  "status" : 5,
  "message" : null,
  "crc" : "X2w="
}, {
  "id" : 137587,
  "timID" : 62,
  "rseID" : 24,
  "activity" : 1,
  "activityDate" : 1439214280000,
  "timData" : null,
  "status" : 0,
  "message" : "[pt25o5][jp3][jl3][fo1]ACCIDENT[fo][nl1][jl3][fo1]MM
46[fo][np][pt25o5][jp3][jl3][fo1]RIGHT CENTER LANE[fo][nl1][jl3][fo1]BLOCKED[fo]",
  "crc" : "pmg="
}, {
  "id" : 137303,
  "timID" : 62,
  "rseID" : 24,
  "activity" : 1,
  "activityDate" : 1439206373000,
  "timData" : null,
  "status" : 0,
  "message" : "[pt25o0][jp3][jl3][fo1]ROADWORK I-66W[fo][nl1][jl3][fo1]MM 62 TO
60[fo][nl1][jl3][fo1]RIGHT LANE BLKD[fo]",
  "crc" : "3pc="
}, {
  "id" : 136476,
  "timID" : 62,
  "rseID" : 24,
  "activity" : 2,
  "activityDate" : 1439166283000,
  "timData" : null,
  "status" : 5,
  "message" : null,
  "crc" : "X2w="
}, {
  "id" : 136461,
  "timID" : 62,
  "rseID" : 24,
  "activity" : 1,
  "activityDate" : 1439165802000,
  "timData" : null,
  "status" : 0,
  "message" : "[pt25o0][jp3][jl3][fo1]DISABLED VEH[fo][nl1][jl3][fo1]MM
48[fo][nl1][jl3][fo1]RIGHT CNT LN BLKD[fo]",
  "crc" : "bkg="
```

```
} ]
```

APPENDIX C: FINAL PRESENTATION AND SOFTWARE TRAINING

Development of Prototype On-Board Unit for Connected Vehicle Initiatives

Virginia Tech Transportation Institute

Final Presentation

October 3, 2017

Dr. Michael A. Mollenhauer
Director
Center for Technology Implementation
Virginia Tech Transportation Institute

Agenda

- Project Schedule
- Project Objectives and Scope
- VCC Overview
- Discussion of Tasks A – E
- Software Training

Schedule

Task	Months after award	Original Deadline	Adjusted Deadline
Task A – Evaluate DDOT Infrastructure	2	3/15/17	3/15/17
Task B – Develop OBE Equipment	5	6/15/17	7/15/17
Task C – Install OBUs	6	7/15/17	8/15/17
Task D – Provide Monitoring Software	9	10/15/17	10/15/17
Task E – Support Data Analytics	9	10/15/17	10/15/17
Summary/Final Report Draft			10/15/17
Final Report Complete			11/11/17

Development of Prototype OBU for CV Initiatives Project

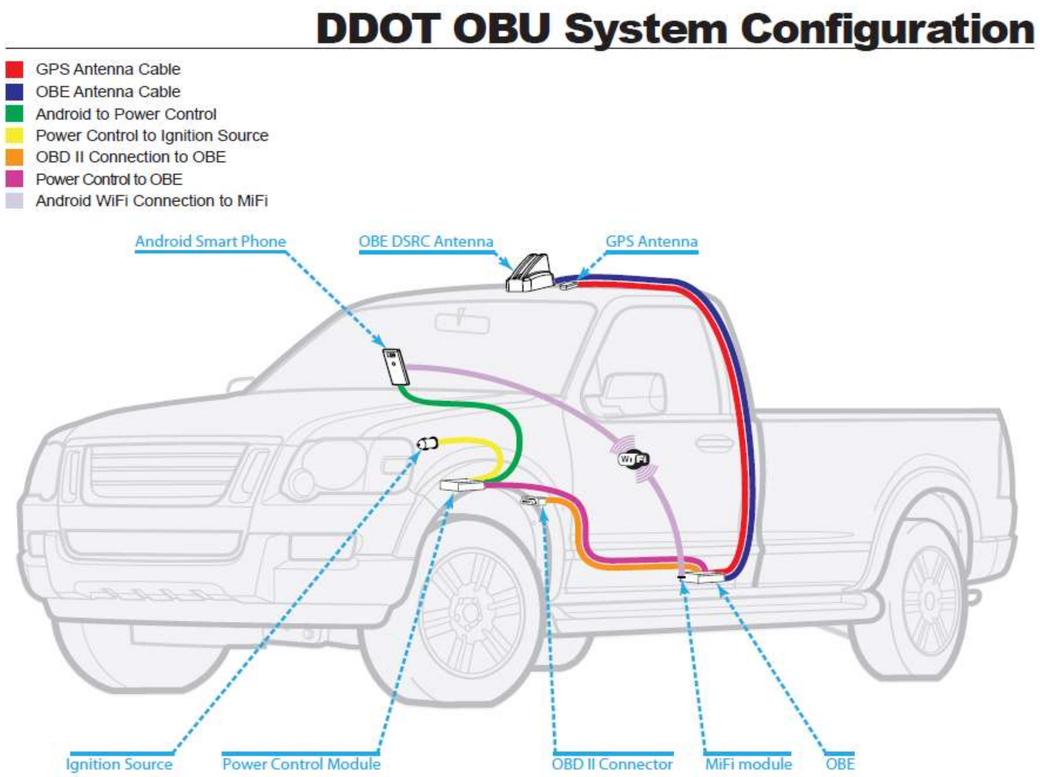
- Project Objectives
  - To develop and deploy connected vehicle (CV) infrastructure in the District
    - Leverage technology to provide a safer and more sustainable transportation system
    - Improve efficiency of identifying and prioritizing maintenance on capital assets

Task A – Evaluate DDOT Infrastructure

- Review and inspect existing systems
- Evaluate adherence to standards and ability to upgrade to support future V2I goals
- Create report that assesses DDOT's ability to deploy V2I systems
- Synchronize work and results to DDOT Cybersecurity Assessment project to extent possible

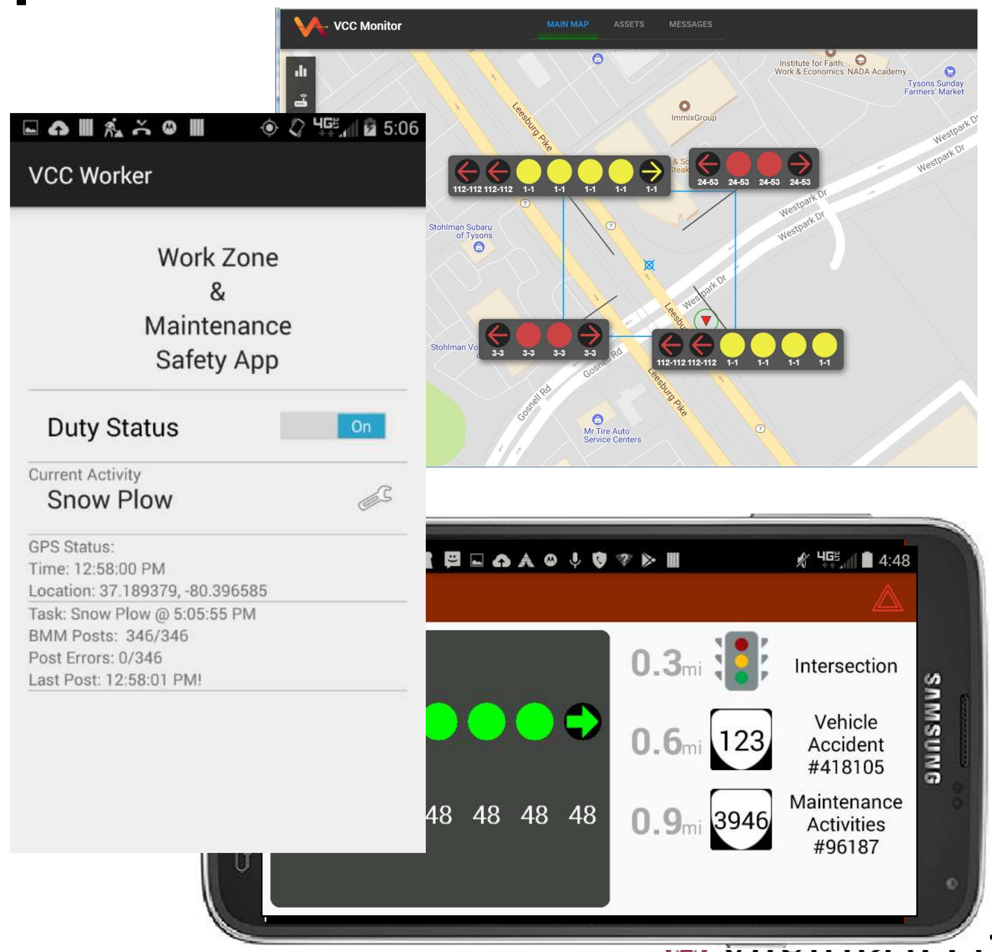
Tasks B & C
Develop and Install OBE Equipment

- Integrate an OBE package for 5 DDOT service vehicles
  - DSRC radio
  - Cellular MiFi
  - OBDII
  - Power control
  - GPS antenna
- Android cell phone
  - VCC Mobile App
  - Pothole detector
  - Weather sensor



Task D - Provide Application Software

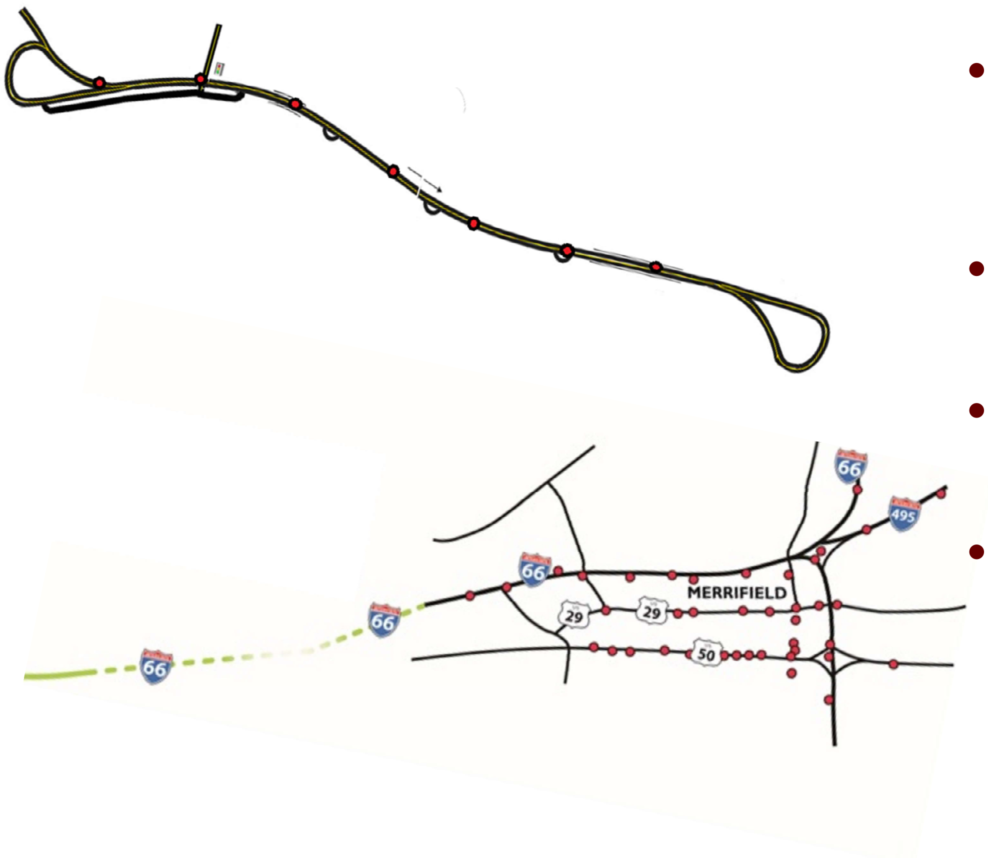
- VCC Monitor
  - Web application
  - Health and status monitoring
  - Data flow and quantification
  - Manual TIM generation
  - Pothole locations
  - Temperature data
- VCC Mobile
  - Pothole detection
- VCC Worker App
  - Registers vehicle and task
- VTTI provide onsite training for software tools



Task E – Support Data Analytics

- Assess DDOT's data analytics aspirations
- Provide access to real-time data from DDOT vehicles
  - Speeds
  - Locations
  - Weather conditions
- Data retrieval interface from archived data
- Light integration of general purpose data analytics tools

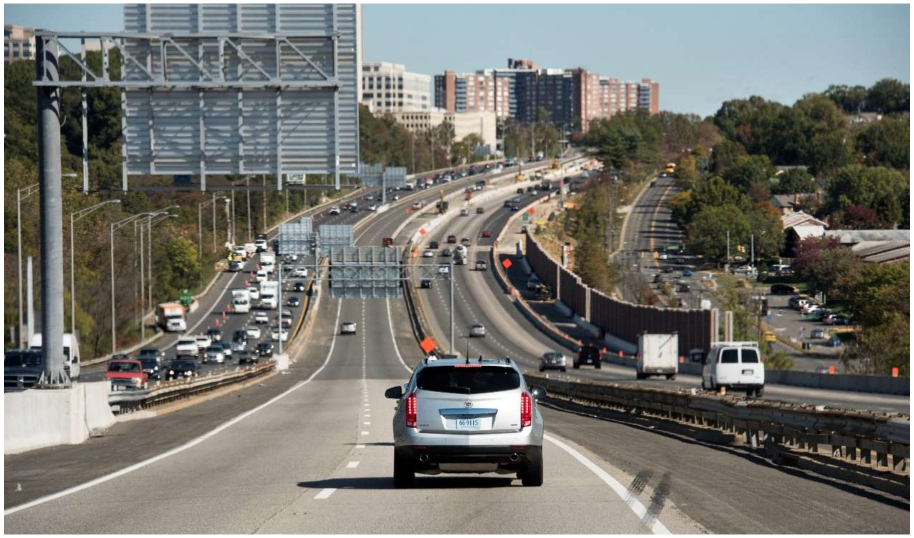
VCC Overview

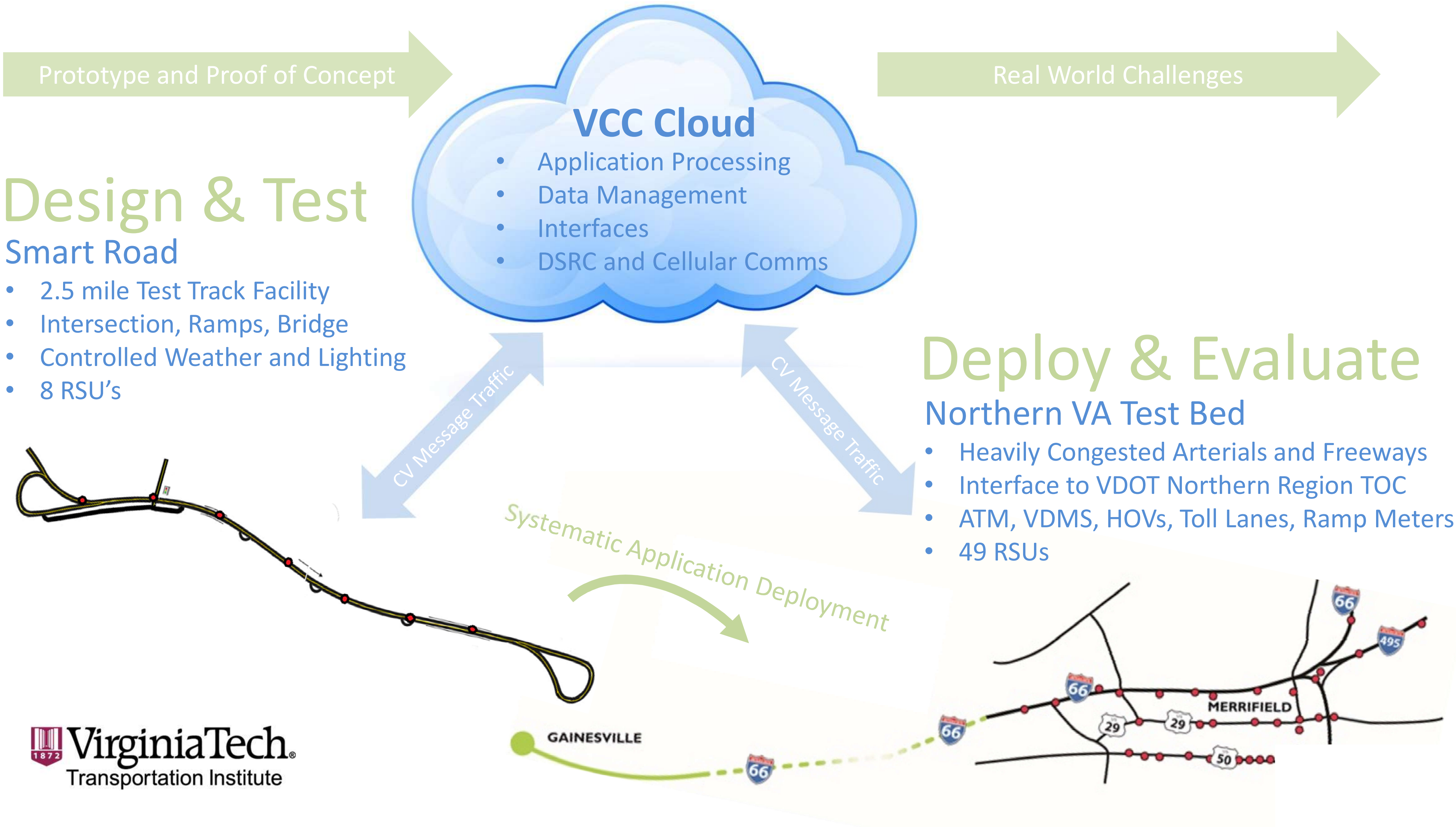


- Open CV test and development environment in Northern Virginia and Blacksburg, Virginia (Smart Road)
- Over 60 RSUs connected via DSRC and cellular technology
- Partnership between VDOT, VTRC, VTTI, and other VA research institutes
- VDOT has allowed DDOT the use of relevant portions of their backend system during this project

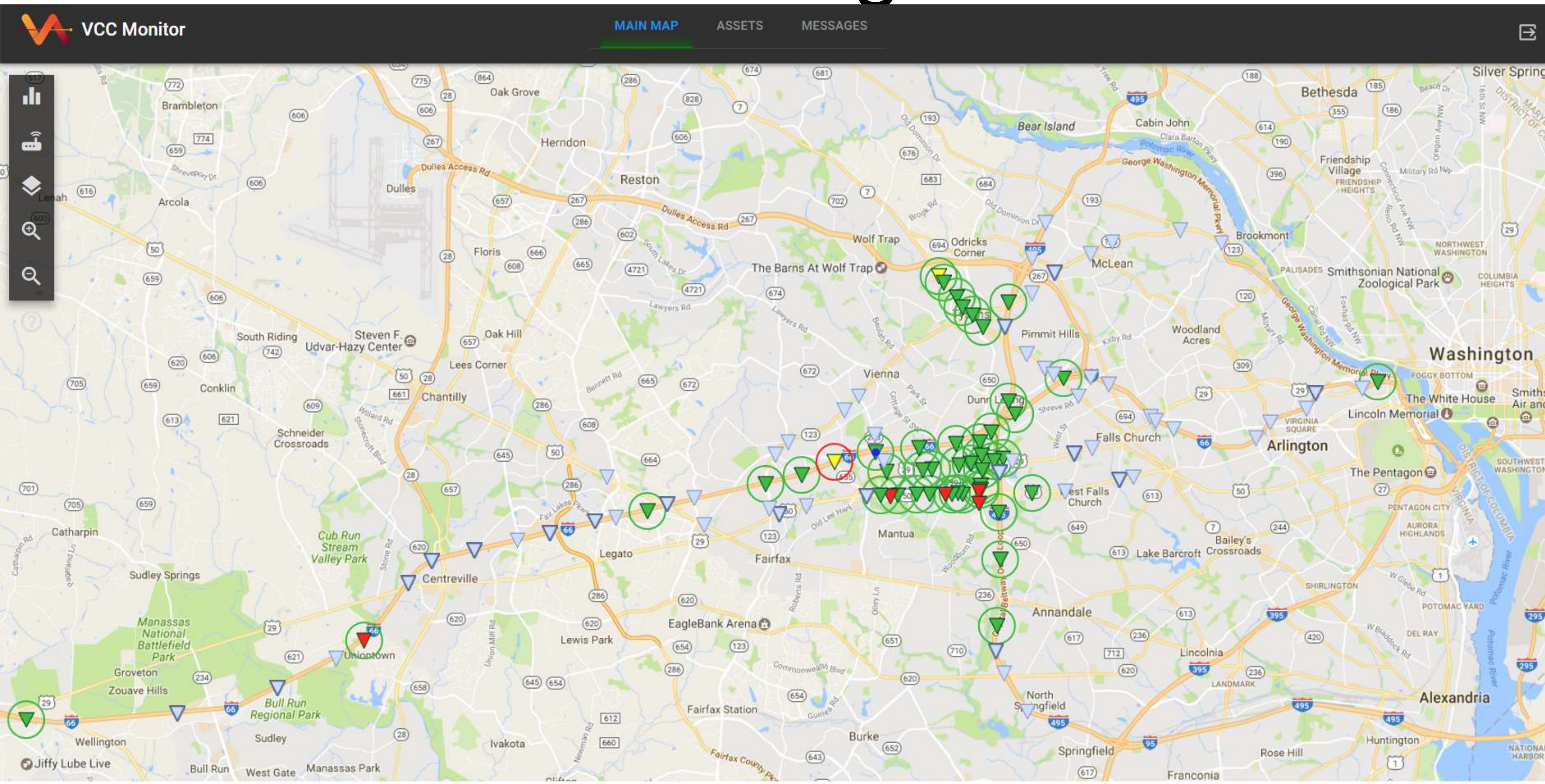
Purpose of VCC

- Leverage CV technology to improve mobility and safety
- Enhance the operation of the nation's roadways to create a better experience for the traveling public
- Enable interaction of automated vehicles with infrastructure and other vehicles to maximize understanding of the environment and minimize crash risk




Prototype and Proof of Concept
Real World Challenges
VCC Cloud
Application Processing
Data Management
Interfaces
DSRC and Cellular Comms
Design & Test
Smart Road
2.5 mile Test Track Facility
Intersection, Ramps, Bridge
Controlled Weather and Lighting
8 RSU's
CV Message Traffic
CV Message Traffic
Systematic Application Deployment
Deploy & Evaluate
Northern VA Test Bed
Heavily Congested Arterials and Freeways
Interface to VDOT Northern Region TOC
ATM, VDMS, HOVs, Toll Lanes, Ramp Meters
49 RSUs
GAINESVILLE
MERRIFIELD
66
495
29
50
VirginiaTech
Transportation Institute

VCC Northern Virginia Testbed



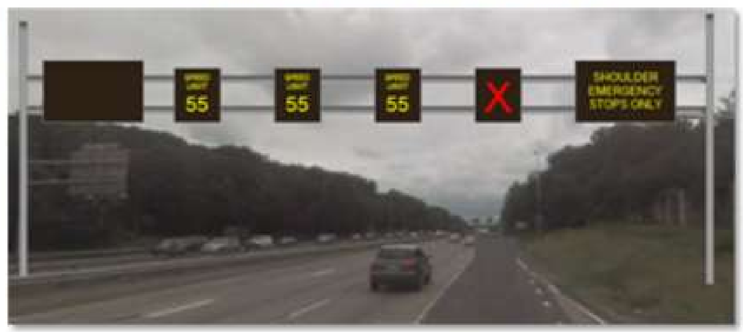
VCC Overview

ATM System Integration

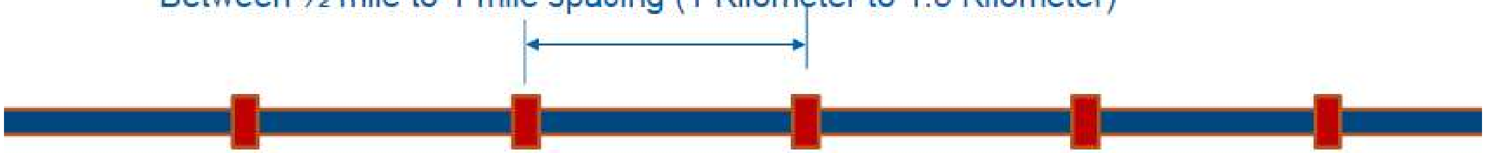
- Exploring opportunities to integrate ATM signage into VCC Mobile app
- Contrast status displays versus notifications
- Opportunities to reduce infrastructure in future

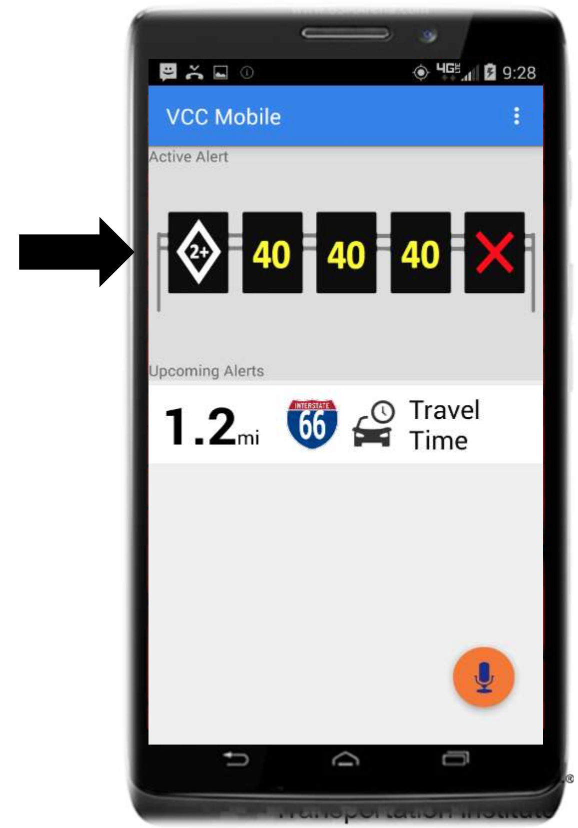
Example with HOV Lane and Shoulder Running

Example with HOV Lane, No Shoulder Running

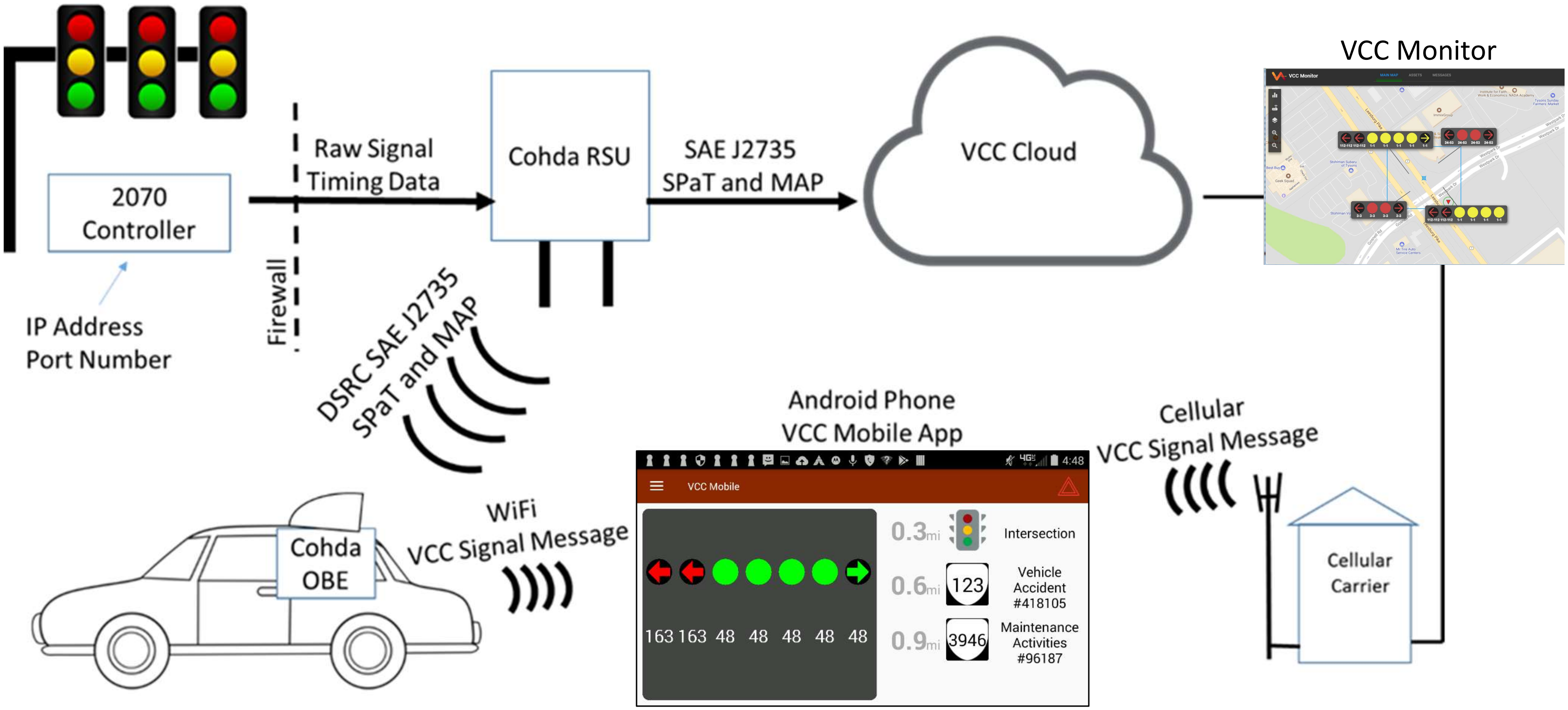


Between ½ mile to 1 mile spacing (1 Kilometer to 1.5 Kilometer)





VCC Overview
SPaT Implementation



Advancing Transportation Through Innovation

Virginia Tech Transportation Institute

Task A – Evaluate DDOT Infrastructure
Key Findings

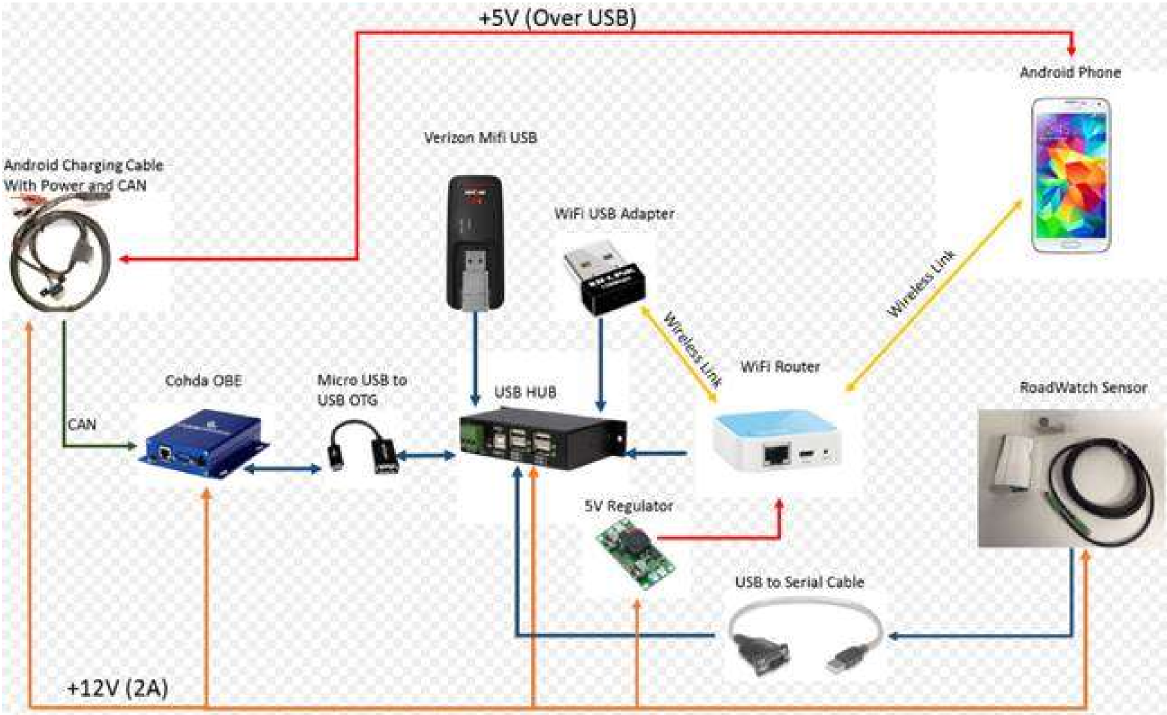
- CVI Deployment Needs
  - Infrastructure
    - Upgrade ITS infrastructure
      - DSRC RSUs
      - Upgrade/modify signal controllers to incorporate SPaT
    - Upgrade backhaul communication network
      - Fiber optic
      - Wireless (DSRC and/or cellular)
    - Relocate TMC building to new facility

Task A – Evaluate DDOT Infrastructure
Key Findings

- CVI Deployment Needs, continued
  - Software
    - Upgrade RSE and traffic management software
  - Security
    - Physical infrastructures (RSUs, signal controllers)
    - Communication infrastructures (DSRC)

Task B – Develop OBE Equipment

System Overview



- Cohda Wireless MK5 OBU
- 4G LTE Global USB Modem U620L MiFi
  - DDOT shipped to VTTI; all arrived by 7/26
- Samsung Galaxy S5 phone
- OBDII Connection to OBE
- Power control to OBE

Task B – Develop OBE Equipment

Weather Sensor

- M.S. Foster RoadWatch Weather Sensor
  - Measures and displays surface temperature and ambient temperature
  - Accuracy of +/- 2 degrees
  - Integrated with OBE and data collection and monitoring package

Task C – Install Equipment

- Met with DDOT to finalize installation plan
  - hardware mounting and wiring locations
- First truck installed 7/13
- Remaining four trucks installed 8/7-8/8 once remaining MiFis were received at VTTI

Task C – Install Equipment

Rear interior wall of truck cab

Task D – Provide Application Software

- VCC Monitor
- VCC Mobile
- Pothole Detection App
- VCC Worker App

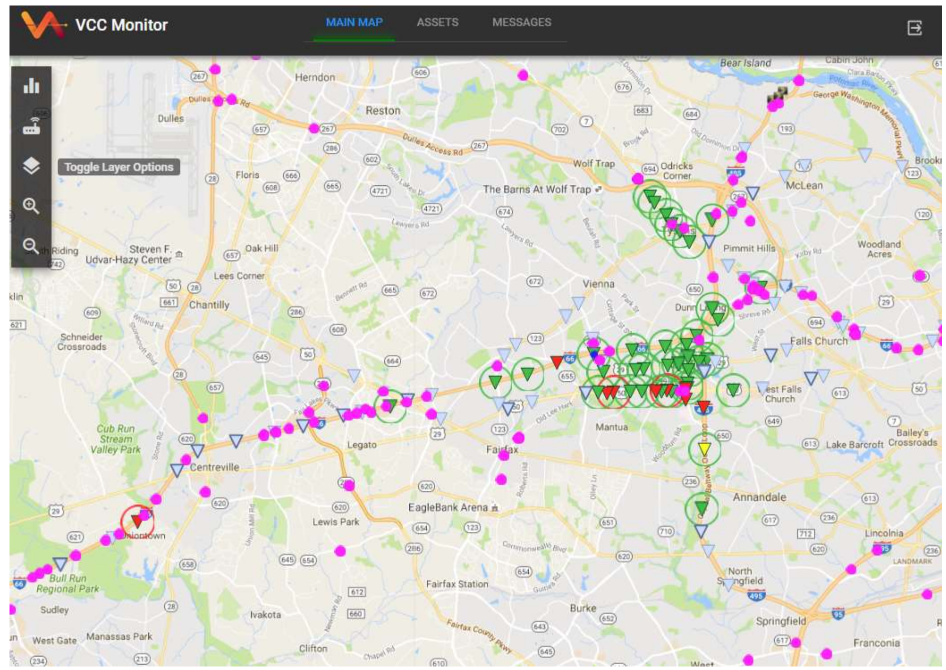
VCC Monitor Training

Prepared August 14, 2017

Virginia Tech
1872
TRANSPORTATION INSTITUTE

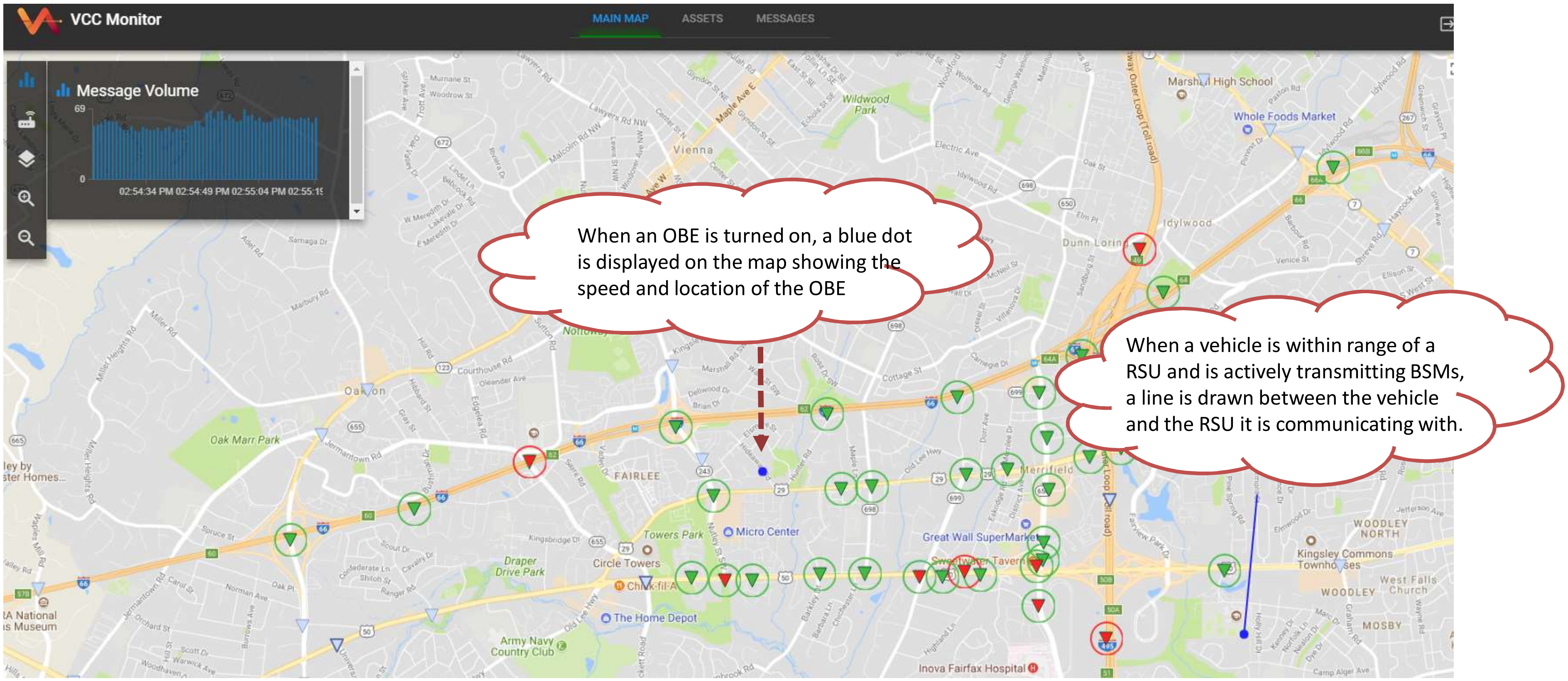
VCC Monitor Overview

- Situational awareness tool
- Real-time information display
  - Weather sensor information
  - Pothole detector information
- RSU status and performance monitoring
- Message flow monitoring and management (BSM, BMM, PDM, TIM, etc.)
- Driver Report Location



Main Map

Real-time Map Display

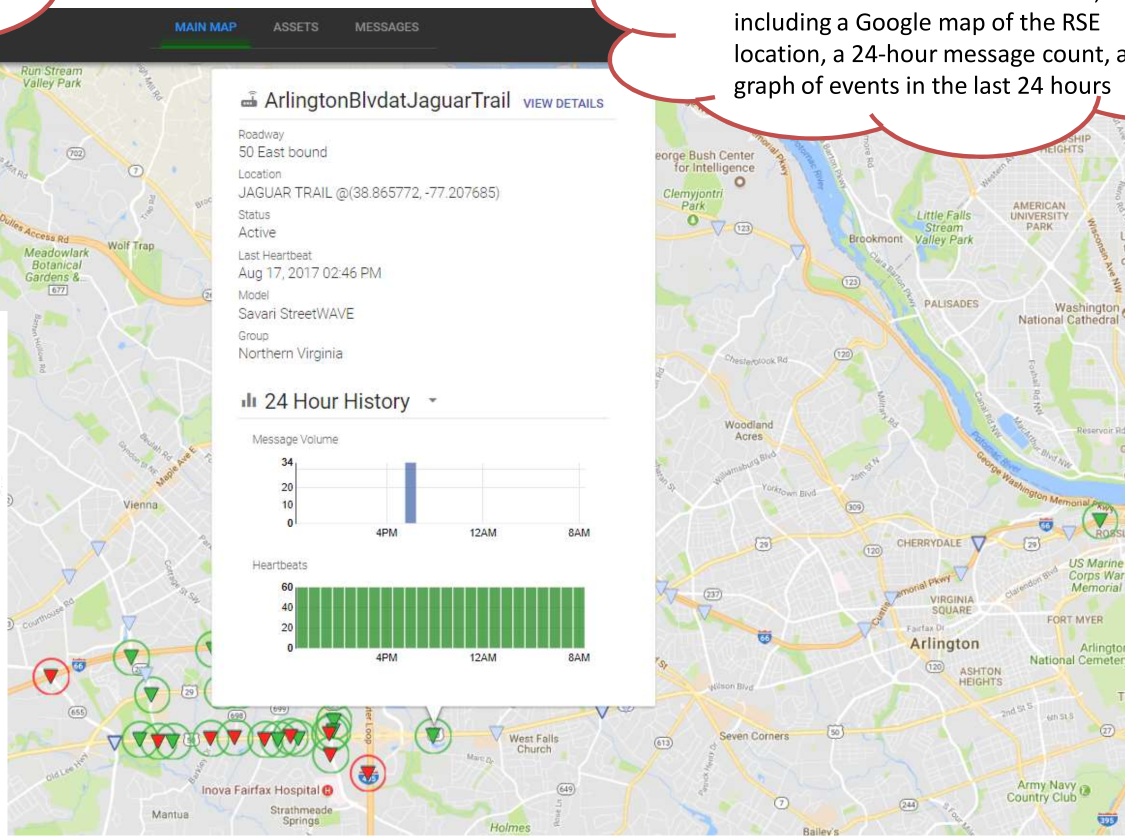


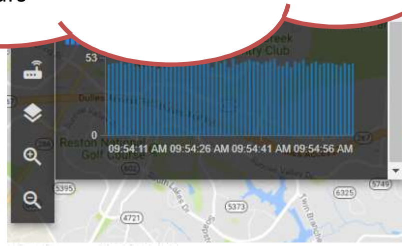
Main Map

RSE Information

When you click on a RSE icon, additional information appears including the roadway and GPS coordinates, the status, the time of the last heartbeat, the model, and a graph of the heartbeats over the last 24 hours

When you click on "View Details", you will be redirected to the Assets – RSU page which will show additional details, including a Google map of the RSE location, a 24-hour message count, and a graph of events in the last 24 hours

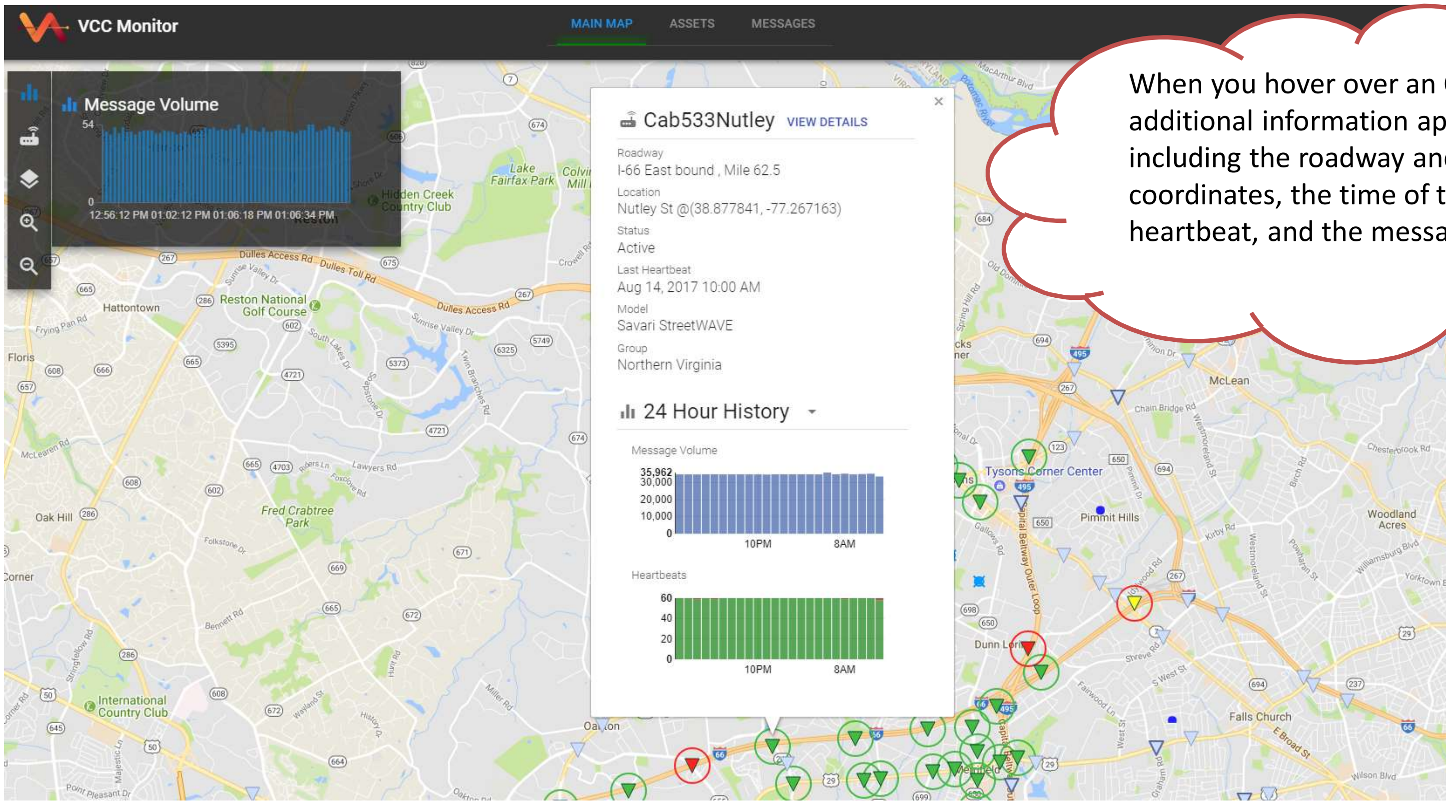




Symbol	Definition
	RSE is communicating reliably
	RSE has communicated reliably in the last 10 minutes, but has had some problems in the last 24 hours
	Caution – RSE has had a few problems communicating
	Warning – RSE is having problems communicating
	Future deployment

Main Map

Vehicle Information

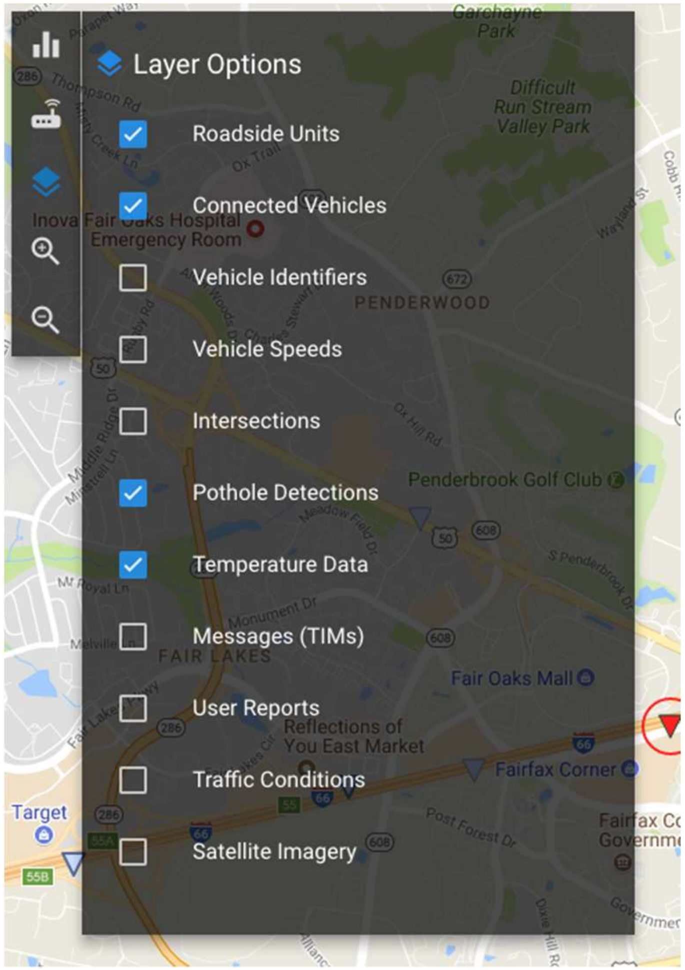


When you hover over an OBE, additional information appears including the roadway and GPS coordinates, the time of the last heartbeat, and the message volume

Main Map

Toggle Layer Options

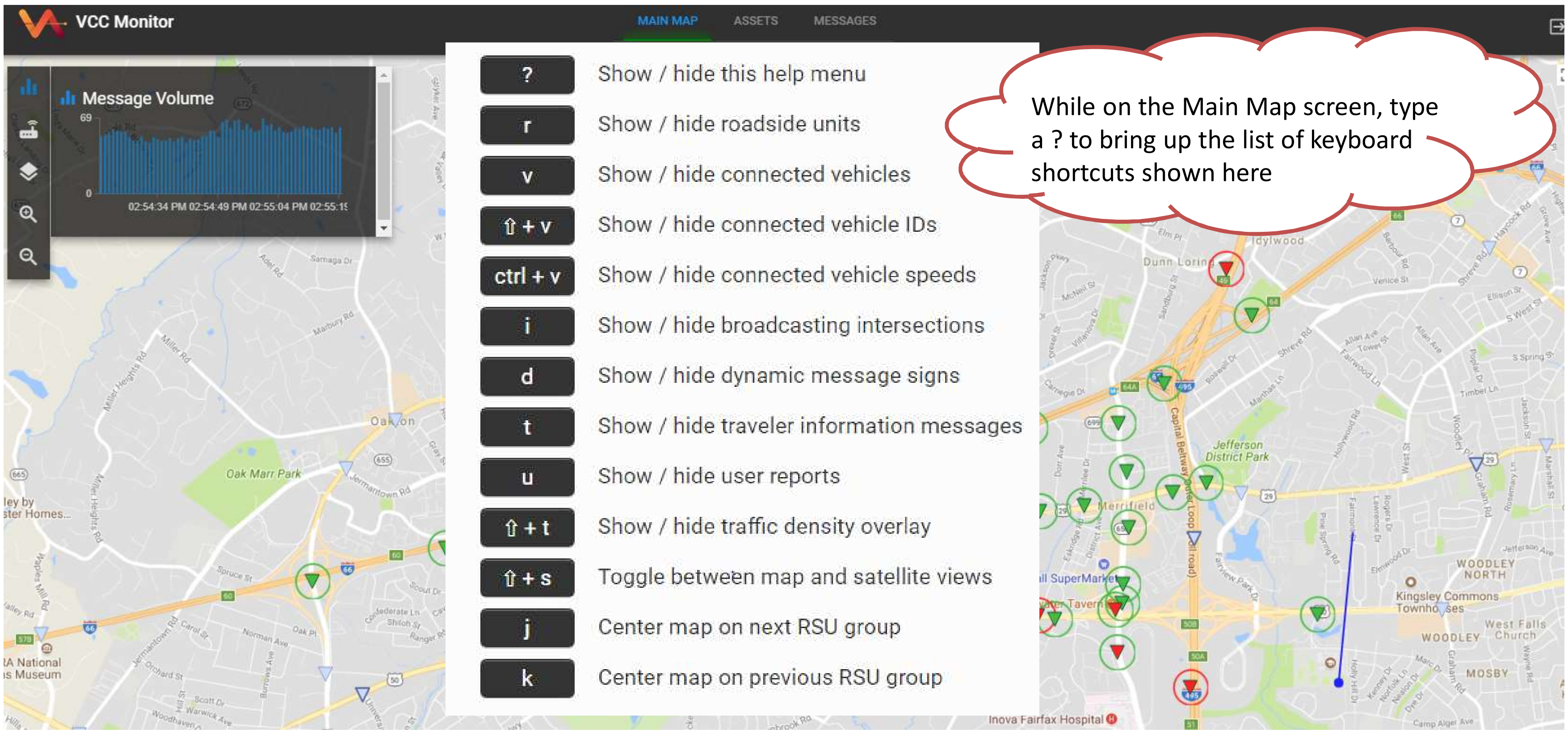
- In the upper left-hand corner of the main map screen, there are a number of controls, including toggle layer options
  - The user can toggle on and off the layers of interest, including “Pothole Detections” and “Temperature Data”



VirginiaTech Transportation Institute

Main Map

Keyboard Shortcuts

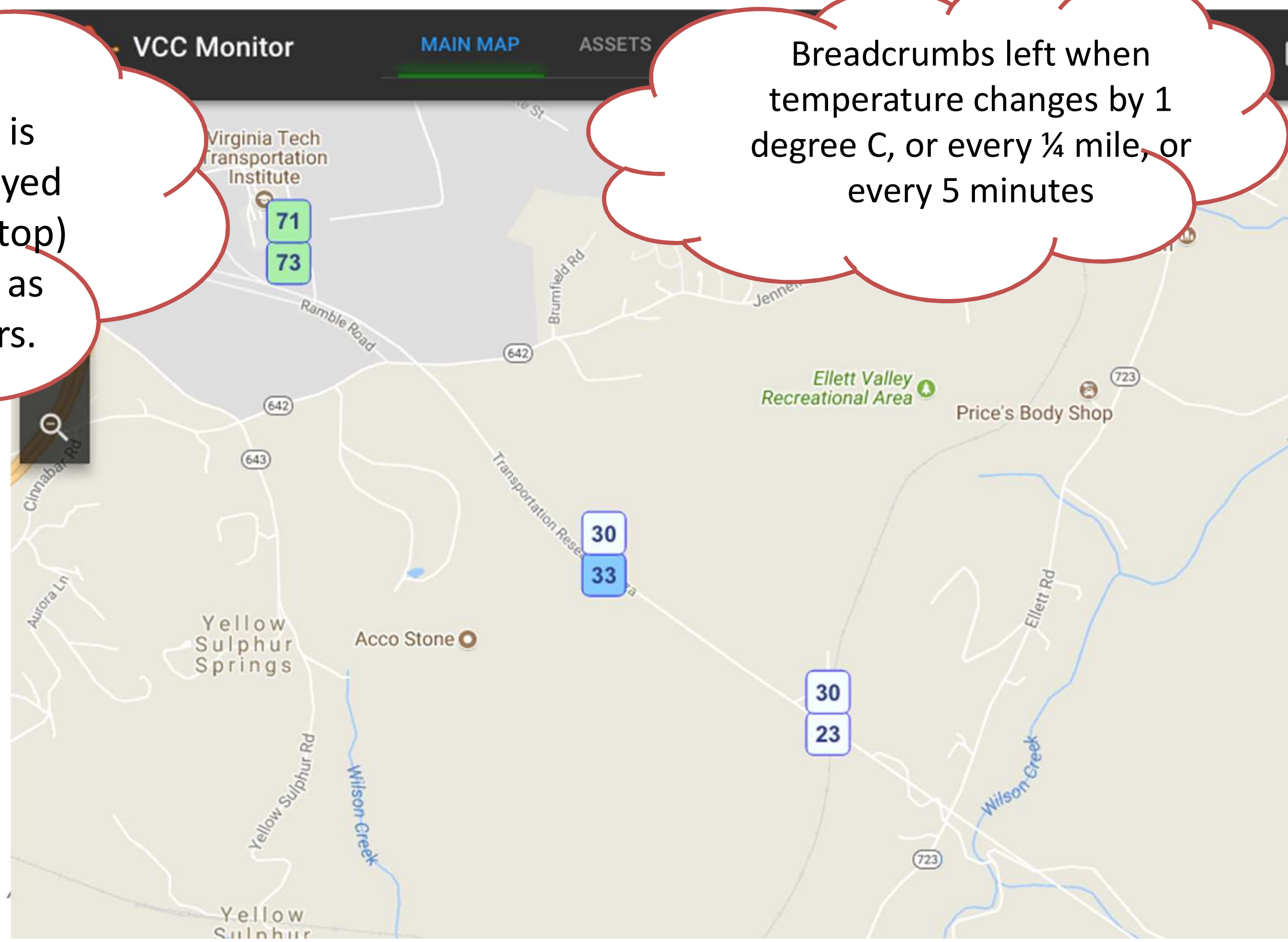


Main Map

Temperature Display

When the Temperature Data layer is active, 'bread crumbs' will be displayed showing the ambient temperature (top) and surface temperature (bottom) as determined by the weather sensors.

Breadcrumbs left when temperature changes by 1 degree C, or every ¼ mile, or every 5 minutes

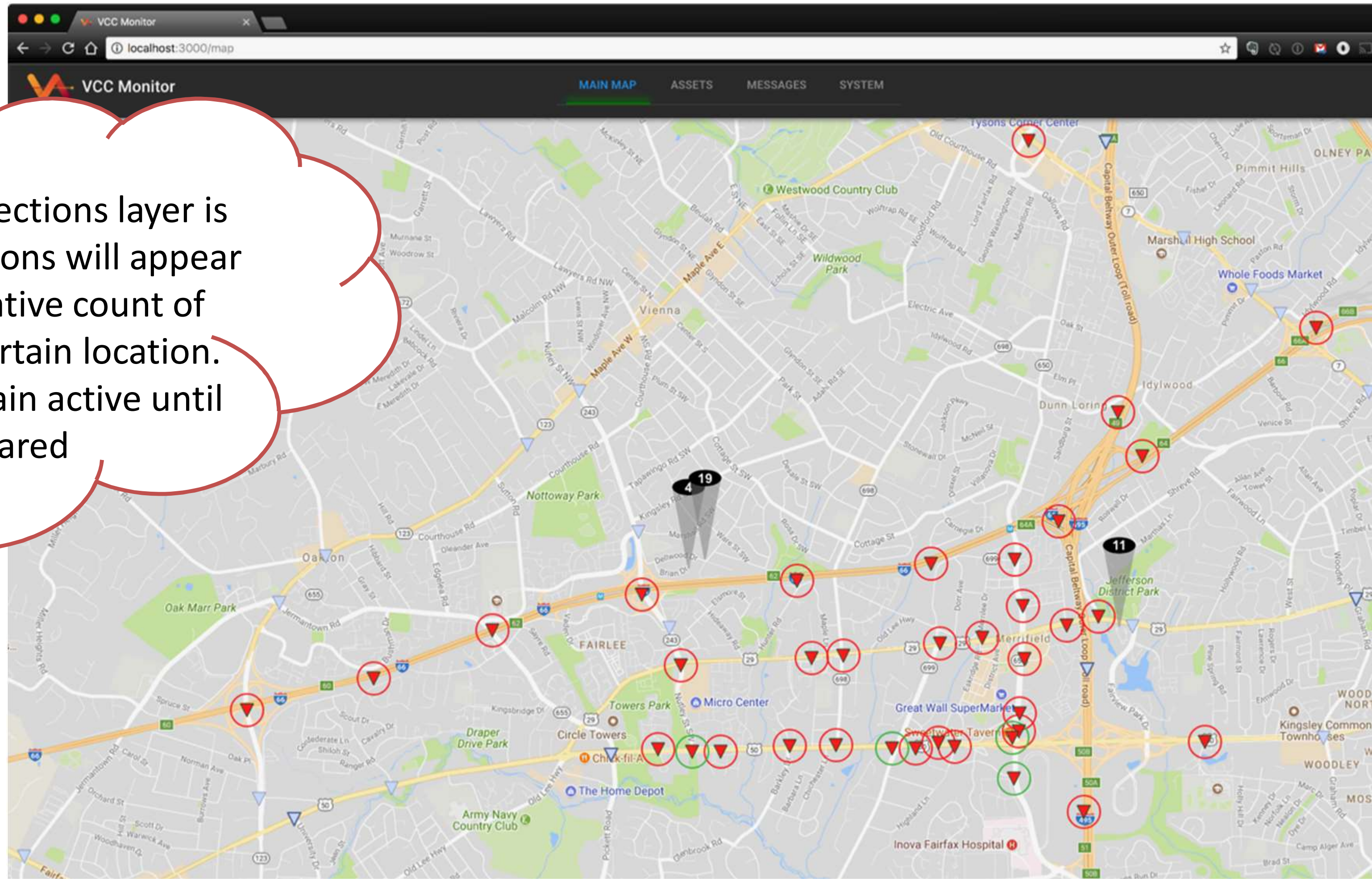


- Temp is below 32˚F
- Temp is 32˚ - 37.4˚F
- Temp is 37.4˚ - 91.4˚F
- Temp is above 91.4˚F

Main Map

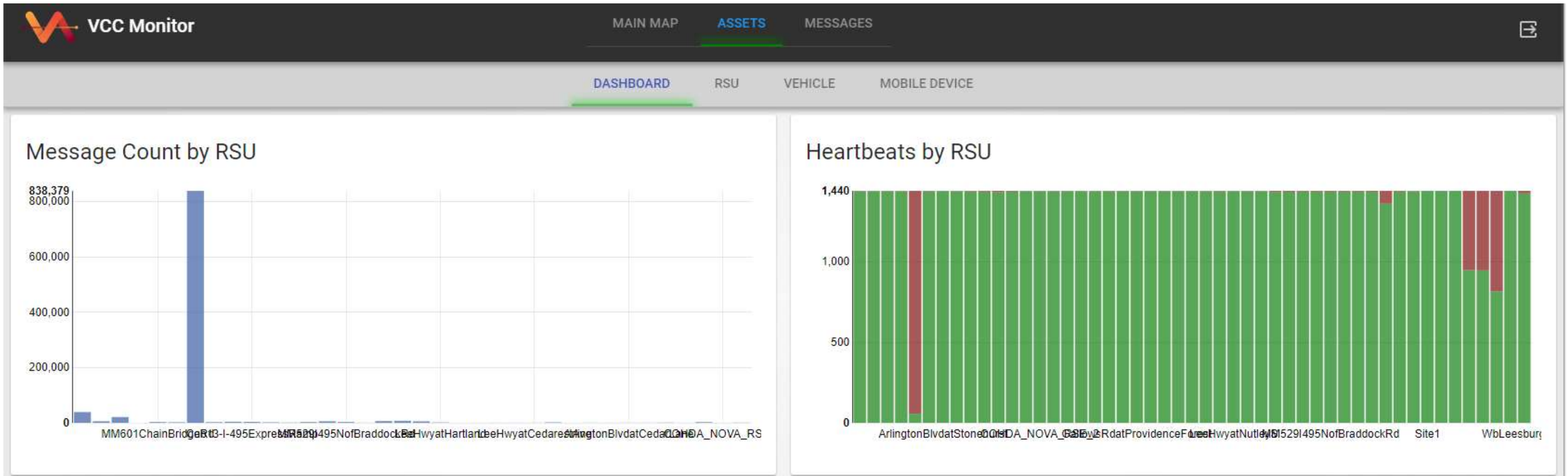
Pothole Detections

When the Pothole Detections layer is active, gray and black icons will appear indicating the cumulative count of potholes found in a certain location. These counts will remain active until manually cleared



Assets

Dashboard

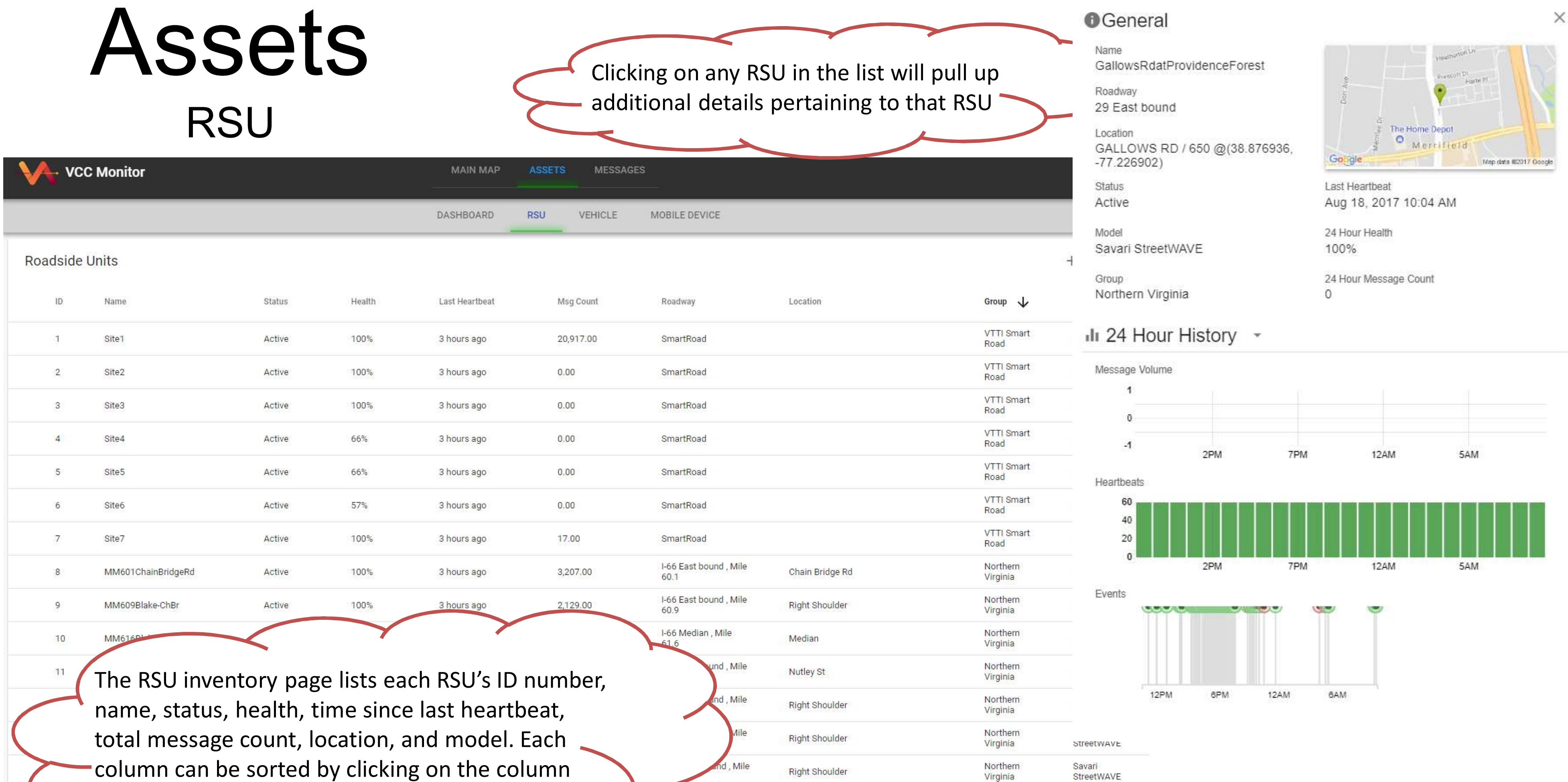


- The dashboard page provides an overview of the VCC activities, including RSU message volume and heartbeats.
- If a RSU is depicted with a green bar, it is active and communicating reliably. Red indicates missing heartbeats or heartbeats that are received but indicate a problem.
- Clicking on one of the bars associated with an RSU will redirect you to the RSU page and highlight more detailed information regarding that RSU.

Assets

RSU

Clicking on any RSU in the list will pull up additional details pertaining to that RSU

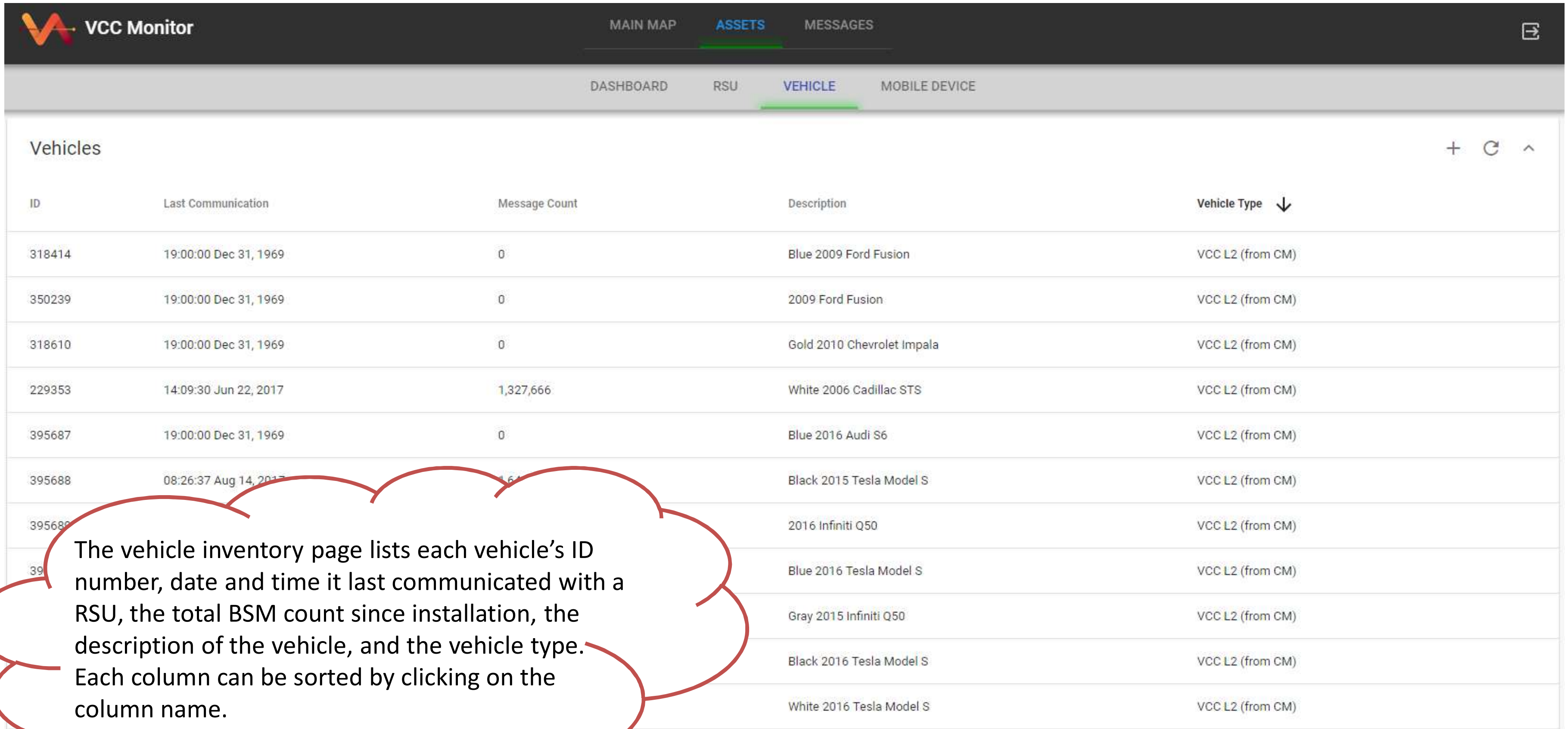


ID	Name	Status	Health	Last Heartbeat	Msg Count	Roadway	Location	Group	
1	Site1	Active	100%	3 hours ago	20,917.00	SmartRoad		VTTI Smart Road	
2	Site2	Active	100%	3 hours ago	0.00	SmartRoad		VTTI Smart Road	
3	Site3	Active	100%	3 hours ago	0.00	SmartRoad		VTTI Smart Road	
4	Site4	Active	66%	3 hours ago	0.00	SmartRoad		VTTI Smart Road	
5	Site5	Active	66%	3 hours ago	0.00	SmartRoad		VTTI Smart Road	
6	Site6	Active	57%	3 hours ago	0.00	SmartRoad		VTTI Smart Road	
7	Site7	Active	100%	3 hours ago	17.00	SmartRoad		VTTI Smart Road	
8	MM601ChainBridgeRd	Active	100%	3 hours ago	3,207.00	I-66 East bound , Mile 60.1	Chain Bridge Rd	Northern Virginia	
9	MM609Blake-ChBr	Active	100%	3 hours ago	2,129.00	I-66 East bound , Mile 60.9	Right Shoulder	Northern Virginia	
10						I-66 Median , Mile 61.6	Median	Northern Virginia	
11							Nutley St	Northern Virginia	
							Right Shoulder	Northern Virginia	
							Right Shoulder	Northern Virginia	StreetWAVE
							Right Shoulder	Northern Virginia	Savari StreetWAVE

The RSU inventory page lists each RSU's ID number, name, status, health, time since last heartbeat, total message count, location, and model. Each column can be sorted by clicking on the column name.

Assets

Vehicle



ID	Last Communication	Message Count	Description	Vehicle Type
318414	19:00:00 Dec 31, 1969	0	Blue 2009 Ford Fusion	VCC L2 (from CM)
350239	19:00:00 Dec 31, 1969	0	2009 Ford Fusion	VCC L2 (from CM)
318610	19:00:00 Dec 31, 1969	0	Gold 2010 Chevrolet Impala	VCC L2 (from CM)
229353	14:09:30 Jun 22, 2017	1,327,666	White 2006 Cadillac STS	VCC L2 (from CM)
395687	19:00:00 Dec 31, 1969	0	Blue 2016 Audi S6	VCC L2 (from CM)
395688	08:26:37 Aug 14, [illegible]	[illegible]	Black 2015 Tesla Model S	VCC L2 (from CM)
39568[illegible]	[illegible]	[illegible]	2016 Infiniti Q50	VCC L2 (from CM)
39[illegible]	[illegible]	[illegible]	Blue 2016 Tesla Model S	VCC L2 (from CM)
[illegible]	[illegible]	[illegible]	Gray 2015 Infiniti Q50	VCC L2 (from CM)
[illegible]	[illegible]	[illegible]	Black 2016 Tesla Model S	VCC L2 (from CM)
[illegible]	[illegible]	[illegible]	White 2016 Tesla Model S	VCC L2 (from CM)

The vehicle inventory page lists each vehicle's ID number, date and time it last communicated with a RSU, the total BSM count since installation, the description of the vehicle, and the vehicle type. Each column can be sorted by clicking on the column name.

Assets

Mobile Device

Clicking on any device in the list will pull up additional details pertaining to that device

VCC Monitor

MAIN MAP | ASSETS | MESSAGES

DASHBOARD | RSU | VEHICLE | MOBILE DEVICE

Phones, Tablets, etc.

ID ↑	Device ID	Vehicle ID	Application	Last Log Upload	MobileBSM Count	Last MobileBSM	Debug Mo
10000000	TestPhone1		vcctest	12:17:45 Aug 31, 2016	169,424	13:11:36 Aug 14, 2017	Yes
10000001	99000530097754				9,822	08:48:38 Jan 29, 2016	No
10000002	990005718187751			19:26:06 Dec 7, 2016	116,064	15:12:32 Mar 31, 2017	Yes
10000003	990004910218894				17,774	14:15:25 Nov 26, 2015	No
10000004	990005717244942				103,787	19:52:29 Oct 5, 2016	No
10000005	android_id				12,303	13:05:46 Oct 11, 2015	No
10000006	990005717484654			12:22:55 Feb 28, 2017	33,305	00:10:08 Jun 24, 2016	Yes
10000007	63788fb7a574156a				6,217	22:44:07 Oct 22, 2015	No
10000008	f2f5ca3308fc146				2,193	10:22:08 Dec 5, 2015	No
10000009	aeb742fea0fb7975				218,781	00:10:45 Jun 24, 2016	No
10000010	3da				149	17:48:30 Oct 11, 2015	No
10000011					79	12:31:07 Oct 20, 2015	No
10					3	08:20:44 Oct 21, 2015	No
10000					32	16:47:42 May 30, 2017	Yes

The mobile device inventory page lists each device's ID, the associated vehicle ID, the total BSM count since activation, the date and time of the last BSM, and what mode the device is in. Each column can be sorted by clicking on the column name.

General

Database ID
10000000

Device ID
TestPhone1

Component ID
Unknown

Last Log Upload
12:17:45 Aug 31, 2016

Mobile BSM Count
179491

Last Mobile BSM
11:15:09 Aug 16, 2017

Notes
This is a fake/test phone.

Active Area

On Duty
No

Work Radius [m]
None

Task
General Labor, In Vehicle

Phase & Mode

Phase 1 Date
8/1/2016

Debug Mode

Enable Pothole Detection

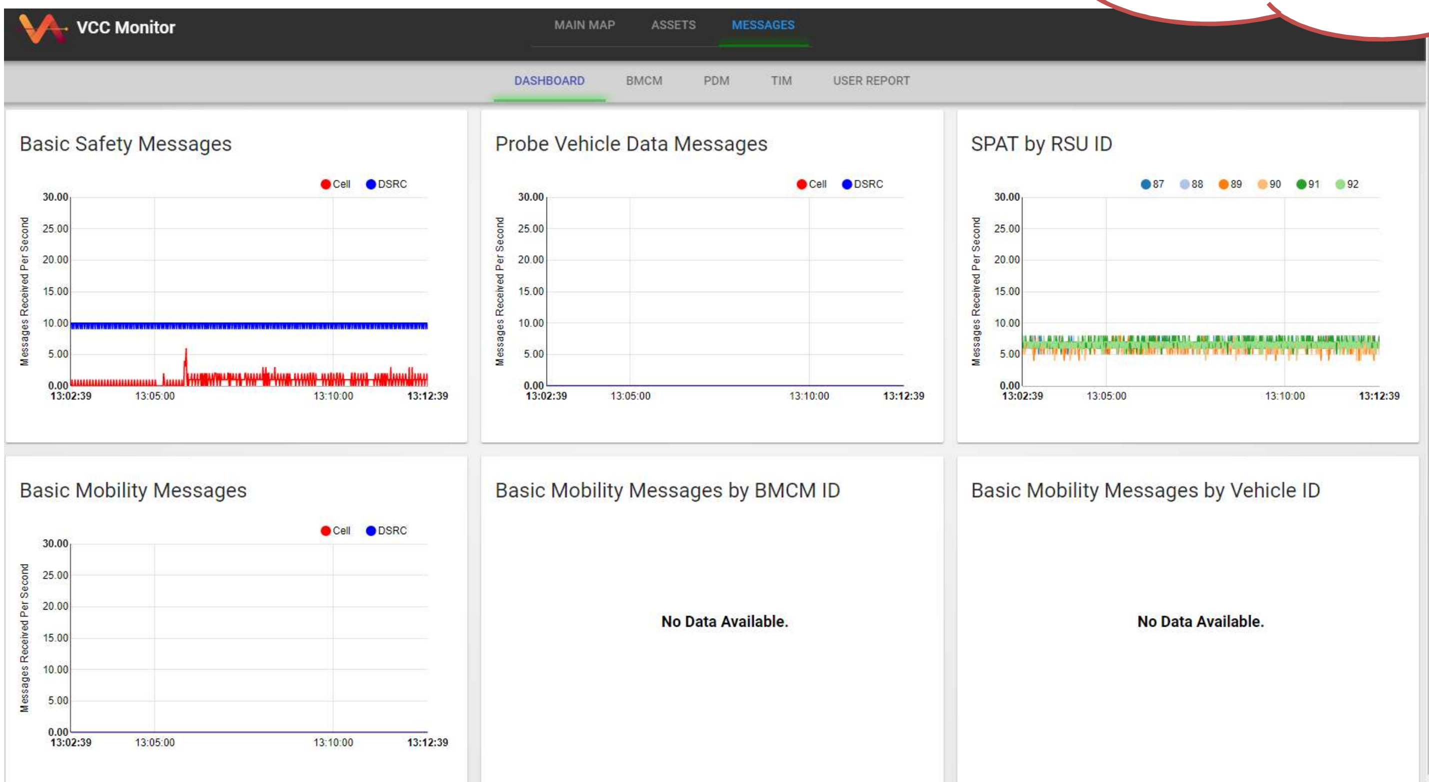
Phase 2 Date
6/1/2017

Phase 3 Date
7/1/2017

Messages

Dashboard

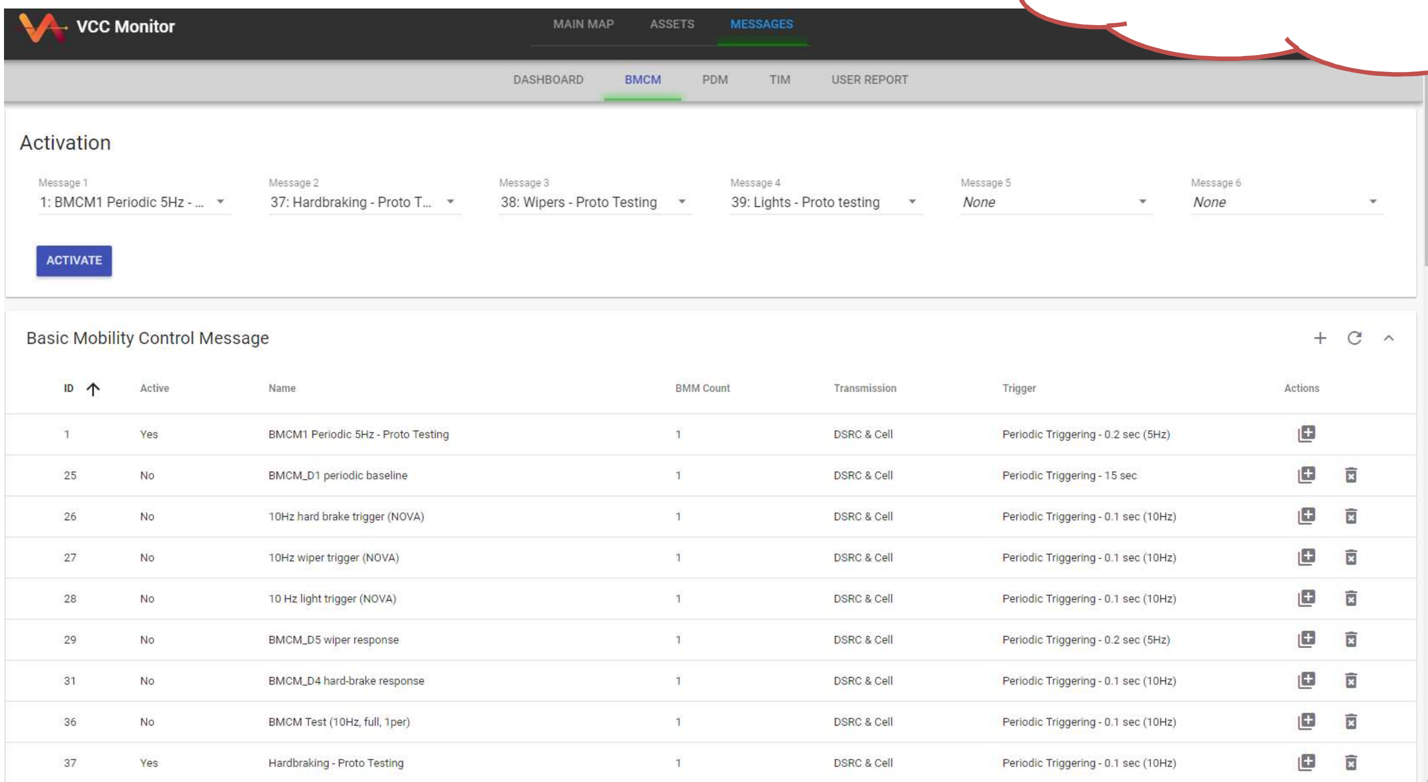
Clicking on the colored circles (e.g., Cell and DSRC in the top left graph) toggle that information on and off so you can isolate certain conditions.



Messages

BMCM

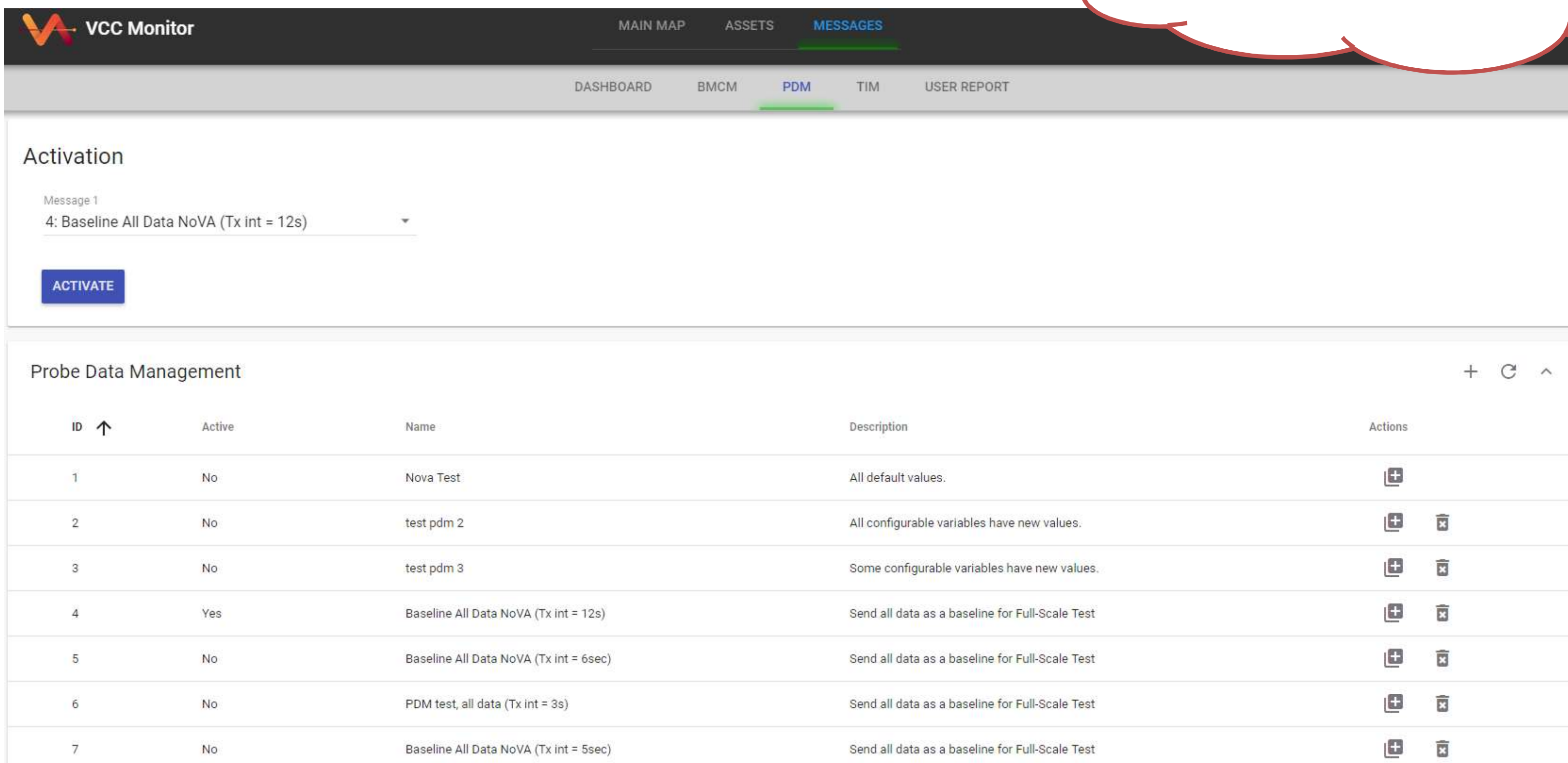
The BMCM page is not supported for this project. However, this is an example of the capabilities of the VCC system.



Messages

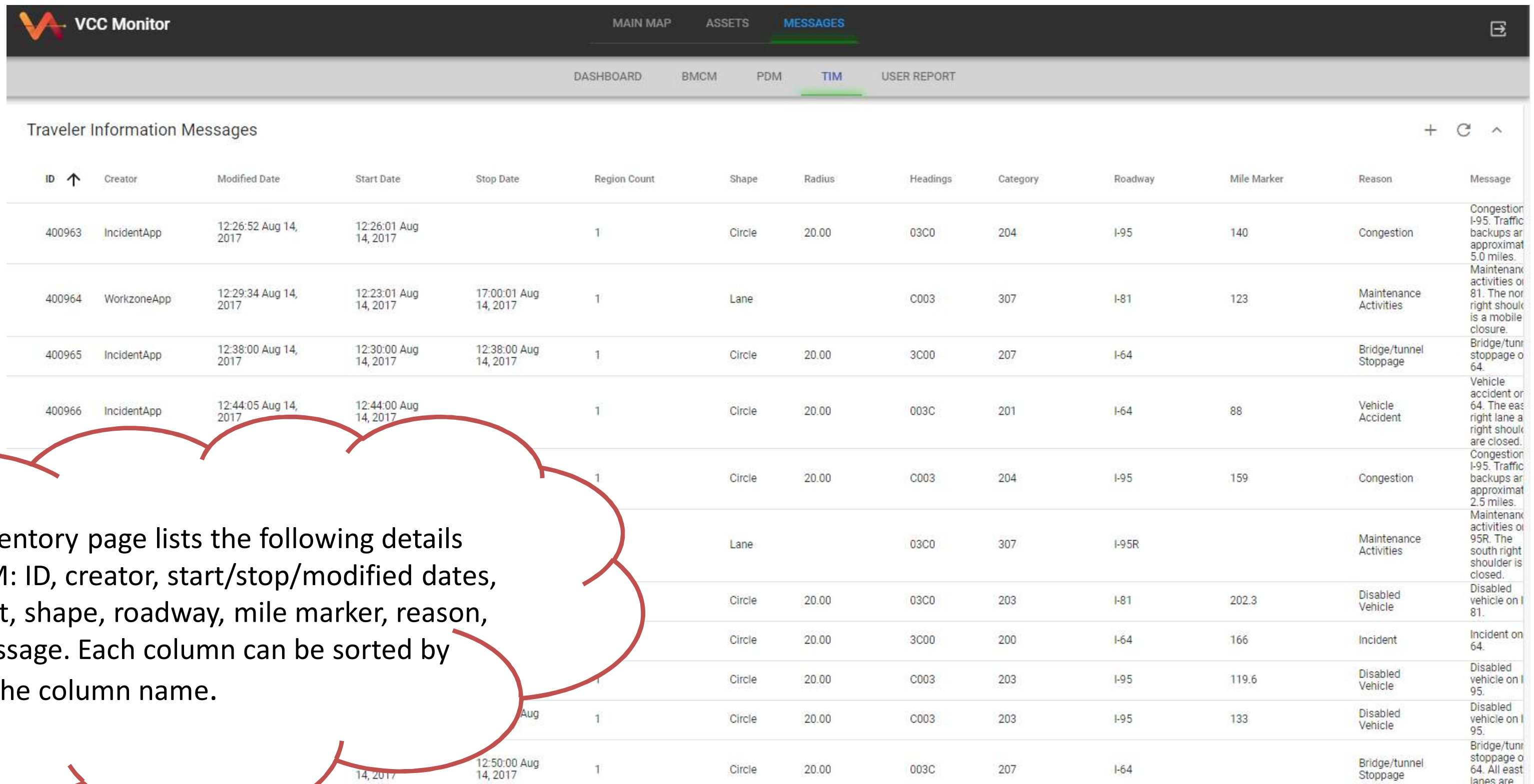
PDM

The PDM page is not supported for this project. However, this is an example of the capabilities of the VCC system.



Messages

TIM

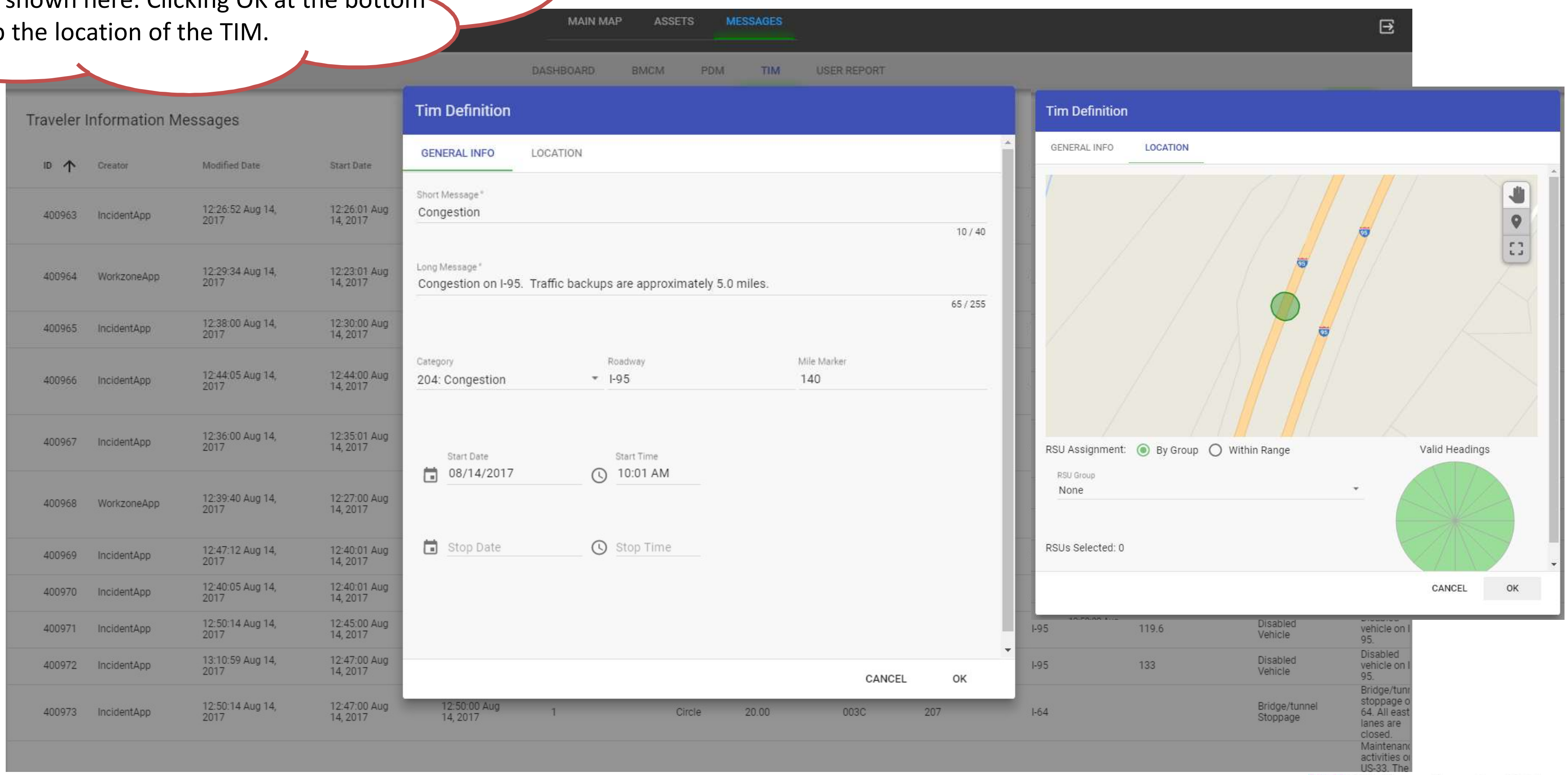


The TIM inventory page lists the following details for each TIM: ID, creator, start/stop/modified dates, region count, shape, roadway, mile marker, reason, and the message. Each column can be sorted by clicking on the column name.

Messages

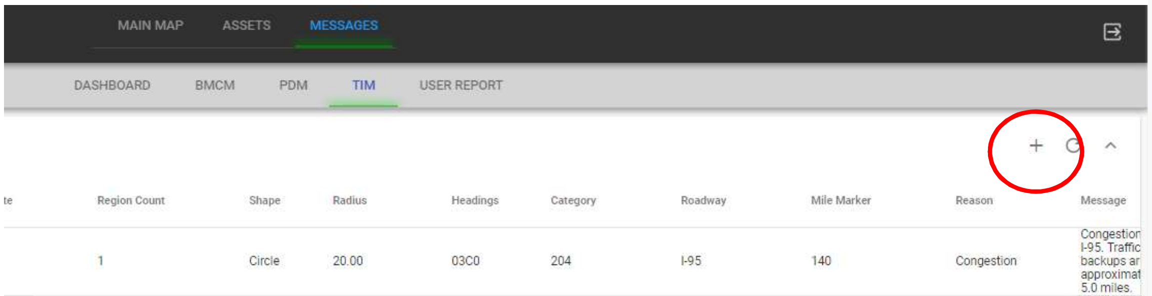
TIM

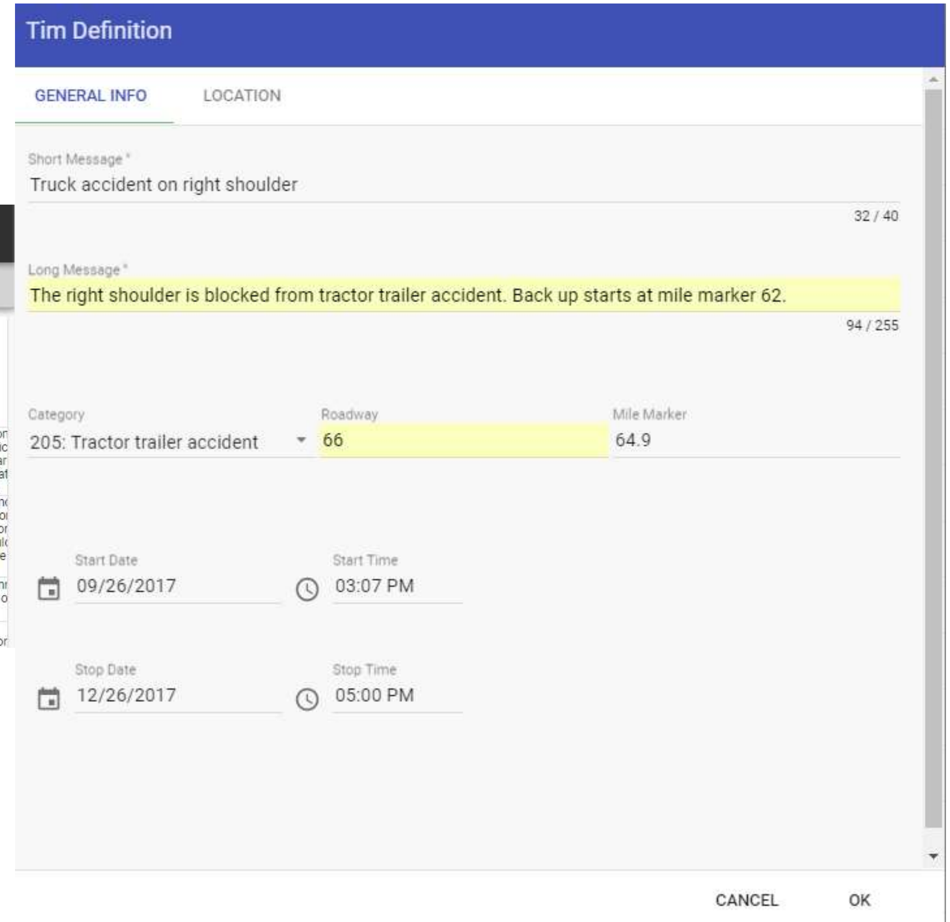
Clicking on any of the TIMs on the inventory page will pull up additional details about that TIM, as shown here. Clicking OK at the bottom pulls up the location of the TIM.



Messages

Creating a TIM





1. Click on the Plus sign in the upper right corner to create a new TIM message
2. Fill in the General Info:
    1. Short message – Displayed in VCC mobile queue
    2. Long message – Read aloud in VCC Mobile when within active area
    3. Select appropriate category
    4. Optional: fill in roadway and mile marker
    5. Select the start and end date/time for the message (if no end time selected, the message will remain active until it is manually cancelled)

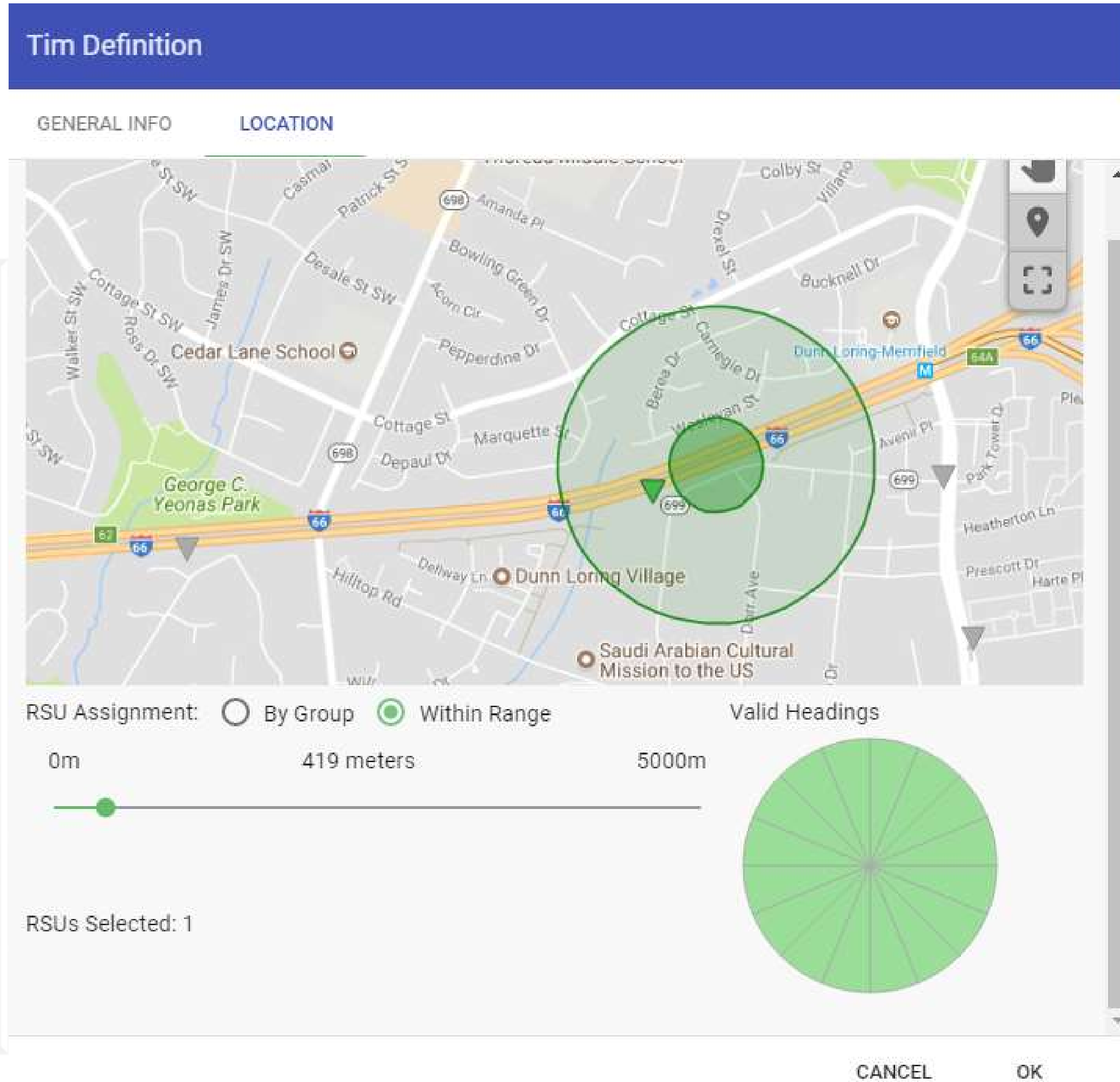
Messages

Creating a TIM

Fill in the Location Info:

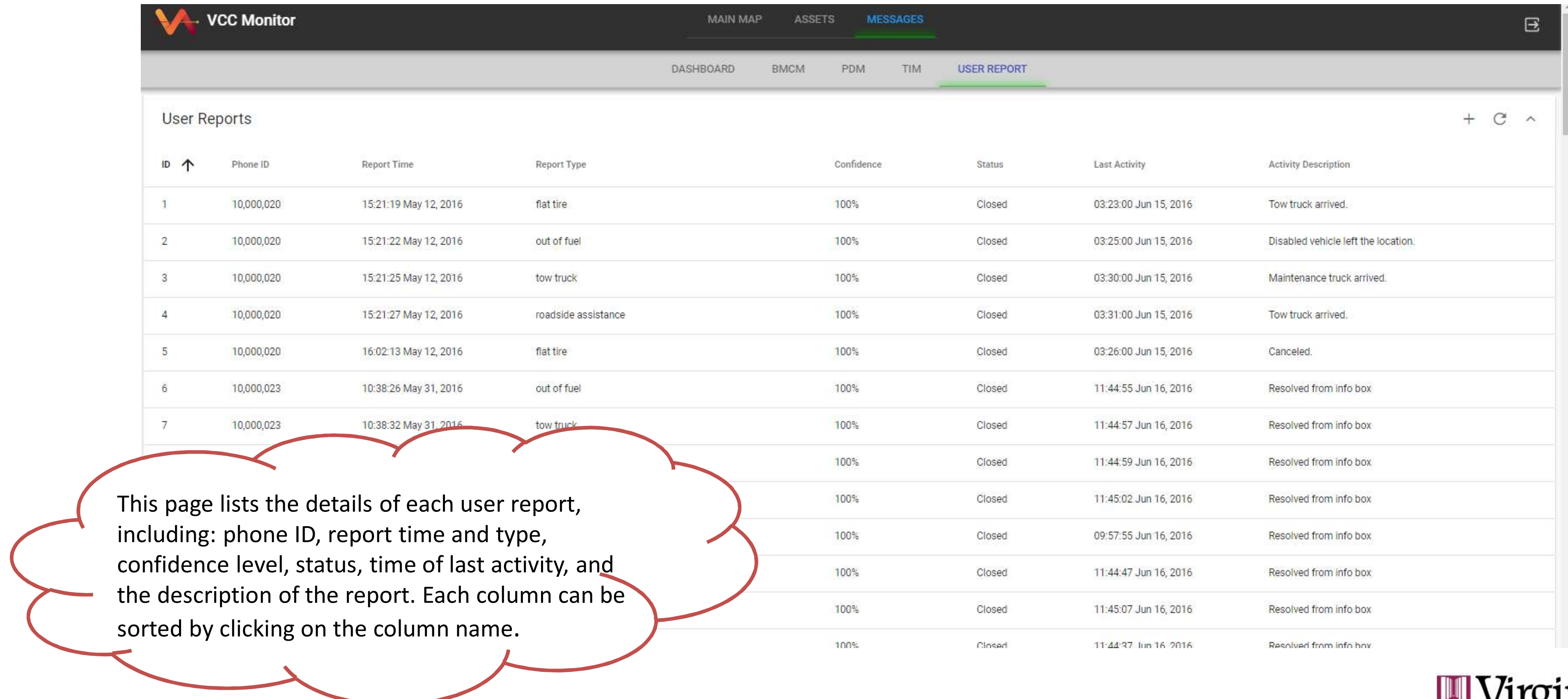
1. Use the Pan option (hand icon) to pinpoint where the TIM should be located on the map
2. Assign to all RSUs within a group (then select appropriate group) or within a certain range to be more efficient (indicate radius by dragging white circle to appropriate range in meters)
   a) All RSUs within selected range will turn green
3. Deselect the appropriate portions in the pie chart to indicate which headings are valid for the message
4. Click OK when done

Messages will then be sent to appropriate cell phones via cellular depending on what coordinate box the phone is located in and sent via DSRC to RSUs within selected range



Messages

User Report



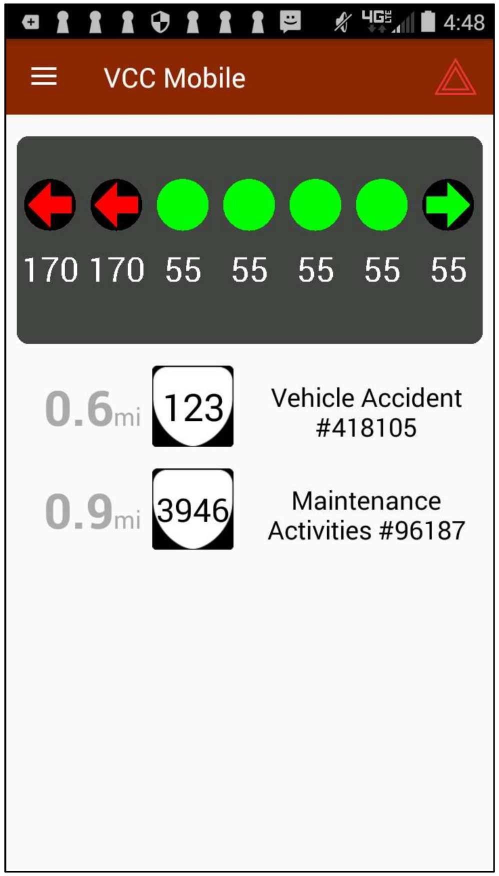
VCC Mobile Application Training

Prepared September 18, 2017

Virginia Tech Transportation Institute

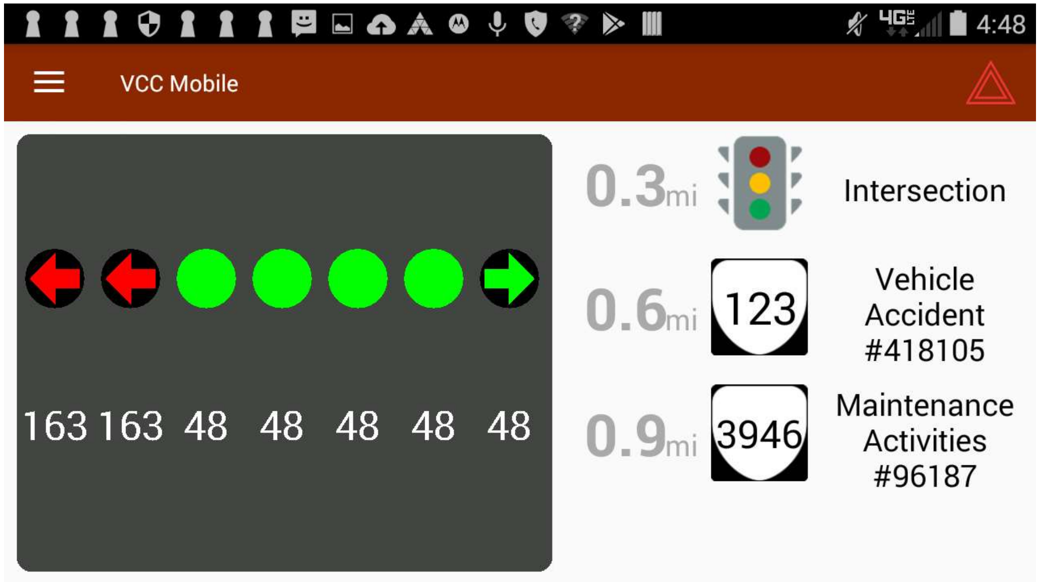
VCC Mobile App Overview

- Android smartphone application
- Provides real-time messaging capabilities to drivers in a format that limits attentional demand
  - Appropriate for use while driving
- Interfaces to OBE via Wi-Fi connection (MiFi)
- Utilizes the OBE's GPS information and communication channels to exchange messages with the VCC Cloud
- Location-based TIMs are received from the Cloud and presented to the driver as applicable



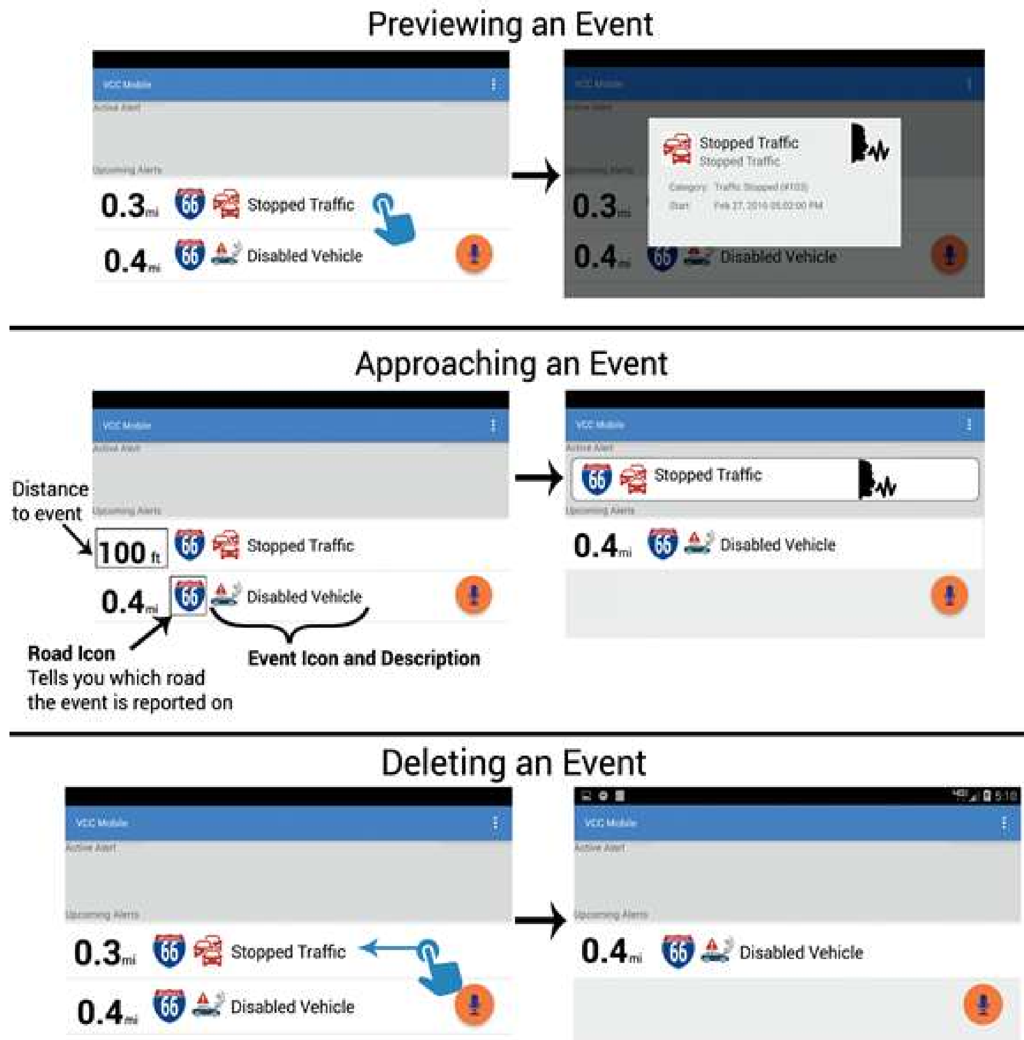
VCC Mobile Capabilities

- Presents real-time location-based driving information
  - Significant weather conditions
  - Traffic conditions (e.g., slowing or clearing of traffic)
  - Roadway incidents (e.g., stalled vehicles or accidents)
  - Dynamic message sign content (ATM/HOV status)
  - Emergency vehicle presence
  - Work zone alerts
  - Other driving related messages
- Driver reporting and Call for Help



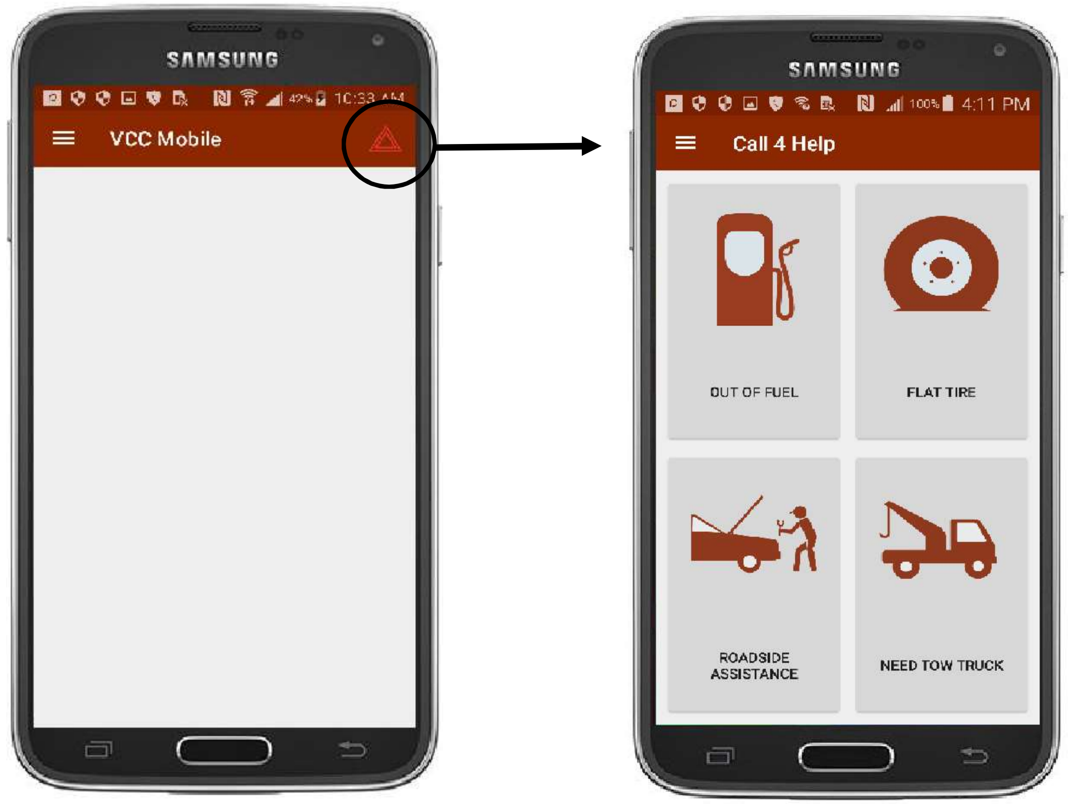
Event Presentation

- Events presented in a list
  - Sorted by distance to the event (Active alerts on top; upcoming alerts on the bottom)
  - Shows which road the event is reported on
  - Brief description of event
- Event will move from upcoming to active when distance threshold reached
  - Active event information is read aloud
- Tapping a message in the list will preview it
  - Reads message aloud
  - Opens window showing message category (e.g., traffic, work zone, etc.)
  - Tap message again to exit
- Delete an event by swiping it to the left



Driver Call For Help

- Press the call for help icon (orange triangle) to bring up the following options:
  - Out of Fuel
  - Flat Tire
  - Roadside Assistance
  - Need Tow Truck
  - Dial 911
- We did not scope in any interface to customer service center or emergency services but this could be done



Pothole Detection

Literature Review

- Road Surface Monitoring Methods and Applications
  - Two main objective methods
    - Image Recognition
      - Relies on cameras to monitor road surface
    - Mobile Sensing
      - Utilizes smartphone sensors to gather real-time data
  - Objective methods may or may not include crowdsourcing

Pothole Detection

Literature Review

Image Recognition

- Advantages:
  - Accurate (Radopoulou & Brilakis, 2016)
- Disadvantages:
  - Requires high computation power/complicated algorithms (Joubert et al., 2011; Radopoulou & Brilakis, 2016)
  - Accuracy can be limited by bad weather, image processing time (Rode et al., 2009)

- Results:
  - Cambridge, UK study with iPhone 5S camera – 84% precision (Radopoulou & Brilakis, 2015)
  - South African study using Microsoft Kinect technology – accuracy TBD (Joubert et al., 2011)
  - Multimodal sensor analysis system with Android smartphone – 93% precision (Orhan & Eren, 2013)
  - Seoul, Korea study using forward-facing camera – 88% precision (Jo & Ryu, 2015)

Pothole Detection

Literature Review

Mobile Sensing

- Advantages:
  - Cost-efficient (Radopolou & Brilakis, 2016; Wang et al., 2015)
  - Provides real-time data (Radopolou & Brilakis, 2016; Wang et al., 2015)
- Disadvantages:
  - Potential problems with accuracy

- Results:
  - Boston's Street Bump – crowdsourced commercial iOS application (Brisimi et al., 2016)
  - RoadBump – crowdsourced commercial application tested in Arkansas in 2015 using Android GPS and accelerometer (Grimmer Software)
  - RoadSense – Android app utilizing C4.5 algorithm with 98.6% accuracy (Allouch, et al., 2017)
  - Pothole Patrol (P^2) used accerometers and GPS mounted to roof with < 0.2% false-positives (Eriksson et al., 2008)
  - Nericell used an array of sensors in Bangalore with a 20-30% high false positive rate; only analyzed z-axis accelerometer (Mohan et al., 2008)
  - Algorithm testing of data collected using Android accelerometers; up to 90% true positives (Mohan et al., 2008)

Pothole Detection

App Development

- Utilize predetermined functions
  - To keep lightweight enough to run on end user's phone
  - Less processor-intensive
- Utilized STDEV algorithm tested by Mednis et al. (2011) as starting point, and incorporated into Mednis' other 3 algorithms
- Applied additional techniques to reduce accelerometer noise

Pothole Detection

Testing Process

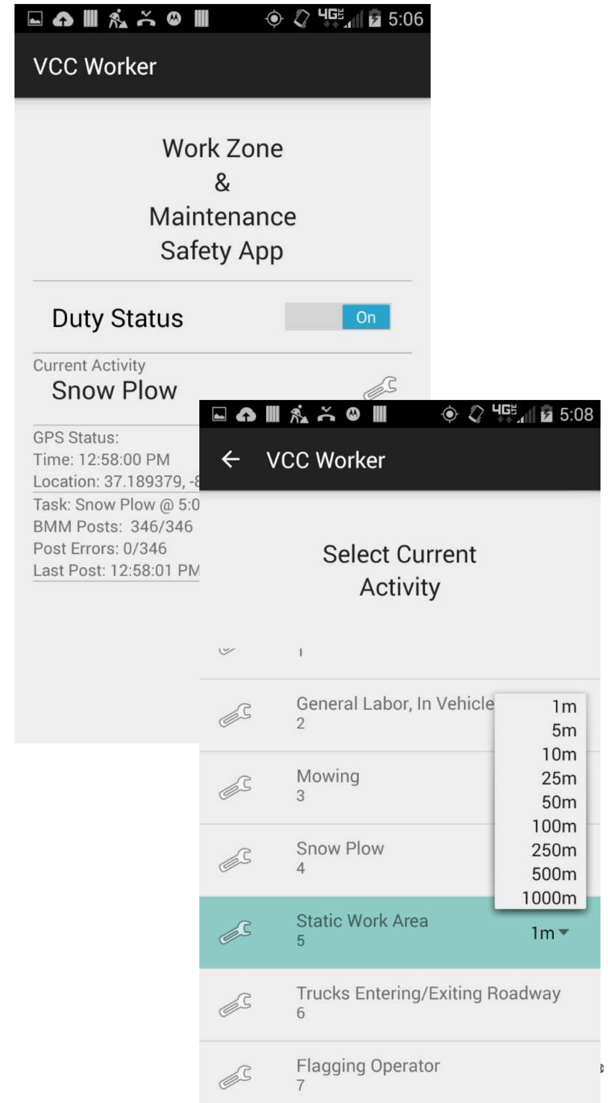
- Worked with DDOT to identify standard for potholes (8/10/17)
  - Large, medium, small
- Collected data using mobile phone in equipped DDOT truck
- Collected data used in machine learning process (8/14/17)

Pothole Detection App

- The pothole detection app is integrated within VCC Mobile
  - Automatically collects pothole data whenever the app is running
  - Data is displayed in real-time on VCC Monitor

VCC Mobile Worker App

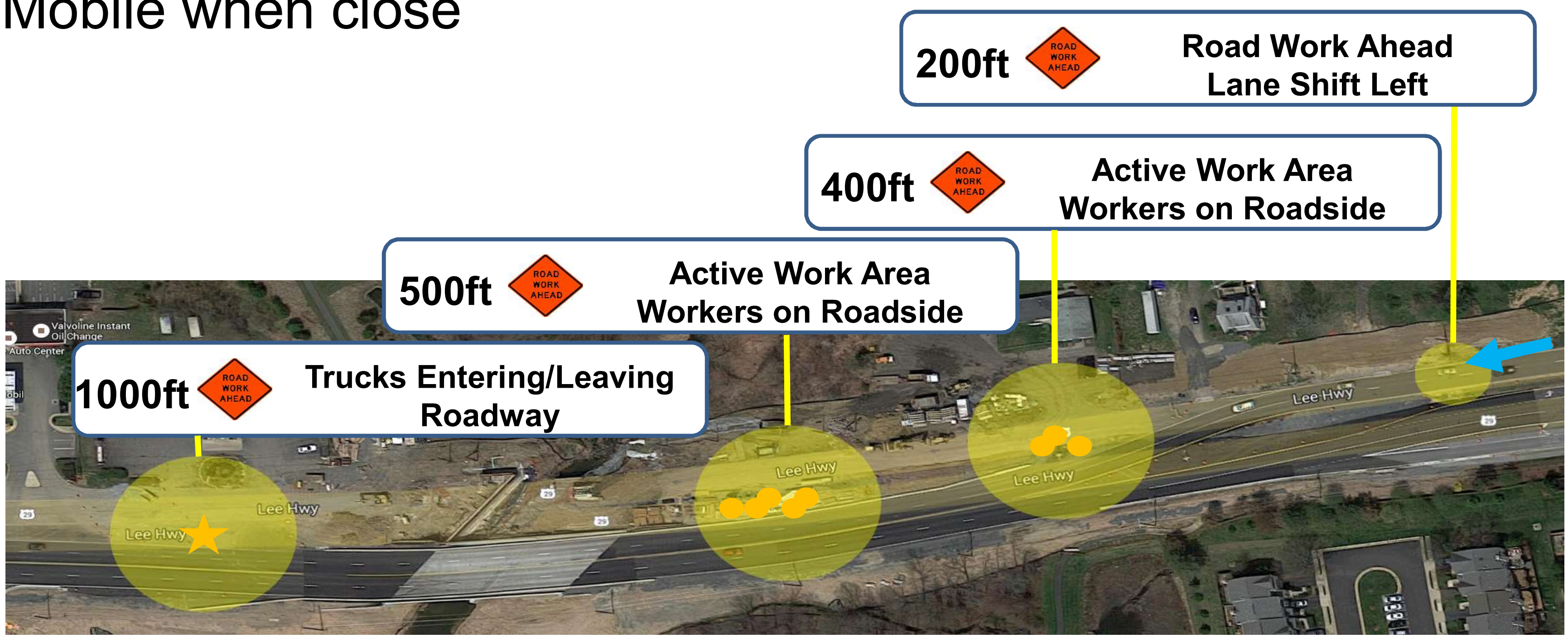
- Location and status updates from smart phone with app (for now)
- User selects an activity and duty status
- Work Zone app sends position and activity data directly to VCC Cloud via cellular comms
- VCC Cloud processes incoming messages and creates advisories and streaming alerts for drivers
  - Clustering of multiple worker reports
  - TIM messages sent via DSRC
  - Proprietary messages sent via cellular to VCC Mobile
- Messages are received on VCC Mobile and displayed to driver based on position, speed, direction, etc.



Work Zone App Components

- VCC Work Zone app
  - Planned work zone data derived from VDOT data sharing
  - Converted to SAE J2735 TIM message and broadcast DSRC
  - Converted to message and sent to VCC Mobile
- VCC Worker app
  - Provides real time worker location and dynamic advisory messages both drivers and road side workers
- VCC Mobile
  - receives and presents messages to drivers as they approach the work zone
- VCC Work Zone Builder app
  - Create and verify digital definition of work zone
  - Locate and define specific work zone features – lane closures, reduced speeds, flagging operations, activity areas, etc.
  - Easy for contractors to use to validate and keep work zone definitions up to date

- VCC Cloud builds dynamic traveler messages and pushes them to drivers via DSRC and cellular
- Clusters multiple workers together, updated every 10 s
- Locations of individual workers or clusters are streamed to VCC Mobile when close

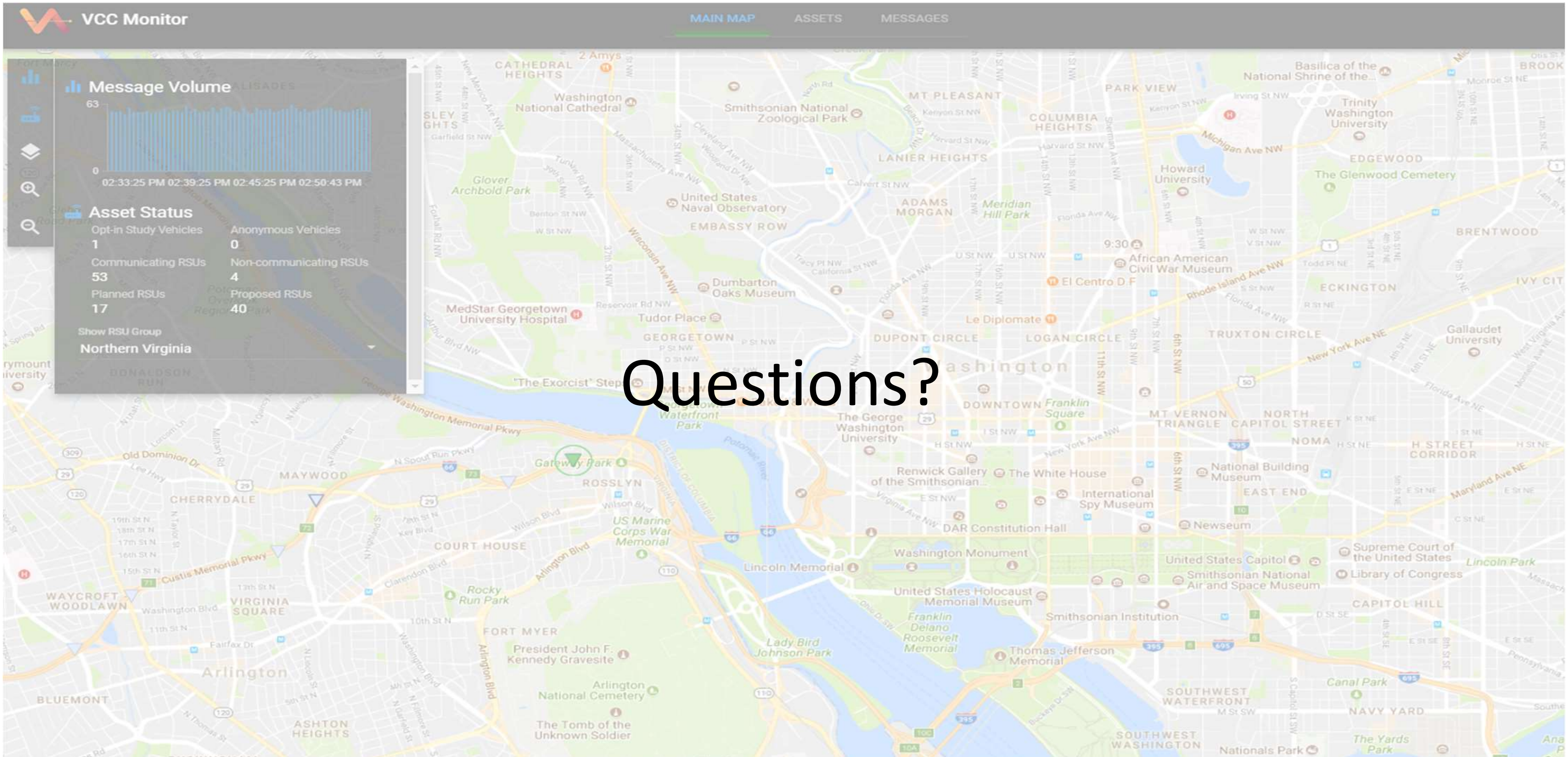


Task E – Support Data Analytics

Public API

- The Public API provides access to:
  - RSU location info
  - Dynamic Message Signs (DMS) location and message info
  - Basic Safety Messages (BSMs) received from RSUs and over cellular
    - Information is a subset of all BSM data including location, speed, temperature, etc.
  - Traveler Information Messages (TIM) - messages broadcast by RSUs
  - SPaT (Signal Phase and Time) - info about specific intersections
  - Map Data - messages describing the lane configuration of specific intersections

Questions?



VirginiaTech
Transportation Institute