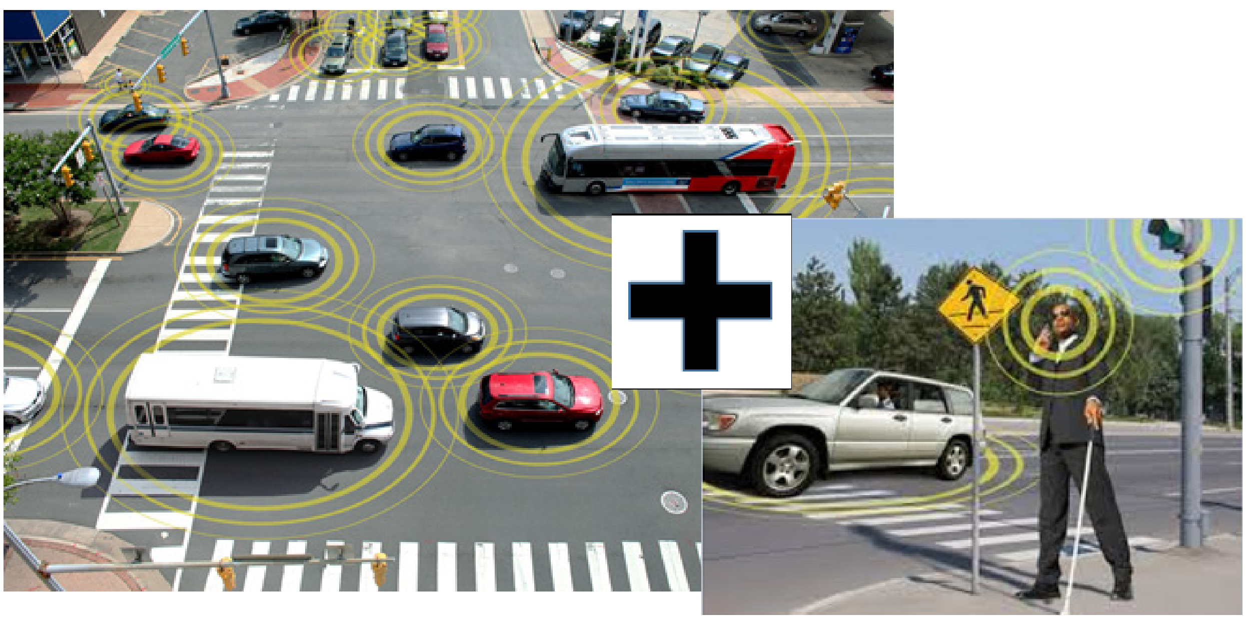
Sharing Data between Mobile Devices, Connected Vehicles, and Infrastructure

Task 4: System Architecture and Design Document - FINAL

www.its.dot.gov/index.htm
Report — October 26, 2016
FHWA-JPO-17-476



U.S. Department of Transportation

Produced by Battelle Memorial Institute under DTFH61-12-D-00046
U.S. Department of Transportation
FHWA Office of Operations Research and Development

Picture Source: U.S. DOT Office of the Assistance Secretary for Research and Technology, Intelligent Transportation Systems, Joint Program Office

Notice

Quality Assurance Statement

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FHWA-JPO-17-476		

4. Title and Subtitle	5. Report Date
Sharing Data between Mobile Devices, Connected Vehicles and Infrastructure Task 4: System Architecture and Design Document (SA/DD)	October 26, 2016
	6. Performing Organization Code
7. Author(s) Matthew Burns, Kristina Guspan, Sudhakar Nallamothu, Margret Hailemariam, Joerg 'Nu' Rosenbohm	8. Performing Organization Report No.
9. Performing Organization Name And Address Battelle 505 King Avenue Columbus, OH 43201	10. Work Unit No. (TRAIS)
	11. Contract or Grant No. DTFH61-12-D-00046
12. Sponsoring Agency Name and Address U.S. Department of Transportation FHWA Office of Operations Research and Development Turner-Fairbank Highway Research Center 6300 Georgetown Pike McLean, VA 22101	13. Type of Report and Period Covered
	14. Sponsoring Agency Code

15. Supplementary Notes

Government Task Monitor – Jon Obenberger

16. Abstract

This report describes the system architecture and design of the Experimental Prototype System (EPS) for the demonstration of the use of mobile devices in a connected vehicle environment. Specifically, it defines the system structure and behavior, the software components, the interfaces and the data necessary for fulfilling the EPS software requirements identified in Task 3: System Requirements Specifications (SyRS). This design document serves as the basis for development of the experiment prototype system that will be used to demonstrate both safety and mobility benefits of using mobile devices including how the interactions and messaging of these devices may be coordinated in order to provide more efficient (and less congested exchange of information) and how these devices might behave in an environment where mobile device coordination does not exist.

17. Key Words	18. Distribution Statement
Mobile Device Message Coordination Prototype, Connected Vehicle, Mobile Device, Coordinated Mobile Device, Personal Mobility Message, Personal Safety Message, system design, system architecture	

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
None	None	94	N/A

Form DOT F 1700.7 (8-72) **Reproduction of completed page authorized**

REVISION HISTORY

Revision	Date	Change Description	Affected Sections/Pages
-	01/04/2016	Document Created	
1	06/20/2016	Pre-Draft Initial Release	All Chapters
2	08/04/2016	Draft Release	All Chapters
3	09/08/2016	Final Release addressing comments received	All Chapters
4	10/26/2016	Final Release addressing 508 compliance	All Chapters

Table of Contents

List of Tables

List of Figures

Chapter 1 Scope

The purpose and vision of this experimental research task order is to enable the support in exchanging messages with mobile devices within the public right of way, to enhance the safety and mobility of these trips, and to enable the public agencies to improve how they manage traffic, which includes travelers using mobile devices.

This document contains the system architecture and system design associated with the experimental prototype system to be developed as part of the "Coordination of Mobile Devices for Connected Vehicle Applications" task order. This document has been written with the assumption that the reader possesses a general knowledge associated with the connected vehicle technologies and applications and the systems engineering process.

This document was developed based on the contents of the Final Concept of Operations (ConOps / dated July 2016), and the Final System Requirements Specifications (SysRS / dated July 2016).

It needs to be noted that the ConOps and the SysRS were developed to provide foundational documents that address the anticipated user needs and requirements of various types. However, funding and time constraints required to limit the number of requirements to be addressed in the CV application software developed as part of this task order. The down-selected requirements are shown in the last column termed "EPS Req" within the requirements tables of the Final SysRS document. This document addresses those down-selected SysRS-defined requirements.

1.1 Background

The United States Department of Transportation (U.S. DOT) has conducted significant research on the use, benefits, and operational issues associated with using dedicated short-range communications (DSRC) and cellular devices in both vehicular and infrastructure-based communications. Specifically, the benefits are intended to improve the safety, mobility and environmental impact on our surface transportation system. When the concept of connected vehicle environment first emerged, DSRC was conceived as an enabler for the mobility-impaired and other travelers with unique needs. However, the unprecedented adoption of smartphones and similar devices in the general population has necessitated a renewed analysis of its role in the broader connected vehicle environment. A better understanding of the flexibility and cohesion of the available technologies can lead to improved traveler safety and user experience. Additionally, key questions and issues exist related to the expected impact that personal mobile devices (e.g., tablets, smartphones.), that are also equipped with DSRC technology, will have on channel congestion and error-rates in the connected vehicle environment.

The Coordination of *Mobile Devices for Connected Vehicle Applications* task order aims to enhance the connected vehicle environment by incorporating the mobile device with DSRC broadcasting capabilities in order to facilitate the transmission of personal safety messages (PSM) and personal mobility messages (PMM) that interact with other systems (i.e., Passenger to Vehicle(P2V) and Passenger to Infrastructure(P2I) exchange of messages). This task order seeks to utilize the mobile device as a medium for messages that complement those transmitted by vehicles by adding the connected "person" fleet dimension to the existing connected vehicle environment with three (3) primary objectives being to:

1. Evaluate the potential benefit and potential issues associated with the transmission of mobility and safety messages from hand-held mobile devices via alternative communications media.
2. Develop and test modifications to mobility and safety messages for mobile devices that complement existing and emerging standardized messages from vehicles, and coordinate those proposed modifications with mobility application developers and standards development organizations.
3. Create and demonstrate potential methods for coordinating safety and mobility messaging linking mobile devices carried by pedestrians into light and transit vehicles that may or may not themselves be capable of generating one or more related safety and mobility messages.

This report documents the high level System Architecture and System Design Document (SA/DD) for the Experimental Prototype System (EPS) being developed to support the planned demonstrations associated with the introduction of mobile devices into the connected vehicle landscape. This SA/DD is a representation of a system/software design that is to be used for recording design information, addressing various design concerns, and communicating that information to the task order stakeholders.

This document provides a representation of the software system created to facilitate analysis, planning, and implementation. It is a blueprint or model of the software, communications, and to some extent, the hardware systems. The SA/DD is used as the primary medium for communicating design information.

The SA/DD shows how the software system will be structured to satisfy the requirements identified in the *Sharing Data between Mobile Devices, Connected Vehicles, and Infrastructure* system requirements report. It is a translation of the requirements into a description of the structure and behavior of the system, the software components, the interfaces, and the data necessary for implementing the software solution.

Most importantly, the scope of this document and the system described herein is limited to an experimental system that will be used to design, test, and demonstrate new communication messages and message types as well as explore the effectiveness and potential mechanisms for coordinating these messages across multiple mobile device, vehicles, and infrastructure. This is intended as a research project and therefore does not seek to identify, define, summarize, or propose a system suitable for immediate wide-scale deployment.

1.2 Terminology

The meanings of the auxiliary verbs used in this document are defined as follows:

- Shall Compliance with a requirement, specification or a test is mandatory
- Should Compliance with a requirement, specification, or a test is recommended
- May Expresses a permissible way to achieve compliance

1.3 Identification

This document is one of the deliverables for Task 4 of the task order "*Coordination of Mobile Devices for Connected Vehicle Applications"*, which is being conducted by Battelle Memorial Institute for the Federal Highway Administration (FHWA) under Contract Number DTFH61-12-D-00046.

The overall approach to this SA/DD is based on the guidance described in IEEE Std 1016-2009, the *IEEE Standard for Information Technology – Systems Design – Software Design Description*.

1.4 Document Overview

The purpose of this document is to further elaborate on the concepts and scenarios described in the re-opened and modified Concept of Operations (ConOps) report. This will be accomplished by describing how the system is to operate in terms of detailed requirement statements.

The remainder of this document consists of the following sections and content:

- Chapter 2 (Applicable Documents) describes any external documentation referenced throughout this document.
- Chapter 3 (System Architecture) includes:
  - an overview of the system of interest
  - a description of the system components
  - the different types of communications technologies used to fulfil the systems requirements
  - the design constraints
- Chapter 4 (Design Concerns) enumerates and expands upon the primary design concerns namely User Interfaces, Networking and Communications, Security, Protection of Personally Identifiable Information, Packaging, and Power.
- Chapter 5 (Components Description) describes the logical and physical components that make up the system.
- Chapter 6 (Internal Interfaces) describes the interfaces between the internal components that make up the system.
- Chapter 7 (External Interfaces) describes any interfaces between the EPS and external systems.
- Chapter 8 (Component Design) documents the design of the various components including the hardware specifications and the software design.
- Appendix A contains the Acronyms and Abbreviations.
- Appendix B contains the Terms and Definitions.

1.5 Intended Audience

The primary audience for this document is U.S. DOT staff and other identified stakeholders who are leading or are interested in understanding the impact of safety and mobility messages from mobile devices within the envisioned connected vehicle environment where DSRC, Cellular, Wi-Fi, Bluetooth and other communication protocols are utilized by both vehicles and mobile devices. Additional audiences include the system developers, engineers, and any others who will assist in the development of a fully deployed Connected Vehicle environment.

Chapter 2 Applicable Documents

This research is sponsored by the U.S. Department of Transportation as part of on-going research related to the connected vehicle program. As such, there are a number of reports, presentations, and documents on the various aspects of the connected vehicle program that can be found at http://www.its.dot.gov/research_documents.htm. The findings, schematics, results, and conclusions in these documents were routinely consulted and are incorporated in this document. Specific references in the following sections pertain only to documents and works that are not included in this public document repository.

2.1 Non-Government Publications

Institute of Electrical and Electronics Engineers (IEEE)

IEEE 1016-2009	IEEE Standard for Information Technology – Systems Design – Software Design Description
IEEE 1609.x	IEEE Family of Standards for Wireless Access in Vehicular Environments (WAVE) define an architecture and a complementary, standardized set of services and interfaces that collectively enable secure vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) wireless communications.
IEEE 802.11™-2012	IEEE Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

Society of Automotive Engineers (SAE)

J2735:2016	Object Dedicated Short Range Communications (DSRC) Message Set Dictionary – published
J2945/1:2016	On-Board System Requirements for V2V Safety Communications– published
J2945.9 - draft	Performance Requirements for Safety Communications to Vulnerable Road Users

National Transportation Communications for ITS Protocol (NTCIP)

1201:2005 V02.27	Global Object Definitions
1202v03 – draft	Object Definitions for Actuated Traffic Signal Controllers (ASC) – version 03 will include data elements and messages that are coordinated with the SAE J2735 standard, but also data elements for addressing pedestrian and bicycle needs at signalized intersections.

Wi-Fi Alliance

Wi-Fi Peer-to-Peer Services Technical Specification Package v1.2
See (http://www.wi-fi.org/discover-wi-fi/wi-fi-direct) for more

Battelle Memorial Institute

FHWA-JPO-16-422	Task 3: Concept of Operations Document for Coordination of Mobile Devices for Connected Vehicle Applications (3rd Revised Report from July 13, 2016)
FHWA-JPO-16-423	Task 3: System Requirements Specifications for Coordination of Mobile Devices for Connected Vehicle Applications (Final from July 14, 2016)

2.2 Order of Precedence

In the event of a conflict between the text of this document and the references cited herein, the inconsistencies should be brought to the attention of the project manager. Nothing in this document, however, supersedes applicable laws and regulations unless a specific exemption has been obtained.

Chapter 3 System Architecture

3.1 System Description

The EPS for the *Sharing data between mobile devices, connected vehicles and infrastructure project* centers on potentially expanding the current connected vehicle environment by coordinating the mobile devices of pedestrians and vehicles. The EPS will include application software which supports several new types of messages: a Personal Safety Message (PSM); a variety of coordination messages; a Personal Mobility Message (PMM) including PMM Response (PMM-RSP), PMM Arrival (PMM-ARRIVE), and PMM Cancel (PMM-CANCEL) messages; and Basic Safety Messages (BSM), which are further described in the next paragraphs.

1. The PSM is patterned after the Basic Safety Message (BSM) for vehicles. The BSM transmits a vehicle's position, speed, and heading, among other information. Surrounding vehicles use this information in various applications to increase safety by, for example, avoiding collisions. The PSM will provide similar information about the position of an individual carrying a mobile device. It will notify vehicles of the presence, for example, of a pedestrian in a crosswalk or of a runner in the street. Applications can be written for use within vehicles so that their drivers are aware of vulnerable road users (VRUs). The use of PSMs also has, for instance, the potential to reduce injuries in situations where drivers look left while they turn right on red and there are VRUs in the crosswalk. Similar to the BSM, PSM will be broadcast on 5.9GHz DSRC frequency band.
   The PSM proposed to be used in this project is virtually the same as the message type of the same name / acronym defined in SAE J2735:2016 except for the addition of a status field to indicate if the pedestrian is in a safe or unsafe zone.
2. Using Wi-Fi Direct, coordination messages will allow mobile devices to directly and efficiently communicate with each other at a "local level" for the purpose of reducing the overall communication burden of the communications system in terms of the throughput on DSRC radio links. In particular, these coordination messages will be used to temporarily "link" travelers together into ad-hoc travel groups so that only a single message, sent by the group leader (the first person to request a transit vehicle), representing the ad-hoc travel group needs to be transmitted to an infrastructure component or vehicle component rather than individual messages from every member of the group.

   The PMM will enable new applications benefitting a variety of users. PMM messages will be sent to the vehicle via 5.9GHz DSRC frequency band (if within range), or cellular to send a single or ad-hoc travel group travel ride/travel request. It is expected to contain information about the traveler's destination and their requirements for travel (schedule constraints or mobility issues) similar to a traveler trip planning request message used in transit applications with the difference that the PMM would be applicable to any type of vehicle and, at the end, be truly multi-modal. It will also contain information about the constraints on the traveler's trip such as requiring a transit vehicle with a wheelchair lift. A vehicle having received a PMM and agreeing to pick up the passenger(s) will send a PMM response (PMM-RSP). The responding vehicle sends a PMM-ARRIVE message to let the traveler or the group leader know that the vehicle is in the vicinity for the purposes of disbanding the traveler or the group, and that they will soon enter the vehicle. A PMM-CANCEL message is issued from a mobile device back to a vehicle if a traveler or group completely dissolves before the vehicle arrives, or if the traveler or group leader receives multiple PMM-RSP messages from different vehicles.

The proposed system will be utilized in a proof of concept demonstration to understand and to test different communication mechanisms, message content, message timing. That is, the proposed system will be used in a controlled test environment.

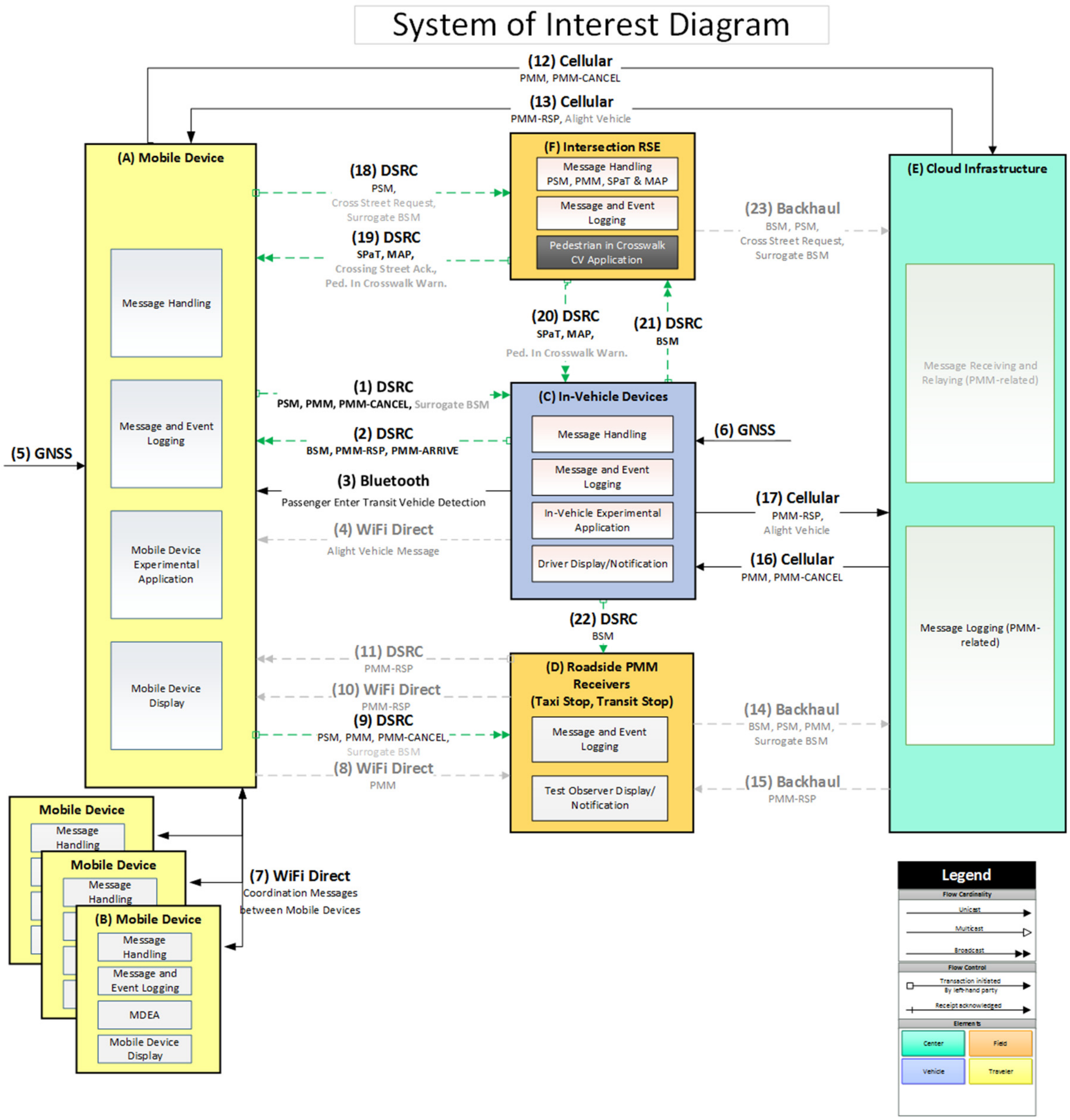
A variety of mobile applications will be developed to support a series of tests, which will explore how multiple technologies can be leveraged to increase traveler safety and improve the traveler experience. The applications are being developed to support the study of two (2) key topics with respect to the inclusion of mobile devices into the Connected Vehicle space. These topics are traveler safety and traveler coordination.

Traveler safety will deal with strategies to detect if a pedestrian is in a situation which is reasonably expected to be safe or if the location or circumstance of the pedestrian is inherently less safe. A pedestrian on the curb is generally in a safe place, whereas a pedestrian in the roadway is not. However, a passenger within a vehicle that is in the roadway inherits the protection of the vehicle's shell, and thus, is also deemed safe. The tests will experiment with a variety of technologies and strategies to ascertain how best to determine the safety of the traveler's location.

Traveler coordination aims to address questions relating to the coordination of groups of travelers and the consolidation of their message to reduce network congestion. If a group of travelers are at a transit stop, and all are utilizing ride-coordination technologies, then the degree of DSRC message traffic generated may adversely affect the system. In situations where several individuals are coordinating with the same target (taxi, transit bus) and itinerary (pick-up location, pick-up time, and destination location), these messages may be consolidated and the load on the communications system reduced.

3.2 System Components

The system consists of four (4) main components, described in the EPS System Component Diagram shown in Figure 1 and exhibits the different components performing certain functions and the communications interfaces between the components including the desired communications media and the data exchange contents. The detailed explanations for this diagram are contained in Section 3.2 - System Description of the "Task 3: System Requirements Specifications (SysRS)."



Note: Communication paths in gray will not be included in the EPS.

Source: Battelle

Figure 1: EPS 'System of Interest' Diagram

3.3 Communications

The four (4) primary components of the system all interact via a combination of communication strategies: DSRC, Bluetooth, Cellular, and Wi-Fi Direct. All of these are used for different functions with the system. Each of these communication technologies are described below along with the message payloads and data signals expected to be transmitted over them.

3.3.1 DSRC

DSRC was developed with a primary goal of enabling technologies that support safety applications and communication between vehicle-based devices and infrastructure-based devices, all with the goal to reduce collisions. - See more at: http://www.its.dot.gov/factsheets/dsrc_factsheet.htm#

DSRC will be used for the exchange of information requiring a low transmission latency. Messages addressing safety issues such as BSMs and PSMs will only be sent via DSRC. The PMMs will also take advantage of DSRC communications. While PMMs do not have the same requirements for low latency, they do benefit from the short range and local aspect of the protocol such as low latency, security and reliability due to designated spectrum when the mobile device and in-vehicle devices are in close proximity to each other (less than 300 meters). In addition to the new messages, mobile devices will also receive Signal Phase and Timing (SPaT) and geographic intersection layout (MAP) messages from an RSU for demonstration purposes only and future use in pedestrian cross road requests.

Below are definitions and a more detailed explanation of each message along with the expected data elements for each message type.

PSM – Personal Safety Message

The PSM message will be broadcasted at a configurable frequency by the mobile device when the travelers are not within a vehicle. By sending a PSM, the mobile device broadcasts its location, direction of travel, and whether it is located within a safe or unsafe zone. Additionally, travelers within an ad-hoc travel group will stop sending their PSMs since their safety is represented by the ad-hoc travel group leader's mobile device. PSM data format is mostly as specified in SAE J2375:2016 with the addition of a field to indicate whether the traveler is a safe or unsafe zone.

- Fields
    - Date [datetime]: date and time that this message information corresponds to.
    - Safety [int, enum lookup]: Safe, Unsafe.
    - Latitude [floating point number]: Latitude where traveler is.
    - Longitude [floating point number]: Longitude where traveler is.
    - Elevation [floating point number]: Elevation where traveler is.
    - Position Accuracy [floating point number]: Position Accuracy of location where traveler is.
    - Speed [floating point number]: Speed of traveler.
    - Heading [floating point number]: Heading of traveler.
    - PSM Number of Peds [int]: Number of pedestrians represented by the PSM.
    - PSM Radius of Protection [floating point number]: Radius from Lat/Long that represents the pedestrian's space. Set in 0.1 meter increments.
    - PSM Path History: The PSM shall specify the pedestrian path history for up to the last 20 seconds. Lat/Long of location, timestamp
    - Path Prediction: The PSM shall specify the pedestrian path prediction for up to the next 5 seconds.

PMM – Personal Mobility Message

These messages are broadcasted at a configurable frequency by traveler's mobile device when they wish to arrange travel with a vehicle and reserve seats. They are continually transmitted by a traveler's mobile device until a response is obtained from a responding vehicle. Travelers with same travel arrangements are grouped, and only one (1) PMM request will be generated to represent entire ad-hoc travel group. PMM message data format is based on the design to meet the requirements of this project.

- Fields
  - GroupId [GUID]: Unique identifier for this travelling group, remains unchanged from first request to final boarding.
  - RequestId [int]: Id within group of iterations of requests, increments with each update.
  - RequestDate [datetime]: date and time that request was created.
  - PickupDate [datetime]: date and time that the travelers wish to be picked up.
  - Status [int, enum lookup]: New, Updated, Done.
  - Latitude [floating point number]: Latitude where travelers will be awaiting pickup.
  - Longitude [floating point number]: Longitude where travelers will be awaiting pickup.
  - Elevation [floating point number]: Elevation where travelers will be awaiting pickup.
  - Destination [string]: destination location expressed in 1/10th integer microdegrees (longitude and latitude with reference to the horizontal datum) where travelers wishes to travel to.
  - Number of Travelers [integer]: Number of travelers in the group.
  - Mode of Transport [integer]: Enumerated preferred method of desired transportation from the following (values are mutually exclusive):
    - Transit
    - Taxi
    - Ride-sharing service
    - No preference
    - A production system would likely include additional transportation modes, but these are intentionally left out of this EPS for simplicity, since it is merely adding more query parameters and does not add value to the proof of concept.
  - Mobility Needs [integer]: Mobility needs requirements from the following (values are not mutually exclusive):
    - Wheelchair
    - Needs Seat
    - No special needs
    - A production system would likely include additional mobility types such as strollers, service animals., but these are intentionally left out of this EPS for simplicity, since it is merely adding more query parameters and does not add value to the proof of concept.
  - ETA threshold [integer]: maximum acceptable deviation value for the originally issued Estimated Time of Arrival (contained in the PMM-RSP Message).

PMM CANCEL – PMM Cancellation

Cancellation of a PMM Request is supported by setting the following field values of a PMM message, fields not specified can be any value and serve no function: PMM Cancel message data format is based on the design to meet the requirements of this project.

- Fields
  - RequestDate = date and time that request was created.
  - RequestId [int] : Return of RequestId of PMM request to be cancelled.

PMM ARRIVE – Vehicle Arrival

Transmitted by In-Vehicle Experimental Application when vehicle arrives at pickup location. PMM Arrive message data format is based on the design to meet the requirements of this project.

- Fields
  - RequestId [int]: RequestId of PMM this PMM-ARRIVE is in response to.
  - Latitude [floating point number]: Latitude of vehicle location.
  - Longitude [floating point number]: Longitude of vehicle location.
  - Elevation [floating point number]: Elevation of vehicle location.
  - ETA [integer]: Seconds until arrival; e.g. zero if arrived.
  - VehicleDesc [text]: a visible description indicator of the vehicle to aid in human identification of the vehicle.

PMM RSP – PMM Acknowledgement/Response

Transmitted by In-Vehicle Experimental Application whenever a PMM request is received. PMM RSP message data format is based on the design to meet the requirements of this project.

- Fields
  - GroupId [GUID]: Return of GroupId of PMM this PMM-RSP is in response to.
  - RequestId [int] : Return of RequestId of PMM this PMM-RSP is in response to.
  - Latitude [floating point number]: Latitude of vehicle location.
  - Longitude [floating point number]: Longitude of vehicle location.
  - Elevation [floating point number]: Elevation of vehicle location.
  - IsDSRCEquipped [Boolean]: True if the vehicle is capable of broadcasting and receiving DSRC messages.

BSM – Basic Safety Message

Transmitted by vehicles at a configurable frequency. Broadcasts their location and direction. BSM data format will be as specified in SAE J2375:2016.

- Fields
  - Date [datetime]: date and time that this message information corresponds to.
  - VehicleId [GUID/null]: unique identifier of the vehicle (application) originating the message. Useful for travelers checking on status of arranged ride.
  - Latitude [floating point number]: Latitude where vehicle is.
  - Longitude [floating point number]: Longitude where vehicle is.
  - Elevation [floating point number]: Elevation where vehicle is.
  - Speed [floating point number]: Speed of vehicle.
  - Heading [floating point number]: Heading of vehicle.
  - Size [integer]: Width and length of the vehicle, expressed in 1 cm increments.

SPaT Data

Transmitted by intersection RSE at a configurable frequency to demonstrate future use with Pedestrian Crossing Street Request. SPaT data format will be as specified in SAE J2375:2016

- Fields
  - IntersectionID [integer]: a unique identifier for the intersection
  - SignalGroup [integer]: an id used to map to a group of lanes which correlates to a particular movement state
  - EventState [integer]: indicates the particular phase for the associated signalGroup
  - ConnectionId [integer]: a value used to indicate which lane in the MAP file is associated with the pedestrian detections
  - pedBicycleDetect [Boolean]: used to indicate if a pedestrian is detected in the lane with the same connectionId

MAP Data

Transmitted by intersection RSE at a configurable frequency to demonstrate future use with Pedestrian Crossing Street Request. MAP data format will be as specified in SAE J2375:2016

- Fields
  - IntersectionID [integer]: a unique identifier for the intersection
  - Reference Point [floating point number]: Latitude and Longitude point which serves as a reference for the roadway geometry
  - LaneId [integer]: a unique id number assigned to the lane
  - ConnectionId [integer]: an identifier used to link lanes to connection information in a SPAT message
  - Maneuvers [bit string]: the permitted maneuvers for the lanes
  - NodeList [NodeSetXY]: list of nodes providing specific x and y offsets for the lane

3.3.2 Bluetooth

With the prevalence of Bluetooth on the vast majority of mobile devices, it makes a good candidate for a communication mechanism for detecting presence of a passenger in a vehicle. Bluetooth will be used for exploring whether a device has entered or exited a transit vehicle. Based on the detected signal strength received from one (1) or more Bluetooth LE (iBeacon or Eddystone devices) located on a transit vehicle, a determination is expected to be made indicating that the pedestrian has entered or exited the vehicle. No custom data other than that provided by the Bluetooth LE protocol specification is expected to be transferred in this scenario.

Bluetooth is also planned to be used internally within the mobile device subsystem for communication between the phone and the portable DSRC radio due to lack of availability of a phone that supports DSRC communication for this project. Therefore, the mobile device component is broken down into an Android based smartphone and a portable DSRC radio. These two (2) devices will exchange information over a Bluetooth communication link.

3.3.3 Cellular

Cellular communication is used by both the mobile device and the in-vehicle device to communicate with the Cloud Infrastructure, either to query data or to retrieve logging entries. Cellular is also used to interact with the Web Service, in which the contents of PMM and PMM-RSP DSRC-compatible messages types are repackaged into a Web request. This is an alternate path for transmitting the PMM and/or PMM CANCEL and/or PMM-RSP message types.

3.3.4 Wi-Fi Direct

Wi-Fi Direct, also called Wi-Fi Peer-to-Peer (P2P), is a Wi-Fi standard enabling devices to easily connect with each other without requiring a wireless access point. Wi-Fi Direct negotiates the link with a Wi-Fi protected Setup system that assigns each device a limited wireless access point. The "pairing" of Wi-Fi Direct devices can be set up to require the proximity of a near field communication, a Bluetooth signal, or a button press on one (1) or all the devices[1].

Wi-Fi Direct is utilized by the system to negotiate travel needs between co-located travelers with the same destinations. Requests among mobile devices of travelers with the same travel itinerary are pooled to reduce network traffic. Communication between members of Wi-Fi Direct group are shown in Table 1.

[1] https://en.wikipedia.org/wiki/Wi-Fi_Direct. 12/2015

Table 1: Communication between Wi-Fi Direct group members

Action	Action Originates From:	Action Description	Response Returned
Query For Matching Group	Follower	New user seeking travel queries existing group with travel criteria to see if there is a match.	True if match exists, otherwise False.
Request To Join Group	Follower	New user requesting to join existing group.	Accept/Reject
Periodic Update	Leader	Leader sends Group Info update on status of group to each Follower	Follower responds with acknowledgement to show they are still present.
Rejection From Group	Leader	Sent from Leader if the vehicle could ultimately not accommodate the seats requested for this user.	None

Source: Battelle

CoordinationRequestMessage

These messages are transmitted by the traveler's mobile devices at a system-wide configurable frequency along with a PMM but via Wi-Fi Direct when they wish to arrange travel with a vehicle and reserve seats to form/join a travel group with similar travel requirements. The CoordinationRequestMessages are continually transmitted until a confirmation response is obtained from a leader's mobile device.

- Fields
  - CoordinationRequestId [int]: Unique identifier for Coordination request.
  - PickupDate [datetime]: date and time that the travelers wish to be picked up.
  - Status [int, enum lookup]: New, Updated, Done.
  - Latitude [floating point number]: Latitude where travelers will be awaiting pickup.
  - Longitude [floating point number]: Longitude where travelers will be awaiting pickup.
  - Elevation [floating point number]: Elevation where travelers will be awaiting pickup.
  - Destination [string]: destination location expressed in 1/10th integer microdegrees (longitude and latitude with reference to the horizontal datum) where travelers wishes to travel to.
  - Number of Travelers [integer]: Number of travelers in the group.

- Mode of Transport [integer]: Enumerated preferred method of desired transportation from the following (values are mutually exclusive):
    - Transit
    - Taxi
    - Ride-sharing service
    - No preference
- Mobility Needs [integer]: Mobility needs requirements from the following (values are not mutually exclusive):
    - Wheelchair
    - Needs Seat
    - No special needs
    - A production system would likely include additional mobility types such as strollers, service animals., but these are intentionally left out of this EPS for simplicity, since it is merely adding more query parameters and does not add value to the proof of concept.

CoordinationRequestConfirmation

These messages are transmitted by the group leader's mobile device in response to a CoordinationRequestMessage to confirm the creation and/or joining of the travel group.

- Fields
    - CoordinationRequestId [int]: Unique identifier for Coordination request.
    - GroupId [GUID]: Unique identifier for this travelling group, remains unchanged from first request to final boarding.
    - RequestId [int]: Id within group of iterations of requests, increments with each update.

CoordinationRequestRejection

These messages are transmitted by the group leader's mobile device in response to a CoordinationRequestMessage to reject a request to join to the group.

- Fields
    - CoordinationRequestId [int]: Unique identifier for Coordination request.
    - GroupId [GUID]: Unique identifier for this travelling group, remains unchanged from first request to final boarding.

CoordinationHeartBeatMessage

These messages are sent by members of the ad-hoc travel group to the ad-hoc travel group leader's mobile phone to maintain membership in the travel group within a configurable frequency.

- Fields
  - CoordinationRequestId [int]: Unique identifier for Coordination request.
  - GroupId [GUID]: Unique identifier for this travelling group, remains unchanged from first request to final boarding.

CoordinationEndMessage

These messages are sent by members of the ad-hoc travel group to the ad-hoc travel group leader's mobile phone to leave the travel group.

- Fields
  - CoordinationRequestId [int]: Unique identifier for Coordination request.
  - GroupId [GUID]: Unique identifier for this travelling group, remains unchanged from first request to final boarding.

CoordinationEndTravelGroup

These messages are sent by the ad-hoc travel group leader's mobile phone to disband the ad-hoc travel group.

- Fields
  - GroupId [GUID]: Unique identifier for this travelling group, remains unchanged from first request to final boarding.

3.4 Design Constraints

3.4.1 General Constraints

- The only known battery-operated DSRC radio, the Arada System's Locomate ME OBU radio contains only a single 5.9GHz radio.
- Wi-Fi Direct is currently only supported on Android devices. The Wi-Fi Direct range is dependent on mobile device transmission power which is typically about 60 meters. If the leader gets out of range, the connection to peers will drop and another device will (need to) assume the leader's role and re-establish the group, which may be a challenge to implement.
- The experiments assume continuous coverage throughout the demonstration environments of both cellular (3G and 4G/LTE) and DSRC.

- Prior investigations and technical literature[2] indicates that the best position accuracy that can be achieved with conventional low cost GPS and Differential GPS (DGPS) technology is three (3) meters. Battelle investigators found that Ublox Precise Point Precision[3] GPS position accuracy was adequate to support which-lane level accuracy of approximately two (2) meters during a test period, but drift in GPS position prevented achieving where-in-lane level accuracy. However, these units required systems comparable to laptop computers in size with similar power requirements. While they could be integrated in mobile devices in the future, we will need to work with the capabilities of the commercially available equipment and monitor the location accuracy we can achieve during experimental demonstrations.
- Prior investigations by Battelle and others suggest that mobile devices may be able to achieve road level position accuracy necessary to support driver advisories. Testing will be required to determine if they can reliably support lane-level accuracy needed to support driver alerts of pedestrians in unsafe zones or roadway lanes. More technology development and engineering will likely be necessary for mobile devices to achieve where-in-lane level accuracy necessary to support driver warnings of pedestrians in the vehicle path and Stop warning of imminent collision.

[2] Beauregard, S. and H. Haas, Pedestrian Dead Reckoning: A Basis for Personal Positioning. International University Bremen School of Engineering and Science. Proceedings of the 3rd Workshop on Positioning, Navigation, and Communication. 2006.
[3] https://www.u-blox.com/sites/default/files/products/documents/NEO-7P_ProductSummary_(UBX-13003350).pdf

Chapter 4 Design Concerns

This section enumerates and expands upon the primary design concerns. Design concerns are specific areas of stakeholder interest that need to be accounted for during the design efforts. Design concerns do not always directly impact tangible design decisions or system features, but sometimes result in systemic properties of a system.

4.1 User Interfaces

The following list enumerates the components of the system and the supported user interfaces.

- Mobile Device Interface System – This component will have a graphical user interface that displays information relevant to the current test that is underway. Users will be able to initiate actions and get feedback on the status of the current test.
- In-Vehicle Device Interface System – This component will have a graphical user interface that displays information relevant to the current test that is underway. Users will be able to initiate actions and get feedback on the status of the current test.
- Roadside Receiver – This component will have a graphical user interface that displays information relevant to the current test that is underway. This interface will be able to show DSRC messages that are being actively transmitted – regardless of whether or not a taxi or transit bus are in range to receive them.
- Cloud Service – The cloud service(s) will provide a user interface to view the PMMs being relayed between mobile devices and in-vehicle devices via cellular.

4.2 Networking and Communications

The design of the networking architecture will strive to minimize the end-to-end latency of a single BSM message on its journey between devices. The bulk of the volume of communications data in the system will in all likelihood be emanating from vehicles sending BSMs and mobile devices broadcasting PSM and PMM and cross street request messages over DSRC. If the networking communications do not operate efficiently, latency introduced at various hops along the way will accumulate into a significant amount by the time a PSM or PMM has reached its final destination. This amount of delay is compounded as more and more vehicles and mobile devices enter the system. An effective design should define a strategy for profiling latency, for instance by recording timestamps at each hop and later using comparative analysis techniques to identify bottlenecks in the system as network volume is increased. However, determination of message broadcast loading is outside the scope of EPS project and will not be investigated.

4.3 Security

Security is fundamental to the success of the Connected Vehicle program, and tremendous efforts on the part of U.S. DOT, vehicle original equipment manufacturers (OEMs), and other industry partners have been performed and continue to be undertaken to ensure that all aspects of the Connected Vehicle environment are secure. As such, no new security techniques will be developed under this task order, but instead, security requirements will be fulfilled by using current approaches.

Specific to connected vehicles, the efforts have primarily focused on the DSRC-based over-the-air (OTA) security between vehicles and between infrastructure and vehicles - approaches, which include signing and encrypting OTA data using public key / private key digital certificates and secure communication layers. This work continues to evolve, but for purpose of the EPS, the OTA security mechanism as it exists today (developed as part of the U.S. DOT Safety Pilot Model Deployment using a security credential management system (SCMS) and a secure DSRC-stack) will be implemented. Details of this approach are available in the *Security Credential Management System Design* – April 12, 2012, available from U.S. DOT, as well as the IEEE 1609 family of standards. For purposes of this prototype demonstration, the Battelle team will serve as the certificate authority and will self-certify the devices used.

The above addresses only the DSRC aspect of over-the-air security; however, the same application of self-signed digital certificates will be used for the cellular-based communications to be implemented along with industry best-practices for secure communications, such as the use of secure socket layers (SSL). Finally, the cloud-based data repository and computing platform will also use industry-based best-practices. Secure network communications between the mobile device and cloud-based server or another device will be using *Transport Layer Security* (TLS) protocol, also referred to as the *Secure Sockets Layer* (SSL).

4.4 Protection of Personally Identifiable Information

These tests do not request nor save any Personally Identifiable Information (PII) or Sensitive Personally Identifiable Information (SPII).

4.5 Packaging

Packaging in this context refers to the hardware components that will be utilized to carry out the demonstrations. In particular, the mobile device is the primary hardware output of this effort, and as currently envisioned, will be comprised of a combination of existing, commercial-off-the-shelf (COTS), components, that when integrated, will provide the functionality required of the said device.

The two (2) primary hardware components that comprise the mobile device are the smartphone and the Bluetooth-connected battery-powered DSRC radio. For this test, the Project Team is using the battery powered LocoMate™ ME on-board unit (OBU) with integrated GPS and DSRC antenna, also known as the “Backpack” the Mobile Device DSRC radio. The backpack form factor is used with the Google Nexus 5X smartphone as an integrated “Mobile Device”.

4.6 Power

Similar to the prior section related to packaging, this section discusses the power requirements of the mobile device and the corresponding in-vehicle network access system.

The mobile device will be battery powered such that it can be removed from the vehicle power source and truly demonstrated as being nomadic. This feature ensures the DSRC radio will continue to transmit Basic Safety Messages at the 10Hz frequency throughout this duration. The main batteries (there are two (2) – one for the smartphone and one for the DSRC radio) are rechargeable and are expected to last at least two (2) hours without being recharged. It should be noted that truly low-powered, mobile DSRC chipsets are only now being developed and are not available for this prototype.

The device will also support operating/charging when installed in the vehicle through the use of a simple accessory port (formerly cigarette lighter) cable which will supply power at 12VDC to the device. The time to fully recharge in this manner has not yet been determined, but is expected to be similar to times typical of current generation smartphones.

Finally, the mobile device will support charging from standard household outlets (110-120 VAC) using an AC-to-DC transformer. Again, the recharge time has not yet been determined, but is expected to be similar to current generation smartphone charge.

The in-vehicle Network Access System will draw power from the vehicle directly through the OBD-II port. The module was designed specifically for semi-permanent installation in the vehicle, and, as such, has a lower-power mode that will detect when the ignition is off and switch to this mode in order to preserve vehicle battery power when the device / vehicle is not in use.

Chapter 5 Components Description

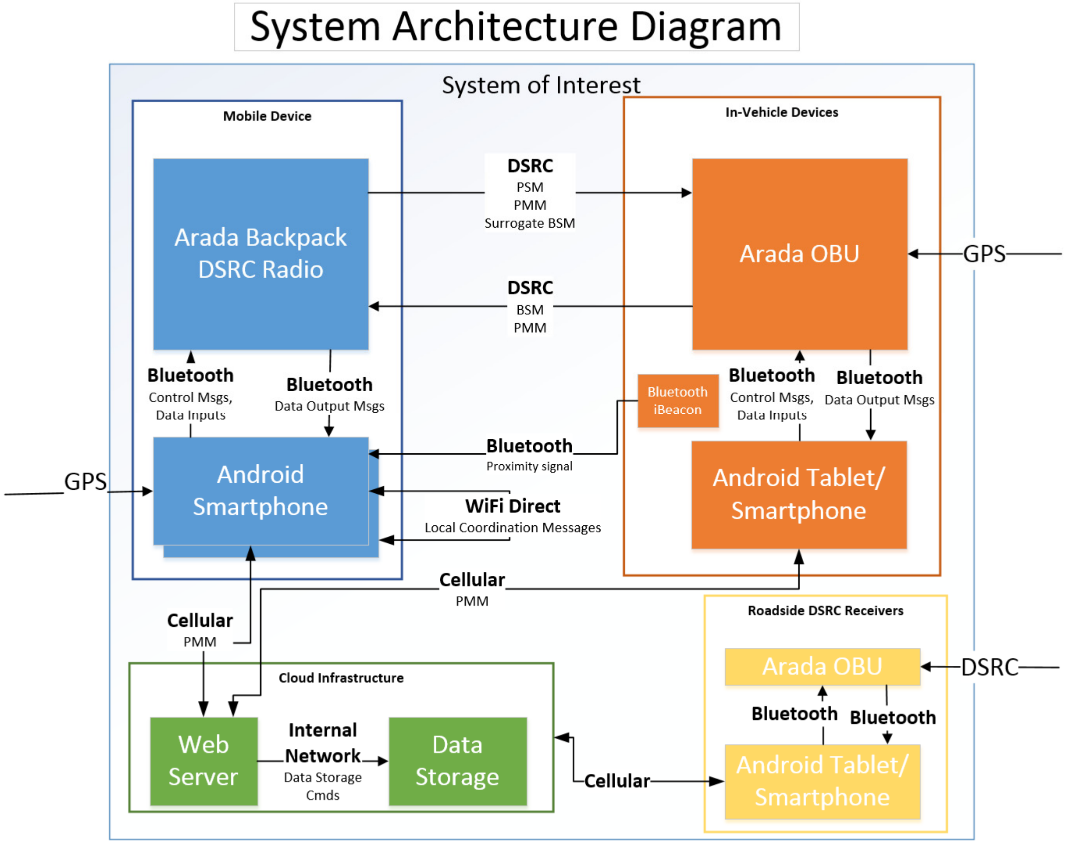
This section identifies the components that will make up the system; both logically and physically. Those components will then be further decomposed into subcomponents where necessary. Finally, software modules will be identified and described for each of the system subcomponents.

5.1 System Decomposition

The objective of this section is to divide the EPS into separate components that can be considered, implemented, modified, and tested with minimal effect on other entities. The system can be broken into a first level decomposition as follows:

- Mobile Device
- In-Vehicle Device
- Roadside Receivers
- Cloud Infrastructure

Figure 2 shows a high level component diagram that models the physical deployment of the above components. Each mobile device, in-vehicle device, and roadside receiver will be comprised of an Android tablet/phone and a DSRC Radio. The Android tablet/phone will have a deployed application that will contain the custom core logic and support modules. The custom core software is unique to that type of component the device represents – e.g. containing the ride request logic when serving as a mobile device, and the ride response logic when serving as an in-vehicle device. The support modules are any of the support libraries needed to support the core logic as discussed in Section 8.2.5 Common Components. The communication between the mobile device and the in-vehicle device over distances exceeding the limits of DSRC will be performed using the cloud infrastructure composed of a cellular network and the database backend.



Source: Battelle

Figure 2: High Level Component Diagram

5.2 Component Decomposition

Extending the decomposition down yet another layer, some of the system components can be broken into subcomponents. The mobile device, the in-vehicle device, the roadside receiver and the cloud infrastructure are sufficiently complex that they can each be broken down into two (2) subcomponents.

As explained earlier in this document, the ideal platform for the mobile device would be a modern smartphone that also includes the capabilities of sending and receiving SAE J2735 compatible messages over DSRC. Unfortunately, at the time of this design, such a device is not commercially available to this project. Therefore, the mobile device component is broken down into an Android based smartphone and a portable DSRC radio. These two (2) devices operate independently and are coupled by a Bluetooth communications channel. The Android smartphone will be the primary computational device for the mobile device.

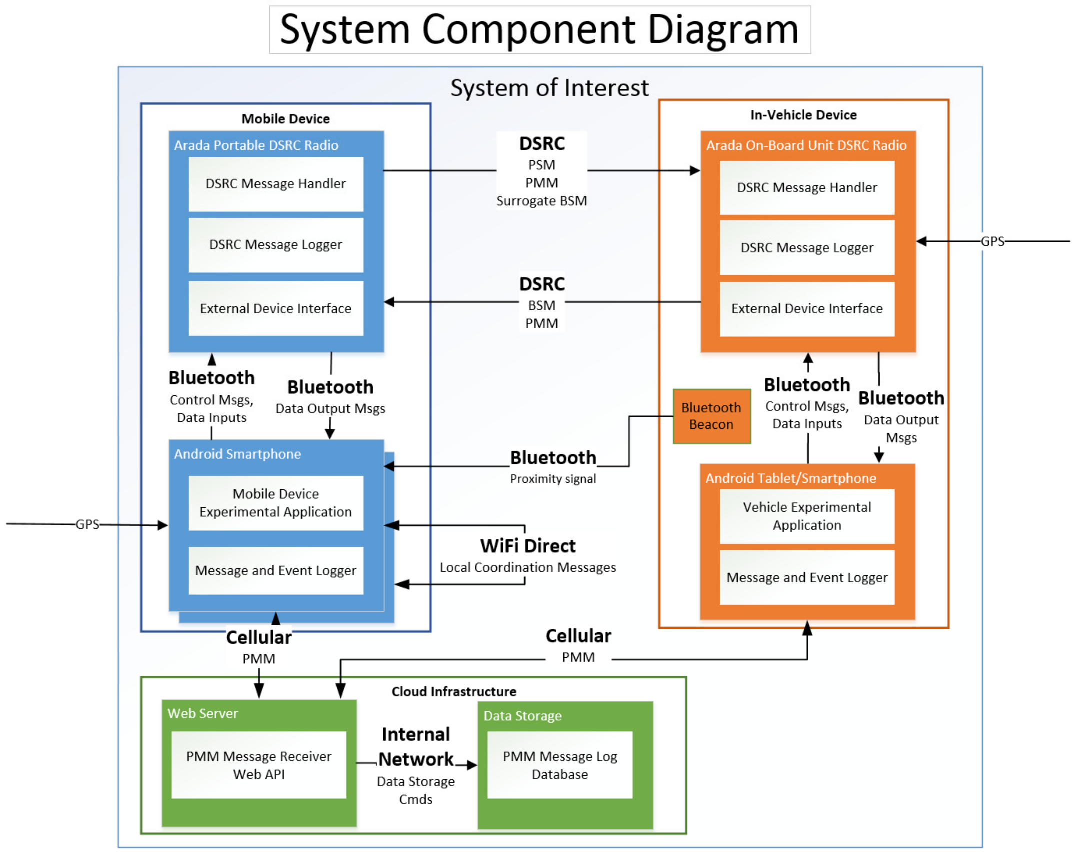
The in-vehicle device is also broken down into two (2) subcomponents in a fashion similar to the mobile device. The need for the two (2) devices are however slightly different. The in-vehicle device component is composed of an on-board unit (OBU) DSRC radio and an Android smartphone. Like the two (2) subcomponents which make up the mobile device, these two (2) components will also operate independently and use a Bluetooth communication channel for inter-device communications. The Android smartphone will be the primary computational device for the in-vehicle device.

The third component, the roadside receiver, will be very similar to the in-vehicle device, other than the Android application targeted to roadside receiver-specific logic instead.

The fourth component to be decomposed into subcomponents is the cloud infrastructure. For this EPS, the cloud infrastructure will have two (2) subcomponents. The first is a Web Server, which will be providing the communications interface for both the mobile device and the in-vehicle device when communicating via cellular. The second subcomponent will be data storage, which will be in place to facilitate the hand-off of message data between mobile and in-vehicle devices. The Web Server communicates with the data storage over the Internet using a URL access point. This data storage will also be critical in preserving the message traffic for later analysis and reporting when evaluating the results of the experiments.

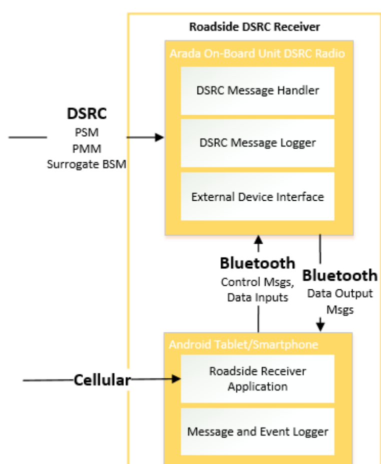
5.3 System Component Descriptions

Each of the system subcomponents represents a physical piece of hardware, or, in the case of the subcomponents of the cloud infrastructure, a virtualized version of a piece of hardware. Each of those subcomponents will contain one (1) or more software modules. The diagram in Figure 3 illustrates the software modules associated with each of the essential system subcomponents. Figure 4 illustrates the software modules associated with the roadside receivers that serve the role of monitoring and recording.



Source: Battelle

Figure 3: Experimental Prototype System Component and Module Diagram



Source: Battelle

Figure 4: Roadside Receiver Component and Module Diagram

The following section will provide a greater description of each system component. These details will cover each of the subcomponents and their associated software modules. In addition, a description of the physical interfaces and expected message traffic for each subcomponents is defined.

5.3.1 Mobile Device

Subcomponents

<u>Android Smartphone</u>

This subcomponent will encompass the functionality provided by the Android smartphone. Most of the required functionality of the mobile device can be satisfied by the Android smartphone with the exception of the DSRC communications. The software modules and interfaces for this subcomponent are described below along with those of the Arada Portable DSRC Radio.

<u>Arada Portable DSRC Radio</u>

The Arada Portable DSRC radio will be responsible for providing the DSRC communication needs of the mobile device. The transmission and receipt of all messages sent over DSRC will be processed on this platform as well as message decoding and formatting. The software modules and interfaces for this subcomponent are described below along with those of the Android smartphone.

Software Modules

<u>Mobile Device Experimental Application (MDEA)</u>

The Mobile Device Experimental Application (MDEA) is the application that resides on the smartphone of the traveler. It has three (3) primary sub-modules:

- Pedestrian Safety Monitor – to monitor incoming BSM messages and transmit PSM messages.
- Ride Request Monitor – to manage PMM messages and travel groups
- MDEA User Interface – to manage user interaction with Safety and Travel monitors.

<u>Message and Event Logger</u>

The Message Logger module will record in storage the time and contents of all Wi-Fi Direct messages that are sent or received by the MDEA. It will also record events and noteworthy decision information of the MDEA.

A SQL Lite database will be used on the mobile device to store Wi-Fi Direct message data. Note: Other messages exchanged over Cellular or DSRC communication medium are stored either by the RSU or the cloud service. Classes and data schema will be established for each type of message to be stored. This will be translated into a database structure via the contract support functionality of the Android code base.

Wi-Fi Direct Message

- Timestamp
- Direction [incoming/outgoing]
- Payload

General

- Timestamp
- [Error/Warning/Info/Debug]
- Message
- StackTrace or Class/Function Name

A Content Provider manages access to a structured set of data. It encapsulates the data, and provides mechanisms for defining data security. Content providers are the standard interface that connects data in one (1) process with code running in another process. The Content Provider leverages the SQL Lite Database for storage and may additionally be necessary to support extracted data.

<u>DSRC Message Handler</u>

The DSRC Message Handler module will receive and parse incoming DSRC Messages. It will also process, package, and transmit outgoing DSRC Messages.

<u>External Device Interface</u>

The External Device Interface will support the communication needs between the DSRC radio platform and that of the Android smartphone. This software module will receive the information requests and configuration commands from the smartphone and respond by making the necessary changes or by providing the requested information.

Physical Interfaces

<u>DSRC Interface</u>

The DSRC Interface module will manage the communication of data through the DSRC Radio. The protocol and frequencies used are provided by the radio manufacturer. The specific format of the messages will follow the J2735 message specification where possible with modifications or alterations made where necessary in support of the required experiments being performed.

<u>Wi-Fi Direct Interface</u>

The Wi-Fi Direct Interface module will manage the Wi-Fi Peer-to-Peer (P2P) library that allows Android devices to connect directly to each other via Wi-Fi without an intermediate access point. This module will wrap the available Wi-Fi P2P Application Programming Interface (API) to provide needed functionality to the in-vehicle experimental application.

GPS Interface

The GPS Interface module will receive and parse GPS information from the integrated GPS receiver on the phone/tablet. This information will include:

- Latitude
- Longitude
- Timestamp
- Speed
- Heading

Bluetooth Interface

The Bluetooth Interface module will handle managing connections to other Bluetooth devices. It will receive and parse incoming Bluetooth data packets from paired devices. It will handle sending outgoing messages out to paired devices. This Bluetooth Interface will be managed by the External Device Interface on the Arada DSRC radio platform and by the MDEA on the Android Smartphone. It will also be communicating with the Bluetooth beacon used for detection of entering and exiting a vehicle.

Cellular Interface

The Cellular Interface module will maintain contact information and implementation details in order to make requests to the Web Server. This interface will be managed by the MDEA.

5.3.2 In-Vehicle Device

Subcomponents

Android Smartphone

The Android tablet or smartphone subcomponent in the in-vehicle device will serve a similar function as the one in the mobile device. In the in-vehicle device, this platform will be the means for the visual display and notification to the driver of the vehicle.

Arada On-Board Unit DSRC Radio

This subcomponent will provide the in-vehicle device with a means of sending and receiving DSRC messages. Transmission and receipt of J2735-compatible Messages sent over DSRC will be handled by this platform as well as message decoding and formatting.

Software Modules

In-Vehicle Experimental Application (VEA)

The In-Vehicle Device Experimental Application (VEA) is the application that resides on the smartphone/tablet of the bus/taxi. It has three (3) primary sub-modules:

- VEA Pedestrian Monitor - to transmit BSM messages and monitor PSM messages.
- VEA Ride Request Monitor – to monitor PMM messages and respond to them
- VEA User Interface - to manage user interaction with Safety and Ride monitors.

<u>Message and Event Logger</u>

The Message Logger module will record in storage the time and contents of all Wi-Fi Direct messages that are sent or received by the VEA. It will also record events and noteworthy decision information of the VEA.

A SQL Lite database will be used in the in-vehicle device to store message data. Classes and data schema will be established for each type of message to be stored. This will be translated into a database structure via the contract support functionality of the Android code base.

Wi-Fi Message

- Timestamp
- Direction [incoming/outgoing]
- Payload

General

- Timestamp
- [Error/Warning/Info/Debug]
- Message
- StackTrace or Class/Function Name

A Content Provider manages access to a structured set of data. It encapsulates the data, and provides mechanisms for defining data security. Content providers are the standard interface that connects data in one (1) process with code running in another process. The Content Provider leverages the SQL Lite Database for storage and may additionally be necessary to support extracted data.

<u>DSRC Message Handler</u>

The DSRC Message Handler module will receive and parse incoming DSRC Messages. It will also process, package, and transmit outgoing DSRC Messages.

<u>External Device Interface</u>

The External Device Interface will support the communication needs between the DSRC radio platform and that of the Android smartphone. This software module will receive the information requests and configuration commands from the smartphone and respond by making the necessary changes or by providing the requested information.

Physical Interfaces

<u>DSRC Interface</u>

The DSRC Interface module will manage the communication of data through the DSRC Radio. This interface for the in-vehicle device will follow the same design considerations similar to what is used on the mobile device.

GPS Interface

The GPS Interface module will receive and parse GPS information from the integrated GPS receiver on the Arada OBU. This information will include

- Latitude
- Longitude
- Timestamp
- Speed
- Heading

Bluetooth Interface

The Bluetooth Interface module will handle managing connections to other Bluetooth devices. It will receive and parse incoming Bluetooth data packets from paired devices. It will handle sending outgoing messages out to paired devices. This Bluetooth Interface will be managed by the External Device Interface on the Arada OBU platform and by the VEA on the Android smartphone.

Cellular Interface

The Cellular Interface module will maintain contact information and implementation details in order to make requests to the Web Server. This interface will be managed by the VEA.

5.3.3 Roadside Receiver

Subcomponents

The Roadside Receiver is a component used as a third party observer during the experiments and demonstrations and as a DSRC-message type generator for SPaT and MAP messages. The approach planned is to utilize the subcomponents from the in-vehicle device to serve the third party observer purpose.

Software Modules

Roadside Receiver Application (RRA)

The Roadside Receiver Application (RRA) is the application that resides on the smartphone/tablet on the roadside, nearby a passenger pick-up stop. It serves to receive information and display it to the user – in particular, it will show DSRC traffic that may be occurring prior to the bus/taxi becoming in range to hear it. It has two (2) primary sub-modules:

- RRA Monitor – to log BSM, PSM and PMM messages received via cellular and DSRC interfaces
- SPaT& MAP message generator - to broadcast simulated SPaT and MAP messages

Physical Interfaces

DSRC Interface

The DSRC Interface module will manage the communication of data through the DSRC Radio. This interface for the roadside receiver will follow the same design considerations as that which is used on the mobile and in-vehicle devices.

Cellular Interface

The Cellular Interface module will maintain contact information and implementation details in order to make requests to the Web Server. This interface will be managed by the RRA.

5.3.4 Cloud Infrastructure

Subcomponents

Web Server

The Web Server subcomponent is to be the entity in the cloud, which the mobile and in-vehicle devices communicate with. This subcomponent will accept HTTP connections from the smartphone devices and respond to offered requests. This Web Server will be the interface point for all communications over cellular.

Data Storage

Because the cellular communication channel between the mobile and in-vehicle devices are not direct connections, the information being exchanged must be stored while it awaits retrieval by the intended recipient. In addition, this project adds additional requirements around the testing and evaluation of these techniques and technologies which require the persistent storage of the message contents as well as the timings associated with the receipt and reception.

Software Modules

PMM Message Receiver Web API

The PMM Message Receiver Web API module will provide a Web API interface that will be utilized by both the mobile devices and the in-vehicle devices to exchange ride request information. The Web API will consist of functions which allow for requests to be added to the system, checking for pending requests, and responding to requests.

PMM Message Log Database

A Microsoft Azure SQL Database module will provide database functionality in order for the Web Service to be able to create and modify database entries. The database will be leveraged to route PMM messages between a traveler and a vehicle prior to being within DSRC range. In addition, the data recorded in this database will also be used during the analysis of the experimental demonstrations. All Cloud message traffic will also be recorded here.

Azure Storage

A Microsoft Azure Table Storage module will be used to record general diagnostic information concerning the health and stability of the Web Service.

Physical Interfaces

Network Interfaces

The interface to the cloud infrastructure will be via a TCP/IP network. Connections made from the Android smartphones within either the mobile device or the in-vehicle devices will be made over the cellular network. Communication between subcomponents in the cloud will be internal TCP/IP network connections.

Modules that exist across different parts of the system will utilize the same design and architecture. For example, the Cellular Message Handler in the in-vehicle device will operate identically to the Cellular Message Handler in the mobile device.

Chapter 6 Internal Interfaces

Internal interfaces consist of the following intra-system component interfaces:

- Mobile Device to Mobile Device
- Mobile Device or In-Vehicle Device to Web API

Each of these are described in detail below.

6.1 Mobile Device to Mobile Device

Smartphones/tablets serve as components of the MDEA, VEA, and the RRA applications. All of these implementations share similar interface support requirements.

Table 2 provides for a description of the origin and destination of each of the messages communicated between the smartphones/tablets of this system.

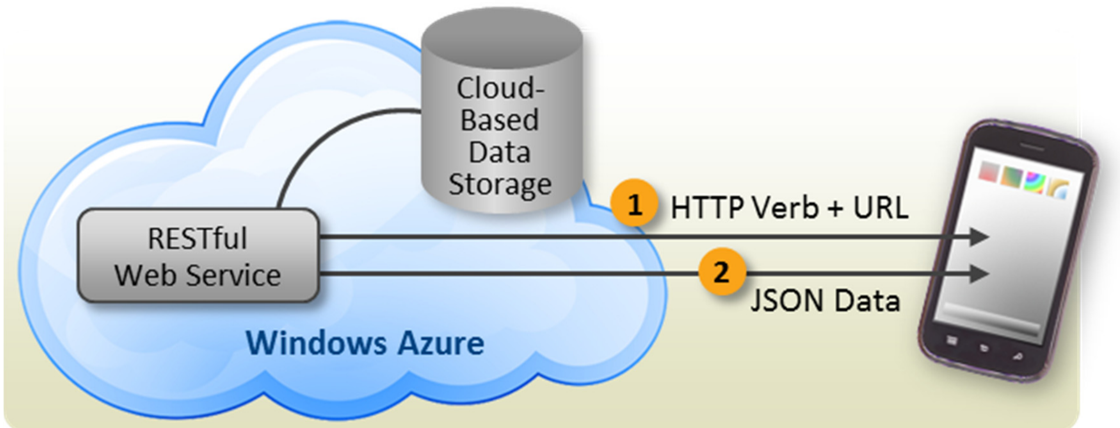
Table 2: Mobile to Mobile Internal Interfaces of the System

Message Types	Origins / Destinations of the Message Types
Basic Safety Message (BSM) Part One	• Transmitted message will conform to the SAE J2735:2016 specification. • Transmitted from the VEA Mobile Device from VEA Pedestrian Monitor. • Received and processed by MDEA Mobile Device Pedestrian Safety Monitor. • Received and processed by RR Mobile Device.
Personal Safety Message (PSM)	• Transmitted message will conform to the SAE J2735:2016 specification. • Transmitted from the MDEA Mobile Device based on Pedestrian Safety Monitor. • Received and processed by VEA Mobile Device
PMM Request	• Transmitted message will conform to the message configuration as used for other message types defined SAE J2735:2016 specification. • Transmitted from the MDEA Mobile Device based on Ride Request Monitor. • Received and processed by Cloud Service and VEA Mobile Device
PMM RSP	• Transmitted message will conform to the message configuration as used for other message types defined SAE J2735:2016 specification. • Transmitted from the VEA Mobile Device based on VEA Travel Monitor. • Received and processed by MDEA Mobile Device
PMM ARRIVE	• Transmitted message will conform to the message configuration as used for other message types defined SAE J2735:2016 specification. • Transmitted from the VEA Mobile Device based on VEA Travel Monitor. • Received and processed by MDEA Mobile Device
SPaT Data Message (SPAT)	• Transmitted message (simulate) will conform to the SAE J2735:2016 specification. • Transmitted from the RR Application at a configurable interval. • Received and processed by RR Application
Map Data Message (MAP)	• Transmitted message (simulated) will conform to the SAE J2735:2016 specification. • Transmitted from the RR Application at a configurable interval. • Received and processed by RR Application
Wi-Fi Direct Coordination Request Messages	• Transmitted between multiple MDEAs. • Received and processed by the MDEAs.

Source: Battelle

6.2 Mobile Device to Cloud API

The Cloud Service will host a secure web server over HTTPS (Hypertext Transfer Protocol Secure). This secure web server will implement a simple Representational State Transfer (REST) architecture to service requests directly from the mobile device via a cellular connection originated from the smartphone component of the mobile device.



Source: Battelle

Figure 5: Cloud RESTful Web Server

A mobile device web client will request a persistent HTTPS connection (Keep-Alive) with the Cloud Web Server to reduce the overhead otherwise required to repeatedly open temporary HTTPS client connections (see Figure 5). The Section *Azure Web Service* provides for details on the specific Web API functions that will be available.

Chapter 7 External Interfaces

No external resources will be leveraged as a part of this system. Thus there are no external references.

Chapter 8 Component Design

As previously discussed, the EPS is comprised of four (4) major components that are integrated into a single system that will enable the execution of the experimental testing plan. These components are as follows:

- Mobile Device – Smartphone and DSRC Radio
- In-Vehicle Device – OBU DSRC with Tablet and Bluetooth iBeacon
- Cloud Infrastructure
- Roadside Receiver – Tablet and DSRC Radio

8.1 Component Selection

This section parallels the previous section, but further specifies the specific make/buy components and the rationale for their use.

Table 3. Component Selection and Rationale

Component	Selection Specifics	Rationale
Android Device	Google Nexus 5X Google Nexus 7 Tablet	• Support for Bluetooth Low Energy • Support for Wi-Fi Direct • Large high resolution display
Arada DSRC Radio	Arada System's LocoMate™ ME OBU Arada System's LocoMate™ Mini 2 OBU	• Only available device that is portable and battery powered • Approved for use in other U.S. DOT programs (Safety Pilot)
Bluetooth Beacon	Estimote Beacon	• Support for Bluetooth Low Energy • Good examples and documentation • Low cost
Cloud Services	Microsoft Windows Azure	• Commonality with other U.S. DOT programs • Ease of use • Low cost • Mature tool chain

Source: Battelle

The components are described in detail throughout the remainder of this chapter.

8.2 Android Smartphone

Each instance of an MDEA, VEA, or RR will be a mobile device (smartphone / tablet) running the Android operating system. The goals / features of this component are:

1. Provide an interface to the cellular network
2. Management of the connection back to the Azure Cloud Services
3. Graphical user interface to communicate the following to the user:

Safety-related message displays

Coordination Request Message-related message displays

4. Interfacing with the DSRC radio module in the mobile device via a Bluetooth connection to send message data to the DSRC radio module and receive safety and travel messages. This will also interface to the Bluetooth beacons located in the transit vehicle.
5. Interfacing with other mobile devices via Wi-Fi Direct interface for forming and coordinating local ad-hoc travel groups.
6. Logging Wi-Fi direct and all PMM related messages

8.2.1 Platform

The target platform is the Google Nexus 5X, which is a 5" Android based cell phone. However, the software developed for the Mobile Device User Interface Module will run on any type of Android device as long as that device supports Android 5.0 Lollipop (API Level 21).

8.2.2 Development Stack

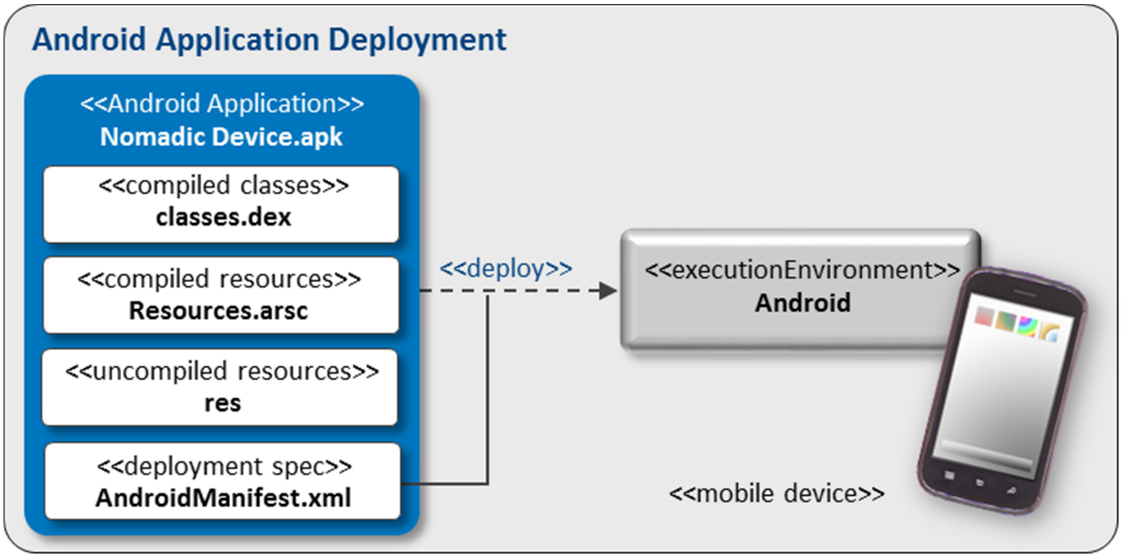
The Mobile Device User Interface Module will be built using Google's Android SDK and targeted at Android devices running at least Android 5.0 Lollipop (API level 21).

8.2.3 Deployment Diagram

Source: Battelle

Figure 6 shows the basic deployment diagram for the Mobile Device User Interface Module. This is a standard deployment model used by most basic Android applications[4] allowing portability of any developed MDEA, VEA, and RRA software.

[4] UML Diagrams website



Source: Battelle

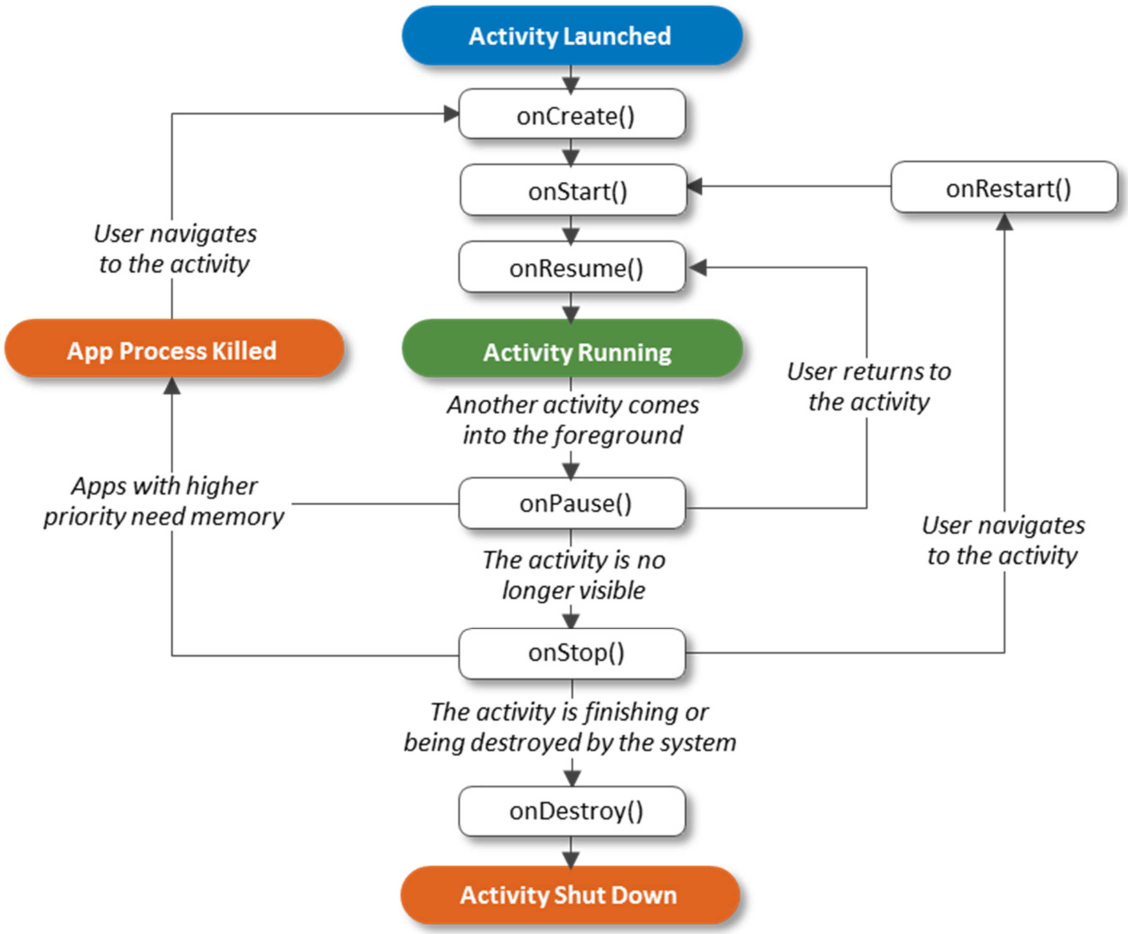
Figure 6: Mobile Device User Interface Application Deployment Diagram

8.2.4 Applications Overview

Most custom Android applications are composed of Activities. The activities normally “live” in a lifecycle controlled by the Android OS. The intent is that the custom application on the mobile device is the only executing application on the device but the system must support the user switching to other activities (applications) or responding to operating system events.

Figure 7 shows the Mobile Device Module lifecycle[5]. This diagram shows the important state paths of an Activity. The square rectangles represent callback methods, which will be implemented to perform operations when the Activity moves between states. The colored ovals are major states an Activity can be in.

[5] Android Developers website



Source: Battelle

Figure 7: Mobile Device User Interface Application Lifecycle

Communication Interfaces

To support communications between the smartphone device and the mobile device's and in-vehicle device's integrated DSRC radio, the application will leverage the Android OS support for the **Bluetooth** network stack, which allows a device to wirelessly exchange data with other Bluetooth devices. The application framework provides access to the Bluetooth functionality through the Android Bluetooth APIs (for both Bluetooth Low Energy and standard Bluetooth). These APIs let applications wirelessly connect to other Bluetooth devices, enabling point-to-point and multipoint wireless features.

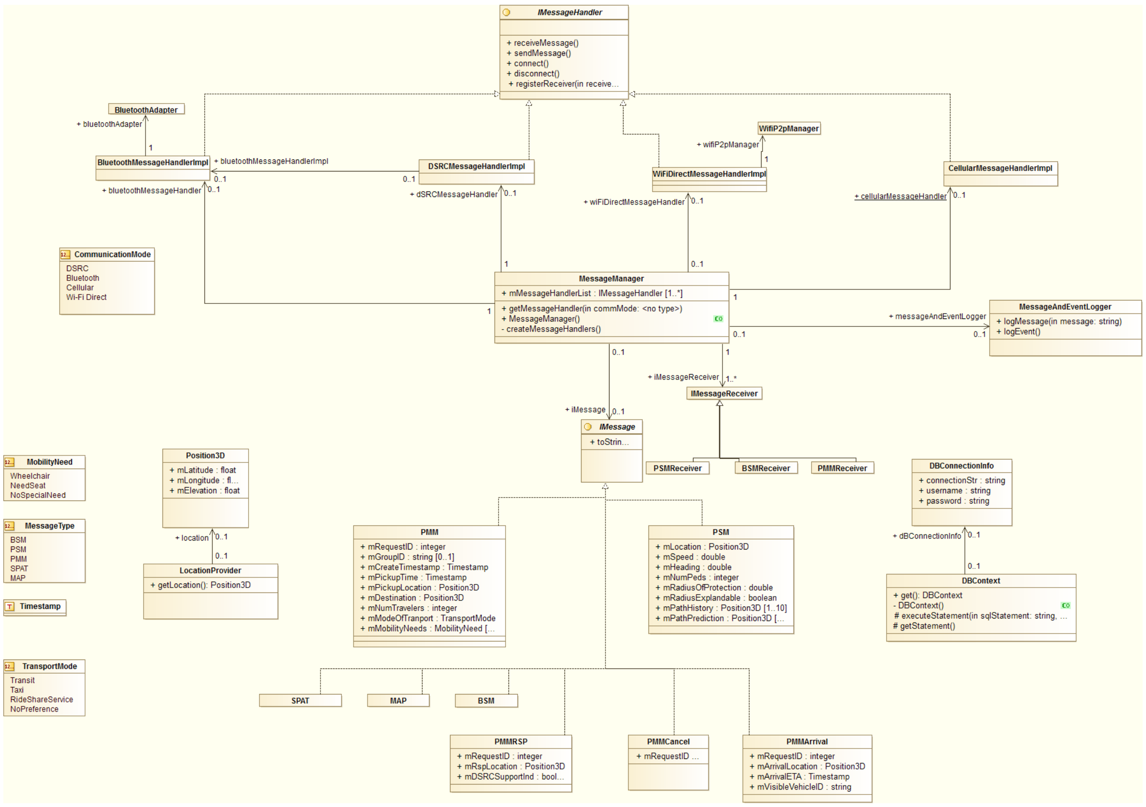
To support communication between the smartphones for **coordination request messages**, the application will use the Android Wi-Fi Peer-to-Peer (P2P) APIs,[6] which allow applications to connect to nearby devices without needing to connect to a network or hotspot.

[6] Creating Android P2P connections with Wi-Fi

Support for interfacing with the integrated **DSRC** radio will be handled by a dedicated Android service. The Android service object supports developers by providing a way to manage long running or persistent operations while not blocking the main user interface activities. In other words, it is a facility for the application to tell the system about something it wants to be doing in the background (even when the user is not directly interacting with the application).

8.2.5 Common Components

Class diagram below shows the design of components that are reused across all major components that make up the experimental prototype system.



Source: Battelle

Figure 8: Common Component Class Diagram

IMessageHandler: the Message Handler interface defines the methods used to send and receive messages over various communication media.

IMessage: the Message interface that defines the methods used to send and receive messages over various communication media.

MessageManager: the Message Manager creates and provides the message hander classes for the various EPS communication modules to receive and send messages over various communication media.

DSRCMessageHandlerImpl: The DSRC Message Handler handles all DSRC communications to and from the EPS application modules and DSRC radio including packaging and un-packaging of

DSRC messages for the EPS applications via a dedicated connection to the DSRC Radio. The DSRC Message Handler receives messages from the EPS application modules, and prepares them to be sent to the DSRC radio, ensuring only properly formatted DSRC messages are sent to the DSRC radio. The DSRC Message Handler receives incoming DSRC messages from the DSRC radio, unpacks the messages, and formats them into the EPS message format.

WifiMessageHandlerImpl: the Wi-Fi Message Handler manages Wi-Fi Direct connections and message communication such as Coordination Request messages between mobile devices within ad-hoc travel groups.

CellularMessageHandlerImpl: the Cellular Message Handler manages the connection and message communication over cellular connections such as PMM and related messages.

MessageAndEventLogger: the Message and Event Logger class is used to log various mobility messages received and sent by the EPS application modules. The data logged will include timestamps, message type and message data specific to each type of message.

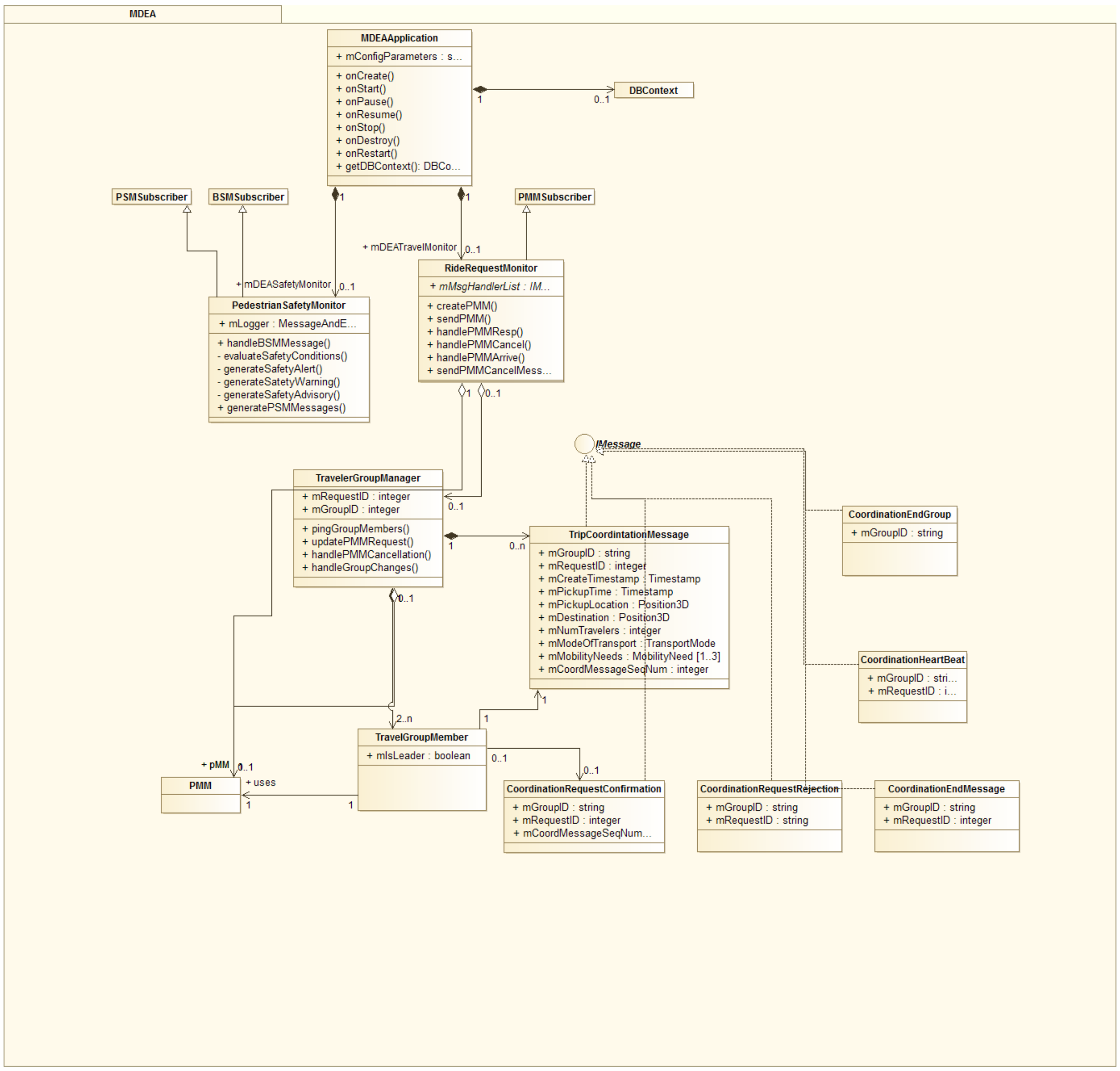
BluetoothMessageHandlerImpl: the Bluetooth Message Handler manages connection and communication over a Bluetooth connection. For EPS, a Bluetooth connection is mainly used to detect presence of a trip requestor's mobile device presence on the transit vehicle.

BluetoothAdapter: the Bluetooth Adapter allows to perform fundamental Bluetooth tasks, such as initiating device discovery, querying a list of bonded (paired) devices, instantiating a Bluetooth Device using a known Media Access Control (MAC) address, creating a Bluetooth Server Socket to listen for connection requests from other devices, and starting a scan for Bluetooth LE devices.

WifiP2pManager: This class provides the API for managing Wi-Fi Peer-to-Peer connectivity. This lets an application discover available peers, setup connection to peers and query for the list of peers. When a P2P connection is formed over Wi-Fi, the device continues to maintain the uplink connection over mobile or any other available network for internet connectivity on the device.

8.2.6 Mobile Device Experimental Application

The class diagram below shows the major components of the mobile devices experimental application.

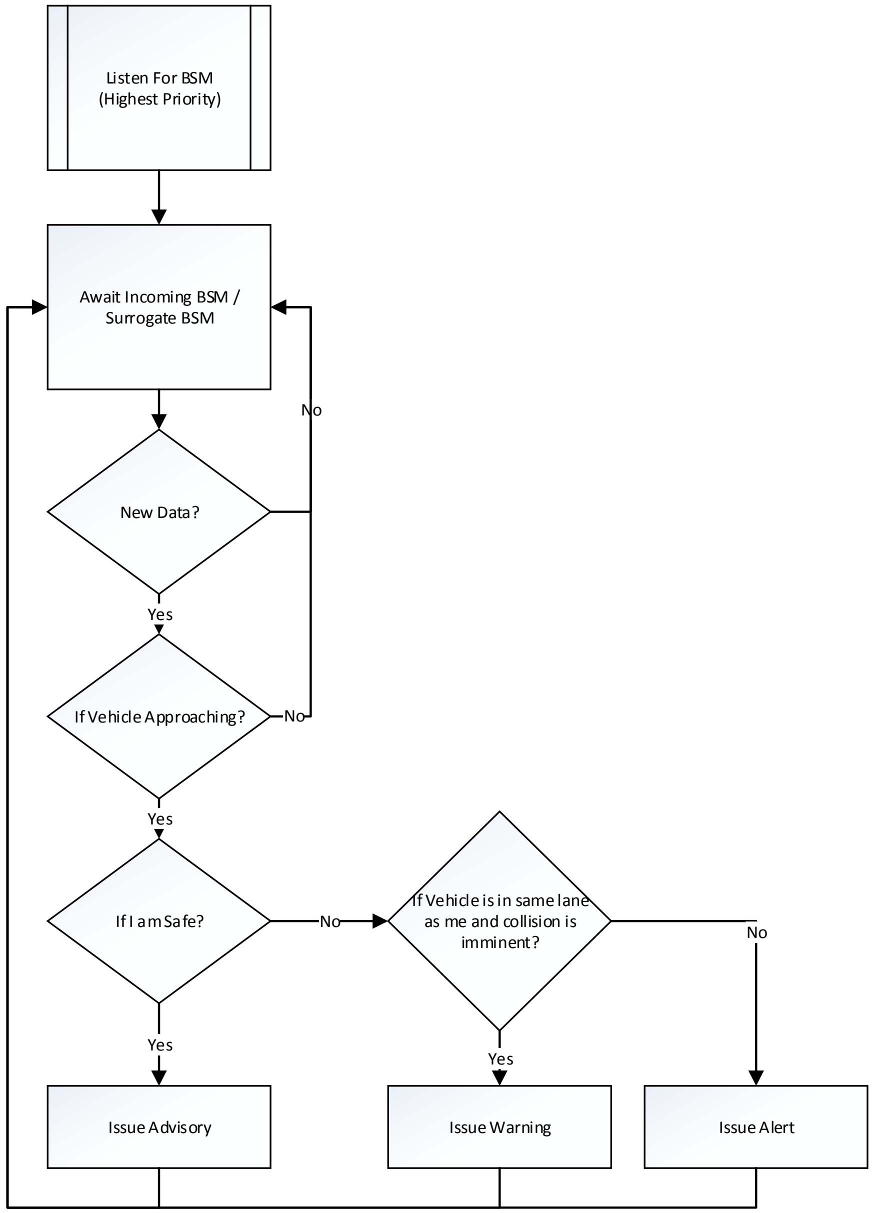


Source: Battelle

Figure 9: MDEA Class Diagram

Pedestrian Safety Monitor

The Pedestrian Safety Monitor is dedicated to always listening for Basic Safety Messages (BSMs) from nearby vehicles, and to notify the pedestrian if there is a threat to their safety from a DSRC-equipped vehicle (Figure 10).



Source: Battelle

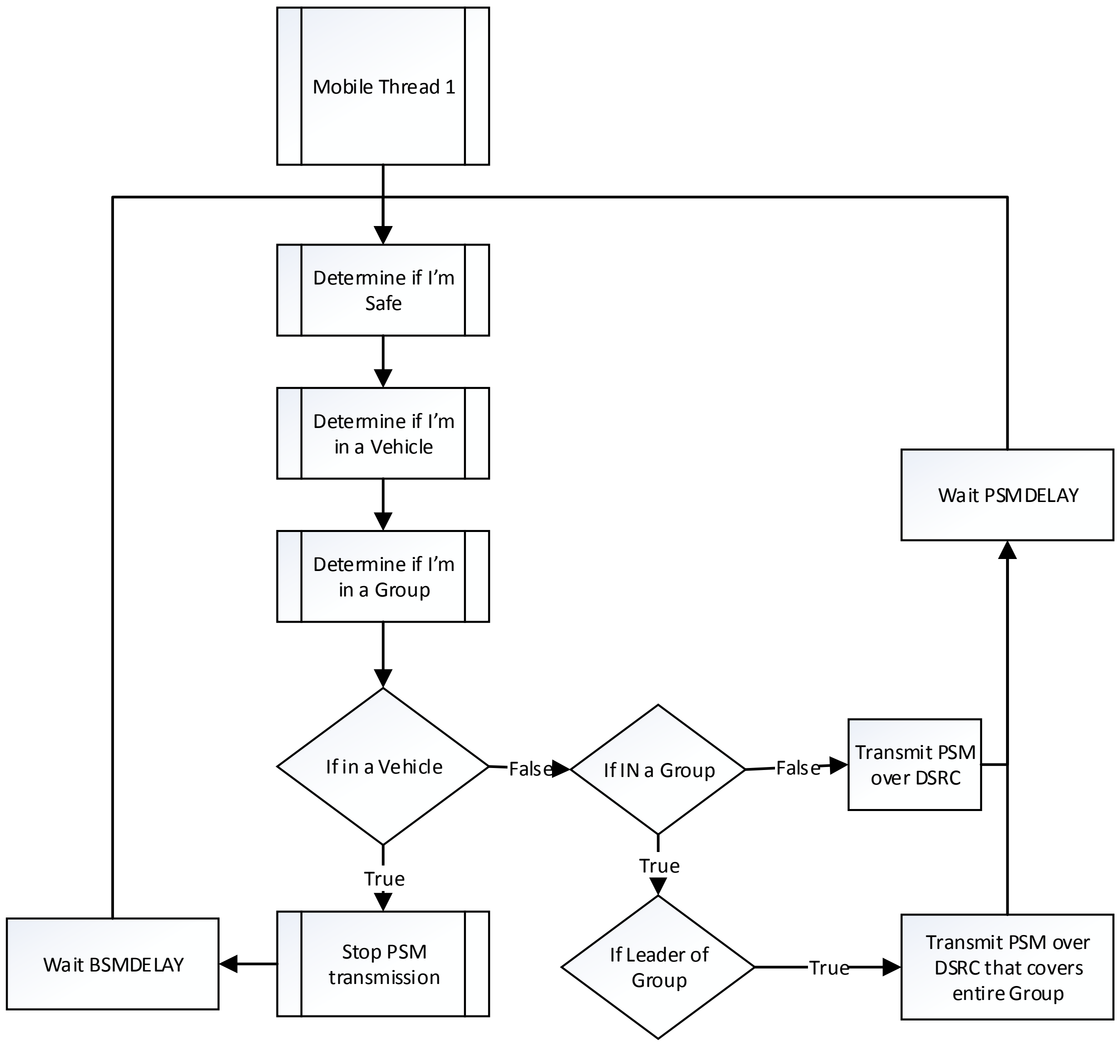
Figure 10: Pedestrian Safety Monitor Logic

Safe Zone Subtest

The approach to determining if a pedestrian is in a safe zone or an unsafe zone is to use polygons to categorize groups of latitude/longitude coordinates. If the latitude/longitude is inside of the four (4) corners of a polygon(s), then it is unsafe. If it is outside, then it is in a safe zone.

The configuration settings will include a place to specify several polygons to be used for the test. The roadways themselves will be subdivided into polygons and identified as unsafe zones.

The MDEA Safety Monitor will also be responsible for sending out periodic Personal Safety Messages (PSMs), so that nearby vehicles can receive these messages and be warned about the location of the pedestrian carrying an equipped mobile device, if necessary. These PSMs will be transmitted at all times except for two (2) conditions: if the person is waiting as a part of a travelling group, or if the person is within a vehicle (Figure 11).



Source: Battelle

Figure 11: Process for Sending PSMs

In the case where the user is not in a vehicle and not in an ad-hoc travel group, the PSM is sent routinely at a fixed interval. Ad-hoc travel groups are discussed in detail in section Ride Request Monitor. The Travel Monitor logic will make globally available constructs to know: of an existing ad-hoc travel group, if the user is an ad-hoc travel group leader or follower within the group, and how many members (and pending members) are within an ad-hoc travel group. When the user is a member of an ad-hoc travel group, the ad-hoc travel group leader assumes the responsibility of sending out PSMs representing the entire ad-hoc travel group. Thus, if the user is identified as a follower, no action is taken. If the user is identified as the ad-hoc travel group leader, a PSM is constructed including the total number of members and pending members as the PSM Number of

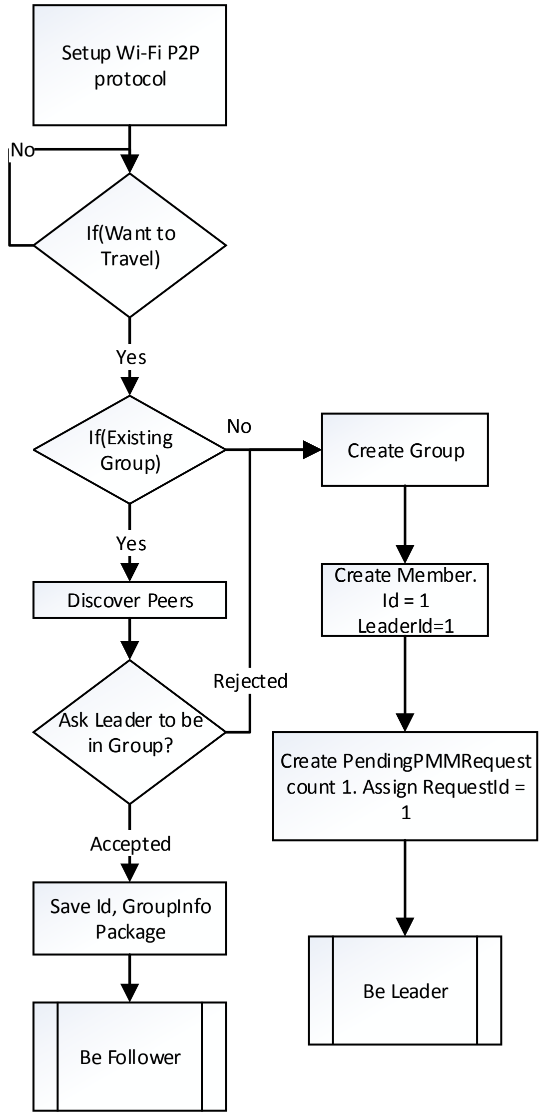
Peds. The Radius of Protection will be calculated via a configuration property [PROTECTION_METERS_PER_PERSON] times this Number of Peds.

Conversely, when the user is in a vehicle, PSMs are not transmitted, because the person is no longer as vulnerable as when outside of a vehicle's shell. Establishment of a Bluetooth connection between the user's mobile device and the iBeacon equipped vehicle is used as an indicator to detect the presence of the user in the vehicle. Vehicles transmit BSMs to communicate the location of the vehicle. PSM transmission will resume when the user gets off the vehicle. *Note that the functionality of Surrogate BSMs will not be integrated into the EPS.*

Ride Request Monitor

The Ride Request Monitor manages functionality and features when a traveler has scheduled a trip. The user will enter trip information into the MDEA User Interface: destination, departure location and time, number of people in party by seat type, and vehicle type. The Travel Monitor then commences two (2) primary ongoing activities: coordinating with other co-located mobile devices that wish to travel from the approximately same pickup location and time to the same place, and coordinating with an incoming vehicle to reserve seating (if leader of group). The approximation of the pickup location and time will be configurable settings in degrees of latitude/longitude and minutes respectively.

Ad hoc travel groups will be established among co-located individuals with identical travel destinations. These groups will be formed automatically among the mobile phones by using the Wi-Fi Direct Interface module to form a Peer-to-Peer group. The Travel Monitor must first ascertain if there is already an existing group to join, or if a new group must be created. If the Monitor joins an existing group, then the Monitor will commence running in Follower mode. If the Monitor finds no matching group and creates its own group, then the Monitor will commence running in Leader mode for the new group. This flow is depicted in Figure 12. Figure 13 outlines the data constructs maintained and utilized during the program flow.



Source: Battelle

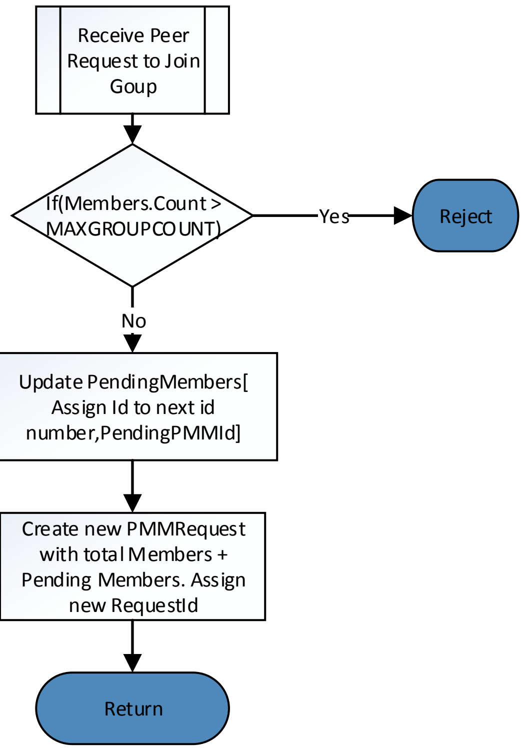
Figure 12: Monitor Decides Role to Play in Group

Source: Battelle

Figure 13: Selected Data Maintained during Travel Monitor

If a matching existing ad-hoc travel group is detected by the Wi-Fi Direct Interface module, then the software will ask to join this existing group. If it receives an acceptance back, then it will store the returned Group Info and then wait in Follower mode. However, if a group is not detected or if the request was denied, the software will create a new group. Members are each assigned a uniquely incrementing ID; thus, as leader, the MemberId assigned to self would be one (1). This is stored in the PendingMembers list. A PMM Request packet is created using the GroupId, the RequestId, and the travel information the user entered via the MDEA User Interface module. A GroupId GUID is created by the leader to represent this group. It will be included in all PMM Update messages as well, which allows vehicles to know that updates supersede previous messages rather than being new, distinct requests. The RequestId, an integer, provides further differentiation among requests originating from within one (1) group. The PMM Request data is stored into the list PendingPMMRequests. The RequestId correlated to the request, which contains the seat(s) for a Pending Member is stored into the PendingMembers list with that member. A member stays a Pending Member until confirmation is received from a vehicle (VEA) that space has been reserved.

Any requests to join an existing group are evaluated by the group leader. Groups have a configurable MAXGROUPCNT that limits the maximum number of members that can be in one (1) group. If the group has already reached this limit, then the request is immediately rejected. If not, the requesting party is added as a Pending Member and a new PMMRequest is created to initiate getting approval from the vehicle for additional space on the vehicle for the party (see Figure 14).

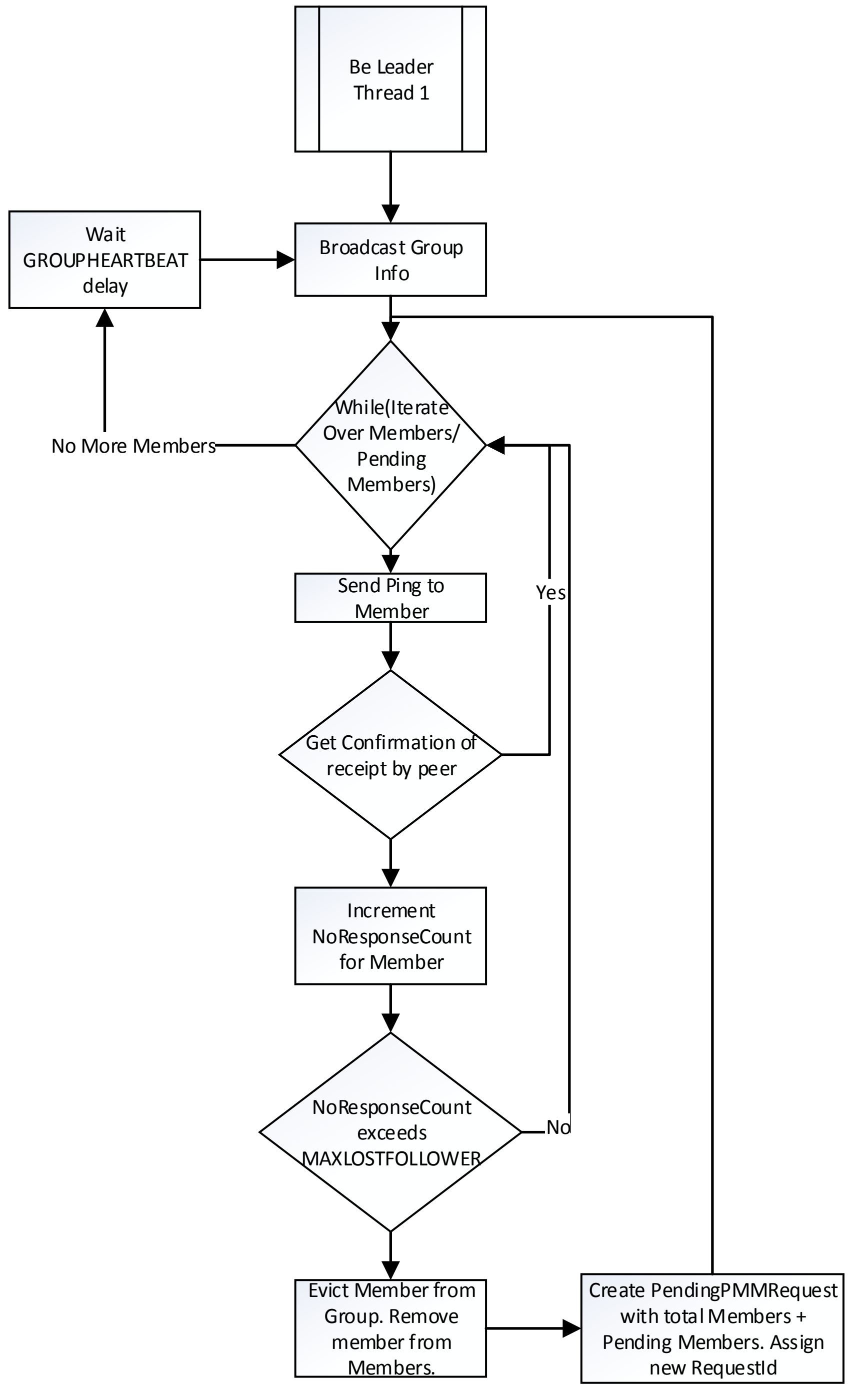


Source: Battelle

Figure 14: Monitor Receives Request to Join Group

The leader of a travelling group will create a dedicated thread in order to handle all the interaction amongst members of the group. The leader is responsible for periodically updating all the participants with the latest travel information. This information contains a list of all the currently accepted Members and the Pending Members of the group. It also includes the LeaderId – e.g., the MemberId of the leader. The leader does more than just broadcast updates – the leader must also ascertain if the traveler is still wishing to travel (present) or has wandered off and no longer needs travel arrangements (Figure 15).

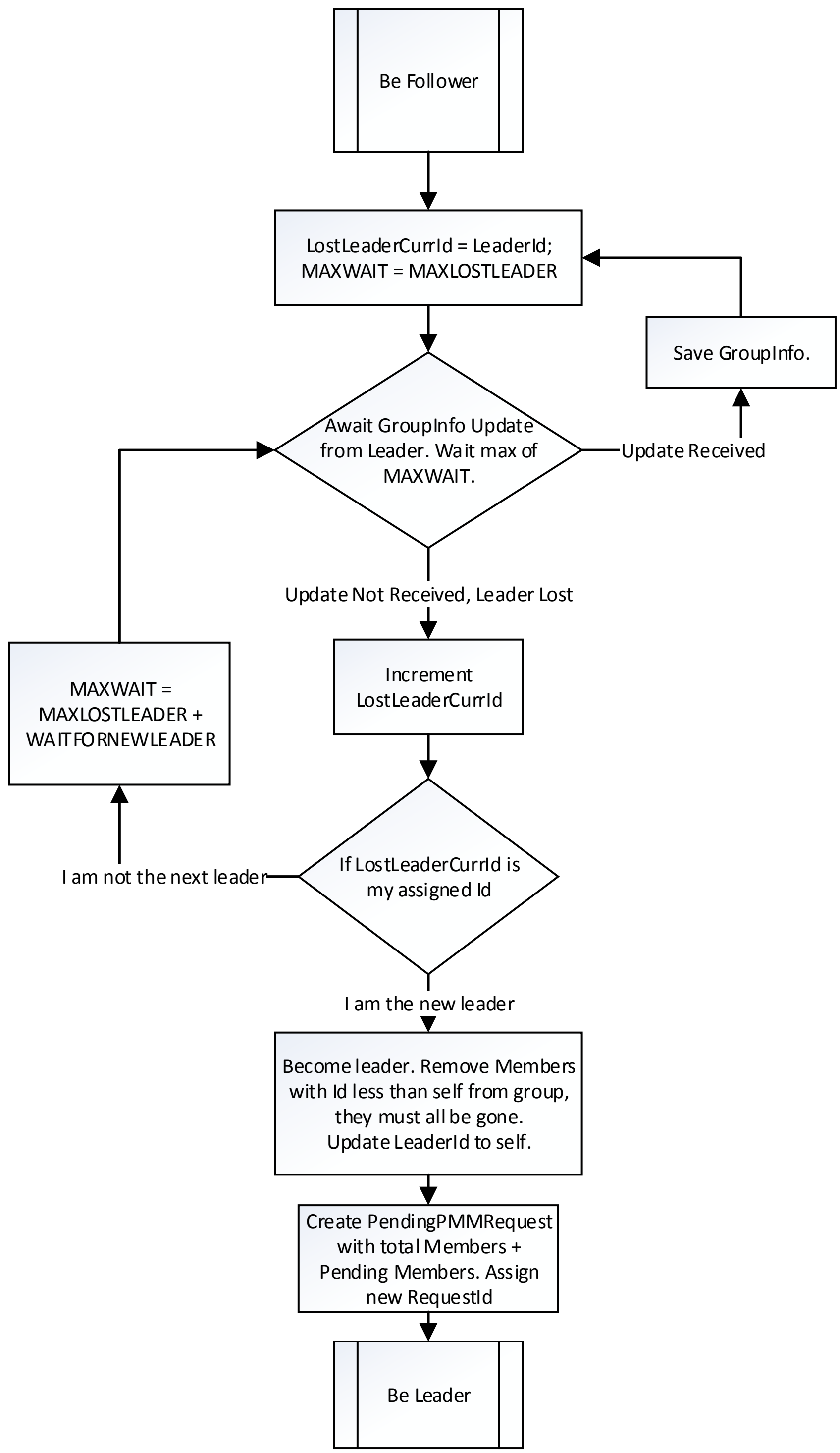
For every Member and Pending Member, the leader must get a confirmation back to validate if all travelers are still present and require travel arrangements. If so, the leader moves on. If not, the NoResponseCount of the member is updated, and if a lack of response repeats MAXLOSTFOLLOWER times, then the member is presumed gone and is removed from the respective list. A new PMM Request will be generated to reflect this reduction in seats requested.



Source: Battelle

Figure 15: Flow of Leader Keeping Track of Group

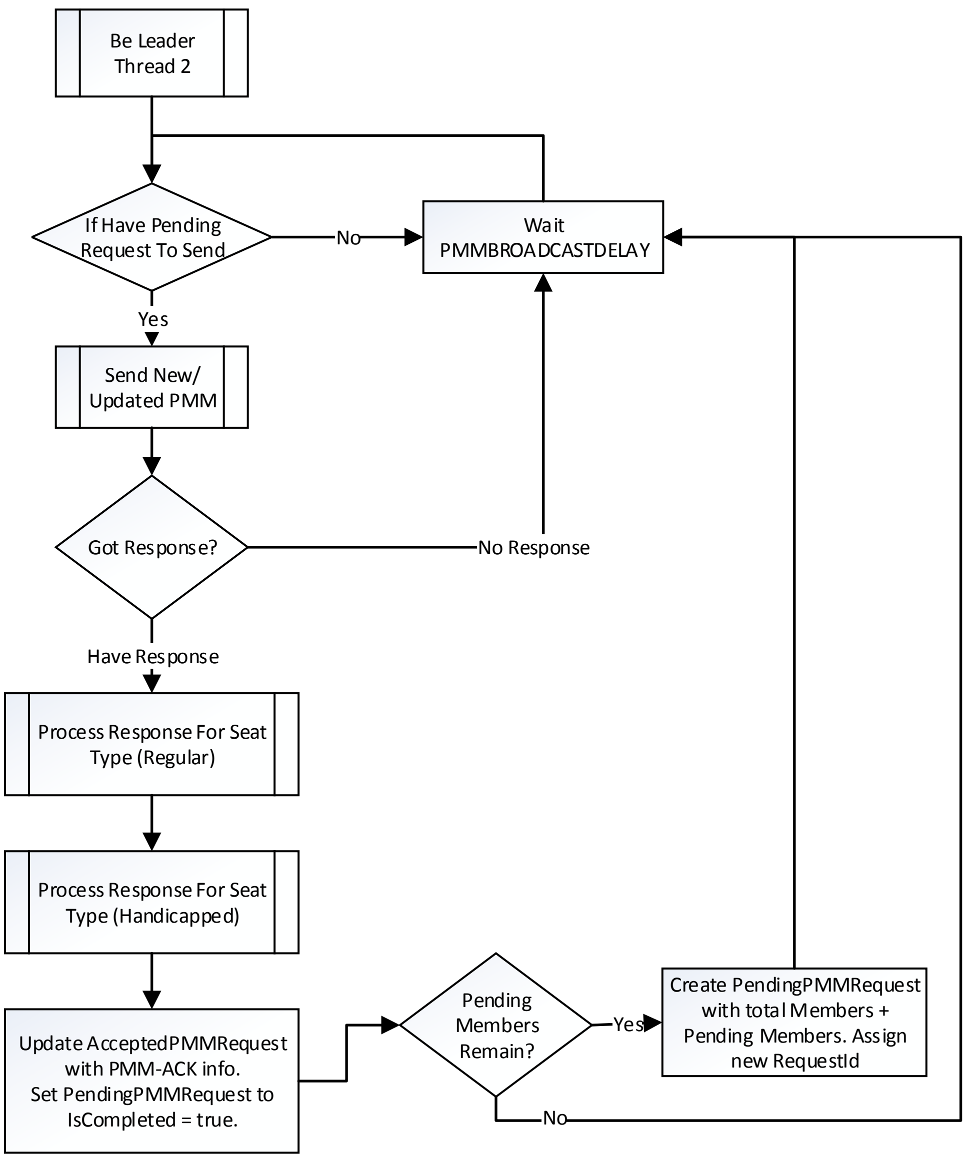
In Follower mode, the software generally just waits for updates from the leader and stores them, as shown in Figure 16. However, in some cases it may be that it is the leader who wandered off, and now the group is without a leader. If a considerable amount of time has passed since the last update from the leader, the LeaderId in conjunction with the MemberId (assigned to each MDEA as a result of being accepted to the group) will be used to promote the next leader. A local variable, LostLeaderCurrId, will be initialized to the existing LeaderId. The LostLeaderCurrId will be incremented by all phones (followers) that are a part of the group. If the newly incremented value matches the assigned MemberId of that follower, then that follower will become the new leader. However, if the LostLeaderCurrId does not match, then the software will continue to wait since theoretically another member of the group found the match and is assuming the role of leader. Once the new leader sends out a new update, then this follower would go back to the simple wait-for-updates routine. If, however, the new leader still has not assumed control after a longer wait, the software assumes that they, too, must have left the group. The LostLeaderCurrId continues to be incremented until a follower successfully takes control of the group.



Source: Battelle

Figure 16: Follower Mode Watches for Loss of Leader

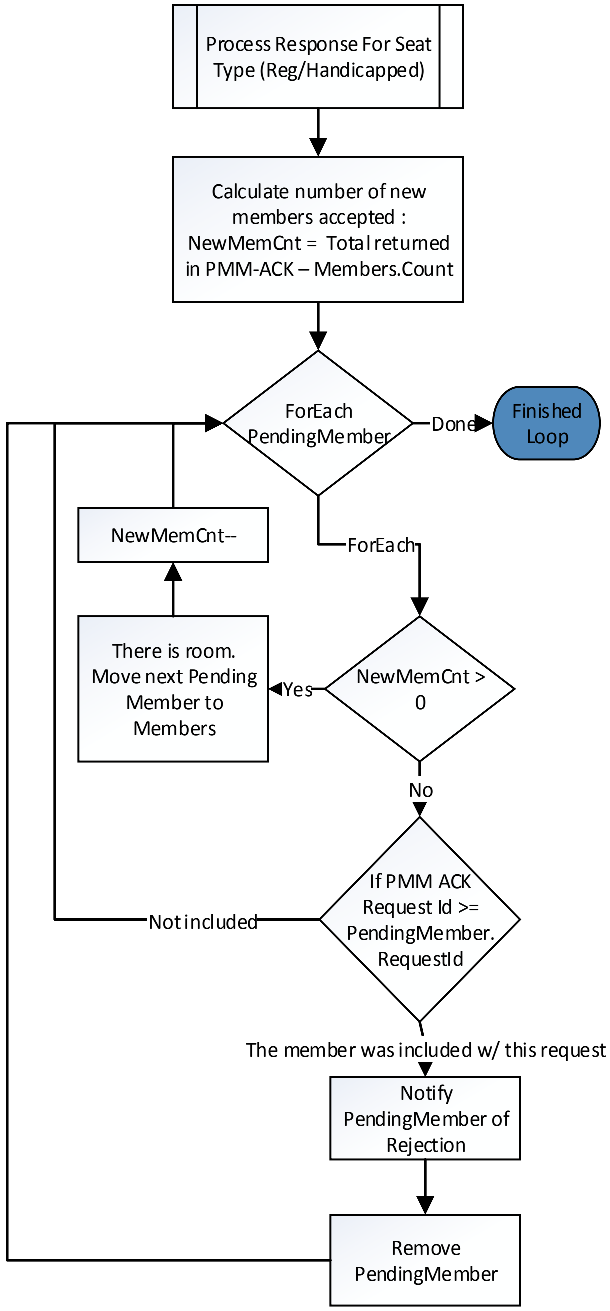
The leader's second central role, beyond managing the communications for the ad-hoc travel group, is to manage the communications external to the group. The leader is the only speaker for the group, and handles broadcasting of the ad-hoc travel group's joint Personal Safety Message, as well as sending the initial and all subsequent updates to the PSM for upcoming travel. The broadcast of PSMs occurs in a dedicated thread and will occur at a configurable periodic rate. Space reservation coordination happens through the exchange of PMM and PMM-RSP messages between the MDEA and a VEA, as shown in Figure 17. Any time that a new Pending PMM Request has been created, the leader will continue to periodically send it and check for a response. Once a response is received, the response is processed (Figure 18). The response is saved as the AcceptedPMMRequest. If Pending Members remain (that were not a part of the request that was just responded to) then a new PendingPMMRequest is created to represent those Pending Members.



Source: Battelle

Figure 17: Leader Mode Processes Accepted Members

Processing the PMM-RSP response received is done by seat type; for the EPS, regular and handicapped seating is addressed. First, the number of new seats accepted is calculated as the delta between the current Members and the total count returned by the vehicle in the PMM-RSP packet. The Pending Members are then iterated over, and as long as new seats remain for that seat type, Pending Members are moved over to the Members list. The Pending Members list is iterated in the order the Pending Members were added, thus providing a first-come, first-serve preference. Once the available new spaces have been consumed, decisions must be made about the remaining Pending Members. This decision is based on if the PMM-RSP being processed was inclusive of a PMM Request that represented the Pending Member or not. The RequestId that is stored with each Pending Member at the time the PMM Request for them was created, is used to determine if they were represented in the request. If the request did not include the Pending Member, then it is unknown if they are accepted, and they remain as a Pending Member. If the request did include the Pending Member, then they have been officially rejected and must be notified of removal from the group and removed from Pending Members.

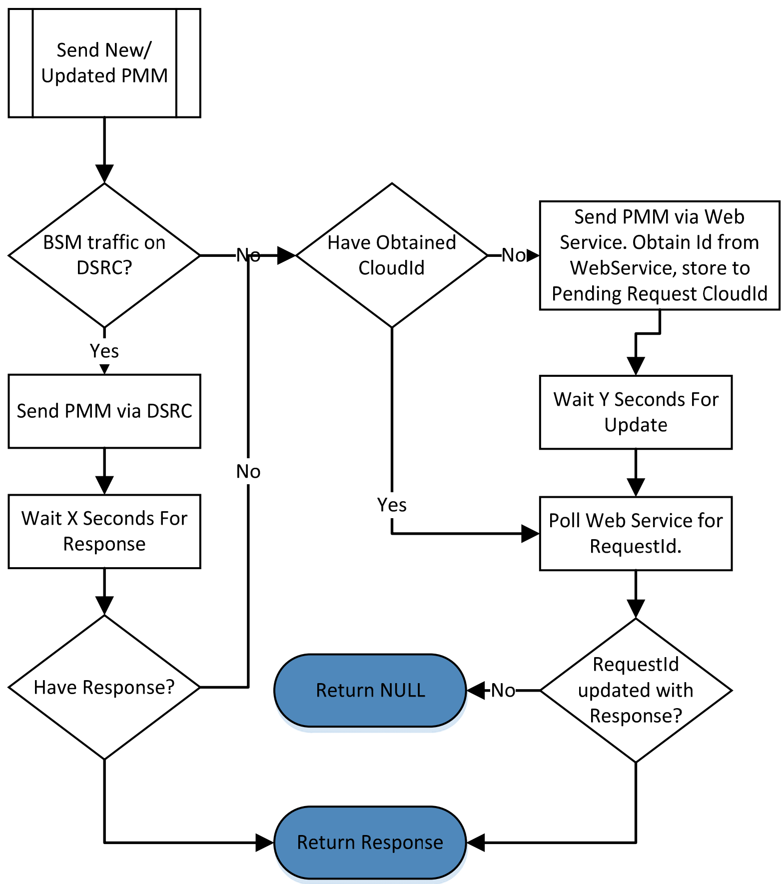


Source: Battelle

Figure 18: Leader Mode Processes PMM-RSP for Seat Reservations

The Travel Monitor utilizes two (2) different communication mechanisms to reserve seating on an incoming vehicle for travel: DSRC and cellular. The flow of these alternative choices is depicted in Figure 19.The software will first check if there is local DSRC activity. If so, the request will be sent over DSRC communication via the DSRC Message Handler module. A PMM message is created and sent periodically at configurable frequencies from once per second to once per ten (10) seconds. This transmission will continue until a PMM-RSP is received from a vehicle.

If no local DSRC activity is present or a PMM-RSP message is not received within a configurable time limit, the request will be sent over cellular communication via the Cellular Interface module to the Azure Web Service. Since the Azure Web Service returns a confirmation of delivery, the PMM information packet only needs to be sent successfully once in order to be stored in the Cloud. The Travel Monitor will then poll the database at a configurable frequency to see if the request has been updated by a vehicle.



Source: Battelle

Figure 19: PMM requests are sent via either DSRC or cellular

A PMM-Arrive will be received by the MDEA of the ad-hoc travel group leader when the vehicle has arrived at the pickup location. The leader will notify the rest of the travel party, and disband the group.

Use Cases

Group Established for One Traveler, Vehicle Out of Range

1. Traveler X arrives at a stop. Schedules a trip.
2. Ride Request Monitor looks for a local group for the same trip. None is found.
3. MDEA establishes a new Group ABC, LeaderId 1, adds self as Pending Member, and enters Leader mode.
4. MDEA creates PMM Request that must be transmitted.
5. MDEA determines that there is No DSRC traffic in the vicinity. Transmit via Web Service AddPMMRequest. Gets back CloudId of 1.
6. VEA of vehicle outside of DSRC range polls Web Service CheckForNewRequests. Sees new request.
7. MDEA continues to poll Web Service CheckOnPendingRequest(1) for updates to its request.
8. VEA accepts request for 1 traveler via Web Service RespondToPendingRequest.
9. MDEA continues to poll Web Service CheckOnPendingRequest(1) – sees response from VEA.
10. MDEA moves Pending Member (self) to Members and stores AcceptPMMRequest.

Group Established for One Traveler, Vehicle in DSRC Range

1. Traveler X arrives at a stop. Schedules a trip.
2. Ride Request Monitor looks for a local group for the same trip. None is found.
3. MDEA establishes a new Group BCD, LeaderId 1, adds self as Pending Member, and enters Leader mode.
4. MDEA creates PMM Request that must be transmitted. DSRC traffic is detected in the vicinity. Transmit via DSRC Message Handler.
5. VEA of vehicle inside of DSRC range receives request and responds via DSRC with acceptance via PMM-ACK.
6. MDEA receives DSRC PMM-ACK.
7. MDEA moves Pending Member (self) to Members and stores AcceptPMMRequest.

Second Traveler Arrives, Existing Group Established By LeaderId 1

1. Traveler Y arrives at a stop. Schedules a trip.
2. Ride Request Monitor looks for a local group for the same trip. One (1) is found.
3. MDEA asks to be added to the existing group.
4. MDEA of Leader receives this request. Adds Traveler Y to Pending Members. Returns acceptance for Traveler Y to become a follower within the group.
5. MDEA of Follower receives this response and enters Follower Mode. Begins loop waiting for periodic updates from Leader.
6. MDEA of Leader creates new Pending PMM Request now reflecting 2 seats requested.

7. MDEA of Leader receives PMM-RSP from the VEA that initially accepted the original PMM to indicate that it accepted 2 seats. Pending Member Y is moved to become a Member.
8. MDEA of Follower receives periodic update from Leader. Id of self (2) is now included as an official Member.

Leader Departs From Existing Established Group of 3

1. Traveler X (who is Leader) walks away and physically departs from group. GroupInfo had been being sent prior to this with LeaderId of 1.
2. MDEA of Follower Y has MemberId of 2. MDEA of Follower Z has MemberId of 3. MDEAs waits for periodic update.
3. MDEAs of Followers have each determined too much time has passed since the last periodic update.
4. MDEA of Follower Y utilized the LeaderId passed in the last received update from the leader, and increments its value. This value, 2, matches the MemberId of 2.
5. MDEA of Follower Z utilized the LeaderId passed in the last received update from the leader, and increments its value. This value, 2, does not match the MemberId of 2.
6. MDEA of Follower Y assumes leadership of the group.
7. MDEA of Follower Z resumes waiting for someone else to become leader.
8. MDEA of Follower Y updates the PMM to reflect the reduced number of members, and sends an update to the group.
9. MDEA of Follower Z receives periodic update from new leader. Stores latest information, including LeaderId value of 2.

Leader And First Follower Depart From Existing Established Group of 3

1. Traveler X (who is Leader) walks away and physically departs from group. GroupInfo had been being sent prior to this with LeaderId of 1.
2. MDEA of Follower Y has MemberId of 2, and also walks away from group.
3. MDEA of Follower Z has MemberId of 3. MDEA waits for periodic update.
4. MDEA of Follower Z has determined too much time has passed since the last periodic update.
5. MDEA of Follower Z utilized the LeaderId passed in the last received update from the leader, and increments its value. This value, 2, does not match the MemberId of 2.
6. MDEA of Follower Z resumes waiting for someone else to become leader.
7. MDEA of Follower Z has determined too much additional time has passed since the last periodic update.
8. MDEA of Follower Z increments the LeaderId value again. This value, 3, matches the MemberId of 3.
9. MDEA of Follower Z assumes leadership of the group.
10. MDEA of Follower Y updates the PMM to reflect the reduced number of members, and sends an update to the group (despite that it is now a group of one).

MDEA User Interface

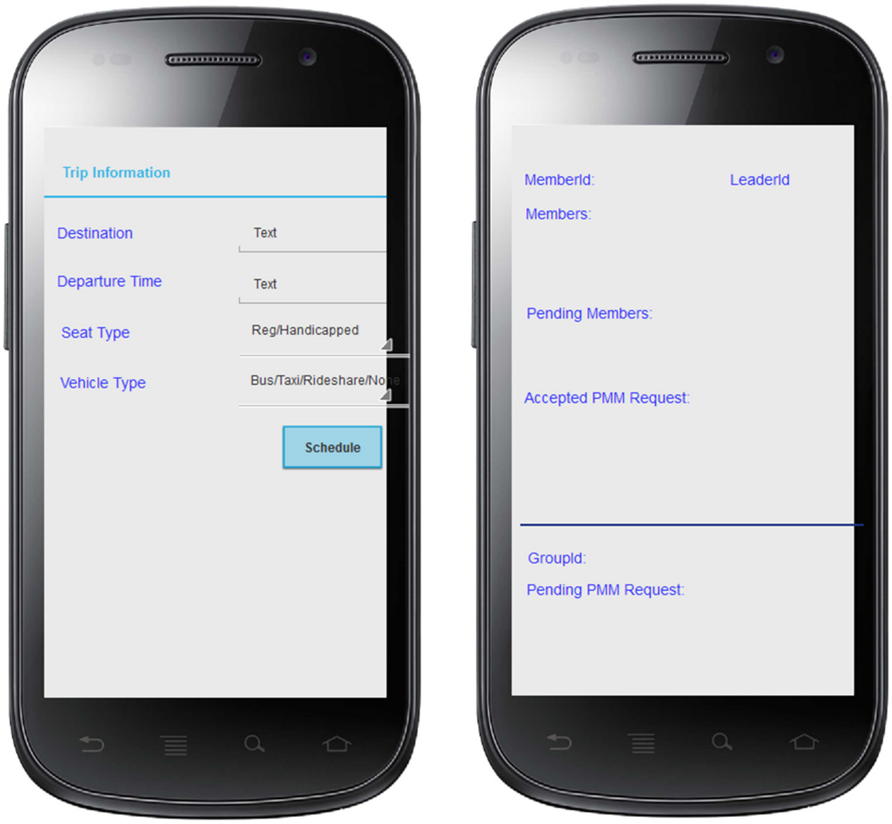
The primary user display (GUI) for the MDEA application will focus on traveler safety (Figure 20). In the mockups the screens are made as simple as possible showing the primary intents of the application.

The GUI will show the relative safety of the user's current location, as well as any advisories/alerts/warnings about oncoming traffic. The GUI will show the status of whether the mobile device (user) is determined to be in a vehicle or not. Overrides will be provided to force the safety level or vehicle status for testing purposes.

Source: Battelle

Figure 20. MDEA Main Display

An action screen will be provided to allow the user to enter the details pertaining to scheduling a pickup for a trip (Figure 21). Destination, departure location and time, number of people in travel party (only for the party associated with this mobile device), seating type, and vehicle type may all be entered. Thereafter, the phone will display the Group Status page where it will show far more information than would be displayed in a production setting. It will show many variables that are used internally in order to facilitate better visibility of the software for the tests being performed. The display will include: MemberId, LeaderId, Current Members, Pending Members, and the current Accepted PMM Request. The leader's phone will additionally display the active Pending PMM Request and the GroupId.

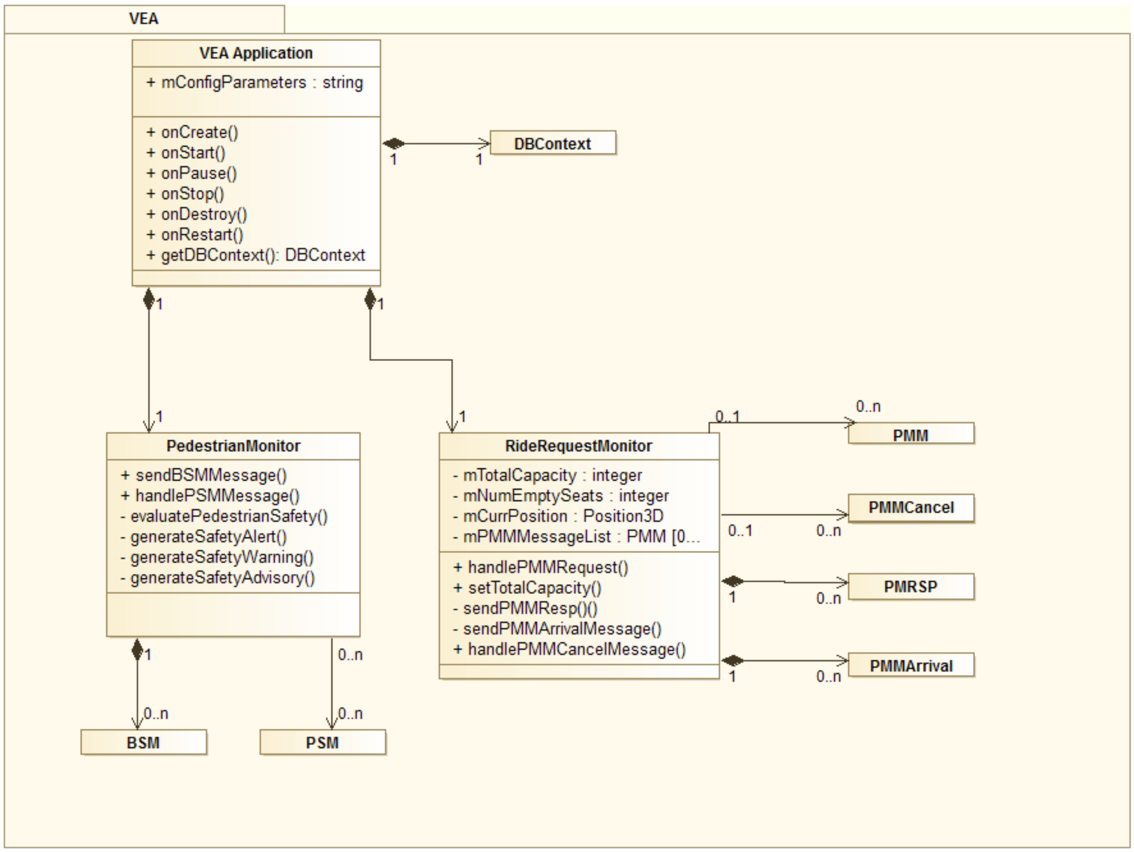


Source: Battelle

Figure 21: MDEA Travelling Group Screens

8.2.7 Vehicle Experimental Application

The class diagram (Figure 22) shows the major components of the vehicle experimental application.

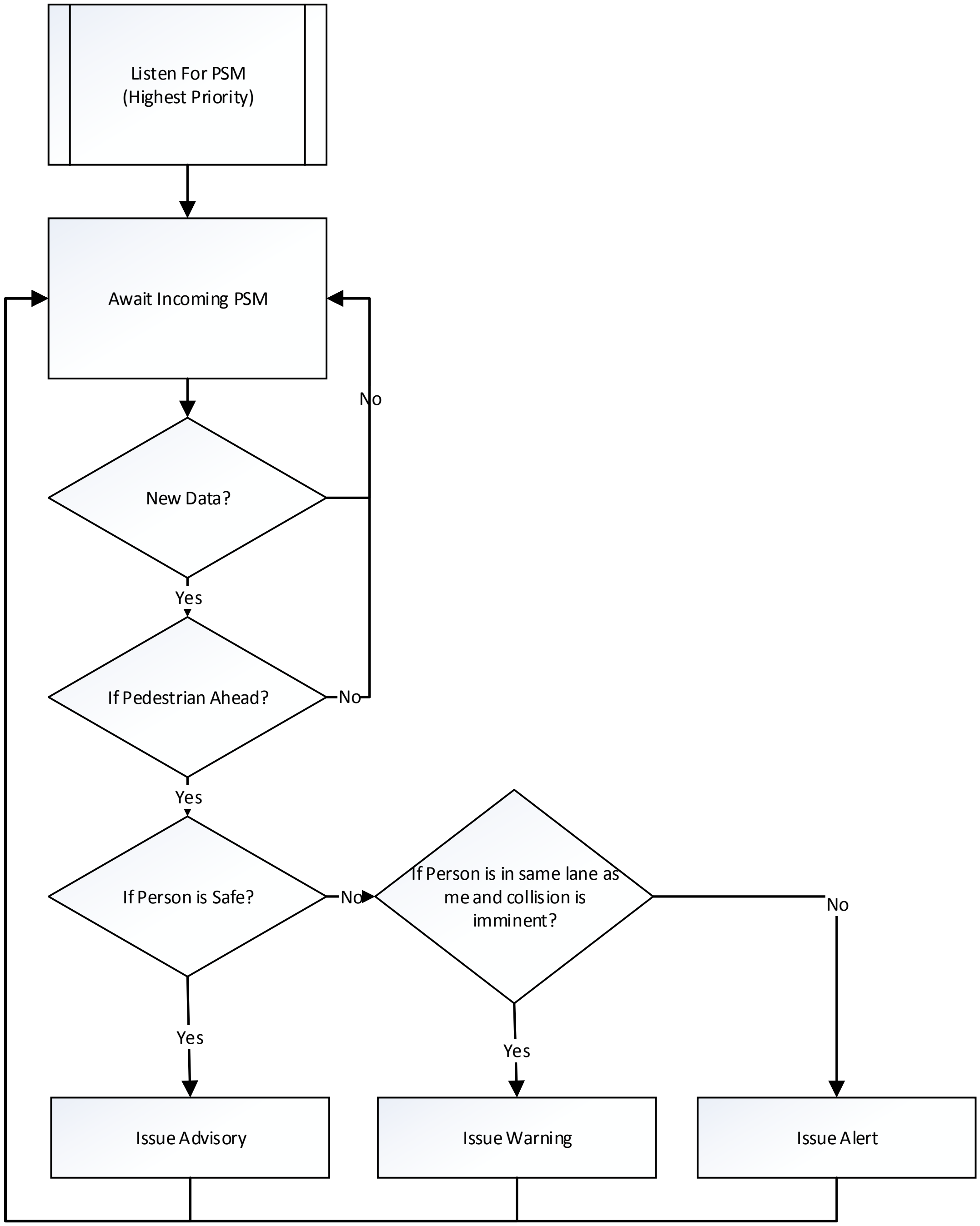


Source: Battelle

Figure 22: VDEA Class Diagram

VEA Pedestrian Monitor

The VEA Pedestrian Monitor is dedicated to always listening for PSMs from pedestrians on the road, and to notify the driver if the pedestrian has a threat to their safety from the driver's vehicle.

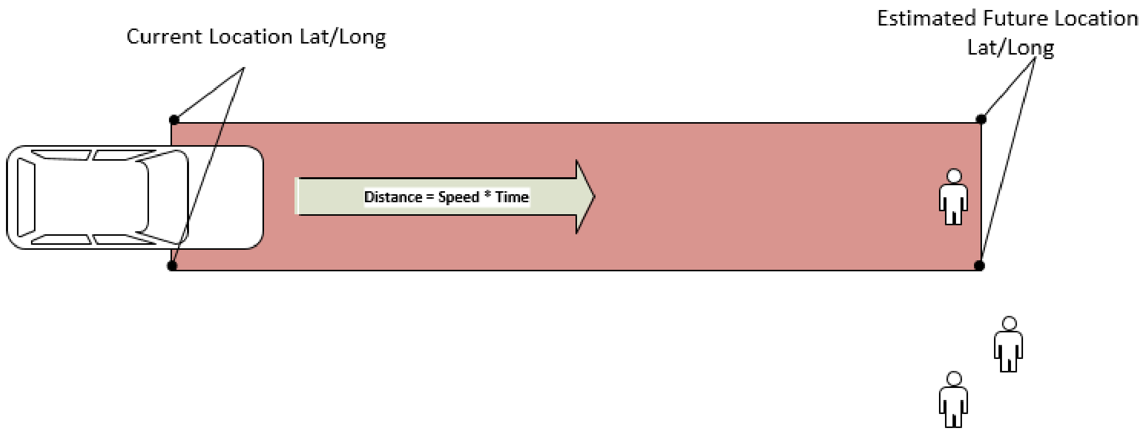


Source: Battelle

Figure 23: VEA Pedestrian Monitor Logic

The vehicle's current location, heading, and speed can be used to create a polygon representative of the trajectory of the vehicle. The current location's latitude/longitude can be expanded to a three (3) meter] width and this can represent the left hand side of the rectangle. The expansion would be performed perpendicular to the heading direction. The speed can be translated to a distance travelled over the next five (5) seconds. In combination with the heading, these values can calculate the latitude/longitude of the right hand corners of the rectangle.

All PSMs will be received by the Monitor, and the latitude/longitude coordinates of each pedestrian will be compared to this polygon. If the coordinates are inside the rectangle, then the pedestrian is unsafe. If the coordinates are outside of the rectangle, then the pedestrian is unsafe and a Warning should be issued to driver.



Source: Battelle

Figure 24: Pedestrian in Vehicle Path

The vehicle pedestrian monitor will determine if a collision is imminent with the pedestrian and warn the driver based on the following logic:

1. Use lat/long of the vehicle and the pedestrian to calculate the distance between the vehicle and the pedestrian.
2. Calculate the driver message display distances using the directives defined in Appendix C. Message Display Equations in the Task 3: System Requirements Specifications (SyRS) document.
3. Calculate the vehicle trajectory, using the Heading by comparing it to the Bearing result from the Haversine formula[7].
4. If the pedestrian is in the trajectory of the vehicle and the distance between the vehicle and pedestrian is less than the advisory message distance and greater than alert message display distance, display the pedestrian warning message.
5. If the pedestrian is in the trajectory of the vehicle and the distance between the vehicle and pedestrian is less than or equal to the alert message display distance, display the pedestrian alert message.
6. Otherwise, display the pedestrian advisory message.

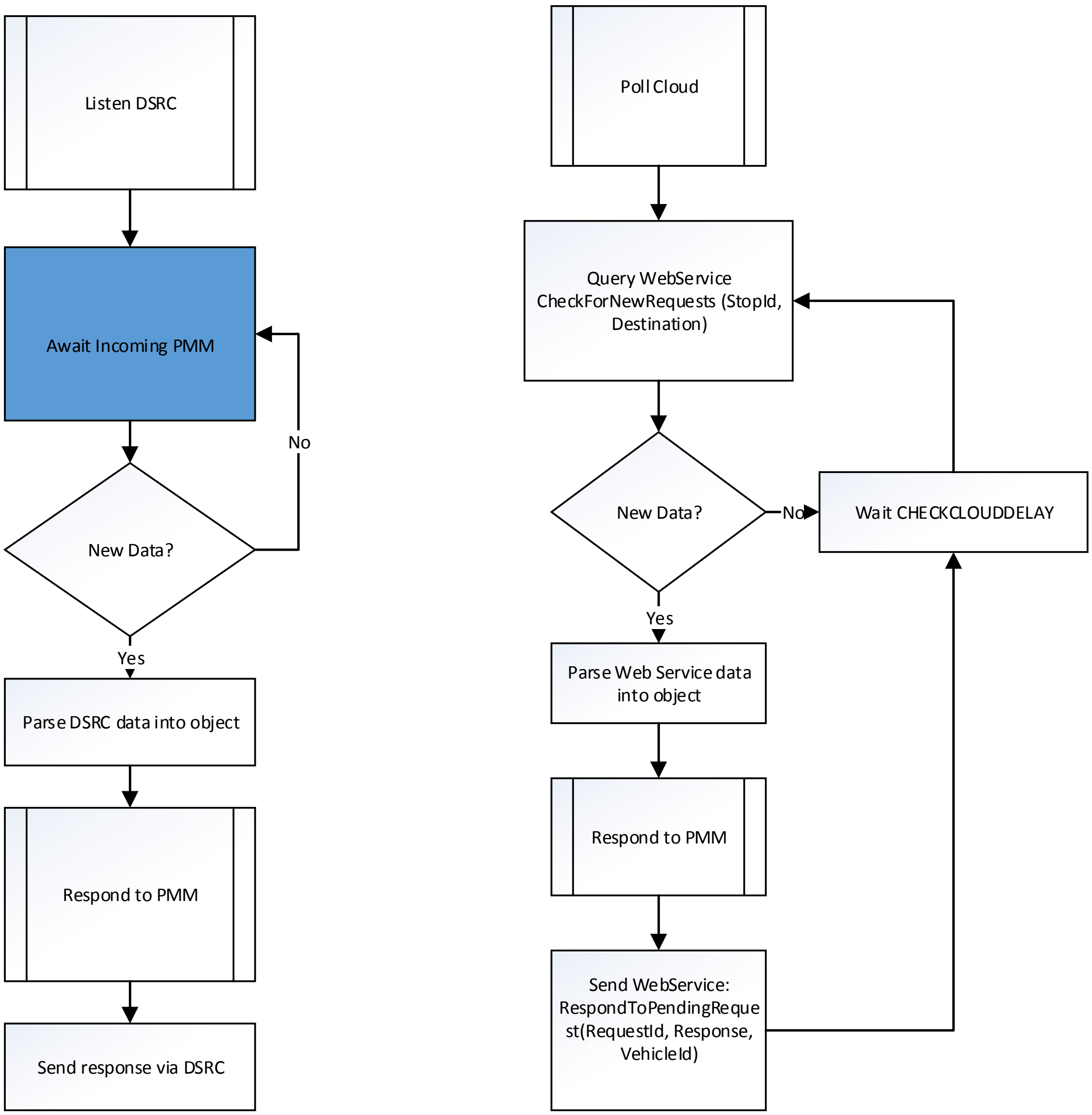
[7] https://en.wikipedia.org/wiki/Haversine_formula

The VEA Pedestrian Monitor will also be responsible for sending out periodic BSMs as well, so that nearby pedestrians can receive these messages and be warned about the location of the vehicle if necessary.

VEA Ride Request Monitor

The VEA Ride Request Monitor is tasked with continually monitoring DSRC traffic and the Azure database to see if new rides have been requested. These requests will be responded to, and if accepted, further updates (such as PMM ARRIVE) will be sent to the requestor.

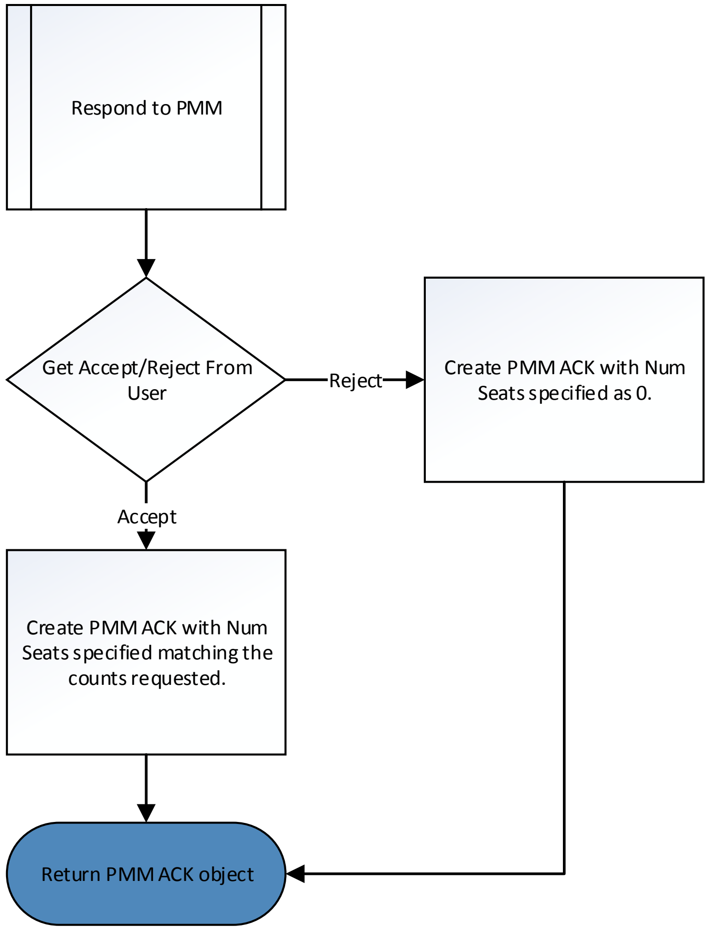
The Ride Request Monitor will spawn two threads in order to perform the required monitoring (Figure 25). One handles incoming DSRC, the other polls the Web API for cloud data, but they both have the same goal of seeing new requests and responding to them.



Source: Battelle

Figure 25: Two Threads for Monitoring Ride Requests

For the EPS, the VEA will not be performing automated management of seat availability. Rather, the user will always manually either accept or decline the request (new or updated). Figure 26 outlines this simplified behavior. Due to this streamlined approach, rejection of seats requested can simply become an acceptance of zero (0) seats.



Source: Battelle

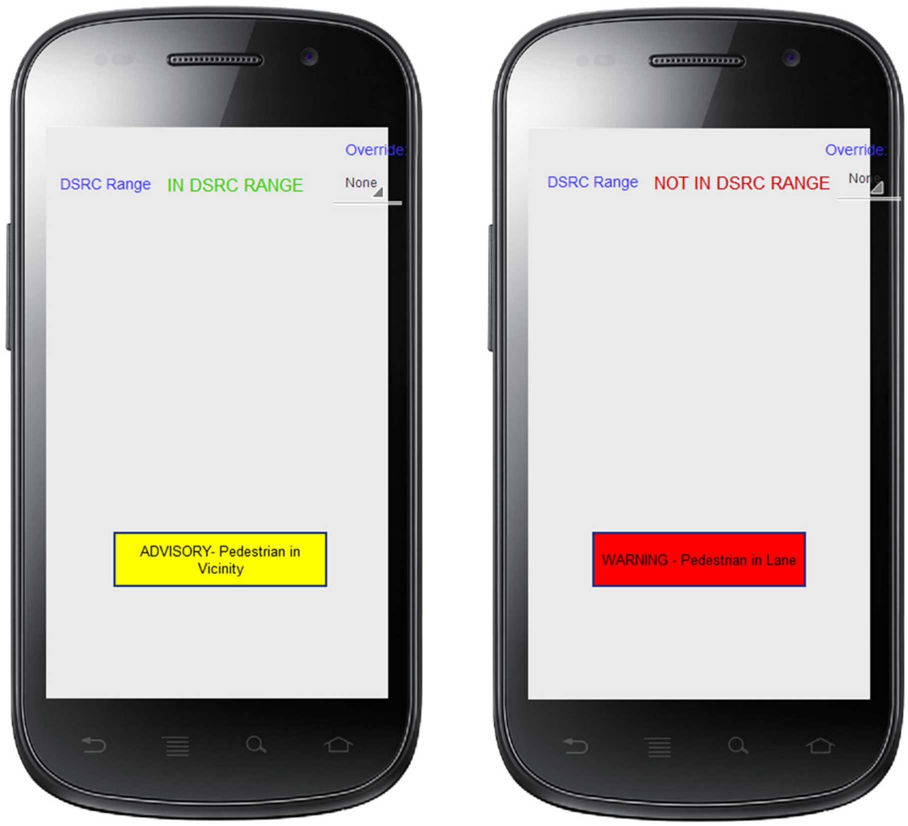
Figure 26: Obtaining Response to PMM Request

The Ride Request Monitor will keep a list of all active requests with the returned number of accepted seats.

VEA User Interface

The primary user display for the VEA application will focus on pedestrian safety (Figure 27). In the mockups the screens are made as simple as possible showing the primary intents of the application.

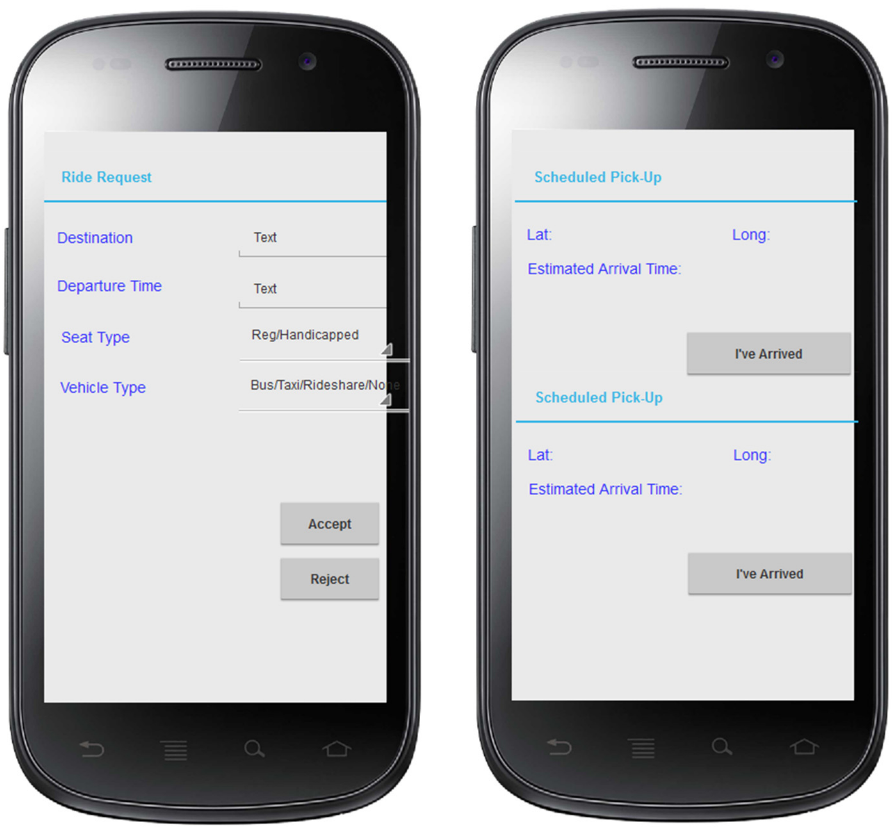
The display will show any advisories/alerts/warnings about nearby pedestrians. Overrides will be provided to force certain conditions for testing purpose, such as preventing traffic over DSRC even if it actually is in range. The Android application's user interface will also contain settings and debug/diagnostic screens as well but those are not shown here.



Source: Battelle

Figure 27: VEA Main Display

When the Ride Request Monitor identifies a new/updated PMM Request, an action screen will be displayed to the driver to accept or reject the number of passengers, seats, and mobility constraints requested (Figure 28). The answer will be passed back to the Ride Request Monitor. Thereafter, the GUI will display all active requests to which the VEA has promised seats. It will display to the driver an "Arrived" button, which may be clicked by the driver when they have arrived at the agreed upon pickup location. After arrival, the request is no longer active and will be cleared from the display.



Source: Battelle

Figure 28: VEA Ride Request Display Screens

8.2.8 Roadside Receiver Application

The Roadside Receiver Application (RRA) is the application that resides on the smartphone/tablet on the roadside, nearby a stop. It serves to receive information and display it to the user – in particular, it will show DSRC traffic that may be occurring prior to the bus/taxi coming in range to receive it.

RRA Monitor

The Monitor will have two (2) threads; one dedicated to retrieving logging any new DSRC data (including SPaT, MAP, PSM, PMM messages), and the other to polling the cloud for new data arriving via that pathway. That way, it becomes a spectator terminal to monitor all processes of the test (other than the Wi-Fi Direct traffic among the coordinating MDEA travelers and other Bluetooth communications).

RRA User Interface

The RRA User Interface screen will be separated into two (2) halves, each scrolling an appending list. One half will be DSRC traffic, and the other will be the table in the cloud containing the PMM requests and responses sent over the cellular network.

8.3 Arada DSRC Radio

The EPS will make use of the Arada DSRC radio as configured in two (2) different packages. The first is the Arada Systems LocoMate™ ME OBU for use in the Mobile Device component. The second unit will be the Arada Systems LocoMate™ Mini2 OBU incorporated into the in-vehicle device. Both DSRC radio units offer similar functionality with the LocoMate™ ME being a small, portable unit that can run on battery power. Both units will send and receive DSRC Messages via an external antenna. When integrated with both mobile and in-vehicle devices, the unit will interface to the Android smartphone via Bluetooth.

When the radio is integrated with the in-vehicle device, it will transmit BSMs (Part I only) to other DSRC Radios that are in range. This unit will also receive and decode PSM and PMM messages and relay those message contents onto the connected smartphone for further processing. PSMs will be used to alert the driver of pedestrians and the PMM will be used for coordination of travel requests.

The DSRC radio included in the mobile device will transmit PSM messages to other DSRC radios that are in range. This radio will also participate in the exchange of PMM messages when coordinating travel requests with other DSRC radios that are part of an in-vehicle device. The location information (GPS coordinates) for both of these messages being sent from the mobile device will be provided by the smartphone. Even though the LocoMate™ ME unit has its own GPS capability, the purpose of the designed experiments are to test the capabilities of the current consumer mobile devices and not those of the DSRC radio products.

The DSRC radio included in the Intersection RSE will transmit MAP and SPaT messages to other DSRC radio that are in range. This functionality for the EPS is intended to be provided by the RSE box and DSRC radio developed for the Integrated Vehicle to Infrastructure Prototype (IVP) project.

Source: Battelle

Figure 29: Mobile Device DSRC Radio Module

8.3.1 Platform

The mobile device DSRC radio will use an Arada System's LocoMate™ ME OBU battery powered unit, shown in Figure 29. The LocoMate™ ME OBU comes in a tiny form factor for use in a portable deployment with a full DSRC Wireless Access in Vehicular Environments (WAVE) software solution and applications for integration with smart phones to ease the human-user-interface. The solution is integrated with GPS (although this capability is not being used for the experiments and instead the GPS capabilities of the smartphone is being used), Bluetooth and high-power 802.11p™ radios. It is fully compliant with Omni-Air's certification and has been used in worldwide deployments including the U.S. DOTs' Safety Pilot in Ann Arbor, Michigan.

Source: Battelle

Figure 30: In-vehicle Device DSRC Radio Module

The in-vehicle device DSRC radio will use an Arada System's LocoMate™ Mini2 OBU unit as shown in Figure 30. The LocoMate™ Mini2 OBU comes in a very small form factor for use with in-vehicle deployments with a full DSRC WAVE software solution and applications for integration with smart phones to ease the human-user-interface. While the LocoMate™ Mini2 does come with an internal GPS, experience from past projects has shown that a more reliably accurate position can be obtained when using an UBlox Neo-7P GPS Receiver. This external GPS receiver is connected to the LocoMate™ Mini2 via USB.

8.3.2 Development Stack

The mobile device DSRC application will be written in C and C++ using interfaces supplied by the manufacturer. The development for the application that resides on the DSRC will use Arada's WAVE API version 1.86. The application will be built to execute on a Linux operating system using the LocoMate™ tool chain version 1.42. This API includes Security, GPS positioning and SAE J2735 messaging.

8.3.3 Application

The application will be based on Arada's example code for the On-Board Equipment deployment. This application implements the messaging, logging and security for the WAVE messages. Additional functionality will be added to this application to incorporate the following:

- Updated handling to support the new PSM and PMM messages
- Support for generation and transmission of the new PSM and PMM messages
- Data exchange with Smartphone applications via Bluetooth

DSRC Message Handler

The DSRC Message Handler handles all DSRC communications to and from the EPS application modules and DSRC radio including packaging and un-packaging of DSRC messages for the EPS applications via a dedicated connection to the DSRC Radio. The DSRC Message Handler receives messages from the EPS application modules, and prepares them to be sent to the DSRC radio, ensuring only properly formatted DSRC messages are send to the DSRC radio. The DSRC Message Handler receives incoming DSRC messages from the DSRC radio, unpacks the messages, and formats them into the EPS message format.

PSM Message Generation

When incorporated in the mobile device component, the DSRC radio will generate and transmit PSMs. Messages will be created and populated with the information available on the DSRC radio itself, such as time, but will also use the most recently provided data from the smartphone such as GPS. The generation and transition rate of these messages will be configurable with the fastest rate being ten (10) messages per second. The transmission of the PSMs will also be able to be started and stopped based on input from the smartphone.

BSM Message Generation

When incorporated in the in-vehicle device component, the DSRC radio will generate and transmit BSMs. Unlike the creation and population of the PSMs, the BSM will be created entirely with data available on the DSRC radio itself. Also, the transmission rate will be fixed to ten (10) messages per second. The ability for stopping and starting the BSM transmissions however, will be made available to facilitate those experiments needing a non-equipped vehicle.

PMM Message Generation

The generation and transmission of PMMs will operate very similarly to that of the PSM and BSM, however, in this case the majority of the data being used to populate the message will be provided from the smartphone. In addition, the PMMs are transmitted at a slower frequency due to them being mobility focused instead of safety.

Message Receiver

All messages are forwarded over Bluetooth channel to the smartphone for further processing.

DSRC Message Logger

The DSRC Message Logger module will record in storage the time and contents of all J2735-conformant DSRC Messages that are sent and received.

External Device Interface

A critical capability of the DSRC radio application will be in its ability to share information with the smartphone. This application will need to manage the Bluetooth connection and quickly respond to requests and promptly provide results. This connection must remain efficient due to the potentially high volume of data being transferred to the smartphone.

8.3.4 User Interface

The user interface to the LocoMate™ ME and Mini2 is through a telnet connection using hardwired Ethernet. Once a user has logged into the device, a command line interface is used to configure the device using standard Linux commands. A command line interface is provided by Arada that will allow the user to manage the operation of the DSRC radio and its external interfaces.

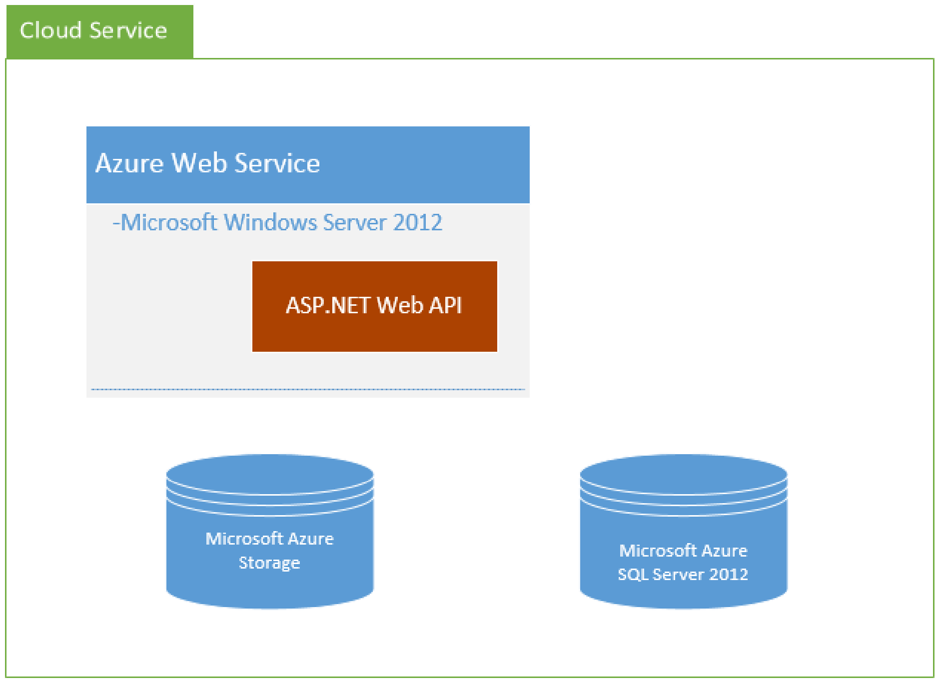
8.4 Cloud Infrastructure

The Cloud Service will be a Microsoft Azure Cloud Service comprised of the following components:

- Azure Web Service
- Azure SQL Database
- Azure Storage

8.4.1 Platform

The computing platform in the Azure Cloud Service will be selected as shown in Figure 31.

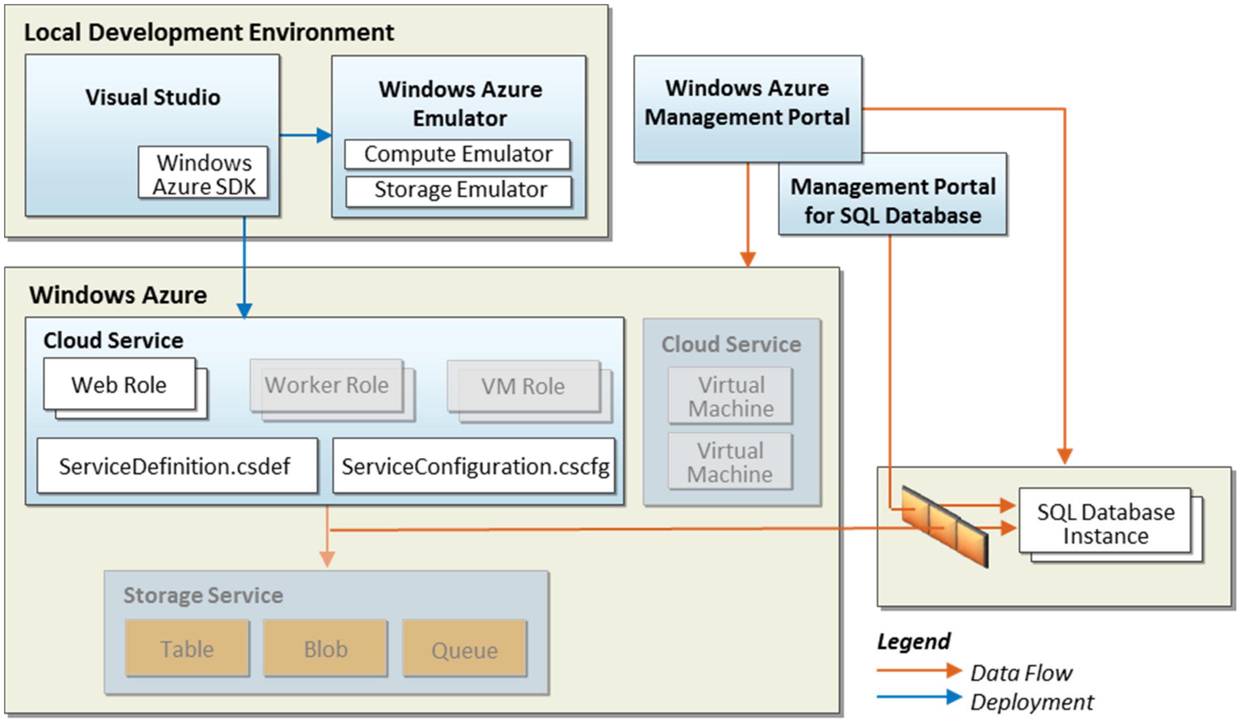


Source: Battelle

Figure 31: Cloud Service Computing Platforms

8.4.2 Development Stack

The Azure software developer will use Microsoft Visual Studio 2015 with the Windows Azure SDK installed. The Azure SDK provides an Azure Emulator for local development and unit testing. The Windows Azure Management Portal provides most configuration capabilities via an interactive web site interface. The Microsoft Azure stack is shown in Figure **32**. Portions of the Microsoft suite that are not being used are greyed out.



Source: Battelle

Figure 32: Microsoft Azure Development Stack

8.4.3 Application

Azure Web Service

The Azure Web Service API will consist of four (4) functions total.

The two (2) following Web API functions will be utilized by the mobile device application:

- AddPMMRequest(Status, GroupId, RequestId, Travel Data)
  - Adds a PMM Request to the database with Status specified. GroupId is unique to each formed travelling group. Updated Requests will maintain the same GroupId throughout until the group is disbanded.
  - Status is New or Updated.
  - RequestId is increasing integer assigned to modification of original travel request.
  - Function will return a unique CloudId so that status of the request can be polled.
- CheckOnPendingRequest(CloudId)
  - Function will return the data from database pertaining to the CloudId specified. This data will include any response that was updated from the In-Vehicle device.

The two (2) following Web API functions will be utilized by the in-vehicle device application:

- CheckForNewRequests(PickupLocLat/Long, Destination)
  - Queries the database for new/updated PMM requests that can be serviced by this vehicle based on pickup location and on destination/route, and that have a Status of New or Updated.
  - A production system would likely include a window of time (duration) in which the request is maintained, but this is intentionally left out of this system for simplicity, since it is merely matching more query parameters and does not add value to the proof of concept.
- RespondToPendingRequest(CloudId, Response Data)
  - Updates the PMM Request in the database specified by CloudId to the Response Data provided. This includes a timestamp and the counts of seats accepted. Also updates the CloudId data item with the VehicleId unique identifier for the vehicle that responded to the request.
  - The Status will be changed to Done. This status aids in querying for new requests that have not yet received responses.

Azure SQL Database

Tables:

PMMRequests

- Id: database assigned unique identifier per row.
- GroupId [GUID]: Unique identifier for this travelling group request, from first request to final boarding.
- RequestId [int]: Incrementing counter assigned per request. Needed in addition to Id, because for DSRC requests that don't involve the Cloud, Id will not be available.
- RequestDate [datetime]: date and time that request was received.
- PickupDate: date and time that the travelers wish to be picked up.
- Status [int, enum lookup]: New, Updated,Done.
- Latitude [floating point number]: Latitude where travelers will be awaiting pickup.
- Longitude [floating point number]: Longitude where travelers will be awaiting pickup.
- Elevation [floating point number]: Elevation where travelers will be awaiting pickup.
- Position Accuracy [floating point number]: Position Accuracy of location where travelers will be awaiting pickup.
- Destination [string]: destination or bus route where travelers wish to travel to.
- RegularSeats [integer]: Number of standard seats requested for the travelling group.
- HandicappedSeats [integer]: Number of handicapped seats requested for the travelling group.
  - A production system could likely include a more types of seating to request, but this is intentionally left out of this system for simplicity since it is merely adding more query parameters and does not add value to the proof of concept.

- VehicleId [GUID/null]: unique identifier of the vehicle (application) that responded.
- ResponseDate [datetime/null]: date and time that response was received.
- TotalRegularSeatsAccepted [int/null]: Total number of Regular Passengers Accepted for Group.
- TotalHandicappedSeatsAccepted [int/null]: Total number of Handicapped Passengers Accepted for Group.

RequestLog

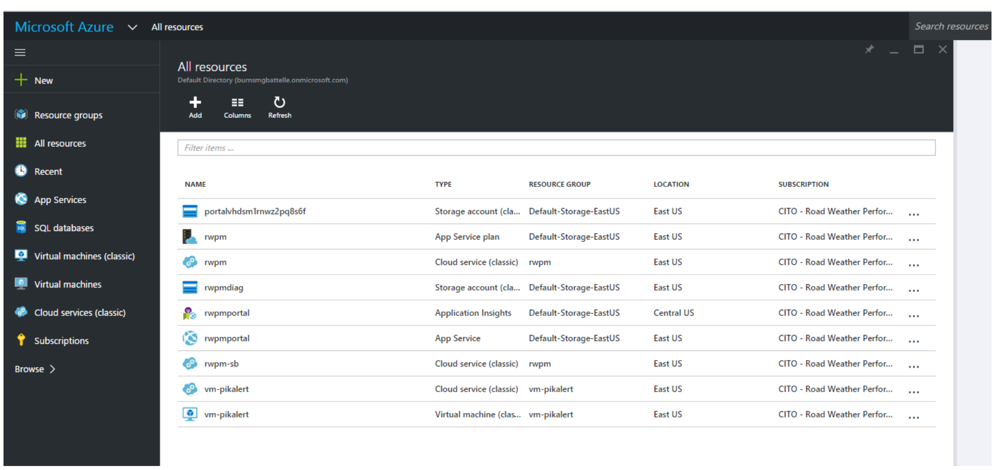
- RequestDate [datetime]: date and time that request was received.
- FunctionName [string]: Name of web service endpoint requested.
- Parameters [string]: Parameters supplied either in message body or url.
- Returned[string]: Stores the value (int or bool) returned from the function, or if the function returns a dataset, stores the count of rows returned.

Azure Storage

Microsoft Azure Table Storage module will be used to record general diagnostic information concerning the health and stability of the Web Service. Azure Storage can be viewed via the Internet using various commercial applications such as Cloud Storage Studio that provide a graphical viewing and query environment.

8.4.4 User Interface

The user interface to the Microsoft Azure Cloud Service is the Microsoft Windows Azure Management Portal, Figure 33 which is an interactive web site located at the following URL: https://portal.azure.com. This portal can be used to view the state of the web service and to start and stop the web service.



Source: Battelle

Figure 33: Example Administration of Services via Azure Portal

Query and viewing of data contained in the database is accessible using SQL Server Management Studio 2012. The server name and credentials are necessary to access this resource data.

8.5 Bluetooth LE Location Beacons

Estimote Inc. Beacons will be used to explore one (1) strategy to detect when a traveler has entered a transit vehicle. By mounting one (1) or more beacons within a transit vehicle, the traveler's phone may detect the presence of the beacon upon entering the bus, and thus determine that the traveler has become a passenger within a vehicle. Different numbers of beacons will be explored to determine which configuration of beacons best identifies when a traveler is on a vehicle versus standing outside of it.

8.5.1 Platform

The Estimote Beacons are small, wireless sensors. Estimote Beacon is a small computer. Its 32-bit ARM® Cortex M0 CPU is accompanied by accelerometer, temperature sensor, and a 2.4 GHz radio using Bluetooth 4.0 Smart, also known as BLE or Bluetooth low energy[8]. Bluetooth Smart does not require pairing to the phone to communicate with the device.

8.5.2 Development Stack

The Estimote Beacon functionality runs on top of the Android SDK, utilizing access to the Android Bluetooth APIs and the Estimote SDK library that is supplied with the beacons.

[8] http://estimote.com/

8.5.3 User Interface

If external visibility of the beacons is necessary, Estimote Inc. provides an Estimote phone app to quickly connect to and access local iBeacons. Through the app you can see any available iBeacons, see identifying information about the beacon, and change the beacon's range or access built-in sensors such as accelerometer.

8.6 Time Synchronization

Synchronization of time across the multiple platforms required of the connected vehicle architecture will be facilitated using the appropriate service or source associated with each of these platforms. All will be based on Universal Coordinated Time (UTC) with corrections for ‘leap’ days included. GPS-enabled devices will acquire their time from that available on the device. Both the mobile device and the roadside equipment will synchronize time using this source. Similarly, the Microsoft Azure cloud services utilize a time reference based on UTC.

Unless otherwise indicated in the design, all timestamp information transmitted or buffered by the applications will be maintained in UTC format.

APPENDIX A. List of Acronyms

Acronym	Description / Explanation
API	Application Programming Interface
ASN	Abstract Syntax Notation
BSM	Basic Safety Message
COTS	Commercial Off The Shelf
DSRC	Dedicated Short Range Communications
EPS	Experimental Prototype System
GATT	Generic Attribute Profile
GNSS	Global Naviation Satellite System
GPS	Global Positioning System
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet Protocol
JSON	JavaScript Object Notation
MAP	Geographic (intersection / stretch of roadway) layout information
MDEA	Mobile Device Experimental Application
NEMA	National Electronic Manufacturers Association
NFC	Near-Field Communications
OBE/OBU	On-Board Equipment/On-Board Unit
OEM	Original Equipment Manufacturer
PII	Personally Identifiably Information
PMM	Personal Safety Message
PSM	Personal Mobility Message
REST	Representational State Transfer
RR	Roadside (DSRC) Receiver
RRA	Roadside (DSRC) Receiver Application
SAE	Society of Automotive Engineers
SDD	System Design Document
SDK	Software Development Kit
SPaT	Signal Phase and Timing
SQL	Structured Query Language
TCP	Transmission Control Protocol

Acronym	Description / Explanation
TMC/TME	Transportation Management Center/Entity
UDP	User Datagram Protocol
URL	Uniform resource locator
U.S. DOT	United States Department of Transportation
VEA	In Vehicle Experimental Application
VM	Virtual Machine
WAVE	Wireless Access in Vehicular Environments
Wi-Fi	Wireless High Speed Internet IEEE 802.11x™

APPENDIX B. Terms and Definitions

Term	Definition
Accelerometer	Hardware sensor that measures the acceleration force on the device along three axes.
Ad-hoc Travel Group	Travel group that is formed though mobile device-to-mobile device communications.
Advisory	As used in Advisory Message vs Alert Message vs Warning Message. Advisory Messages are issued when a pedestrian is in the area.
Alert	As used in Advisory Message vs Alert Message vs Warning Message. Alert Messages are issued when a pedestrian is in the travel lane ahead.
Basic Safety Message (BSM)	Connected vehicle message type which contains vehicle safety-related information that is broadcast to surrounding vehicles;
Bluetooth	Short range wireless technology used to exchange data between enabled devices
Coordinated	Messages are coordinated when one or more mobile devices have boarded a single vehicle (i.e., multiple passengers have boarded a bus), and are interpreted as a single, cohesive sender/recipient.
Destination	The end point of a traveler's trip.
DSRC	Dedicated Short-Range Communications; a low-latency, high-reliability, two-way communications tool used for sending transportation safety messages.
GNSS	The term GNSS includes satellite systems deployed by various nations and regions including the U.S.-based GPS, the Russian GLONASS, and European Union's Galileo.
Gravity (Sensor)	Software sensor that estimates the force of gravity along the three axes.
Gyroscope	Hardware sensor that measures the rate of rotation of the device along three axes.
Linear Acceleration	Software sensor that estimates the acceleration force of the device along three axes, excluding gravity.
Magnetometer	Hardware sensor that measures the geomagnetic field surrounding the device along 3 axes.
MAP Message	J2735-defined message called MAP, a reference to the function of this message to define the geographic layout of an intersection.
Message Type	Type of personal safety or personal mobility message that is transmitted based on the technology used and level of coordination available.

Term	Definition
Mobile Hardware Sensor	Reports raw data from a particular sensor on the mobile device
Mobile Network	A wireless radio network distributed over a large geographic area with fixed location transceivers spread across it. These receivers work together to provide radio coverage over the entirety of the geographic area allowing a large number of mobile devices to communicate with each other.
Mobile Software Sensor	Interprets data from one or more hardware sensors to provide an imputed output
National ITS Architecture	Common framework for the planning, development and integration of ITS deployments.
NFC	Near Field Communications; short-range communications technology (typically 1-2 inches) that may be used to make payments via mobile devices.
Origin	The starting point of a traveler's trip.
Personal Mobility Message (PMM)	Similar to PDM, message intended for the exchange of mobility messages between individual travelers and vehicles/infrastructure, via mobile device.
Personal Safety Message (PSM)	Similar to BSM, message intended to transmit low-latency, urgent safety messages between individual travelers and vehicles/infrastructure, via mobile device
Proximity	Hardware sensor that measures the distance between the sensor and a nearby object.
SPaT Message	J2735-defined message called SPaT, which stands for 'signal phase and timing', which contains the current traffic light indications for an intersection as reported by the local traffic signal controller.
System Engineering Tool for Intelligent Transportation (SET-IT)	A single software tool that integrates drawing and database tools with the Regional Unified Model Architecture so that users can develop project architectures for pilots, test beds and early deployments.
Transmitting	The state in which a traveler has opted in and is sending/receiving messages via mobile device
Traveler advisory message	Connected vehicle message type which Provides congestion, travel time, and signage information.
Uncoordinated	Messages are coordinated when one or more mobile devices have boarded a single vehicle (i.e., multiple passengers have boarded a bus), and are interpreted as a single, cohesive sender/recipient.
Warning	As used in Advisory Message vs Alert Message vs Warning Message. Warning Messages are issued when a pedestrian is in the travel lane ahead.
Wi-Fi	A local area wireless technology, based on IEEE 802.11™, which has been tested by the Wi-Fi Alliance.

APPENDIX C. Hardware Technical Specifications

Nexus 5X Hardware Technical Specifications

Hardware	Specification
Operating System	Android 6.0, Marshmallow
Display	5.2 inches FHD (1920 x 1080) LCD at 423 ppi Corning® Gorilla® Glass 3 Fingerprint- and smudge-resistant oleophobic coating
Cameras	Rear camera • 12.3 MP[1] • 1.55 µm pixels • f/2.0 aperture • IR laser-assisted autofocus • 4K (30 fps) video capture • Broad-spectrum CRI-90 dual flash Front camera • 5 MP • 1.4 µm pixels • f/2.0 aperture
Processors	Qualcomm® Snapdragon™ 808 processor, 1.8 GHz hexa-core 64-bit Adreno 418 GPU
Storage[2] & Memory	Internal storage: 16 GB or 32 GB RAM: 2 GB LPDDR3
Size[3]	147 x 72.6 x 7.9 mm
Weight	136 grams
Colors	Carbon, Quartz, Ice
Media	Single front-facing speaker 3 microphones (1 front, 1 top, 1 bottom)

Hardware	Specification
Battery[4] & Charging	2,700 mAh non-removable battery Standby time: up to 420 hours Talk time: up to 20 hours Internet use time (Wi-Fi): up to 9 hours Internet use time (LTE): up to 8 hours Video playback: up to 10 hours Audio playback: up to 75 hours Fast charging: up to 4 hours of use from only 10 minutes of charging[5] USB Type-C 15W charging
Wireless & Location	LTE cat. 6 Wi-Fi 802.11 a/b/g/n/ac 2x2 MIMO, dual-band (2.4 GHz, 5.0 GHz) Bluetooth 4.2 NFC GPS / GLONASS Digital compass *Wi-Fi use requires 802.11 a/b/g/n/ac access point (router). Syncing services, such as backup, require a Google Account.*
Sensors	Nexus Imprint sensor Android Sensor Hub Accelerometer Gyroscope Barometer Proximity sensor Ambient light sensor Hall sensor
Network	Phone is carrier-unlocked and works on major carrier networks. Check with your service provider for more information. North America: GSM/EDGE: 850/900/1800/1900MHz UMTS/WCDMA: B1/2/4/5/8 CDMA: BC0/1/10 LTE (FDD): B1/2/3/4/5/7/12/13/17/20/25/26/29 LTE (TDD): B41 LTE CA DL: B2-B2, B2-B4, B2-B5, B2-B12, B2-B13, B2-B17, B2-B29, B4-B4, B4-B5, B4-B7, B4-B12, B4-B13, B4-B17, B4-B29, B41-B41
Ports	Micro USB Type-C™, 3.5 mm audio jack, Single Nano SIM slot
Material	Premium injection molded polycarbonate housing

Notes

[1]Final resolution may be lower than 12.3 MP.

[2]Storage specifications refer to capacity before formatting. Actual formatted capacity will be less.

[3]Size and weight may vary by manufacturing process.

[4]Actual battery performance will vary and depends on many factors including signal strength, network configuration, age of battery, operating temperature, features selected, device settings, and voice, data, and other application usage patterns.

Testing was conducted by LG and Google using preproduction Nexus 5X devices and software.

- Talk time tests used default settings with Wi-Fi off and LTE on.
- Standby time tests used default settings with WiFi off and LTE on.
- Wi-Fi internet tests had Airplane Mode on with Wi-Fi connected to a test access point, while loading 20 popular websites cached on a local server. The device loaded a page, waited 40 seconds, and then loaded a page from the next site and so on, in a continuous loop.
- LTE internet tests had Wi-Fi off and LTE on, and used the same testing method as the Wi-Fi internet tests.
- Video playback tests used a 10 minute 1080p video playing natively in a continuous loop, full-screen in landscape mode. Device was put in airplane mode, auto-brightness was turned off and brightness was set to 200 nits.
- Audio playback tests used a 3 minute 256-kbps mp3 track playing natively in a continuous loop. Device was put in airplane mode, auto-brightness was turned off and brightness was set to 200 nits.

[5]Only applies to optimized Nexus devices, such as Nexus 6P and Nexus 5X, charged using the included USB Type-C 15W (5V/3A) charger. Battery must be substantially depleted; charging rate slows as charging progresses.

All battery life claims are approximate and based on an average mixed use profile that includes both usage and standby time.

Proximity Beacons

Identification (model number etc.)	Estimote model REV.D3.4 Radio Beacon.
Frequency range	2400 MHz to 2483.5 MHz
No. of preset switchable channels	40
No. of voice/data/TV channels	40 Data channels (including 3 advertising channels)
Tx-Rx channel separation	2 MHz
Adjacent channel separation	2 MHz
Frequency stability	<20ppm
2nd Harmonic radiation's	<25 dBuV
Mode of emission	not more than 20 DB
Bandwidth of emission	500 KHz
Type of modulation to be required	GFSK
Power output	4 dBm
Sensitivity	-93 dBm
CPU	32-bit ARM® Cortex M0
Flash memory	256 kB

U.S. Department of Transportation
ITS Joint Program Office-HOIT
1200 New Jersey Avenue, SE
Washington, DC 20590

Toll-Free “Help Line” 866-367-7487
www.its.dot.gov

FHWA-JPO-17-476

U.S. Department of Transportation