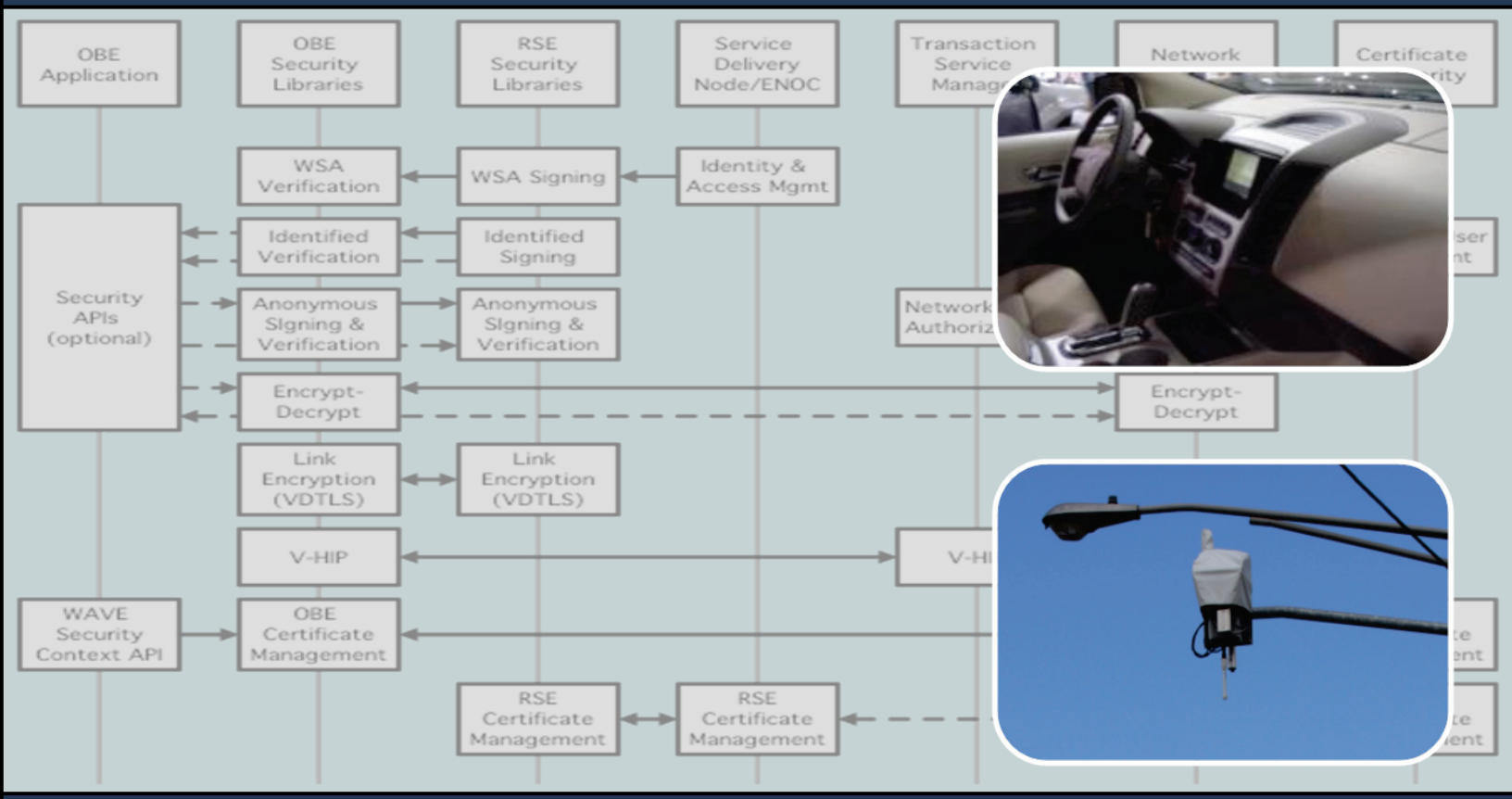
Final Report:

Vehicle Infrastructure Integration
Proof of Concept
Technical Description – Vehicle



Submitted to the
Research and Innovative
Technology Administration,
US Department of Transportation
by
The VII Consortium
May 19, 2009

Notice 1

This material is based upon work supported by the United States Department of Transportation under Cooperative Agreement # DTFH61-05-H-00003.

Notice 2

This report includes references to "Vehicle Infrastructure Integration" (VII). This program name was in official use at the inception of and during the execution of the work described in this report. The United States Department of Transportation has initiated a new program entitled "IntelliDrive℠" which now encompasses all activities that were previously part of VII.

Notice 3

Notice 4

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FHWA - JPO-09-017		EDL 14458

4. Title and Subtitle	5. Report Date
Final Report: Vehicle Infrastructure Integration Proof of Concept Technical Description -Vehicle	May 19, 2009
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Scott Andrews, Michael Cops	
9. Performing Organization Name and Address	10. Work Unit No. (TRAIS)
VII Consortium, Suite 600, 39555 Orchard Hill Place, Novi, MI 48375	11. Contract or Grant No. DTFH61-05-H-00003
12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
U.S. Department Of Transportation Research and Innovative Technology Administration 1200 New Jersey Avenue, SE Washington, DC 20590	Technical description of implemented VII system. Oct 2005 to Dec 2008
	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

This report provides the technical description of the VII system developed for the Cooperative Agreement VII Program between the USDOT and the VII Consortium. The basic architectural elements are summarized and detailed descriptions of the hardware and software systems are provided along with the descriptions of the applications used to assess the system performance and operation.

17. Key Words	18. Distribution Statement
ITS Architecture, IA, Intelligent Transportation Systems, ITS, Vehicle Infrastructure Integration, VII, Architecture, Transportation, Transportation Systems, Infrastructure, Integration, IntelliDrive[SM]	No restrictions. This document is available to the public.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	96	

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

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

GLOSSARY OF TERMS AND ACRONYMS

3G	Third Generation Cellular (Wireless Data System)
AASHTO	American Association of State Highway and Transportation Officials
AOCA	Anonymous OBE Certifying Authority
AMDS	Advisory Message Delivery Service
API	Application Programming Interface
BAH	Booz Allen Hamilton
BER	Burst Error Rate
BMW	BMW of North America, LLC
BSP	Board Support Package
CA	Certificate Authority
CAN	Controller Area Network
CCL	Channel Coordination Layer
CCH	Control Channel
CEP	Circular Error Probable
CM	Certificate Manager
COTS	Commercial Off-the-Shelf
CPU	Central Processing Unit
CRL	Certificate Revocation List
DGPS	Differential Global Positioning System
DSRC	Dedicated Short Range Communications
DTE	Development Test Environment
ECC	Elliptic Curve Cryptography
ENOC	Enterprise Network Operations Center
ESB	Enterprise Service Bus
XML	Extensible Markup Language
FCC	Federal Communications Commission
FHWA	Federal Highway Administration
FPGA	Field Programmable Gate Array
Gbps	Gigabit Per Second
GHz	Giga-Hertz
GID	Geographic Intersection Description
GM	General Motors Corporation
GPS	Global Positioning System
HANDGPS	High Accuracy National Differential GPS
HB	Heartbeat
HIP	Host Identity Protocol
HMI	Human Machine Interface
HPSAM	High Performance Security Accelerating Module
HTTP	Hypertext Transfer Protocol
ILS	Information Lookup Service
IP	Internet Protocol
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
ISO	International Standards Organization
IVI	Intelligent Vehicle Initiative

I2V	Infrastructure to Vehicle
IVPS	In-Vehicle Payment Service
IVTP	In-Vehicle Toll Processing
ITS	Intelligent Transportation Systems
JMS	Java Message Service
JVM	Java Virtual Machine
LIN	Local Interconnect Network
LLCF	Low Level CAN Framework
LTP	Local Transaction Processor
LTTP	LTP Toll Processing
MAC	Medium Access Control
MDOT	Michigan Department of Transportation
MEDS	Map Element Distribution System
MEG	Map Element Generator
MHz	Mega-Hertz
MINAP	Michigan Network Access Point
MTU	Maximum Transmission Unit
NTPDA	Network Trip Path Data Accumulator
NUC	Network User Component
NUG	Network User Gateway
NUPS	Network Users Payment Service
OAA	OBE Authorizing Authority
OBE	On-Board Equipment
OCM	OBE Communications Manager
OEM	Original Equipment Manufacturer
OBNA	Off-Board Navigation Application
OS	Operating System
OSGi	Open Services Gateway Initiative
PC	Personal Computer
PCI	Peripheral Component Interconnect
PDC	Probe Data Collection
PDCS	Probe Data Collection Service
PDDS	Probe Data Distribution Service
PDSS	Probe Data Subscription Service
PDM	Probe Data Management
PDU	Protocol Data Units
PDS	Probe Data Service
PDVC	Probe Data Vehicle Component
PER	Packer Error Rate
PKI	Public Key Infrastructure
POC	Proof of Concept
PSC	Provider Service Context
PSID	Provider Service Identifier
PSN	Probe Sequence Number
RCOC	Road Commission for Oakland County
RF	Radio Frequency
RITA	Research and Innovative Technology Administration
RSE	Roadside Equipment
SAE	Society of Automotive Engineers

SCH	Service Channel
SDN	Service Delivery Node
SDRAM	Synchronous Dynamic Random Access Memory
SIT (Tunnel)	Simple Internet Transition (Tunnel)
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SPAT	Signal Phase and Timing
SRS	Software Requirement Specification
TCP/IP	Transmission Control Protocol/ Internet Protocol
TMT	Technical Management Team
TPGA	Trip-Path General Application
TPT	Trip-Path Transmission
TSM	Transaction Service Manager
UDP	Universal Datagram Protocol
URL	Uniform Resource Locator
USDOT	United States Department of Transportation
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
VAPI	Vehicle Application Programming Interface
V-DTLS	VII-Datagram Transport Layer Security
VEG	Vehicle Expert Group
V-HIP	VII-Host Identity Protocol
VGA	Video Graphics Array
VIDMT	Vehicle Interface Device Management Tree
VII	Vehicle Infrastructure Integration
VIIC	Vehicle Infrastructure Integration Consortium
VIN	Vehicle Identification Number
VIS	Vehicle Interface Service
VPN	Virtual Private Network
VSC	Vehicle Signage Component
VW	Volkswagen of America
WAAS	Wide Area Augmentation System
WAVE	Wireless Access in Vehicular Environments
WME	Wave Management Entity
WO	Work Order
WSA	WAVE Service Advertisement
WSC	WAVE Security Context
WSMP	WAVE Short Message Protocol

1 Abstract

This report provides the technical description of the Vehicle Infrastructure Integration (VII) system developed for the Cooperative Agreement Program between the United States Department of Transportation (USDOT) and Vehicle Infrastructure Integration Consortium (VIIC). The basic architectural elements are summarized and detailed descriptions of the hardware and software systems are provided along with the descriptions of the applications used to assess the system.

2 Organization of the Final Report

The VII Cooperative Agreement Program final report is organized into five volumes:

Volume 1a -- Final Report: VII Proof of Concept Executive Summary – Vehicle
This volume provides an overview of the program goals and objectives, program organization, program technical direction and the key findings and recommendations. The report does not detail test results. This report is recommended for executives and managers of VII communities concerned with the deployment of VII systems.

Volume 2a -- Final Report: VII Proof of Concept Technical Description – Vehicle
This volume describes the technical approach of the program and specifically describes the VII POC system architecture, the system component design and the sample applications designed to enable some of the system testing. In addition the deployment of the system to the test track and Development Test Environment (DTE) is described. This report is recommended for engineering managers and practicing engineers concerned with the deployment of VII systems.

Volume 3a -- Final Report: VII Proof of Concept Results and Findings Summary – Vehicle
This volume describes the test objectives and approach and presents a summary of results and findings for both the system testing and the application testing. Detailed results are not presented. This report is recommended for engineering managers and engineers concerned with the deployment of VII systems. It assumes the reader has knowledge of the VII POC system architecture as described in Volume 2a.

Volume 4a -- Final Report: VII Proof of Concept System Detailed Test Results – Vehicle
This volume describes the system test objectives, the system test approach and details the results of the individual components and the end-to-end system tests. This report is recommended for engineers concerned with the deployment of VII systems. It assumes the reader has knowledge of the VII POC system architecture and components as described in Volume 2a.

Volume 5a -- Final Report: VII Proof of Concept Application Detailed Test Results – Vehicle
This volume describes application test objectives, the application test approach and details the results of the individual application tests. This report is recommended for engineers concerned with the deployment of VII systems and the design of VII applications. It assumes the reader has knowledge of the VII POC system architecture and applications as described in Volume 2a.

Volumes 1a, 2a and 3a have complimentary reports: Volumes 1b, 2b and 3b, which describe the development and testing of the POC Infrastructure written by Booz Allen Hamilton (BAH).

3 VII POC Program Overview

3.1 Background

During the 10th World Congress held in Madrid, Spain (November 2003), the USDOT announced a new initiative, namely, VII. This initiative represents the confluence of three areas of high interest to transportation policy managers: the Intelligent Vehicle Initiative (IVI), an emphasis on improved traffic operations, and the continuing evolution in telecommunication technology.

Regarding the latter item, the Federal Communications Commission (FCC) has allocated 75 MHz at 5.9 GHz for the primary purpose of improving transportation safety. In addition to safety of life and public safety applications, the FCC's Final Report and Order also allows private and non-safety applications to make use of the spectrum on a lower priority basis. Dedicated Short Range Communications (DSRC), the wireless medium, will allow vehicles to communicate with a low-cost roadside infrastructure, as well as with each other, in real time. This communications capability, in combination with a nationwide data collection and processing network, will facilitate improvements to safety, mobility and productivity/convenience.

Reducing the number and severity of roadway transportation incidents is a top priority of the USDOT. Development of a system supporting communication between vehicles and between vehicles and a roadway infrastructure has the potential for positively contributing to the government's goal of improving transportation safety. Such real-time communications would enable a range of crash avoidance and crash mitigation applications with the potential to reduce traffic deaths and injuries, while simultaneously enabling a host of additional applications with secondary benefits, such as optimized traffic and incident management systems.

To enable this vision, it was proposed to undertake a project to specify, design, build and test a small-scale instantiation of the envisioned national system to determine if the concept was sound and could support the intended use. Pending the anticipated results of the system's testing, a nationwide system would be deployed. It was understood that the success of the project required close collaboration between the USDOT, the State Departments of Transportation through the American Association of State Highway and Transportation Officials (AASHTO), and light-duty vehicle manufacturers. These primary stakeholders were brought together by the USDOT in the National Vehicle Infrastructure Integration Coalition.

3.2 Program Goals and Objectives

The original program goals included development and testing of a concept system that could be nationally deployed beginning some time around 2010, to provide a mechanism for wirelessly sending and receiving roadway information to and from vehicles, and between vehicles to satisfy the following viability criteria*:

Safety

- Provides for infrastructure initiated safety applications.
- Supports vehicle initiated safety applications.

Mobility

- Provides for collection of various mobility data from vehicles.
- Provides for use of collected mobility data by state and local authorities.
- Exhibits sufficient benefit in terms of road and traffic management and transportation efficiency.

Private Services

- Vehicles can access private services through the system.
- Private services can access vehicles through the system.
- Co-existence of private services with safety and mobility services is economically viable.
- Private services can be implemented in a manner that does not interfere with safety and mobility applications.

Security

- System is resistant to denial of service, replay and intrusion attacks.
- Security compromises can be identified and mitigated.
- Security credentials can be properly distributed and managed at all levels of deployment.

Maintainability

- Roadside Equipment (RSE) software can be remotely managed through the network.
- VII-related vehicle software can be securely maintained over the vehicle life cycle.

Privacy

- Cannot track an individual vehicle over any road segment longer than 2 km.
- Cannot identify any individual vehicle as violating a traffic law through publicly collected data.
- Cannot identify a vehicle or a vehicle occupant or owner from messages sent to, or through, the infrastructure.

* Note: These criteria were developed by the VIIC at the start of the program. They were agreed upon between the VIIC members and the USDOT. Other criteria have been proposed, but as of this date, none have been fully agreed upon by all of the VII stakeholders. The other criteria are generally a simplified and less comprehensive set, relative to the criteria presented here. These criteria are used for completeness, and in general, any differences between this set and the others discussed are minor.

3.3 Project Roles and Responsibilities

The system concept was understood to consist of a roadside network component and an on-board vehicle equipment component. The responsibility for the network and roadside equipment was assigned to BAH, a USDOT contractor, and the On-Board Equipment (OBE) to light-duty vehicle manufacturers, represented by the VIIC. Additionally, it was anticipated that typical applications would be designed and tested as a part of the project. The responsibility for public applications, i.e. those to be used by the Federal and state governments, was assigned to BAH and those likely to be used by the vehicle manufacturers and providers of the commercial services were assigned to the VIIC and their suppliers for development.

The USDOT's Intelligent Transportation Systems (ITS) Joint Program Office provided oversight and program management. Funding for the program was shared between USDOT and the VIIC, with the USDOT providing the majority share.

3.4 VII Consortium Formation

The VIIC was formed in February 2005 by three manufacturers of light-duty vehicles for the specific purpose of actively engaging in the design, testing and evaluation of a deployable VII system for the United States. The Consortium was also formed to provide a contracting

mechanism for the Cooperative Agreement that was later established with the USDOT Federal Highway Administration (FHWA). Initial membership included Daimler-Chrysler, Ford Motor Company and Nissan Technical Center North America, Inc. Subsequently, the membership increased to include BMW of North America, LLC, General Motors Corporation, Honda R&D Americas, Inc, Toyota Motor Engineering and Manufacturing North America, Inc. and Volkswagen of America, Inc.(VW). Since the split of the Daimler-Chrysler organization, both Mercedes-Benz Research and Development North America, Inc. and Chrysler LLC have retained memberships resulting in the VIIC membership totaling nine light-duty vehicle manufacturers at the time of this report publication. The Consortium is a Michigan 501(c)(6) non-profit organization.

3.5 Cooperative Agreement between VIIC and USDOT

A Cooperative Agreement between the USDOT FHWA and the VIIC was executed on December 7, 2005. The objectives of this agreement are:

- Analysis of the requirements to permit the auto industry to provide a coordinated input to the Vehicle Infrastructure Integration Coalition
- Analysis of the requirements and definition of specific design elements of the VII architecture
- Design of specific hardware to facilitate the implementation of the VII system
- Develop software that could be employed either on the vehicle or in the infrastructure
- Fabrication or procurement of equipment to be used in the test and evaluation of the VII program
- Testing of specific elements and /or combinations of elements of the VII architecture
- Integration of elements of the VII architecture to permit evaluation of the design
- Evaluation of the effectiveness of specific designs with respect to the stated objectives of VII
- Analysis of data and results of the VII program.

The period of performance of the Cooperative Agreement is 60 months.

3.6 VIIC Organization

The VIIC organization is detailed in Figure 3-1.

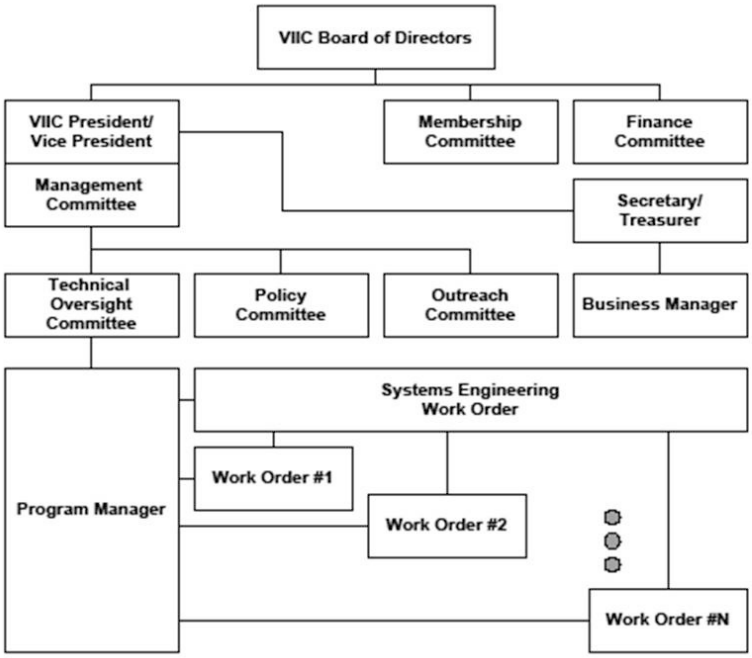


Figure 3-1 VIIC Program Organization

The VIIC Management Committee consisting of members from each of the VIIC participating vehicle manufacturers report to the VIIC Board of Directors, which is responsible for carrying out the charter outlined in the Articles of Incorporation. The Management Committee manages the daily operations of the VIIC through three committees: the Technical Oversight Committee, the Policy Committee and the Outreach Committee. The program is subdivided into a series of twelve (12) Work Orders detailed in Table 1 for executing the technical work policy and outreach programs.

The Policy and Outreach Committees have jurisdiction over the Policy Work Order (WO) and the Technical Oversight Committee has jurisdiction over the remaining WOs dealing with technical content of the program through the office of the Program Manager. Each WO is managed by a Technical Management Team (TMT) leader who is responsible for the technical direction, delivering the work products and maintaining the schedule for the deliverables.

Work Order #	***Work Order Title***	***Work Order #***	***Work Order Title***
1	Program Management	7	Positioning
2	Systems Engineering	8	Security Framework
3	Radio	9	Testing Lab and Facilities
4	Policy Support	10	Field Operational Test*
5	OBE Subsystem	11	Alternative Analyses*
6	Application Development	12	Private Service Enablers

* Not initiated

Table 1 Cooperative Agreement Work Orders

3.7 VIIC Work Order Description

As mentioned previously, the POC program portion of the Cooperative Agreement work was broken down into twelve (12) WOs. The following is a description of each WO as they pertain to the POC. The suppliers in each WO are shown in Table 2.

3.7.1 WO 1: Program Management

Tasks included program management of the overall project including the delivery of all work products on time and within budget. Tasks also included the oversight of the Technical, Policy and Outreach Committees by the Management Committee. Other tasks not included in other WOs were included in this WO.

Deliverables included weekly and quarterly reports detailing progress and expenditure, and quarterly Program Management Reviews.

3.7.2 WO 2: Systems Engineering

Tasks included development of system requirements, allocation of requirements to system components and interface specifications, oversight of interface documentation between WO initiatives, development of test plans and procedures, system and application testing, reports analysis and reporting.

Deliverables included systems engineering and test master plans, VII POC architecture documentation, requirement specifications, requirements allocation documentation, test plans and procedures, test results and final test reports.

3.7.3 WO 3: Radio

Tasks included designing a compatible radio, based on the IEEE 802.11(p) and IEEE P1609.2, P1609.3 and P1609.4 specifications. Also included are the development of specifications and development of all software and testing of the completed work product. The tasks also included feedback to the standards committees and conducting a scalability study.

Deliverables included radio design specifications, source code for all layers, physical delivery of dual antennas, test plans and results and feedback to the standards committees.

3.7.4 WO 4: Policy Support

POC tasks included guidance on Human Machine Interface (HMI) implementation and analysis of testing implications, antitrust issues, patent analysis and patent guidance.

Deliverables included HMI guidelines, guidance to WO teams concerning federal, state and local law compliance, antitrust guidelines, patent infringement and patent defense analysis, deployment plan and business case recommendations and legal obligations.

3.7.5 WO 5: On-Board Equipment Subsystem

Tasks included the development of various design specifications, selection and procurement of a hardware platform, selection of various operating systems and the development of some of the base services including the HMI, power management and vehicle interface. Also included was the

responsibility for integration of all service components resident on the platform, but designed in other work orders.

Deliverables included OBE hardware and HMI specifications, production of a number of dual-purpose antennas, vehicle interface specifications, protocol documentations, interface and final test procedures.

3.7.6 WO 6: Application Development

Tasks included the development of requirements, design specification and the development and testing of on-board and off-board network components for various applications. The applications developed were In-Vehicle Signage, Probe Data Collection (PDC), Off-Board Navigation, Tolling and Parking Payments and Trip-Path Data Collection.

Deliverables included updated use cases, Software Requirements Specifications (SRS), executable code, unit level and integration test reports.

3.7.7 WO 7: Positioning

Tasks included the development of requirements for a Commercial Off-the-Shelf (COTS) positioning system, procurement and testing of same, and the development of a Positioning service and system clock for integration into the OBE.

Deliverables included positioning requirements, Application Programming Interfaces (APIs), executable software, physical hardware, High Accuracy National Differential GPS (HANDGPS) reference station and final test and qualifications plans and reports.

3.7.8 WO 8: Security Framework

Tasks included the development of a security system in accordance with IEEE P1609.2. This included the development of requirements and design specifications for the framework and the development and integration testing of the system components for both the OBE and RSE. The system components included the Certificate Authority (CA), Certificate Manager (CM), Security Cryptographic libraries and a hardware accelerator to support the libraries. Also included was a study of security system vulnerability.

Deliverables included security concept of operations, scalability and threat analysis white papers, CM process specification, POC software architecture and SRS, executable code, software libraries, hardware accelerator boards and probe data encoder.

3.7.9 WO 9: Testing Lab and Facilities

Tasks included the completion of all test plans and procedures, the testing of Communication, Positioning, Vehicle Interface and Security services, integration testing and final testing of applications as part of system end-to-end testing. Also included in the WO were tasks to analyze all results and develop associated reports.

Deliverables included critique of all test plans prepared by others, subsystem component and test equipment supply, test plans and procedures for all services and applications integration test plans, test results for all services testing and final test reports.

3.7.10 WO 10: Field Operational Testing

This WO has not been developed or funded.

3.7.11 WO 11: Alternative Analysis

This WO has not been developed or funded.

3.7.12 WO 12: Private Service Enablers

Tasks included the development of requirements for methods to connect private services to the network in order to conduct point-to-point transactions across multiple RSEs. Included in the work order were the development of OBE and network components and associated protocols to facilitate point-to-point transactions.

Deliverables included requirements specifications, VII POC architecture design, component analysis and selection, software specifications, executable code and acceptance test plans and results.

3.8 VIIC Supplier Selection

The design of a complex system required the expertise of companies with a wide range of diverse skills ranging from automotive electronics to networking systems. It was also recognized that the inclusion of a significant number of suppliers in the program would involve related industries and help to accelerate industry involvement. Invitations to submit proposals were sent to approximately 100 suppliers known to have an interest and the capabilities to develop the VII system. Proposals were reviewed and selections of the final supplier candidates were made by the program team including the VIIC membership. Table 2 details the supplier involvement for each WO.

Supplier	WO Number*									
	1	2	3	4	5	6	7	8	9	12
ABSS Inc.	X									
ARINC Inc.							X			
Battelle Institute		X							X	
BMW	X		X	X	X					
Chrysler LLC.	X		X	X	X	X			X	
Cogenia Partners LLC.	X	X						X	X	
Delphi Automotive Systems Inc					X	X				
Denso International America,			X		X					
Dykema Gossett				X						
Ford Motor Company	X			X		X			X	
Honda R&D Americas Inc.	X			X					X	
Intel Americas Inc.										X
Mark IV IVHS Inc.						X				
Mercedes-Benz	X		X	X	X					
Moser Racing LLC.	X									
MTS LLC.						X				
Navteq North America LLC.						X				
Nissan Technical Center NA	X			X			X			
Ntru Cryptosystems Inc.								X		
Parvus Corporation					X					
Prosyst Software Gmbh.					X					
Raytheon Company						X			X	
Roush Industries									X	
Sirit Technology Inc.			X							
Technocom (now Kapsch TrafficCom)			X					X		X
Telcordia Technologies								X	X	
Toyota Motor Engineering and	X			X					X	

Manufacturing NA Inc.										
Transcore LP.			X							
Volkswagen	X			X	X					
WFET Group		X							X	
Wind River Systems Inc.					X					

*WOs 10 and 11 not initiated

Table 2 VIIC Supplier Involvement in Work Orders

3.9 Collaboration Agreement between VIIC and Suppliers

The Collaboration Agreement was developed to protect the intellectual property right of the WO Participants. The Collaboration Agreement was used in WOs where intellectual property was likely to be used or developed. Each WO was supported by a number of suppliers as detailed in Table 2. As a result of the Collaboration Agreement, these teams of suppliers were able to work collaboratively in a pre-competitive environment.

3.10 Infrastructure Program

BAH acted as the system integrator for the infrastructure and public applications, while the VIIC led the development and testing of the OBE and private applications including test applications to determine the systems capability to support safety applications. The BAH team consisted of Iteris, Raytheon, Sirit, Technocom and Telcordia. Local support in Michigan was provided by the Michigan Department of Transportation (MDOT) and Road Commission for Oakland County (RCOC).

	Prgram Mgmt	System Eng	SDN	ENOC &CA	RSE	DTE	SIT	Public Apps	POC Testing	O&M	Stds
Booz Allen Hamilton	X	X	X	X	X	X	X	X	X	X	X
Iteris		X						X			X
MDOT/RCOC						X			X	X	
Raytheon		X			X	X	X				X
Sirit					X		X				
Technocom/Kapsch					X		X				
Telcordia		X	X	X			X				

Table 3 Roles and Responsibilities of BAH Team

3.11 Combined Program Management Process

Formal program management processes were applied as appropriate for the size and complexity of the program. The need for this was accentuated by the number of WOs, (the relatively large number of suppliers spread across two continents) and the assignment of responsibilities between VIIC and BAH as described in Section 3.3. To achieve this goal, the program was broken down into the following tasks:

1. Collect stakeholder requirements.
2. Develop a concept of operations for the system, based on stakeholder requirements.
3. Develop requirements for the system concept and its components.
4. Develop or procure the components according to the requirements.
5. Assemble and deploy a small scale version of the system.
6. Perform integration testing.
7. Perform system testing against performance specifications.

8. Analyze results and determine if the viability criteria for the system had been met.
9. Report on the findings of the program.

The formal processes employed to manage the program included:

- Weekly WO program progress meetings with the TMT leaders chaired by the Program Manager.
- Weekly WO supplier progress and technical meetings chaired by TMT leaders.
- Weekly coordination meetings between VIIC and BAH.
- Weekly reporting to the USDOT.
- Quarterly Program Management Reviews with the USDOT and supply community.
- Engineering Review Board Meetings (VIIC only).
- Configuration Control Board Meetings.
- System Integration Team meetings to review and discuss component interfaces and operation. These meetings were, on occasion, held on a daily basis to resolve difficult and urgent technical issues.

Tools used for program management included Microsoft Project for project scheduling and task completion assessment and QuickBooks for project accounting.

4 VII POC Technical Overview

4.1 POC System Architecture Description

The POC system includes mobile terminals that were typically installed in vehicles. In the POC, these units are known as On-Board Equipment (OBE). OBEs exchange messages with each other for Vehicle-to-Vehicle (V2V) applications and with the stationary roadside terminals known as Road Side Equipment (RSE) for Vehicle to Infrastructure (V2I) applications. The link between OBEs and between OBEs and RSEs is the Dedicated Short Range Communications (DSRC) Radio system. The RSEs are connected to, and are remotely managed from a Service Delivery Node (SDN) and an Enterprise Network Operations Center (ENOC). The SDN provides a variety of services that are described in more detail in subsequent sections.

A critical aspect of the VII architecture is the management of scale. The system needs to be designed to support 100% vehicle deployment, which translates to just over 200 million vehicles. In operation, this means that applications such as PDC may be handling tens of millions of messages per second, across the entire network. The system must allow a single user to post, for example, a warning sign in the vicinity of a particular hazard. To manage these large-scale extremes, the system uses a tiered tree-like architecture (See Figure 4-1). As can be seen in the figure, any given RSE needs to be capable of interacting with up to 250 vehicles at any given time. This is determined primarily by the number of vehicles that can fit inside a typical Radio Frequency (RF) footprint, which provides a range of approximately 250 m.

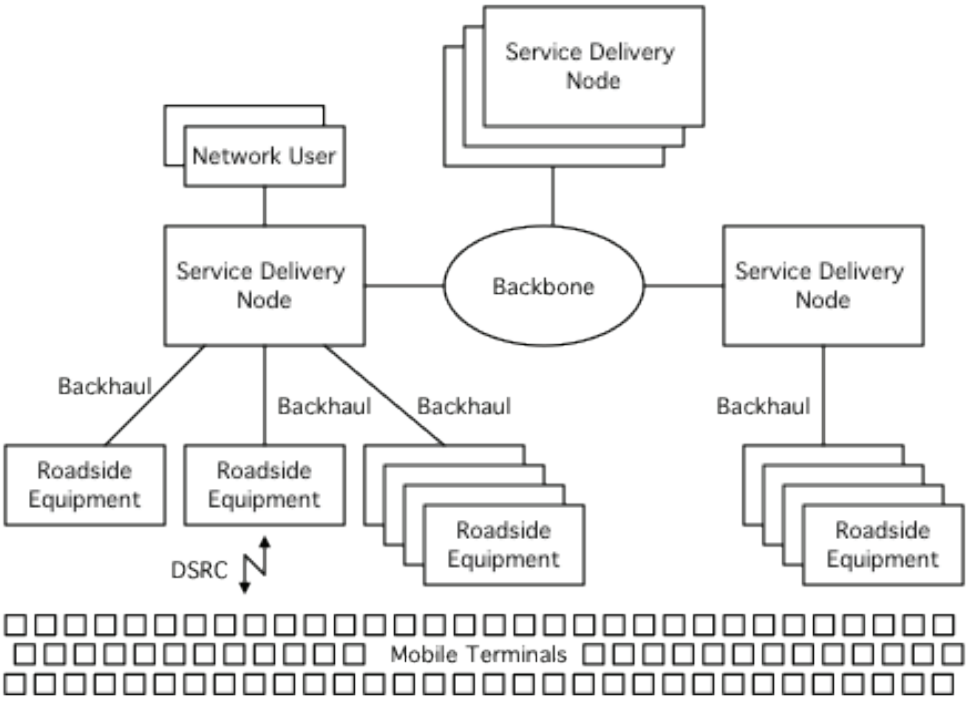


Figure 4-1 Overall System Structure

Each RSE is connected to a regional SDN via a backhaul link, and each SDN is connected to all other SDNs via a wide band backbone network. Using this architecture, any RSE is accessible from any SDN, and this is a key feature of the scalability of the system, since any user connecting to the local SDN can interact with any RSE.

A typical SDN is expected to support between 1000 to 2000 RSEs, so, for a nationwide deployment there might be between 100 and 200 SDNs. The SDN provides a variety of services, but key to the discussion of scaling are the Advisory Message Delivery Service (AMDS), the Probe Data Collection Service (PDCS) and the Network User Gateway (NUG).

The AMDS serves as a link between network users who have advisory messages to distribute and the entire installed base of RSEs (and therefore the entire OBE population). In one possible deployment scenario, it is expected that the number of signage providers will be less than about 10,000, so for signage, the system allows 10,000 providers to efficiently interact with 200,000 RSEs and deliver signage messages to 200 million OBEs. In the reverse direction, the PDC collects probe data from all of the RSEs attached to the SDN. This data is parsed into data "topics," and then data for any given topic is distributed to network users who have subscribed to that particular topic. A typical topic might be instantaneous road speed at a particular location on a particular road. This scheme allows the system to collect vast amounts of data from vehicles on the road, and sub-divide the data passing only those parts of interest to any given subscriber. It is expected that there may be about 10,000 to 100,000 probe data subscribers, and as with AMDS, the system effectively scales from over 200 million vehicles (generating roughly 50 Gbps) to about 50,000 users of this data. The SDN also provides a simple routing system that links vehicle users with private service providers in either one-to-one or one-to-many relationships. In this role,

the SDN is effectively like a mobile Internet system linking users to web service centers such as navigation providers.

The POC implementation of the system includes 55 RSEs placed at various locations in the northwestern Detroit suburbs. These RSEs are linked to two different SDNs using a variety of different backhaul technologies. One SDN is located in Novi, Michigan, and the other is located in Herndon, Virginia. The Herndon facility also includes an ENOC and the CA required to support security functions. The POC implementation is thus a minimalist version of the national system architecture allowing the program to assess the operational behavior of the system as if it were a full-scaled deployment.

4.2 Concept of Operations

Conceptually, the system provides several core functions from which a suite of applications may be created. The POC system:

- Delivers broadcast messages from network providers to OBEs at specified geographic locations
- Delivers broadcast messages from local systems such as traffic signals or toll stations to OBEs at specified geographic locations
- Delivers broadcast messages between OBEs
- Collects data from OBEs and distributes topical information extracted from the data to network subscribers
- Provides OBEs access to remote private service providers, and this access can be carried over from one RSE to the next without disrupting the service
- Provides security functions to protect against attacks and to protect the privacy of the individual users.

As part of the overall VII program, a set of approximately 100 use cases or applications were developed by various stakeholder groups. In general, these descriptions did not fully articulate the use cases in the context of the system, but they did provide insight into the needs and priorities of the various stakeholders. From this initial set, 20 use cases were expected to be available at the system's initial deployment and were identified and articulated in more detail. This group is known as the "Day-1 Use Cases."

Because developing and testing all 20 Day-1 Use Cases would have been impracticable, the POC program identified a subset of use cases that exercised the core functions described above. These were then implemented and tested in ways to assess both the functionality of the system and the baseline performance, under the assumption that the system would provide these core functions in the same way regardless of the specific details of the application.

In several of the safety applications, the use cases were scaled back to allow the assessment of key architectural and system aspects without requiring development of a full-blown application.

This report focuses on the VIIC's developed and tested POC applications described in the following sections.

4.3 Dedicated Short Range Communications

The 75 MHz band in the 5.9 GHz frequency range allocated by the FCC offers significant data transfer capacity. However, to make use of this spectrum in a mobile environment required development of new communications protocols. The core radio protocol used is based on the well-known IEEE 802.11a/b/g wireless Ethernet standard, often referred to as WiFi. Because of the unique mobile environment, the IEEE 802.11a standard was modified to allow what is known as an "association-less" protocol, identified as IEEE 802.11p. This means that the system does not establish a conventional network with all of the mobile terminals as nodes, all of which know about each other. The reason this is not done is that the mobile terminals (OBEs in the POC) are entering and leaving the hot spot rapidly, and there is insufficient time available to set up a new network identity for each new arrival, and inform all other nodes in the network before the network changes again because a terminal has left the footprint of the RSE, or a new one has arrived. On the surface, this approach may seem to limit the functionality of the system since it means that any given mobile terminal cannot interact uniquely with another terminal (the way computers on an office network might), but this is not the case. Because the system is radio based, all terminals can hear all messages sent. Since, under most circumstances one can simply broadcast a message in the local area, and all terminals (OBEs and RSEs) can receive it, there is no need to establish a unique low-level network identity for each communicating device.

The higher levels of the protocol are defined in a suite of standards known as IEEE 1609 Wireless Access in Vehicular Environments (WAVE). This suite addresses security (IEEE P1609.2), networking and messaging (IEEE P1609.3) and channel management (IEEE P1609.4). In particular, IEEE P1609.3 defines a WAVE Short Message Protocol (WSMP) that allows a simple way for a terminal to send messages in the local vicinity of other terminals within local radio range. WSMP allows for direct message addressing based on the Medium Access Control (MAC) address of the intended recipient, but in practice most WAVE Short Messages (WSMs) are broadcast and therefore, are not addressed to any specific recipient.

The current DSRC standards divide up the 75 MHz spectrum into 10 MHz channels. This allows RSEs in local proximity of each other to provide services without causing interference. Also, because the physical layer protocols are based on IEEE 802.11a, the DSRC standard allows for use of existing commercial IEEE 802.11a radio components. Since it is critical for safety reasons to ensure that all terminals can hear each other, and the standards developers did not want to assume the use of multiple radio receiver systems (or very wide band receiver systems), a method for channel management was developed and described in IEEE P1609.3 and P1609.4. The approach separates terminal operations into two modes: "Provider" mode and "User" mode, and splits the use of channels into two time intervals (of 50 ms each). The Control Channel (CCH) interval and the Service Channel (SCH) interval are illustrated in Figure 4-2. All terminals are required to monitor the CCH during the CCH interval. In Provider mode, the terminal transmits a WAVE Service Advertisement (WSA) on the CCH during the CCH interval, and since all terminals are monitoring this channel at that time, they all receive the WSA. The WSA contains a list of the services that the provider (typically an RSE) will provide during the SCH interval along with the SCH channel number they will be using. The services are identified by a code number known as a Provider Service Identifier (PSID). If a terminal in User mode (typically an OBE) receives a WSA that contains a PSID of interest (for example, a message associated with an application that is active on that terminal), the terminal will switch to the appropriate SCH during the SCH interval, and make use of that service.

.

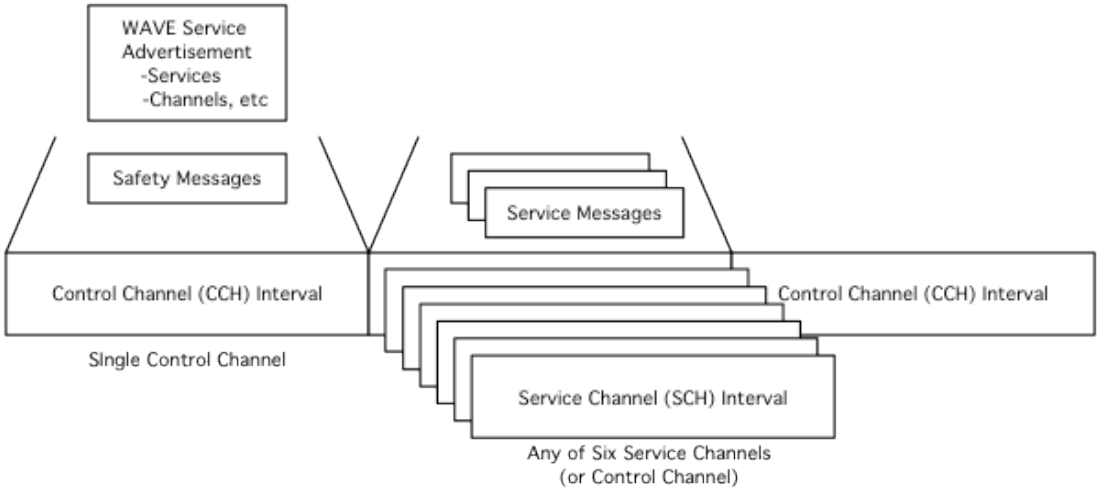


Figure 4-2 DSRC Channel Management Concept

Because all terminals are required to monitor the CCH during the CCH interval, all high priority safety messages are sent on the CCH during the CCH interval.

All low priority services and other services using Internet Protocol (IP) are restricted to use the SCH during the SCH interval. The result of this method is that all terminals have a high probability of receiving important messages, and less important message traffic is distributed across the other channels, thereby reducing congestion.

IP transactions typically require some form of network setup, and, as described previously, the DSRC protocol does not establish this. To support this type of traffic, the WSA also contains the IP address of the provider. In general, the standard does not describe the use of IP between OBEs because OBE-to-OBE messaging is safety-related and will use WSMP on the CCH. This avoids any issues with OBEs needing to route packets (although this sort of usage is not prohibited, it is just not defined in the standards). Once a user terminal has acquired the RSE IP address, it can then create its own IP address using Internet Protocol version 6 (IPv6) rules and can then send IP packets to remote service providers. These packets are routed by the RSE through the backhaul network to the SDN and from there, through the network gateway to the Internet (and then to the service provider).

While somewhat more complex than typical protocols, DSRC achieves the unusual feat of administering communications resources in real time to assure that critical safety messages will have top priority, also allowing lower priority messages, both local messages and messages bound for distant servers, to simultaneously use the system.

4.4 Security Subsystem

The VII Security subsystem is a complex set of functions and services that operate in parallel with the other elements of the system to ensure safe and verifiable system behavior and to prevent misuse of, and attacks on the system.

4.4.1 Security Subsystem Objectives

The VII Security subsystem is aimed at ensuring three basic objectives: privacy, authenticity and robustness. The basic structure of the Security system is also designed to provide assurance, relative to the confidentiality of private message traffic, the authenticity of public message traffic and the anonymity of private generators of public messages.

Privacy

Privacy is addressed in two ways in the VII Security subsystem. Fundamental to the system operation is the assurance of anonymity and confidentiality. While service providers outside the system may need to know the identity of a specific OBE, the VII system itself has no reason to know this information. The system has been specifically designed to avoid requiring any form of traceable or persistent identification of any OBE. In addition, when identifying information is passed through the system to trusted service providers, the system provides mechanisms to encrypt this information so that none of the system elements or operators can access it. Finally, these encryption schemes are also used to suppress the opportunity for observers to correlate operational information (e.g. vehicle speed information) with physical observation, and thus the system also protects against misuse by external attackers.

Authenticity

In any system, it is desirable to require users to prove their authorization to access and/or use the system's resources. The VII system is unique since, for the OBE, this authentication must be accomplished without violating the user's privacy. The VII Security system provides a sophisticated means for validating an OBE's legitimacy without identifying the OBE. This approach allows users to be assured that information provided by the system is legitimate and truthful, and it allows the system to prevent access to the system by users with no authorization, or OBEs that appear to have been tampered with.

Robustness

It is inevitable that the system will be attacked. These attacks may be full-scale sophisticated attempts to disrupt the system, or they may be small-scale pranks. In any case, the system must make it very difficult to mount an attack; it must be capable of identifying and terminating a severe attack in progress, and it must provide a means for rapid recovery of full capability following any actions to terminate the attack.

4.4.2 Security Constraints

The system must perform all of the functions summarized in Section 4.2 while subject to the following operational constraints:

Anonymity

Anonymity was discussed briefly as an element of privacy in Section 4.4.1. The system must perform all of its required functions without identifying the OBE and without disclosing any private information being passed through the system between trusted users and providers.

Inability to Track

The system is designed to assure anonymity and to protect private information while inside the system. However, this is not sufficient to assure that the system cannot be used for improper purposes. Since the system will be used by vehicle users as they move about geographically, it must be impossible to use anonymous and encrypted vehicle messages to track a vehicle from place to place. This means that the messages must not only be non-identifying, but they must also

have limited and controllable relationships to each other, so that they cannot be linked together to form a trace of the movements of the vehicle. In other words, message transactions occurring at different geographic locations in the system must not contain any information that allows an observer (legitimate or not) to know the path of the vehicle they are associated beyond a certain distance.

Scalability
Since the deployed system will include hundreds of millions of vehicles, the security solutions used must be scalable to these large volumes without incurring excessive costs or performance degradations. The increases in hardware and processing should scale at worst, linearly and preferably sub-linearly with the number of users. Similarly, the impact of low levels of misbehavior should not result in increasing disruption of the system as the user population increases.

Lifecycle Management
The system must be manageable without imposing any special or unusual service requirements over this vehicle life span. Security updates and re-authorizations should, under normal conditions, occur transparently to ordinary users, or at worst, occur concurrently with other service and maintenance activities.

4.4.3 POC Security Architecture

The VII Security system is based on the well-known asymmetric cryptography system. This approach uses pairs of public and private keys to encrypt and decrypt information. The keys are mathematically designed so that each key will decrypt what the other key encrypts (a so-called asymmetric key pair). These pairs typically are separated into a public key (one made generally available) and a private key (one kept secret). In many cases, the communicating parties do not have an established trusted relationship. As a result, the parties need to send their public keys to each other. To assure the authenticity of these keys, the key is digitally signed using the key of a well-known CA (that presumably both parties know and trust). Digital signing is a process whereby a checksum (called a “digest” or “hash”) of the signed document (in this case, the key) is encrypted using the signer’s private key. The signer’s public key is typically sent with the certificate, but it is also verifiable by checking with the CA. Using the known public key, the receiver can check that the received signature "decrypts" giving the checksum of the received message. This “verification" process assures the receiver that the sender of the message is certified by the CA and that the message was unchanged and sent by the claimed sender. In many cases, the recipient is already in possession of the CA’s public key.

Conventional public key systems use very large keys. Since all messages in the VII system will eventually be transmitted over the limited bandwidth radio link, the VII Security system uses keys based on Elliptic Curve Cryptography (ECC). These keys are typically about 1/4 the size of conventional keys. The resulting certificates are about 1/8 the size of conventional certificates as defined in the X.509 standard (the certificates used in Internet security). The certificates for VII vehicle security are defined in the IEEE P1609.2 standard.

The various security operations are shown in Figure 4-3. This figure illustrates the main security relationships between the various elements of the system that use the IEEE 1609.2 standard. This includes signing and verification of WSAs, WSMPs and IP traffic between OBE applications, as well as certificate management functions and methods to secure probe data and network transactions. The POC project included development and test of all elements of this subsystem except the VII Host Identity Protocol (V-HIP) function.

The Security Services in the OBE are described in Section 4.5.3.4, and the CA structure for the VII system is described in Section 4.11. Volume 2b, Infrastructure System Technical Description, provides an overview of the network user authorization and other security mechanisms used to secure the various SDN, RSE and ENOC transactions.

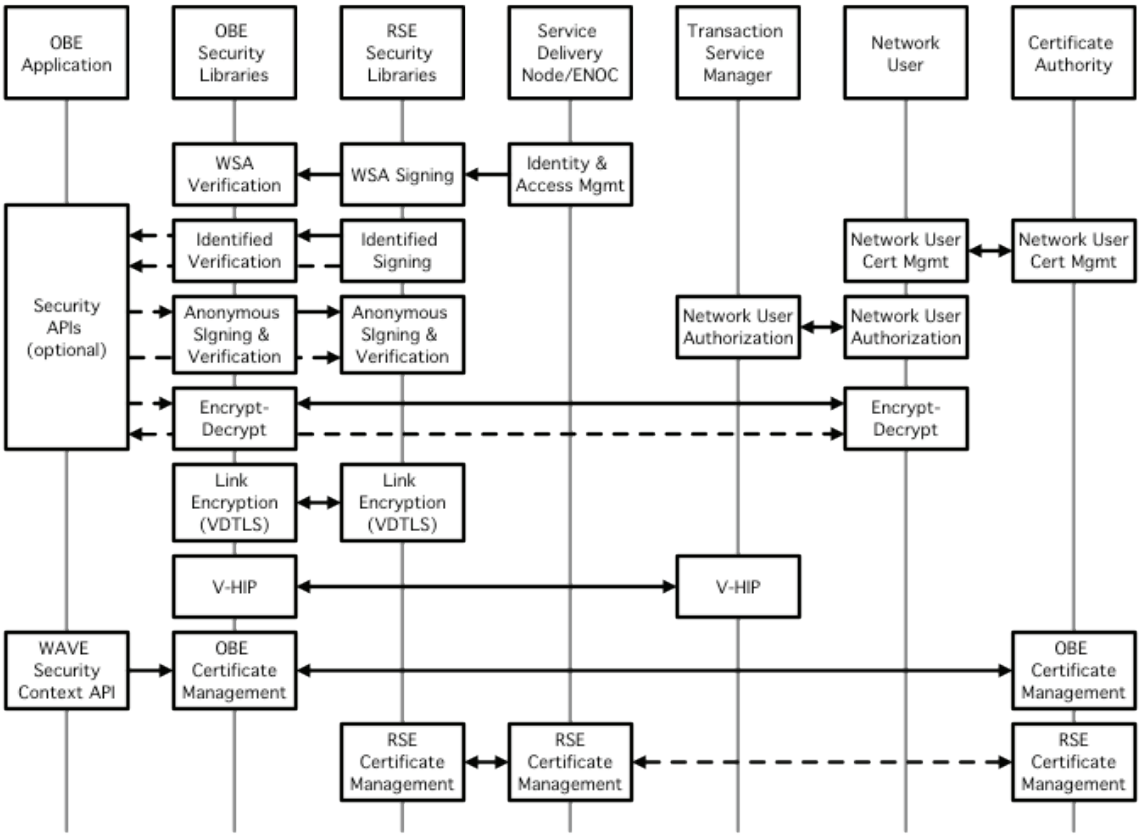


Figure 4-3 Security Subsystem Transactions

4.5 On-Board Equipment Description

The OBE is a self-contained computing system that supports a wide variety of applications and services. It is typically intended to be used in a vehicle, although it is also capable of bench-top operation. It was not intended to be a deployable platform but as a test platform for use in the POC.

The OBE computing platform hardware is the central piece of hardware responsible for vehicle interactions within the VII network. The hardware supports communications with other VII components, exchanges data with Original Equipment Manufacturer (OEM) vehicle systems through a Controller Area Network (CAN) interface, and accommodates driver interaction through a HMI. In addition to providing the hardware implementation of VII OBE interfaces for the POC, the OBE computing platform hardware also provides daughter card slots and assorted local interfaces which provide feature, control and test flexibility during POC. Figure 4-4 provides a diagram showing the OBE computing platform hardware within the context of the POC vehicle and related VII components.

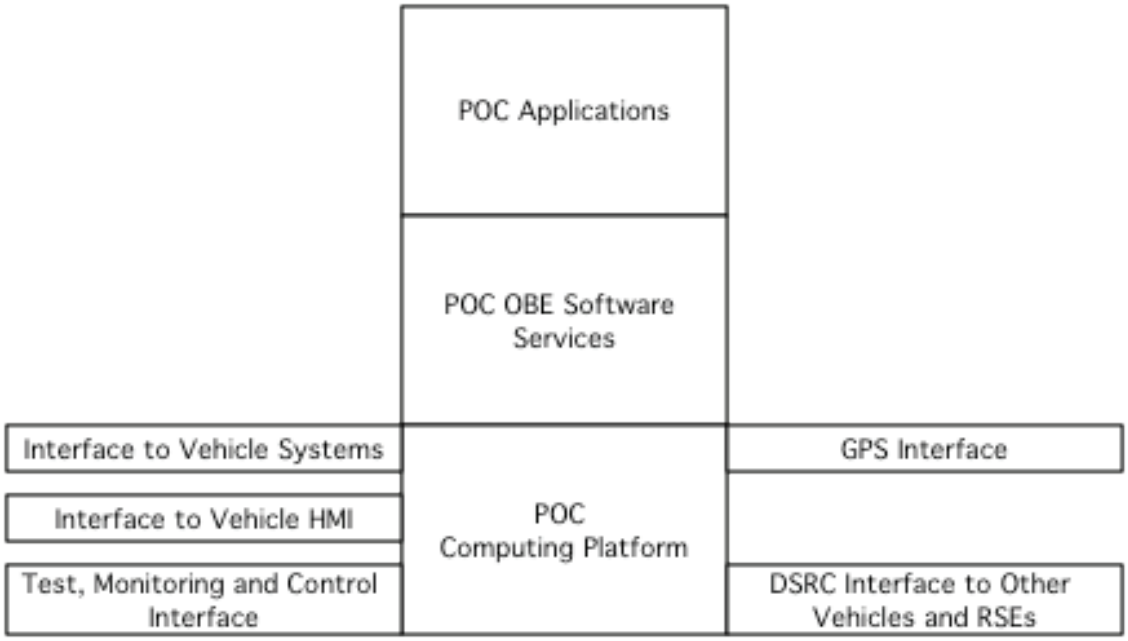


Figure 4-4 OBE Subsystem Interface Diagram

The OBE subsystem is shown in Figure 4-5. This subsystem is based on an Intel processor based computer (OBE processing unit) running the Linux Operating System (OS), and configured with a variety of software services, as described later in this section. In support of the processing unit, the OBE subsystem also includes a touch screen display device, an external combined Global Positioning System (GPS) and DSRC antenna, a programmable power management system and an external positioning unit.

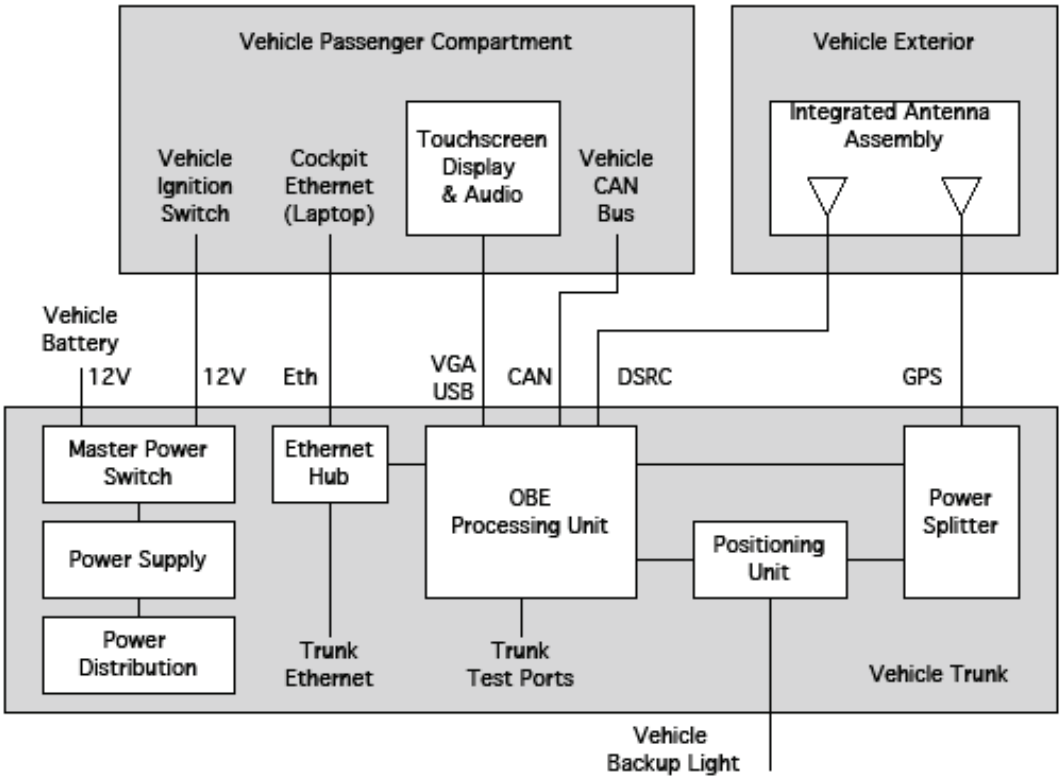


Figure 4-5 OBE Subsystem Block Diagram

4.5.1 OBE Processing Unit

To minimize design and development time, the hardware computing platform for the OBE was selected from a range of off the shelf ruggedized computers designed for mobile application. A commercial WiFi radio was added to provide the physical layer of the DSRC Radio and a hardware accelerator was added to augment the processing speed required for security functions. A Linux OS was selected to match the various system requirements.

4.5.1.1 EuroTech DuraCOR

The primary processing functions for the OBE are performed by the EuroTech DuraCOR unit. This is a rugged self-contained, convection cooled (fan-less), embedded computer-based on a standard x86 architecture.

The DuraCOR unit was selected from among dozens of candidate units based on its unique combination of size, packaging, processing capability and the number and type of I/O interfaces. EuroTech has supplied transit and rail onboard computing modules in Europe and had some experience in the U.S. market as well. The DuraCOR uses an Intel Celeron 400 MHz processor with 256 MB Synchronous Dynamic Random Access Memory (SDRAM). Program memory is provided by a 2 GB solid-state disk for reliability and robustness to the environmental conditions expected in the vehicle environment. While this unit is modest by Personal Computer (PC) standards, it is substantially more capable than current automotive embedded production systems which operate at lower speeds and support less memory.

One of the key factors in selecting the DuraCor was the availability of numerous I/O options. The DuraCOR has eight serial ports and two CAN interfaces, as well as four Universal Serial Buses (USBs) (2x USB2.0 and 2x USB1.1) interfaces, a Local Interconnect Network (LIN) I/O, and 10/100 BaseT Ethernet ports. In addition, the usual Video Graphics Array (VGA), audio, and keyboard/mouse interfaces are provided.

The DuraCor also supports two Mini-PCI expansion ports. One port is used for the DSRC Radio card and the other supports the High Performance Security Accelerating Module (HPSAM) security processor/accelerator. An internal Wide Area Augmentation System (WAAS)-enabled GPS device is included, and this is used to provide backup positioning as well as the precise Pulse Per Second timing used for channel synchronization (See Section 4.3).

The module accepts 12/24V supply and specifies an ambient operational temperature range of -20C to +55C (the unit was tested successfully to +65C). Vibration meets the EN 50155 category 1 class B, and Electromagnetic Capability (EMC), supported by (European Standards) EN 50155 and Economic Commission for Europe/ Organisation des Nations Unies (ECE)/(ONU) regulation No. 10/2 as well as the essential constraints defined in EN 60690. The basic unit is shown in Figure 4-6.

Photograph provided by Eurotech, Inc.

Figure 4-6 DuraCOR Processing Unit

The internal architecture, showing the motherboard and the supporting expansion cards are shown in Figures 4-7 and 4-8.

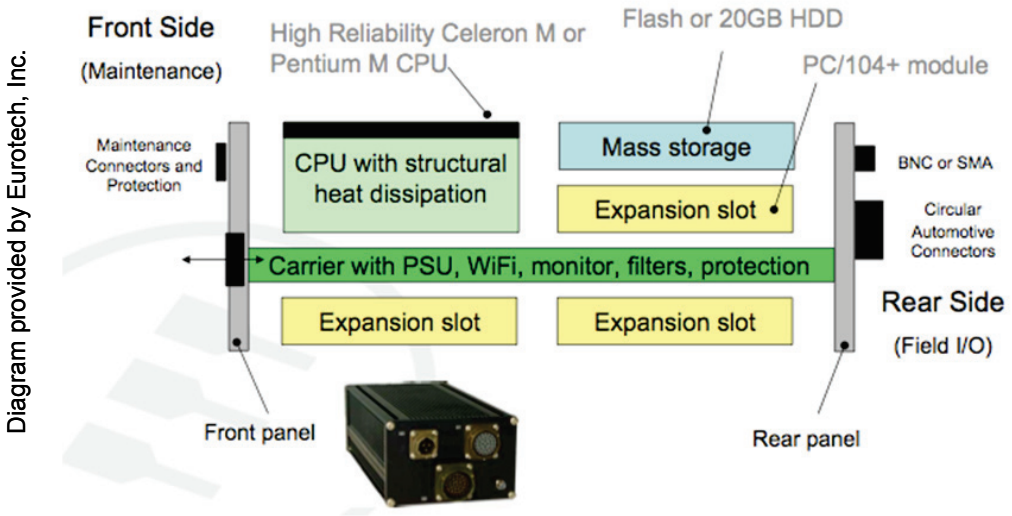


Figure 4-7 DuraCOR Unit Physical Architecture

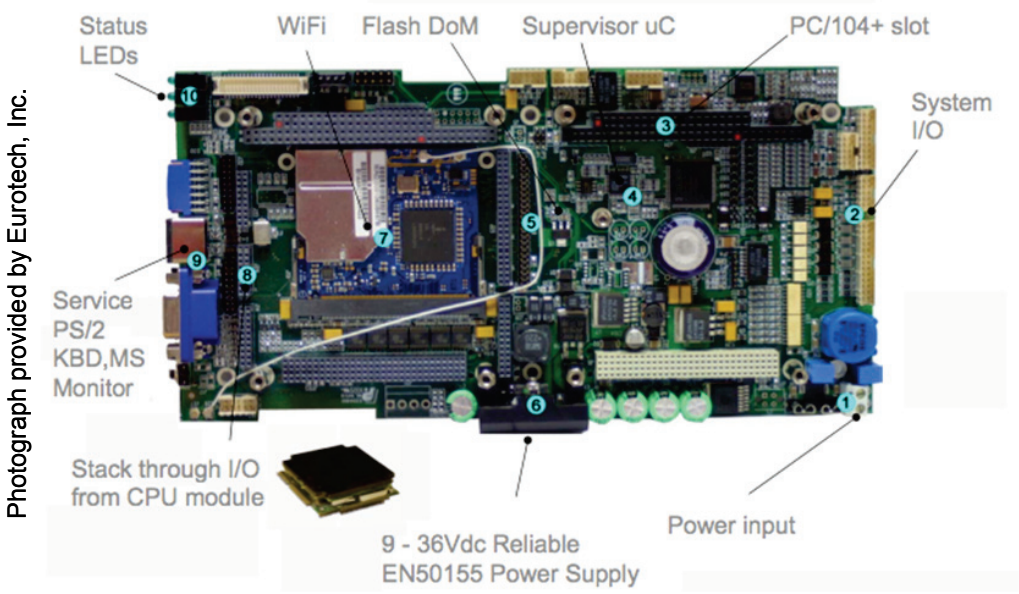


Figure 4-8 DuraCOR Unit Motherboard

4.5.1.2 Wind River Linux Operating System

The operating system used on the EuroTech DuraCOR unit is the Wind River distribution of the Linux OS. Wind River's Linux platform is a fully tested and validated distribution based on Linux 2.6 kernel technology.

All components of the platform, including the kernel, integrated patches and packages, and supported hardware architectures and boards, have been exhaustively tested and validated by Wind River. Some of the key benefits achieved by the OBE team, from standardizing on Wind River's Linux platform, include:

- **Saved time and expenses** - Reduced cost and schedule by eliminating the burden of building, supporting and maintaining a custom VIIC/OBE Linux distribution.

- **Focused effort** - Enabled the limited OBE team resources to focus on the essential and critical competitive differentiators, rather than on platform infrastructure.
- **Access to industry leading expertise** - Wind River's, commercially accepted, cross-build system and layered development methodology.

The Wind River Linux platform licensed for the OBE team also included Wind River's Workbench Development Suite. This Eclipse-based suite offered full capability across the development process in a single integrated environment. This suite came complete with integrated tools for debugging, code analysis and testing.

Wind River's Linux distribution and the associated bootloader and Board Support Package (BSP) for the DuraCOR hardware was made available to the OBE team members.

4.5.1.3 DSRC Radio Implementation

As described in Section 4.3, the DSRC/WAVE system is based on IEEE 802.11p and IEEE 1609 standards. The OBE DSRC Radio is implemented as a hybrid hardware and software system as illustrated in Figure 4-9.

The physical layer and the supporting IEEE 802.11p protocols are implemented using a commercial WiFi radio packaged on a Mini-PCI card. This card contains firmware that was modified to conform to the IEEE 802.11p standard. The upper layers of the DSRC protocol defined in IEEE 1609, known as WAVE protocols, are implemented in software running within the OBE software system.

The upper and lower layers of the radio subsystem are managed by a software element known as the WAVE Management Entity (WME). This forms what is known as the "management plane" of the radio, while the layers that operate on the messages themselves are called the "data plane."

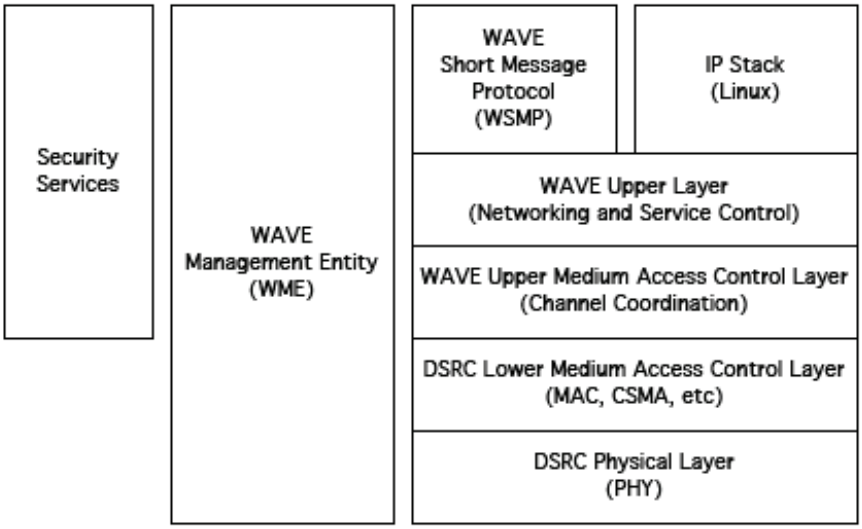


Figure 4-9 DSRC/WAVE Radio POC Architecture

DSRC Layers

The DSRC Radio physical layer and lower Medium Access Control (MAC) layer are responsible for physically generating and receiving the RF signals and for controlling the basic operations associated with sending and receiving these signals. The requirements for this operation are specified in the IEEE 802.11p standard which defines DSRC Radio.

The physical and Lower MAC layer are implemented using an Atheros Radio subsystem implemented on a Mini-PCI card. The base radio card is designed to support the IEEE 802.11a WiFi standard that operates at a slightly lower frequency band, and operates using slightly different protocols. The basic IEEE 802.11a operation has not changed, but key elements have been added to allow the system to operate effectively in the high speed vehicle environment where it is not possible (or necessary) to set up a full blown network prior to communicating. The changes to the protocol stack are summarized in Figure 4-10.

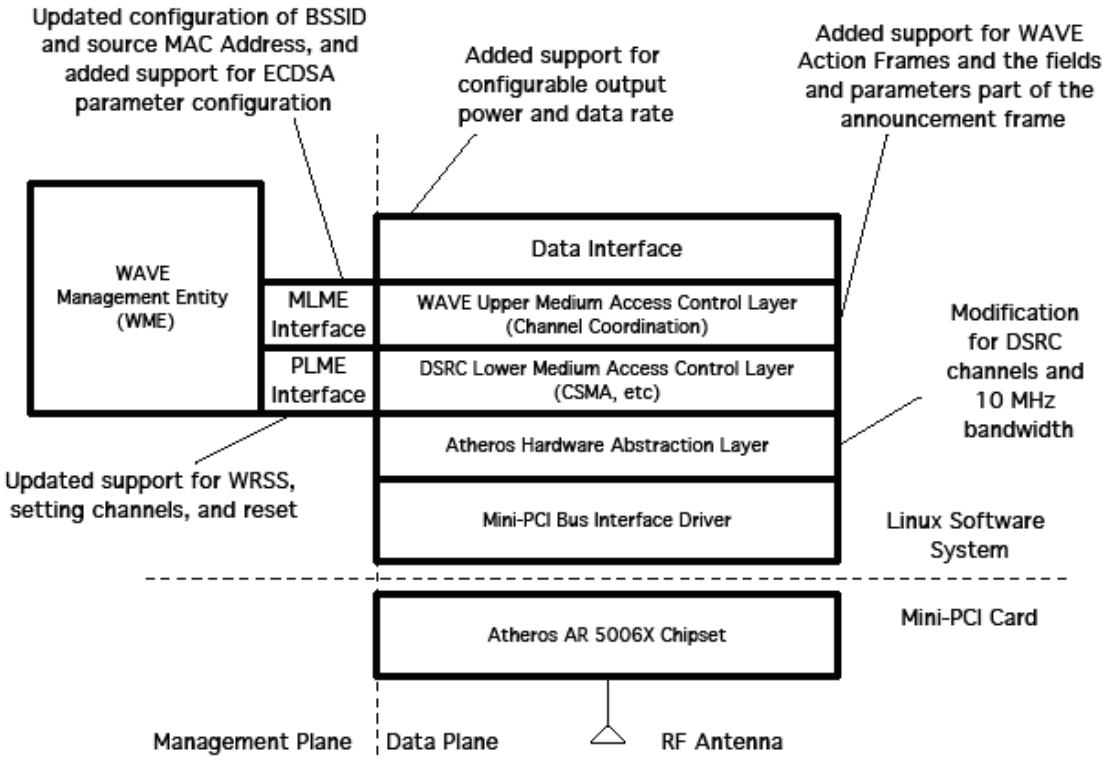


Figure 4-10 DSRC Radio POC Architecture

The Atheros Mini-PCI radio card is shown in Figure 4-11.

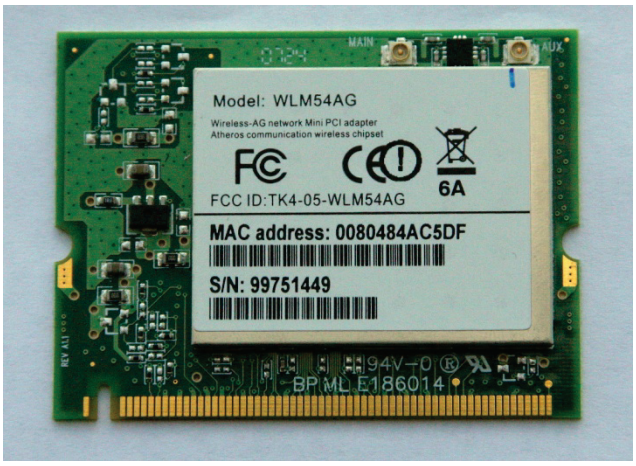


Figure 4-11 DSRC Radio Mini-PCI Card

WAVE Layers

The upper layers of the DSRC/WAVE Radio implement the WAVE part of the overall protocol as described in the IEEE 1609 standards. This includes the overall service management logic that determines how a WAVE radio decides what services from which providers to use, the WSMP, and logic to manage the seven different DSRC channels defined for use in the U.S. by the FCC.

The WAVE layers support two different types of message elements, conventional IP packets, and WSMs. Illustrated in Figure 4-9, this complicates the upper layers since normal WiFi radios simply pass incoming packets to the IP stack provided by the OS. In the case of IP communications, the VII implementation is not particularly different from this, but in the case of WSMs, there is no native function to route packets to the intended applications. As a result, the upper layer WAVE implementation also provides an API that allows the user applications to register as both a User or Provider, for support service and channel decisions (See below), and to send and receive WSMs. This is illustrated in Figure 4-12.

As described in Section 4.3, the DSRC/WAVE protocol uses a CCH and SCH interval concept (See Figure 4-2). By requiring that any radio monitor the CCH during the CCH interval, the system assures that any given radio is tuned to the right channel at the right time to hear important messages regarding safety and service announcements. A DSRC/WAVE radio may operate as either a User or a Provider. While both a User and Provider operate the same from a message communications perspective, a Provider is also able to issue a special type of message known as a WSA. The WSA is broadcast on the CCH during the CCH interval to announce or advertise the services that the provider is offering, and indicates which DSRC channels these services may be found. In general, OBEs operate in the User mode, although this is not a requirement.

The WME is responsible for receiving any WSAs and for deciding which channel (if any) to use during the SCH interval. This is done on the basis of what services the OBE applications have registered for, what services have been advertised by RSEs and the relative priority of those services. The WME also interacts with the Security Services (See Section 4.5.3.4) to verify the digital signatures of any received WSAs and in Provider mode, to digitally sign any outgoing WSAs.

The Channel Coordination Layer (CCL) is responsible for controlling the channel used by the radio, and for routing messages into queues for the channels on which they are intended to be

sent. This is a key function since the radio must be synchronized to all other radios so that the CCH and SCH intervals line up, and messages intended for a specific service must be held until the radio is tuned to that channel. The channel switching operation is synchronized to the Pulse per Second provided by the GPS receiver.

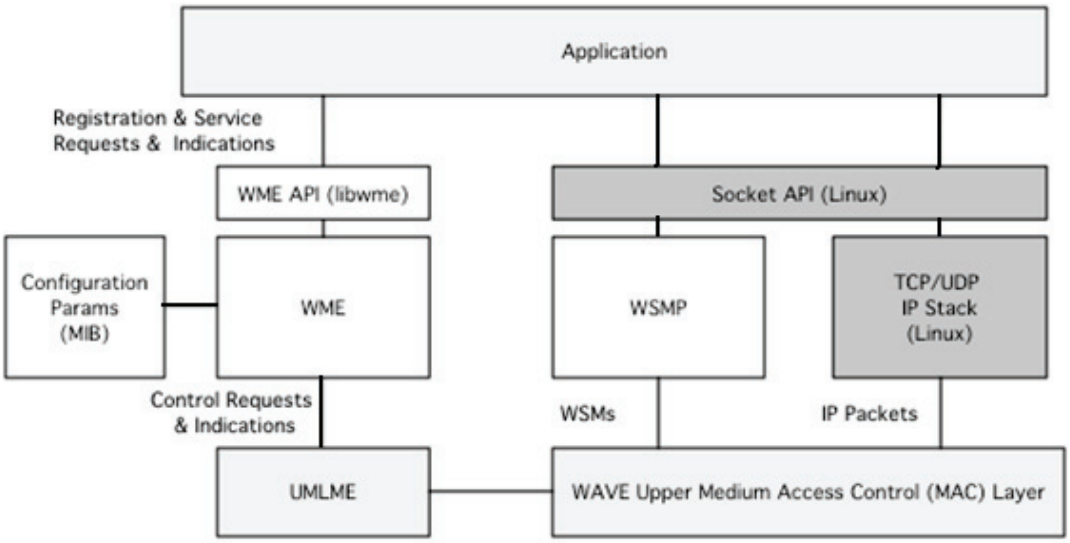


Figure 4-12 WAVE Upper Layer Software POC Architecture

4.5.1.4 Security Accelerator

The IEEE P1609.2 Security Protocol requires the use of ECC. This approach has a significant advantage in that it results in substantially smaller keys for a given level of security compared to other systems (e.g. RSA keys). However, being an asymmetric operation, and being relatively new, the software based solutions for encryption and decryption are slow and non optimal.

A typical VII vehicle is expected to be in range of about 223 other vehicles under worst case load situations (Typical eight-lane freeway, vehicles placed in all lanes at 2 m spacing, each vehicle 5 m long; range of 100 m forward and behind; ≈28 vehicles per lane; 224 total vehicles within 100 m of each other). If every message is signed, and every OBE sends Heartbeat messages at 100 ms intervals, each OBE will be required to encrypt 10 messages per second, and decrypt 2230 messages per second (10 per second from each of the other 223 vehicles). As a result, the worst-case security processing load is about 2240 operations per second.

It was decided that this represented a significant processing load on the OBE and might impact other software functions. As a result, the OBE was configured to provide a Mini-PCI slot that was used to support a hardware accelerator specifically designed to perform ECC operations.

The Crypto accelerator card (also known as the HPSAM) is shown in Figure 4-13. This card contains two special purpose chips. One runs the Peripheral Component Interconnect (PCI) bus interface, and the other is a high speed Field Programmable Gate Array (FPGA) that executes the ECC functions. A special software driver resident on the OBE provides a software interface that

allows the OBE Security Services to pass byte fields for encryption and decryption to the accelerator card.

The HPSAM was specified to support up to 2500 ECC operations per second.

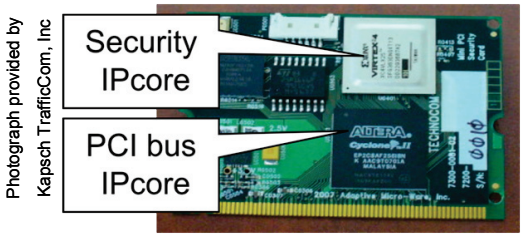


Figure 4-13 HPSAM Security Accelerator Card

4.5.2 POC OBE Software Architecture

As shown in Figure 4-14, the OBE uses shared services architecture. This means that key services expected to be used by most applications are provided as resources in the OBE. Any application needing these resources can then make use of them through simple software interfaces. Since many VII applications involve similar kinds of data and operations, the shared services approach avoids the need to implement these functions within each application.

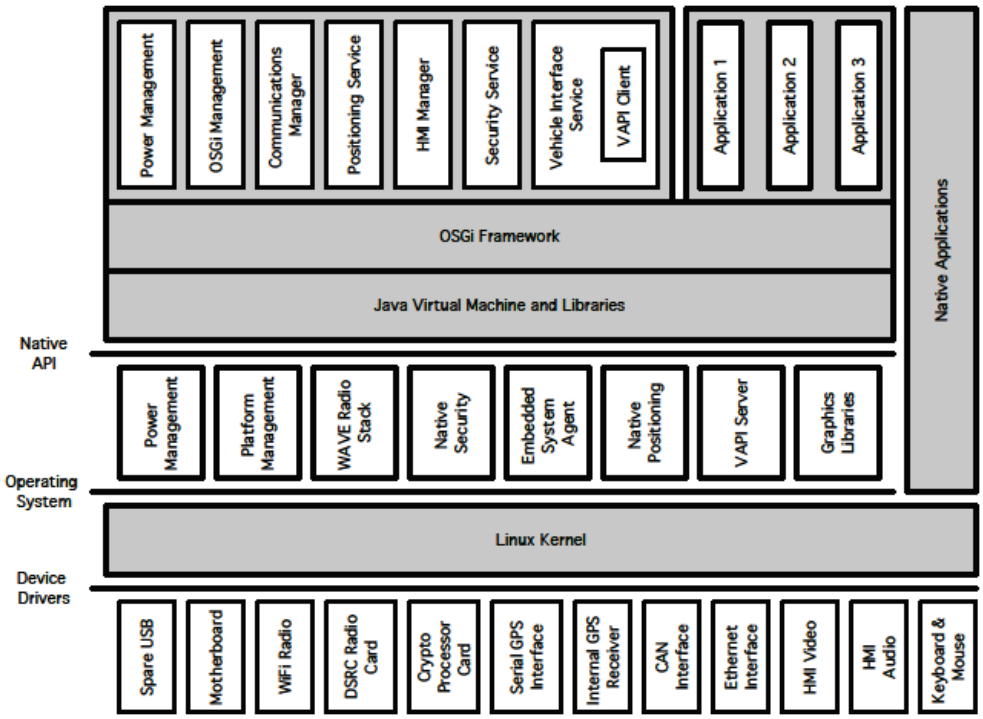


Figure 4-14 OBE POC Software Architecture

The most important OBE services are described in more detail in the following sections.

4.5.3 OBE Software Services

A description of the use of the Open Services Gateway Initiative (OSGi) framework and the description of the major OBE services are provided in the following sections.

4.5.3.1 Open Services Gateway Initiative Framework

The OSGi Service Platform is an open Java-based component framework that allows the rapid and safe installation and operation of services and applications (known as "bundles" in the OSGi specification). The OSGi Service Platform provides a service oriented architecture that allows applications to dynamically discover and use services provided by other applications running in the same environment. This service oriented architecture allows OSGi applications to be much smaller than conventional applications because they do not need to implement all of the various services they might otherwise need (e.g. positioning, security, etc).

Starting, stopping and updating bundles can be performed without the need to restart the system. The OSGi architecture allows the operator to control the Service Platform in fine detail by using a model where operators can ensure that their required policies are implemented.

As shown in Figure 4-14, the applications and most higher level services were implemented as OSGi bundles. This allowed very rapid re-configuration of the OBE using different applications, and also simplified software development and system integration by making installation of new versions of these higher level software bundles very fast and simple.

Although not tested during the POC program, the OSGi Framework also allows remote installation and control of bundles. While this was a key, longer term element of the rationale to use this system, it is not necessarily an essential part of the deployment architecture. Ideally, in the longer term, use of this framework could allow the OBEs to be remotely managed and thereby acquire new capability without requiring the vehicle to be brought in for servicing. Adoption of this approach would be at the discretion of the automakers.

4.5.3.2 Network Services Enablers Subsystem

The Network Service Enablers subsystem simplifies the communications process between OBE applications and network side services.

The VII system supports a variety of different modes of communications tailored to specific types of system functions. In addition, the DSRC system including security can involve rather complex processes from an OBE application perspective, and because the vehicles are moving through the network with only intermittent connectivity, there is no way for a network service to know how to get a message to a particular vehicle. Also, the OBE is designed to support a variety of simultaneous applications, and these applications need to share the communications resources. Since the applications are all developed by separate organizations, it was decided that the OBE should provide a common mechanism for sharing the communications resources. To do otherwise would have likely resulted in a variety of conflicting approaches developed by the independent application teams.

The Network Service Enablers subsystem effectively creates a single interface for all applications to use to access services from the DSRC Radio and to use the DSRC security functions. It also supports an Internet style back-end system that allows the addition, modification and combining

of services at a service provider without requiring changes at the OBE. This is important as a scalability feature. As described earlier, this Service Oriented Architecture (SOA) is well known in the internet, where connectivity is stable, but it was not known if the same approach would be feasible in the intermittently connected concept of the VII.

The POC Network Services Enablers architecture is shown in Figure 4-15.

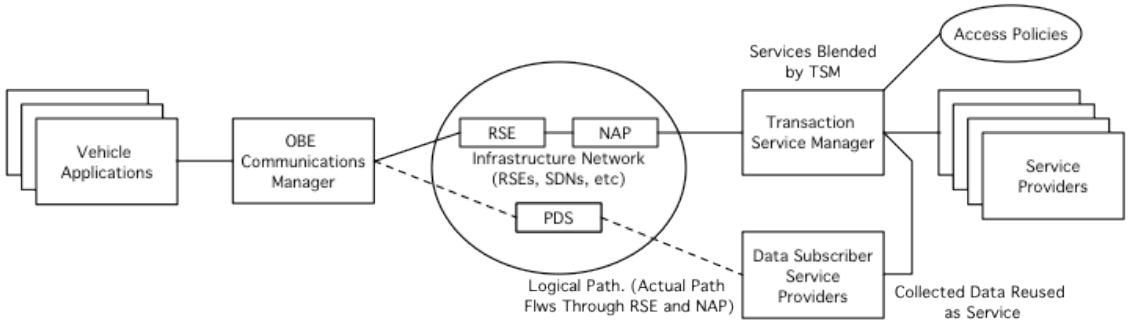


Figure 4-15 POC Network Services Enablers Architecture

The Network Services Enablers subsystem is composed of two primary elements. In the OBE, a "middleware" system known as the Communications Manager provides message routing and service/message registration and security functions for the OBE applications. In an analogous manner, at the other end of the system, outside the access gateway, the Transaction Service Manager (TSM) aggregates services from different sources into a single suite of services. The Communications Manager and TSM also work together to bridge between service sessions as the OBE-equipped vehicle moves out of coverage for one RSE and then re-connects some time later at another RSE. This "hand off" between RSEs is a new, key feature of the VII system that has not been done before in the context of Internet style services, and it substantially reduces the impact of the intermittent connectivity coverage characteristic of the VII POC architecture.

4.5.3.2.1 OBE Communications Manager

The OBE Communications Manager (OCM) facilitates the interaction between in-vehicle applications and external services by providing means to ensure the transparency and appropriate security of communication from an application perspective. The goal of the Communications Manager is to abstract applications from any details related to communications and communication security, with the understanding that the more functionality is encapsulated inside the service, the easier is the task of writing applications. By isolating the communications process from the applications, they are shielded from any changes in communications protocols and infrastructure.

As shown in Figure 4-16, the Communications Manager is composed of three primary elements: the Application Manager, the Message Manager and the Transport Channel. The Communications Manager has two operational modes from an application perspective. First, it provides an interface through which applications can register for services and the messages associated with those services. This administrative process includes establishing security credentials, and communications preferences, and it is typically performed once during the application start up. Second, during regular operation, the Communications Manager interacts with the DSRC Radio and notified applications when any of their registered services are available (i.e. they have been advertised by an RSE), and passes messages between the applications and the

DSRC Radio. One key aspect of the Communications Manager is that it is multi-threaded; it is thus capable of supporting multiple applications. It also effectively serves as a multiplexer, by collecting messages to be sent from many applications and distributing received messages to those applications.

The detailed operation of the elements is described in the following section.

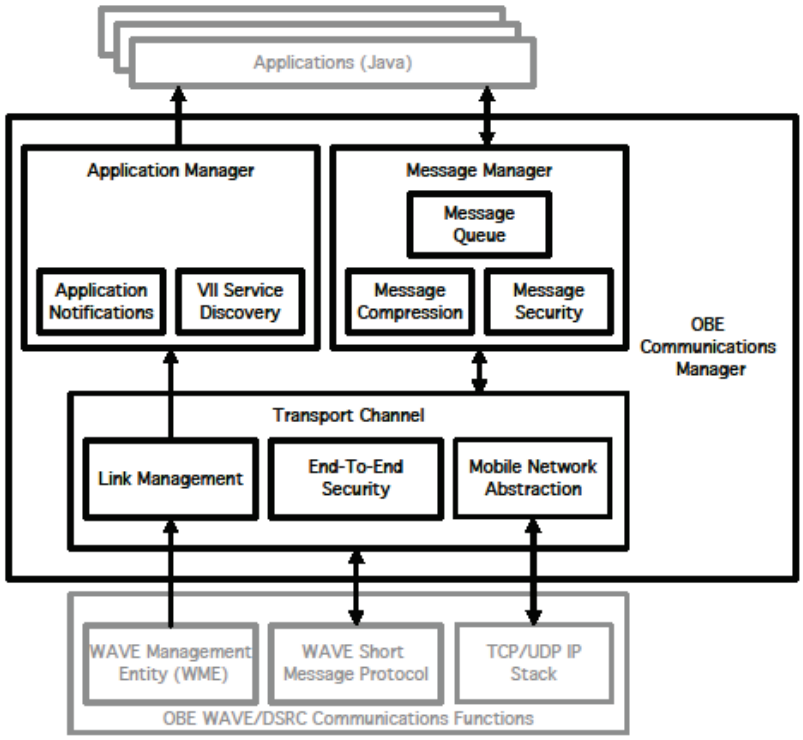


Figure 4-16 OCM

Application Manager

The Application Manager provides a set of interfaces to the OBE applications to set up and manage services. At start-up, the OBE applications register with the Application Manager and provide both security information and the PSIDs for the services they plan to use. The Application Manager informs the Security Services about the needs of the application, and sets up the DSRC Radio to respond when those services are available.

The Service Discovery component of the Application Manager is designed to facilitate dynamic interaction between various external services and in-vehicle applications. The current discovery mechanism is based on a PSID number that identifies a type of service. This type of service is quite generic and does not provide specific service-related details to applications. A second number, the Provider Service Context (PSC), is a vendor-specific identifier used to denote specific protocols, service versions, or other key service-related information.

As shown in Figure 4-17, the Application Manager function within the Communications Manager provides an interface that allows applications to query and discover the required services in a generic way. Having an intermediate layer provides a uniform way for the discovery of services that simplifies the design of in-vehicle applications.

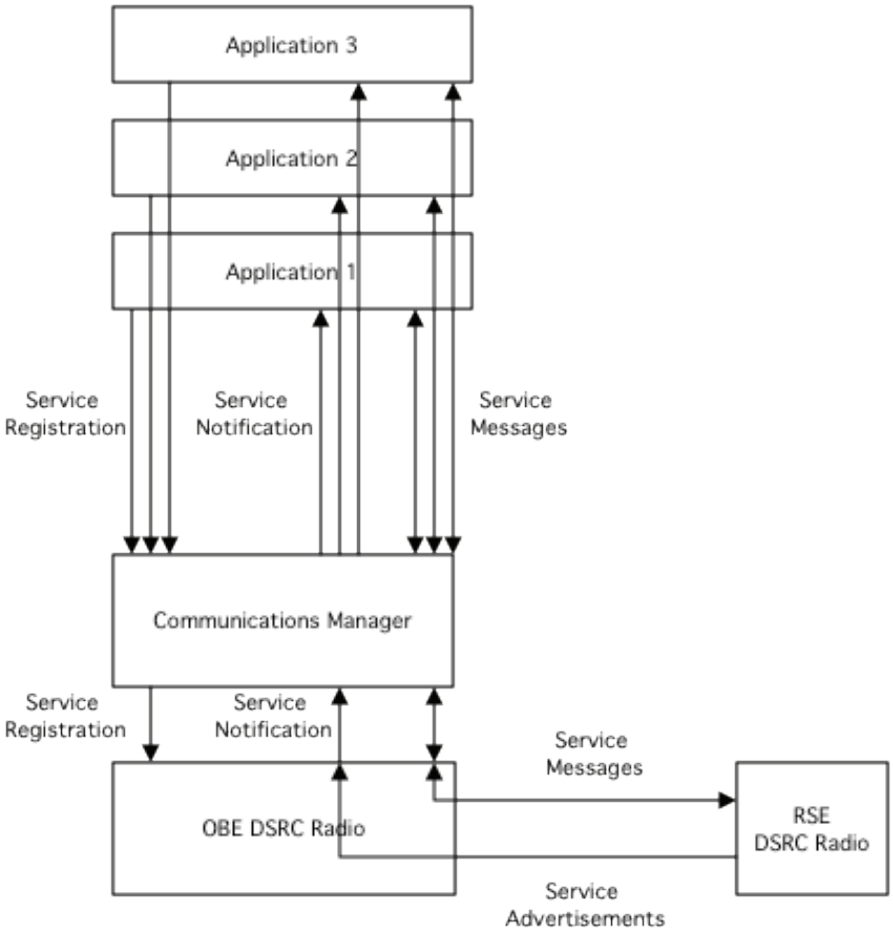


Figure 4-17 Communications Manager Service Discovery Scheme

Message Manager

The Message Manager functions within the Communications Manager and handles incoming and outgoing message traffic for applications using the WSMP through a socket interface similar to that provided by the Linux OS for IP packets. On the outbound side, OBE applications provide messages to the Message Manager via the socket interface. The Message Manager then processes the message according to the security operations defined by that application at registration, and then passes the message to the Transport Channel for transmission (See next section). On the inbound side, the Message Manager receives messages from the Transport Channel, passes them to the security libraries for verification and/or decryption, and then delivers them to the socket assigned to that message type. The socket assignments are based on the PSID. Each WSM includes the PSID and this is used to bind the messages to the application via the socket assignment.

The Message Manager is able to handle messages in parallel since the flow of messages may be such that messages are received before prior messages have been fully processed. Lastly, the Message Manager avoids delivery of duplicate incoming messages to registered OBE applications by saving the message ID for received messages and comparing newly arrived messages with the message ID cached by the Communications Manager. This feature can be selected by an application during initial registration.

Transport Channel
The Transport Channel function within the Communications Manager interfaces to the lower layers of the system, specifically the various DSRC Radio interfaces and security libraries.

In its simplest operations, the Transport Channel receives messages from OBE applications via the Message Manager, and provides them to the appropriate DSRC Radio interface for transmission, and performs the converse operation for messages received from the DSRC Radio. This operation is exclusively performed on WSMs since IP-based message traffic uses the IP stack provided in the Linux OS.

Under some IP exchanges; however, the Transport Channel is responsible for setting up a special type of IP session. Using the Host Identity Protocol (HIP), the Transport Channel exchanges a set of handshake messages with the TSM. These exchanges result in a secure and anonymous session identity known only to the Transport Channel and the TSM. Using this session, IP data exchanges take place as usual using the Linux IP stack. However, if the DSRC link is lost, for example, because the OBE host vehicle drives away from the RSE, the session state is maintained at the TSM and in the Transport Channel for a period of time. If the OBE encounters another RSE before the session times out (the timeout was about 10 minutes in the POC), then the Transport Channel and the TSM re-establish the session, and agree on a new secure session identifier. This allows the system to pick up the data exchange where it left off only using the packet routing for the new RSE (which is the new network location for the OBE). The result of this approach is that the OBE can carry out long data transfers or extended transactions with the TSM even though the OBE is coming to and leaving many different RSEs. This hand-off mechanism effectively creates a semi-seamless network usable for long transaction even though the network does not provide geographically-continuous connectivity to the OBE.

The Transport Channel also interacts with the security libraries (See Section 4.5.3.4) to set up secure links for IP-based applications. This means that the application simply needs to provide security information at registration instructing the Transport Channel that a secure link is needed, and the Transport Channel takes care of the rest of the process. Through this architecture, the applications (and application developers) do not need to include extensive code to interact with the security system, and in fact can successfully use the security system with little or no knowledge of the details of its operation.

4.5.3.2.2 Transaction Service Manger
The TSM is the server-side or off-board portion of the POC Network Services Enablers architecture. It acts as an intermediary for network-based transaction services communicating with applications on the OBE. An intermediary is necessary to act as an access control point as well as a queuing station to mediate asynchronous communication. The TSM also serves as the integration point for cooperative services defining an extensible service framework that may prove useful well beyond the scope of the POC.

The TSM is shown architecturally in Figure 4-18.

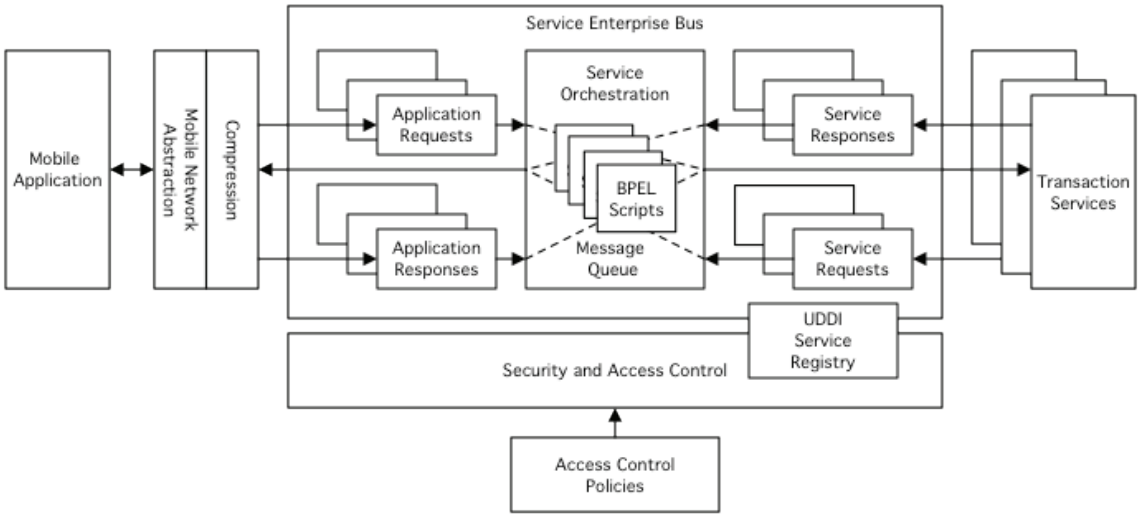


Figure 4-18 Transaction Services Manager

The TSM provides Mobile Network Abstraction, Service Orchestration, Message Queuing, and Security Interaction capabilities to ease the development and integration of traditional web services.

Mobile Network Abstraction

Mobile Network Abstraction isolates the problems of dynamic routing and network session discontinuity from the transaction services. All transactional communications between the VII system and the network services are facilitated by Mobile Network Abstraction.

Mobile Network Abstraction provides two related services. First, it works with the Communications Manager in the OBE to bridge service gaps of reasonable size, and second, it maintains the ability to route packets despite changes in the vehicle's location in the RSE network with consequential changes in the OBE's IP address. This overcomes the key limitations of the IP relative to mobile operation. These limitations arise because during the initial design of the IP, almost all machines capable of supporting IP were physically large and therefore stationary. There was no obvious need to design a system capable of supporting mobile connections, and the problems inherent in such an approach would realistically serve no purpose while introducing considerable complexity and processing requirements to their designs.

In this context, the standards organizations made one decision in particular that made supporting mobile systems rather difficult: the IP address concept overloaded the notions of location and identity, in that, "who am I" and "where am I" were both features of the IP address as defined in Internet Protocol Version 4 (IPv4). Unfortunately, in mobile systems, it is an absolute necessity to separate identity from location and routing information. Mobile Network Abstraction implements the server-side of the HIP described earlier as part of the Communications Manager Transport Channel.

Service Orchestration

Service Orchestration is the concept that multiple network-side services can participate in a single OBE application request. A request can be processed by or affect one or more services. A purchase request is one example. A single request to purchase an item may interact with an inventory service, a payment service and a shipping service. In fact, the inventory service could

be interacted with twice; first, to verify availability of an item and second, after payment is validated to update the number of those items. This approach is illustrated in Figure 4-19.

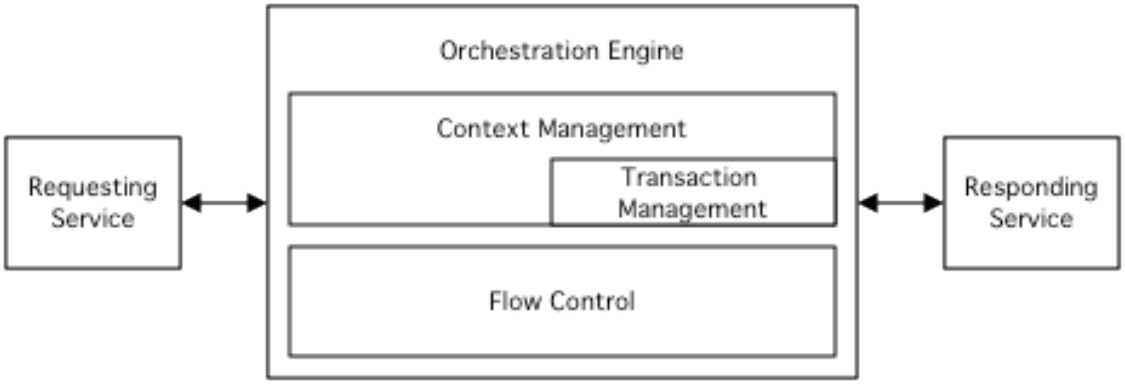


Figure 4-19 Service Orchestration

Another important aspect of orchestration is reuse. Not only can customers of one service provider make use of those services, but different service providers (or service provider brands) can orchestrate those same services in other ways.

In the POC implementation, this orchestration is managed by a subsystem known as the Enterprise Service Bus (ESB). The ESB provides two main functions. First, it manages flow control that defines which services are called and when. Second, it provides context management, so that services participating in a single request participate within the same context. Exceptions or faults that occur at any individual service are reported to and managed by the context. A specialized case of this context coordination is transaction management. Transactions across web services are more complicated than transactions within a single database, for example. Multiple systems may be involved crossing thread and connection boundaries.

Message Queuing

The TSM message queuing service manages the exchange of data between service consumers (OBE applications) and service providers (Network User/Providers). The requests and resulting responses are transformed and controlled by a series of workflows residing in the service orchestration layer. Additionally, service access is arbitrated by an authorization mechanism controlled by policies that are expected to be determined in the future. Message Queuing is a mechanism internal to the TSM which is fully isolated from services and applications. In accordance with the design goal of using open standards, the message queue implementation uses Java Message Service (JMS). Messaging with external entities is conducted using Simple Object Access Protocol (SOAP) over Hypertext Transfer Protocol (HTTP).

4.5.3.3 Human Machine Interface Manager

The HMI Manager arbitrates HMI resources amongst applications and provides a toolbox of graphical components to support the user interface for the applications. The HMI Manager is capable of presenting both visual and audible information, including warning symbols and signals.

By defining a common HMI service for the OBE, the HMI is uniform across applications and is usable in a wide variety of vehicles. In addition, all applications are isolated from the HMI design, so the application developers need not be concerned with the details of the specific HMI implementation. This approach simplified the POC development and testing, but the HMI is not expected to be standardized in the actual deployment environment.

POC HMI Manager Architecture and Operation

The HMI Manager architecture is shown in Figure 4-20. The HMI Manager is constructed around a set of core graphical operations and structures known as widgets and templates. These structures define basic screen layouts into which application-specific content can be placed. These structures interact directly with the basic HMI drivers and tools provided by Java.

The HMI Manager provides a plug-in interface that allows each application to define, in a single file, its unique graphical elements such as icons and labels, and application-specific text elements.

In operation, an application interfaces with the HMI Manager via the HMI Manager API. This provides a set of tools that the application can use to write context-specific information into the template, and to specify content from the plug-in to be used on a specific screen. By separating the changeable context-specific content from the more stable graphical elements, the application interface and graphical operations are much simpler. Essentially, the application specifies the content and the HMI Manager constructs the view.

The HMI Manager also arbitrates the use of the HMI resources between applications as described in more detail below.

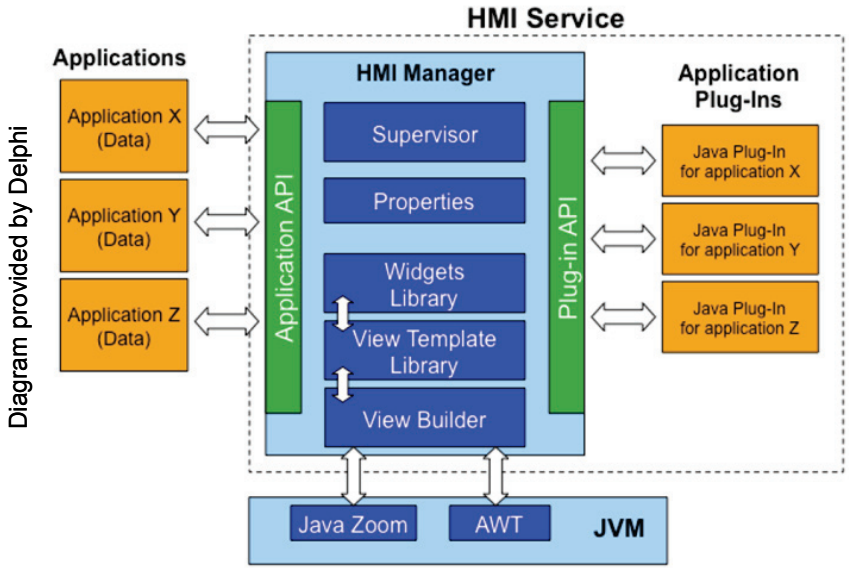


Figure 4-20 POC HMI Manager Architecture

Figure 4-21 provides a brief look at the operation of the HMI Manager. In this example, the "View 12" template is being used. The application specifies, via the API, that it wants to display a sign type ("SignID= 0x24"), and specifies two textual variables that relate to that sign type. The HMI Manager Supervisor goes to the application's plug-in and retrieves the content of sign type "0x24", which consists of a graphical icon showing a curve warning symbol. Also retrieved are the text strings "Curve Ahead" and "Recommended Speed 40 mph" and the template type to be used. In this case, the plug-in has combined variable data provided by the application ("Curve

Ahead" and "40") with fixed text strings associated with that sign type. The View Builder then compiles the view by placing the icon in the proper location of the template and writing the text strings into the defined locations of the template. The result is the display shown in Figure 4-21. Note, that if the curve had been the other direction, or the recommended speed had been different, the application would have provided this information, and the resulting sign, based on the same processes and template, would be changed to conform to the specific details provided by the application.

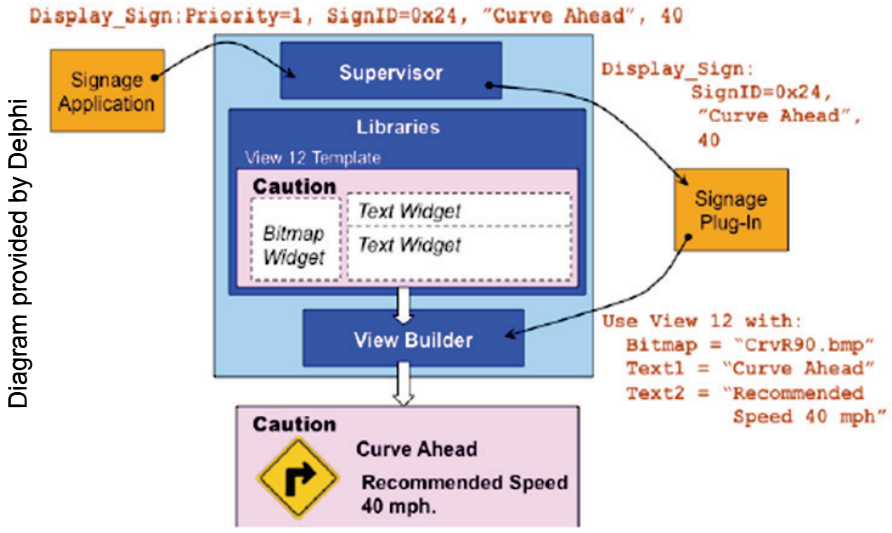


Figure 4-21 HMI Manager Example

Prioritization Scheme for Displaying Messages

As mentioned, the HMI Manager also provides arbitration of the HMI resources. This is necessary because the OBE supports many simultaneous applications, and there is only one HMI. As a result, the HMI Manager must decide, based on priority, which application gets to use the display (or audio) at which time.

HMI arbitration is performed using a prioritization scheme developed by the International Standards Organization (ISO), ISO 16951 uses a Priority Index calculation. The Priority Index approach uses measures of criticality, urgency, user context and scenario to compute a priority of any given message. This priority value is compared to the priority of other applications seeking to use the HMI, and the HMI resource is awarded to the application with the highest priority. This approach is shown conceptually in Figure 4-22.

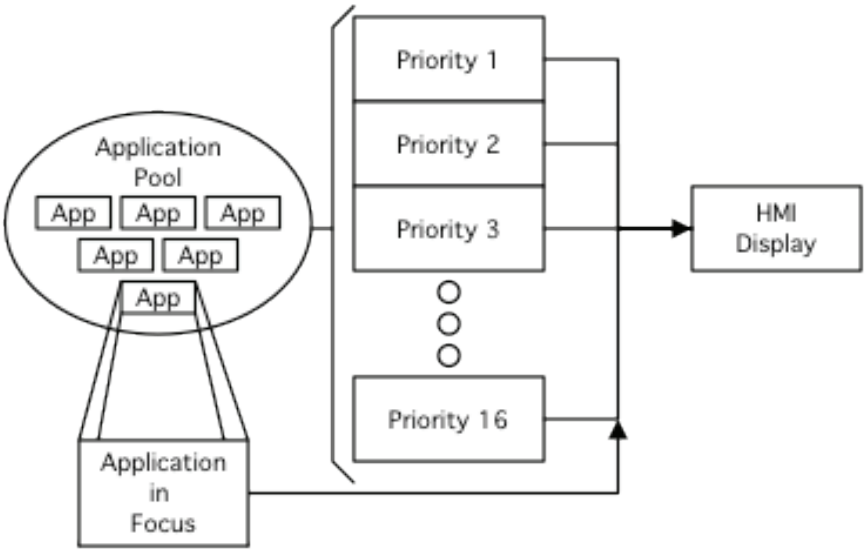


Figure 4-22 HMI Display Prioritization

Priorities are defined in a special HMI Priorities properties file, which is accessible to all applications. At startup, each application reads this central property file to obtain the application-specific priorities for each screen the application will eventually display.

For each screen, the HMI Priorities file contains one priority value for each of the following:

- **User Initiated** -- This value (0 or 1) will be set by the application and defines whether the screen was requested/ initiated by the application. (For example, if the user selects "Billing" in the Parking Payment Application, the follow-up screen is user-initiated).
- **Criticality** -- Severity of the impact, of the most likely accident or malfunction that can occur when the message is not received or is ignored by the driver.
- **Urgency** -- Time within which driver action or decision has to be taken if the benefit intended by the system is to be derived from the message.
- **Scenario** -- Explanation of the driving context and situation (in terms of the road environment and the traffic condition in which the message is likely to be presented) for the message.

The Priority Index is calculated by the following formula:

Priority = Weight_1 * User Initiated + Weight_2 * Criticality + Weight_3 * Urgency + Weight_4 * Relevancy

The relative weights are defined by a configuration file and may be adjusted to fine tune the way priorities are determined. When multiple HMI requests are made, the HMI Manager computes the Priority Indexes for each and based on their relative values, awards HMI access to the application that has the highest Priority Index. This approach assures that the prioritization is context-specific rather than simply giving a particular application higher priority even when the use of the HMI is not high priority. This method was adequate for POC but is not likely to be the method of choice for final deployment.

Advisory Message Template

Figure 4-23 illustrates the Advisory Message template. This image represents a general version of a road sign for a class of signs. For example, an orange diamond represents all signs describing Work Zones and Road Work. "Text Line 1" displays the major "TITLE" of the road sign. "Text Lines 2-5" are utilized to display specific context of the sign. In another example, the image may represent all R2 Speed Regulation Signs, "Text Line 1" displays "SPEED LIMIT," and the following text lines will convey the actual speed limit of "50 MPH" (See Figures 4-24 and 4-25).

The screen areas labeled "btn" on the right hand side of the screen are available to allow the user to select which advisory screen they wish to display, as defined by an appropriate category icon. This is useful when multiple advisory signs have been received. The highest priority sign will always be displayed automatically, but the user may then choose to view a lower priority sign manually.

The screen area labeled "btn1" through "btn5" across the top of the screen are populated with button icons of active applications with HMI content. The user can shift the display between applications by pressing the appropriate buttons.

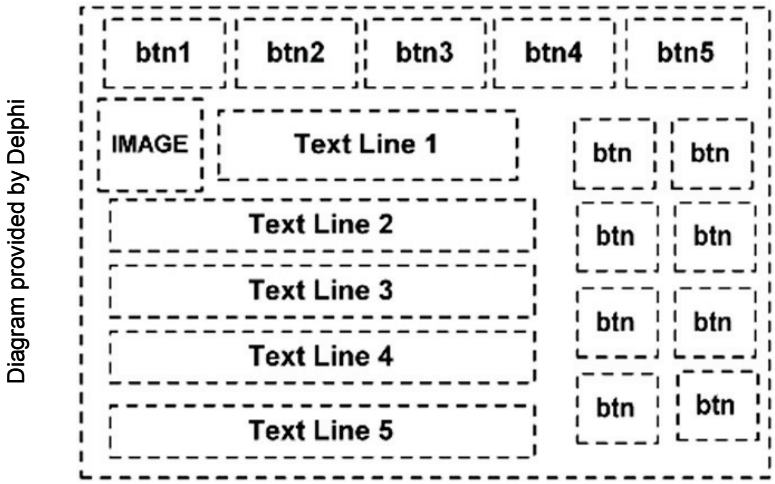


Figure 4-23 HMI Road Advisory Template

Examples of this template in use are shown below in Figure 4-24 and 4-25.

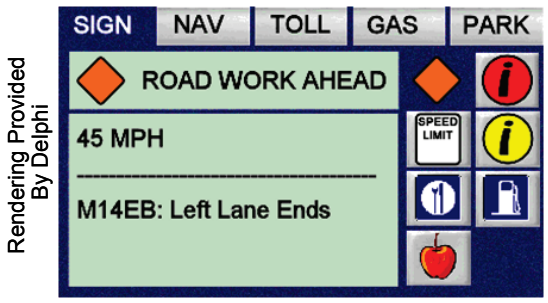


Figure 4-24 Road Work Advisory Example

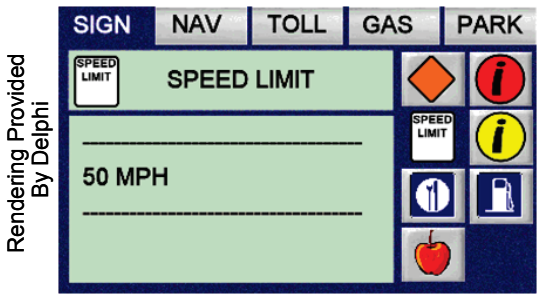


Figure 4-25 Speed Limit Advisory Example

Next Exit Services Template

Next Exit Services Template is shown in Figure 4-26. This template is slightly more complex than the road advisory template since the signs contain more graphical and textual information. Each individual sign is split into services to the left and right of the exit ramp. The two data columns shown in the template are used to place left or right side services as appropriate.

As with road advisories, the button icons across the top indicate the active application and allow the user to select the active application. The buttons on the right allow the user to select the category of advisory message to display.

Each Next Exit Services field is populated with a left or right arrow as appropriate, a small text field indicating the distance to the service and either an icon or text describing the service.

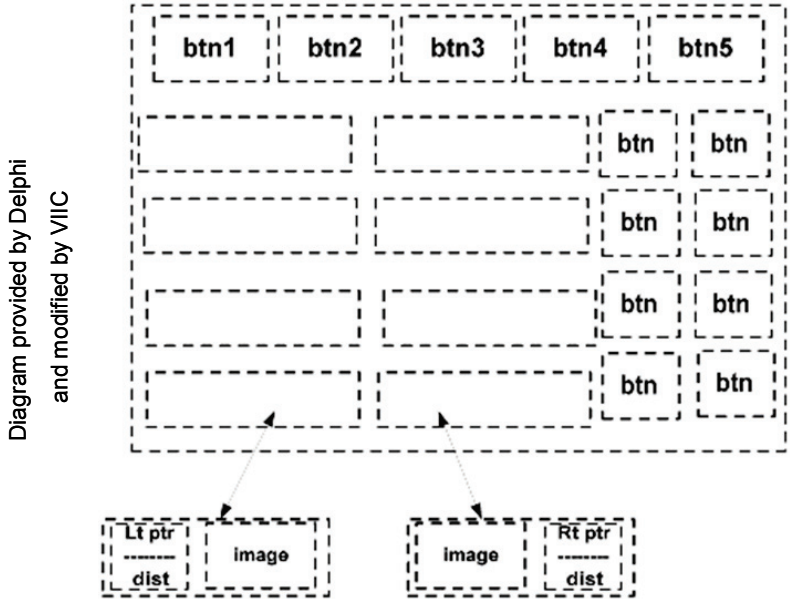


Figure 4-26 Next Exit Services Template

Figure 4-27 shows an example of a screen generated using this template.

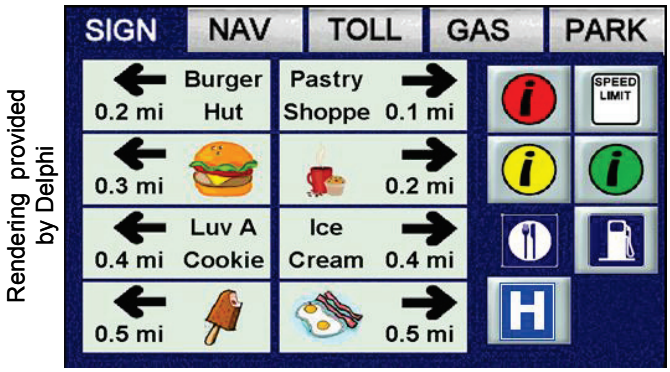


Figure 4-27 Next Exit Services Screen Example

Traveler Advisory Template

The Traveler Advisory Template is general in that it can be extended and adjusted to display a wide variety of different types of information. Two of several types of templates and typical example corresponding message displays are shown in Figures 4-28 to 4-31.

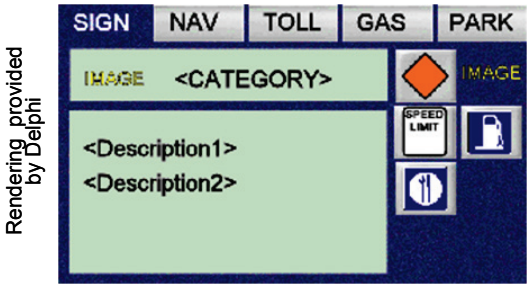


Figure 4-28 General T/A Template

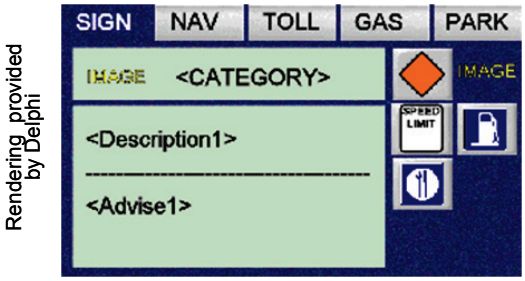


Figure 4-29 Driver Advice Template

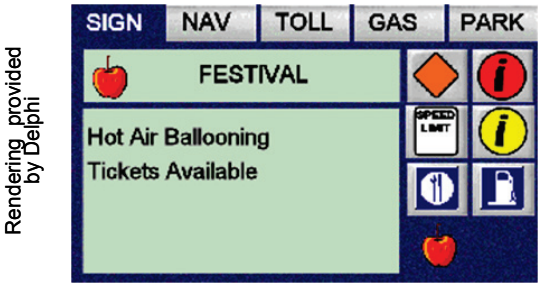


Figure 4-30 General T/A Example

Figure 4-31 Driver Advice Example

Off-Board Navigation Template

The Off-Board Navigation Application (OBNA) uses several different templates depending on the state of the application. For brevity, we have shown only a few examples of typical screens. The same template approach described previously, is used for this application. The user must have the ability to choose a destination via a scrollable list of destinations. Figures 4-32 and 4-33 show typical destination setting screens with up/down buttons on the right side to scroll between the pages.

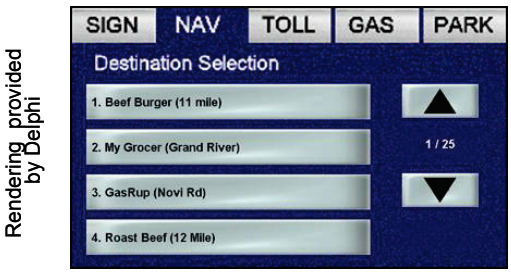


Figure 4-32 Destination Set Screen

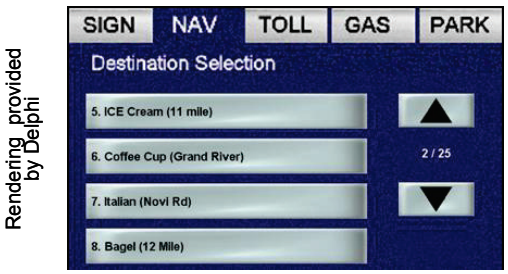


Figure 4-33 Destination Screen (Page 2)

Once the destination has been selected and the system has obtained driving directions, a turn list is provided to guide the driver's maneuvers to reach the destination. This is shown in Figure 4-34.

Figure 4-34 Off-Board Navigation Turn List Screen

The OBNA also provides a route overview screen that shows the entire route on a graphical map display as shown in Figure 4-35. It is useful to note that this screen also includes buttons/icons on the right side to access the turn list (See above) and to update or cancel the route.

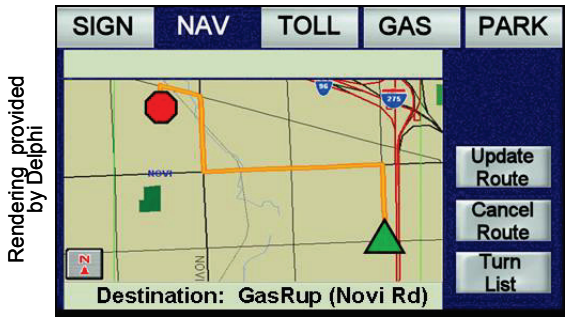


Figure 4-35 Route Overview Screen

Toll Payment Displays

The Toll Payment Application displays are very basic with only two general screens. Figure 4-36 shows the screen used to allow the driver to turn the Tolling Payment function on and off. Figure 4-37 shows the screen used to inform the driver when a toll has been paid. This is automatically displayed if the priority is high enough and a tolling payment transaction has just been carried out.

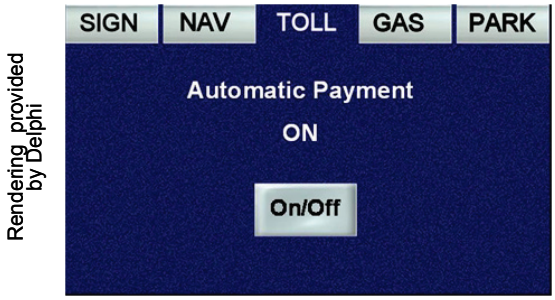


Figure 4-36 Toll Payment On/Off Screen

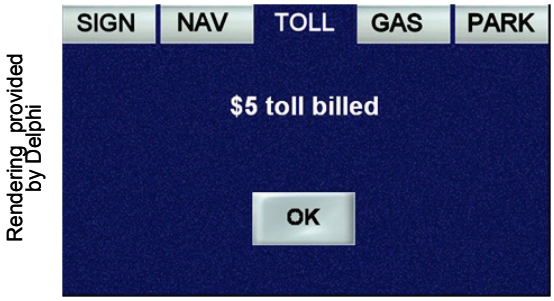


Figure 4-37 Toll Payment Info Screen

Parking Payment Displays

The Parking Payment system behaves similarly to the Toll Payment system, although the screens are slightly more involved. These are shown in Figures 4-38 to 4-40.

Figure 4-38 Parking Announcement Screen

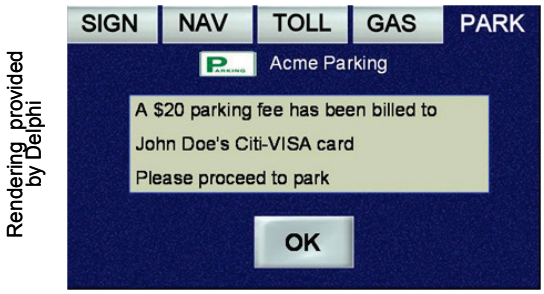


Figure 4-39 Parking Payment Screen

The last parking screen allows for configuration of billing options (i.e. changing the account/ credit card which is used to pay for the transaction).

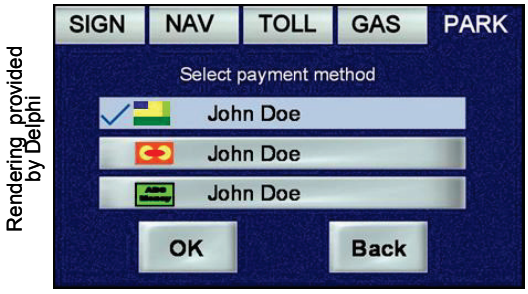


Figure 4-40 Parking Payment Billing Selection Screen

4.5.3.4 Security Services

The OBE contains two security elements: the certificate management subsystem, and the Security protocol implementation.

Certificate Management

As described previously, the certificate management subsystem has two elements. The first element interacts with the CA to request and process certificates. The second element controls access to and use of the certificates in the vehicle system.

When an application is added to the OBE, the OBE must verify that the application is legitimate, namely, that it has come from an approved vendor. OSGi supports the mechanism known as "code signing" to meet this requirement. Using code signing, a platform can be required to accept only certain applications. This provides individual OEMs with control over what applications their OBE will run, thus greatly improving system assurance. As part of the installation process, the application registers with the Security Services. The Security Services then generate an application and OBE-specific key pair, which can be used to request certificates for that specific instance of that application. This key pair, and all other cryptographic and security material for the application, is stored in an application security file on the OBE known as a WAVE Security Context (WSC).

Based on the WSC information, the CM then requests the appropriate security credentials from the CA. In the case of an anonymous application (one that will send anonymous public messages), the CM will interact with the Authorizing Authority and Certifying Authority as described in Section 4.11. In the case of an identified application (one that sends private messages), the CM would go to the CA responsible for providing identified certificates. The CA verifies that the OBE in question is entitled to run the appropriate application before issuing the certificate.

The CM continually checks the state of the certificates for each active application, and routinely requests replacement certificates when older certificates expire or are found to be revoked.

For each anonymous application, the CM maintains a pool of anonymous certificates provided randomly by the Certifying Authority. As the application sends messages, the CM randomly rotates the certificates in the pool, so that the same certificate is not used again and again.

The CM maintains its own security credentials as well. These are used to interact with the identified Certifying Authority (or Authorizing Authority) during the various certification operations.

In addition to the certificate management operations described above, the CM is also responsible for assuring that the OBE has not been tampered with in any way. The first line of defense in this area comes from the installation process. If any attempt is made to install an unauthorized application, it will fail the code signing check, and the installation will stop. In addition, the OBE will need to authenticate itself each time the system starts up. In this approach, the OBE is not trusted by the security system. Instead, the OBE must provide information attesting to its legitimacy, and when this information has been verified as correct, the security system will release the security credentials. By this approach, any other participant in the system that receives a signed anonymous message can be assured that the message originated from an OBE that was able to successfully prove its authenticity, and that no unauthorized software was installed. For example, in a simple implementation, the vehicle (OBE) might provide the Vehicle Identification

Number (VIN) and if the VIN matches that stored in the CM, then the CM will allow the system to operate. If the VIN has changed, (e.g. the system has been transported to a different vehicle), then the system will refuse to unseal the keys, and the applications will be unable to sign any messages. Due to the OEM unique nature of this approach, this system was not implemented in the POC. In future developments, it is expected that this type of verification process will be elaborated and reflected in a requirement for OBE validation.

OBE Security Protocol

Once the identified and anonymous certificates have been established in the OBE, and the OBE has established its validity to the CM subsystem, the OBE Security Protocol implementation may use them to sign, and encrypt outgoing messages in accordance with the IEEE P1609.2 DSRC Security Protocol. This same standard is used to verify, authenticate and decrypt incoming messages.

The mechanisms used in IEEE P1609.2 are based on general Public Key Infrastructure (PKI) security principles. The mechanics of this will not be described here.

Of relevance to this subsystem description are the mechanisms used in IEEE 1609.2 to counteract various threats. Specifically, in addition to basic signing, the IEEE 1609.2 protocol includes "scope" elements that restrict the time, geography and function associated with a signature. To accomplish this, the IEEE P1609.2 headers include transmission time, transmission location and the PSID of the originating application. If the message is subsequently received at a different time or at a different location, then it will be considered invalid. In addition, if the message is sent from an originator who is not authorized to send messages of that type, then the PSID of the message will not match any PSID in the certificate, and the message will be considered invalid.

Also included in the OBE Security Protocol are mechanisms for localized encryption. This mechanism, known as VII-Datagram Transport Layer Security (V-DTLS) is used to encrypt public and private data sent from an OBE to a roadside unit. The function is specifically intended to protect probe data messages from being intercepted from the radio link. While the content of this data will eventually be made public, there is a risk that a bystander on the roadside could intercept the delivery of this data and correlate the content with a particular vehicle (for example, if there is only one vehicle passing the roadside unit, it would be obvious where the data came from). Since probe data carries a history of speeds and locations, the information could compromise privacy when tied to a specific vehicle. For this reason, V-DTLS provides a mechanism for encryption over the radio link.

OBE Security Services are a critical element of the Security subsystem. These functions are used to manage the security certificates in the OBE on behalf of the radio and the various applications, and to perform the various message security operations such as signing, verification, encryption and decryption.

The POC Security Services architecture is shown in Figure 4-41. This figure illustrates that the Security Services software operates at multiple levels in the OBE software system. This requirement derives from the fact that different users of the Security Services exist at different levels in the software system. Specifically, the DSRC Radio code, and the Crypto processing card driver exist in kernel space, and the various POC applications are implemented in Java in user space. In addition, the system was designed to be available to native applications (e.g. C++ code applications) running in user space.

As a result, the code to implement the Security Services has complex interfaces between these levels, and some services, for example the Secure Messages and CM function use code implemented in both C++ and Java.

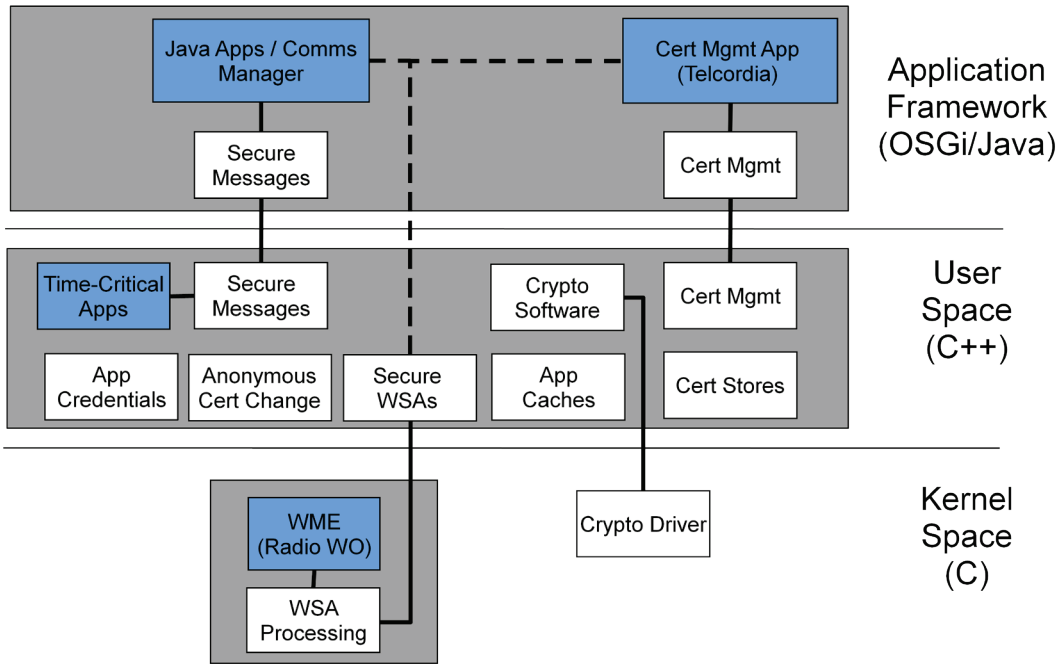


Figure 4-41 POC Security Services Architecture

The Security Services operate as follows: the Security system is initially set up with a root certificate that will be recognized by the CA (See Section 4.11). When an application registers for Security Services, the Application Credentials element determines that the application has no certificates in the Certificate Stores. This causes the CM to contact the CA (when DSRC Radio connectivity is available) to obtain certificates for the application. To do this it uses the pre-stored root certificate to authenticate itself and to establish a secure session with the CA. When the CA returns the application certificates, the Certificate Manager stores them in the Certificate Stores.

Inbound Security Operations

When a WSA is received, it is sent to the Secure WSAs element that checks its signature. If the signature is valid, the Secure WSAs element notifies the radio, and the radio joins the advertised service (assuming there is a service for one of the registered OBE applications). This activity involves communications between kernel space (where the radio operates) and user space (where the Security Services operate). This communication is necessary because otherwise all of the code used to verify signatures would need to be duplicated in both user space and kernel space.

When a signed or encrypted WSM is received, it is sent to the Security Services by the Communications Manager for verification or decryption or both (depending on the message). For a signed message, the Secure Messages element checks the signature and verifies that the certificate used on the signature is also valid (by checking it against the CA certificate). If the signature is valid, Secure Messages notifies the Communications Manager and the Communications Manager passes the message to the application in which it was intended. For encrypted messages, it simply decrypts it and passes the decrypted message to the Communications Manager.

At times, an application may choose to access the Security Services directly. This is primarily used when an OBE application has established a secure IP connection with another remote application. In this situation, the OBE application simply passes the message to Secure Messages and the result is passed back to the application.

Outbound Security Operations
If the OBE is operating as a Provider (See Section 4.3), it must broadcast signed WSAs. In this case, the WSA is formed by the radio and it is then passed up from kernel space to the Secure WSAs element in user space for signing. In this case, Secure WSAs uses a signing certificate it has obtained from the CA. The signed WSA is then passed back to the radio for broadcast.

When an OBE application wants to send a signed or encrypted WSA, it passes the message to the Communications Manager which passes it to Secure Messages in user space. Depending on the needs of the application, Secure Messages either signs the message or signs and encrypts it, and passes it back to the Communications Manager for transmission.

Similar to the inbound leg, OBE applications can also use the Security Services directly to sign or encrypt WSMs and IP packets.

4.5.3.5 Positioning Service

The OBE Positioning Service provides a suite of positioning services to the other on-board services and applications based on the output of the external (primary), or internal (secondary) GPS receiver. The Positioning Service uses position information from the GPS receiver as well as extrapolated positions that it computes every 100 ms when requested.

The Positioning Service provides two separate APIs: one Java API, implemented as an OSGi bundle, and a native C/C++ API, implemented as a native library. A GPS daemon is used to share the access to the GPS port between at least two application processes: the Java Virtual Machine (JVM) process that runs the VII applications and at least one native process that runs the native applications. The POC Positioning Service architecture is illustrated in Figure 4-42.

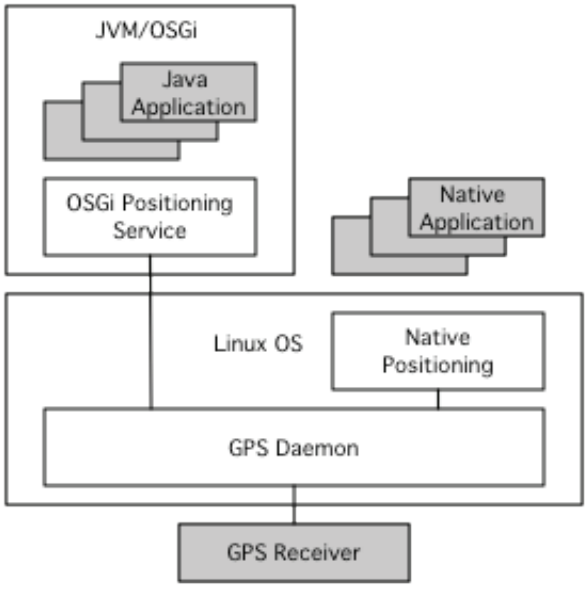


Figure 4-42 POC Positioning Service Architecture

Positioning Service Functions
The Positioning Service is available as an OSGi bundle, with an activator class. The API is built upon a main Java Interface (implemented by the activator), a set of classes representing the data provided or handled by the positioning service and a set of listener interfaces to be implemented by the application.

The Positioning Service Interface provides the following features:

- Access to the latest GPS position: a getPosition() method returning a Position object
- Position listeners for both the GPS position (received every second) and the extrapolated position (calculated every 100 ms)
- Polygon listeners - A polygon is represented as the list of points that define its edges. All polygons must be simple in the mathematical sense: two edges must never intersect
- Circle listeners- A circle is represented as a center location and a radius.

These features are based on three basic data classes: the Point, Position and Area classes:

- A Point object represents one point on earth
- A Position object represents the position of a vehicle at a given time
- An Area object represents an arbitrary area on earth that includes a set of polygons or circles. The Area class is used to implement the grouping of polygon listeners.

Vehicle Position
The Positioning Service makes the assumption that the position of the GPS antenna defines the position of the vehicle. This is called the vehicle's reference point. While the position is always derived from the GPS receiver, there are several different types of position solutions, depending on how the position was computed: GPS, GPS-augmented with Differential Global Positioning System (DGPS) or dead reckoning. In addition, on request, the Positioning Service may extrapolate a position every 100 ms, in which case the type is an "extrapolated" type derived from the three types listed above.

The Point Class
The Positioning API defines a Point Class. Each Point object represents one point on earth. A point cannot move. The longitude, latitude and altitude items are final once the object has been created. A point has no direction, speed or accuracy information.

The Point Class defines methods for doing operations on points, such as obtaining the latitude, longitude or altitude of the point, computing the distance between two points, obtaining the true north azimuth of the line connecting two points, and so forth. The Point Class also provides a method for determining if a specified point is inside or outside a polygon specified by a set of three or more points. In the POC project, this was used for payment events in the Tolling and Parking Applications, for relevance checks in the Advisory Message (Signage) Application, and was used for security certificate checks.

The Position Class
The Positioning API defines a Position class to extend the Point class with vehicle and measurement information. Each Position object represents the vehicle positioning information at a given time. It is a Point object, since the vehicle is at one point on earth, however it also has speed, heading information (since the vehicle might be moving), and it has accuracy information since the actual measurement method is known to have errors.

A Position object cannot be modified. The coordinates, heading, speed, measurement time and accuracy cannot be changed. When a new vehicle position is generated, a new Position object is created. Therefore applications can keep a reference to a Position object without incurring time-sensitive abnormal behavior.

The Position class provides methods to get vehicle speed, direction and various accuracy measurements.

The Positioning Service provides two classes of vehicle position:

- The GPS position is measured by the GPS receiver every second and is received on the OBE and translated by the Positioning Service into a Position object.
- The extrapolated vehicle position is calculated by the Positioning Service every 100 ms.

In addition to simply requesting the position, the estimated position can be obtained by registering a listener. This approach automatically provides the current position each time the position is updated.

Region and Area Class Listeners

Most VII applications are concerned with localizing the vehicle's position relative to physical features in the real world, such as gas station pumps, parking entrances and exits, etc. The applications typically access a map database that defines polygons for each feature of interest. These features are naturally grouped in areas: the gas pumps belong to one gas station; the entrances belong to one parking lot, etc.

This natural grouping is of importance to the Positioning Service for two reasons:

- It provides a coherency unit. For example, it does not make sense to have registered a listener for only one parking entrance if there are two actual entrances.
- It provides an optimization opportunity, as most of these areas do not intersect. Therefore, if a vehicle is found to outside an area, it is outside any of the area's zones.

The Positioning Service provides support for registering region listeners. The listener is called when the vehicle enters the region and (optionally) when the vehicle exits the region. A region is a geometric figure that defines an exact portion of space. Regions are described as polygons with Points (See *Region and Area Class Listeners*) as the vertices, or as a circle with a Point center and radius.

The Positioning Service also defines an Area class that is used to group Region listeners and to notify the registered application when the vehicle is inside the area defined by the regions.

4.5.3.6 Vehicle Interface Service

The Vehicle Interface Service (VIS) provides a common referencing scheme and means for accessing vehicle data. It allows the OBE to be used in a variety of vehicle types without needing to customize each application to interface with each vehicle type.

Figure 4-43 shows the main logical layers of the Low Level Vehicle API Module:

- Layer A (CAN-bus Access) implements the access to the raw data from the device drivers.
- Layer B (Low Level Connectors) implements a set of connectors to different data sources in the vehicle.

- Layer C (Mappings and Conversion Rules) is responsible for the conversion of the raw data (e.g. from the CAN frame to meaningful CAN messages).
- Layer D (Object Management) defines the data structures that will be used at the higher application layers.

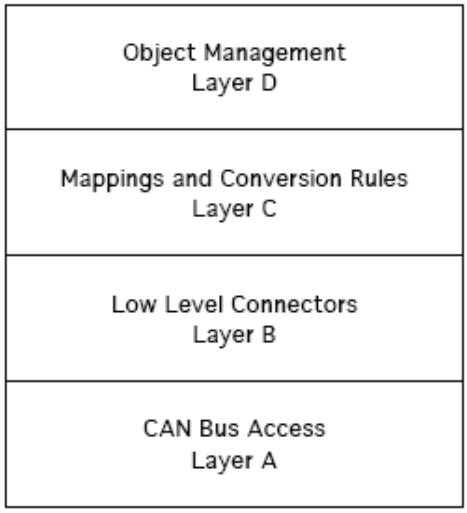


Figure 4-43 Logical Layers of the Vehicle Interface

4.5.3.6.1 Low Level CAN Framework (LLCF)

Layers A and B were implemented using a code set developed by VW. This is known as the Low Level CAN Framework (LLCF). The LLCF is based on a network CAN driver that abstracts the different bus protocols over standard Linux interfaces (sockets). It defines a new protocol family PF_CAN, similar to the IP PF_INET between the network layer and the socket layer of the Linux Transmission Control Protocol (TCP)/IP stack. Figure 4-44 shows an overview of the POC architecture of LLCF.

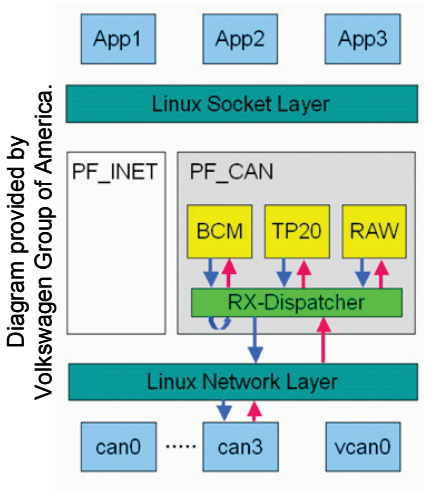


Figure 4-44 POC Architecture of the Low Level CAN Framework

The POC architecture re-uses the Linux Network and Socket Layers, ensuring ease of use for the application developers, who can access the CAN-bus over standard Linux sockets. The "can0" to "can3" modules are CAN drivers for different proprietary CAN-busses that have been re-worked in the form of network drivers. The RX-Dispatcher is responsible for the forwarding of the data from the CAN-busses to the different protocol modules. Because of the particularities of the CAN addressing it can happen that there are several modules that want to receive one and the same message with a given CAN ID. The protocol modules can register in the RX-dispatcher for which CAN IDs they want to listen to. The RX-Dispatcher then forwards the received frames to all entities that have registered listeners. In the POC, the VIS operates on behalf of the various applications.

The aim of the LLCF architecture is to make the communication with the CAN-bus as close as possible to standard TCP/IP communication over sockets.

4.5.3.6.2 Vehicle API

Because the OBE applications are Java bundles running in the OSGi Framework, it is necessary to provide a Java API so that these applications can efficiently and easily access the information provided by the LLCF. The Virus Application Programming Interface (VAPI), also developed by VW, runs on top of LLCF.

As shown in Figure 4-45, the VAPI is separated in two parts, the VAPI Daemon (server) and the J-VAPI Client Stub. The VAPI Daemon works as a native application on top of the LLCF. The J-VAPI Client Stub provides Java API to access the data, which VAPI Daemon handles from the LLCF and the underlying CAN network.

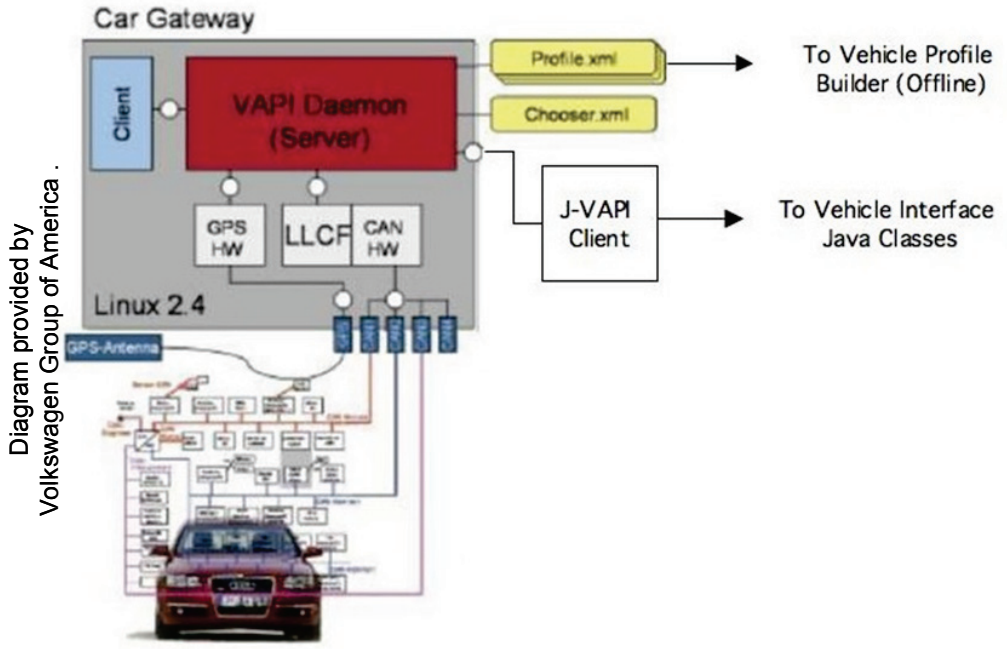


Figure 4-45 VAPI Architecture

The VAPI Daemon works on the device, in which the vehicle network is attached. The VAPI Daemon has a VAPI Profile (shown as “Profile.xml” in Figure 4-45), which it loads at its startup,

and this profile uniquely specifies what vehicle sensors are accessible through the VAPI, thereby tailoring the VAPI to the specific vehicle in question. The VAPI defines three logical elements in describing the VAPI Profile:

- **Data objects** – describes the various vehicle sensors, which are visible to the applications. It defines the Uniform Resource Identifier (URI) at which the vehicle sensor information is available and the type of the data that it provides.
- **Protocol Data Units (PDU)** – describes the meaning of the physical signals that are transmitted through the vehicle bus.
- **Mapping of PDU and Data objects** – describes how data objects are constructed from PDU and vice versa. The first description is used to read vehicle sensor information from the bus and the second one is used when the data has to be sent through the vehicle bus.

All of this information is mixed in a single VAPI Profile Extensible Markup Language (XML) file, which uniquely specifies what vehicle sensors are supported on a given vehicle. There is a tool, called Profile Generator, which can be used to construct this VAPI Profile XML. The VAPI profile often contains proprietary information about the structure of a given car-maker's CAN-bus implementation. Since this information is considered proprietary, it was made difficult to read by human form via an "obfuscator". The obfuscator essentially scrambled the profile in a manner that made it effectively impossible for a casual user to observe or copy the content of the profile.

VI Device Management Tree Admin
As described, the VAPI provides a Java API for a specific set of parameters available in a particular vehicle. An additional problem that must be overcome is that the POC uses different vehicle types, and yet the OBEs are all the same. To avoid needing to port each application to each vehicle, the OSGi Vehicle Interface Device Management Tree (VIDMT) was used (Layer D). The VIDMT Admin provides a standard naming scheme for high level access, which then is mapped to the different vehicle ontologies by the OEMs themselves. The mapping can be provided in XML form and will hide the proprietary internal structure of the vehicles. This approach also makes the naming scheme independent from the actual sensor networking; therefore, the sensors can be grouped in a logical way, which will be more easily understandable by the application programmers. The OSGi Vehicle Expert Group (VEG) scheme follows a structural approach using six sub-trees:

- VehicleInformation (static parameters like VIN)
- VehicleOperations (operational parameters like speed)
- VehicleBody
- VehicleChassis
- VehiclePowerTrain
- VehicleInterior

Each sub tree contains vehicle component primitives that describe the basic naming scheme for that particular type of component. Using this scheme, the specific resource names for any and all instances of a vehicle component can be consistently named. The Device Management Tree Meta Data provides the basic information about what is available in the particular car, and the application can easily determine the name of each parameter.

4.5.4 DSRC/GPS Antenna

The POC antenna posed a special set of design constraints. The OBE uses both GPS and DSRC Radio functions. Since these two systems operate in different parts of the radio spectrum, each

requires its own antenna. In addition, the coverage patterns are very different. DSRC requires good coverage in all azimuth directions but requires very limited range of elevation performance (because the other cars and RSEs are all generally on the same more or less planar road). The GPS system is seeking signals from space, so the GPS antenna must provide good gain in the vertical and azimuth axes. In addition, because GPS signals are sent from satellites in space orbit, the signals are very weak, so it is typical to include low-noise amplifiers at the antenna to limit the noise contribution from cabling to the receiver.

Most DSRC work done prior to the POC used vertical post antennas. These provide excellent radial gain, but they protrude vertically from the vehicle. One of the goals of the VII POC program was to demonstrate that such a system is feasible for production vehicles, and this meant that the antennas needed to have a little impact as possible on the vehicle profile and esthetics.

The solution was to develop a magnetically attached planar antenna module that provided both GPS and DSRC functions. The magnetic mount allowed the antennas to be quickly installed with no modifications to the vehicle, and it also allowed the antennas to be moved easily from vehicle to vehicle.

The antenna design, shown in Figure 4-46 uses a single planar element that has patterns for both DSRC and GPS. The patch itself forms the GPS antenna which is tuned by the location of the feed posts and the corner cuts (See Figure 4-46). This creates a Right Hand Circularly Polarized antenna with good vertical and azimuth gain performance. The DSRC antenna is formed by a ring slot structure etched into the substrate above the GPS patch. This structure has similar performance to the monopole post used in prior DSRC work, but it is co-planar with the surface of the vehicle. The DSRC pattern was optimized by the addition of circumferential slots around the ring.

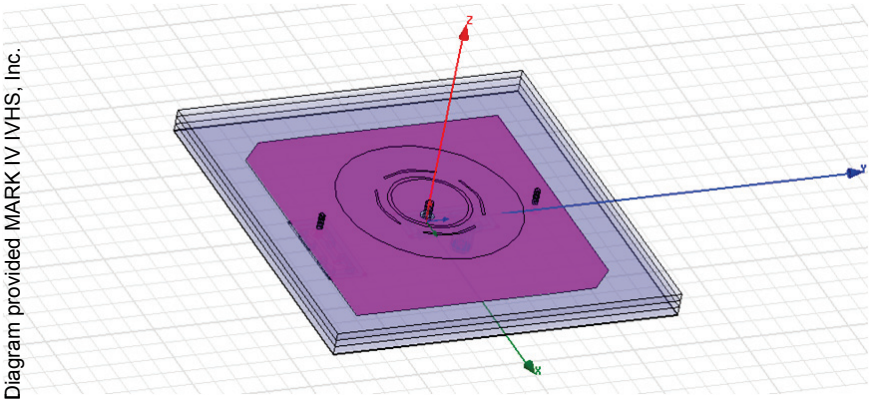


Figure 4-46 Planar Dual GPS/DSRC Antenna Element

The measured gain plots for the two antenna structures are shown in Figures 4-47 and 4-48.

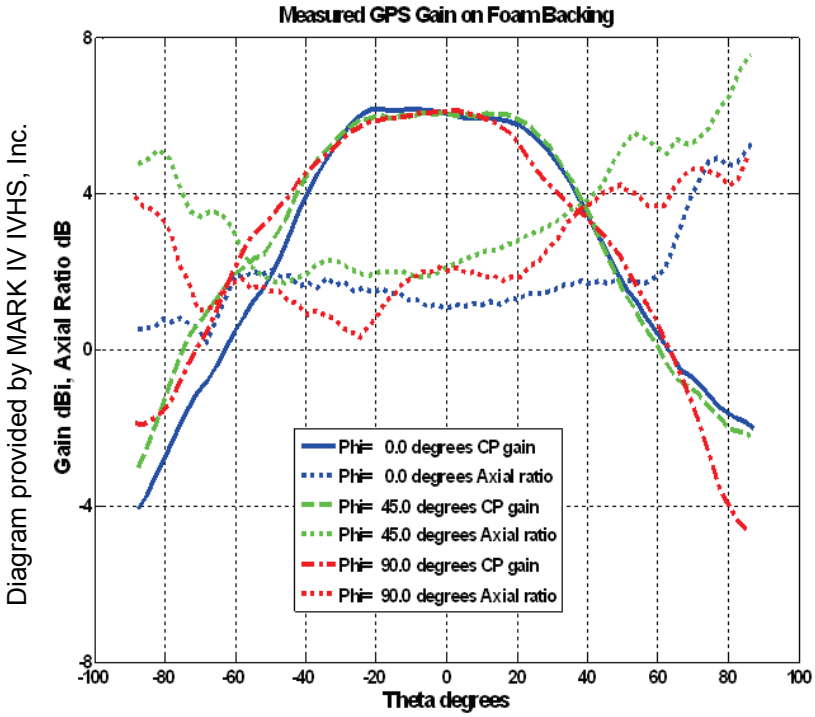


Figure 4-47 GPS Antenna Gain

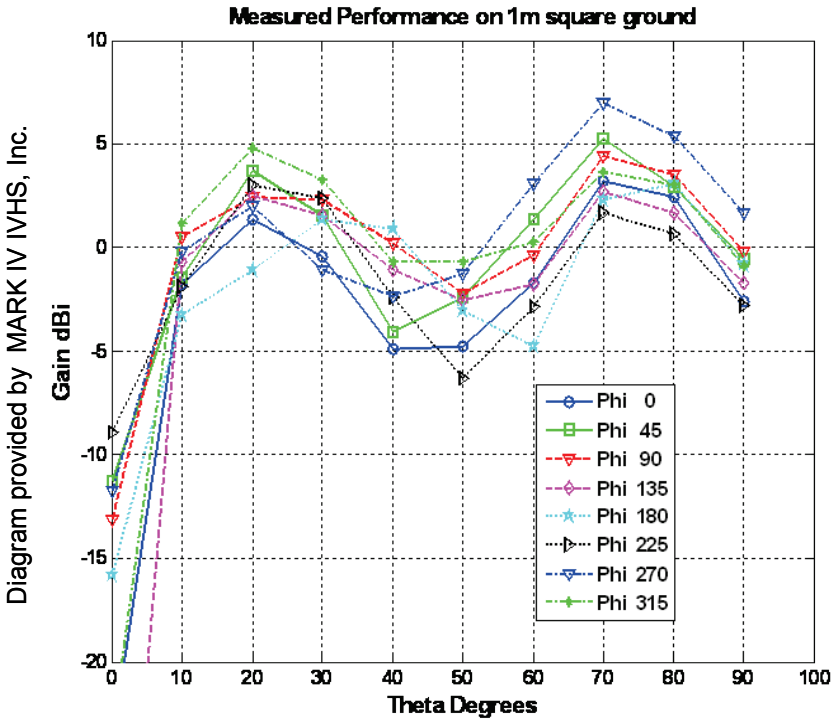


Figure 4-48 DSRC Antenna Gain

The antenna was packaged in a low profile plastic package that included room for the GPS low-noise amplifier. Power for the amplifier was passed through the RF coaxial cable. The package and cabling are shown in Figure 4-49, and the unit mounted to a POC vehicle is shown in Figure 4-50.

Figure 4-49 Dual DSRC/GPS Planar Antenna Package and Cabling

Figure 4-50 Antenna Mounted on POC Vehicle

4.5.5 External Positioning Unit

The OBE uses a combination of internal and external GPS receivers. The internal receiver is part of the Eurotech DuraCOR unit and is described in Section 4.5.1.1. Because the POC applications generally require higher levels of accuracy than the DuraCOR GPS receiver can provide, a provision was made for an external unit. The external GPS receiver includes a dead reckoning sensor that is intended to track the position of the vehicle between GPS position reports by keeping track of distance and heading changes, and extrapolating a position fix from the last

known fix. The dead reckoning sensor is a low-cost gyroscopic device that provides motion change information to the external GPS receiver. The output of this device is calibrated by the receiver while in GPS coverage, and used to update the vehicle location during short GPS outages. The internal GPS receiver is used primarily to provide an accurate time base for the DSRC Radio; however, it also is available for use as a back-up source of positioning information.

In the POC program, two types of external positioning units were used; a SiRFStar II unit and a U-Blox unit. Both of these systems included internal, dead reckoning sensors and the associated filtering software. These systems obtained a GPS signal from the dual DSRC/GPS antenna via a power splitter that also routed the signals to the internal GPS card in the DuraCOR unit.

The overall positioning system is shown in Figure 4-51.

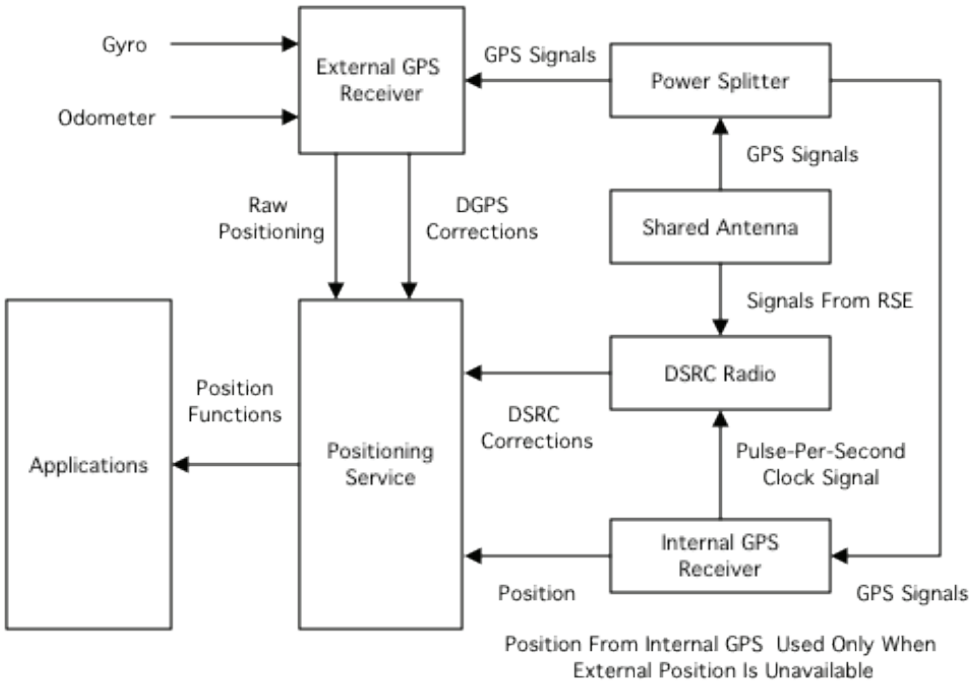


Figure 4-51 OBE Positioning Subsystem

The SiRFStar II unit is shown in Figure 4-52, and the two units are compared in Figure 4-53.

Figure 4-52 SiRFStar Positioning Unit

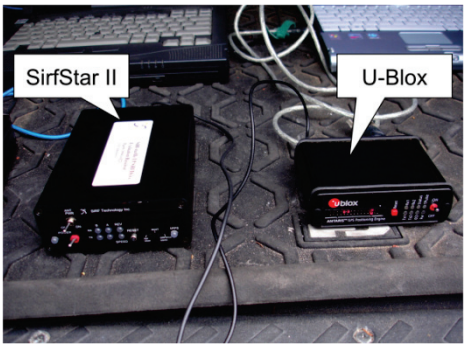


Figure 4-53 SiRFStar II and U-Blox Positioning Units

4.5.6 Power Management Unit

Because the vehicle power system is both noisy and subject to abrupt interruption, the vehicle integration included a power management system. The CarNetix power control module, shown in Figure 4-54 provides filtering as well as programmable timing for various power buses. This allows the system to power on and off the supply voltages for various components, in a specific and prescribed order.

Diagram provided by CarNetix.

Figure 4-54 CarNetix Power Management Unit

The CarNetix unit provides a USB port which allows a laptop and CarNetix client software to configure, manage and if needed, monitor the power supply. The power supply hardware configuration is controlled by jumpers internal to the CarNetix unit. Figure 4-55 shows the setup screen used to set and monitor the supply lines and their respective on/off sequencing.

Rendering provided by CarNetix.

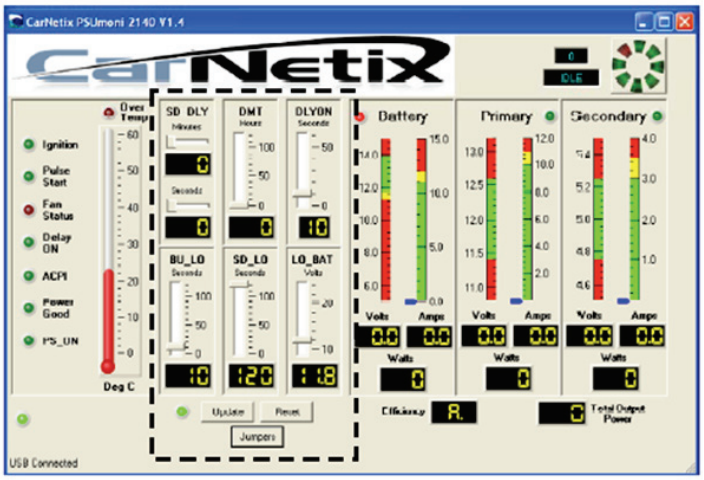


Figure 4-55 CarNetix Power Management Controls

4.6 VIIC POC Vehicle Integration

The POC included a variety of vehicles, each of which presented its own system integration challenges. For the VIIC-equipped vehicles, a standard setup was used that allowed the system to be efficiently installed in any of the vehicles. A few of VIIC car maker members also developed their own custom installations.

Complete installation of an OBE subsystem includes installing and interconnecting the following equipment in the vehicle:

- Passenger Compartment
  - Xenarc 700TSV HMI / Touch screen
  - Test / maintenance laptop
  - Vehicle CAN-bus connection(s)
  - Cabling to and from trunk
- Vehicle Exterior (Roof)
  - Integrated GPS / DSRC Antenna, Mark IV 801836-001
  - Cabling to trunk
- Vehicle Trunk (or rear, if no trunk)
  - OBE Computing Platform
  - Carnetix P2140 DC-DC power supply
  - U-blox or SiRFStarIIe/LP SiRFDRive External Positioning Hardware
  - GPS antenna splitter
  - Ethernet switch (Netgear GS605)
  - Power distribution block
  - Cabling between in-trunk components
  - Cabling to passenger compartment
  - Cabling from Integrated Antenna

The OBE subsystem assembly is composed of a main aluminum chassis that holds the multiple components that make up the assembly. This is shown in Figure 4-56. The purpose of the raised panel on which the components are mounted is to allow a convenient place to store the large cable harness. This harness was necessary since each car is slightly different in layout, and the program wanted to avoid developing specialized harnesses for each vehicle. As a result, use of

the “one size fits all” cable harness often meant that there was excess cable length that was stored in the area under the OBE chassis platform. A portion of this cabling can be seen in Figure 4-57.

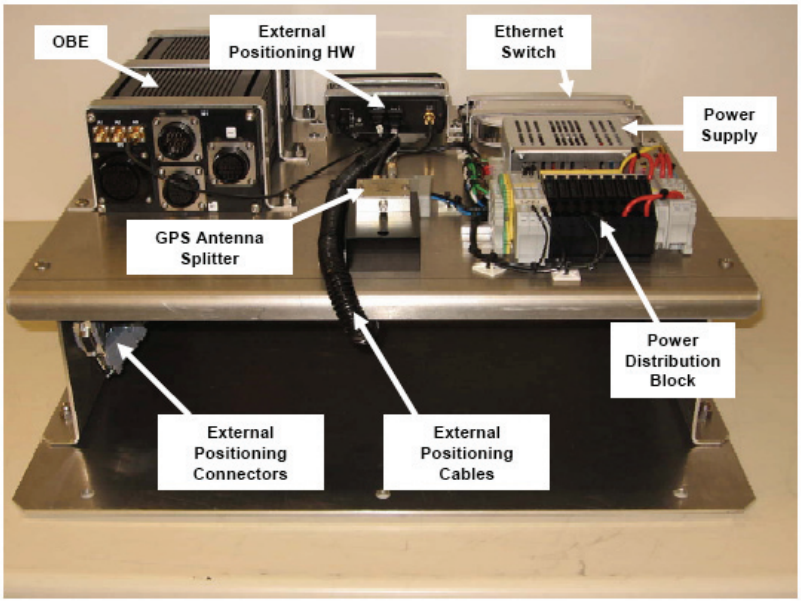


Figure 4-56 OBE Subsystem Assembly

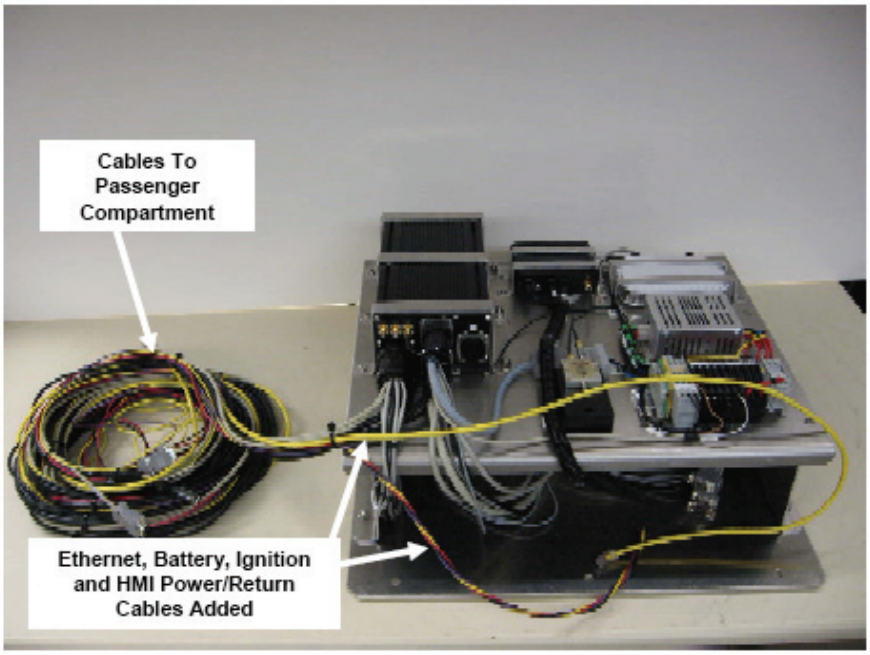


Figure 4-57 OBE Cabling Assembly

A typical trunk assembly is shown in Figure 4-58. This approach allowed the OBE to be installed with minimal disruption of the vehicle and allowed the OBE subassembly to be easily accessible. As an example of the complexity of the vehicle integration, the hinges shown in the figure are necessary to allow the OBE assembly to be tilted up to access the vehicle’s spare tire, a necessary precaution for a vehicle traveling thousands of miles in tests on the open road.

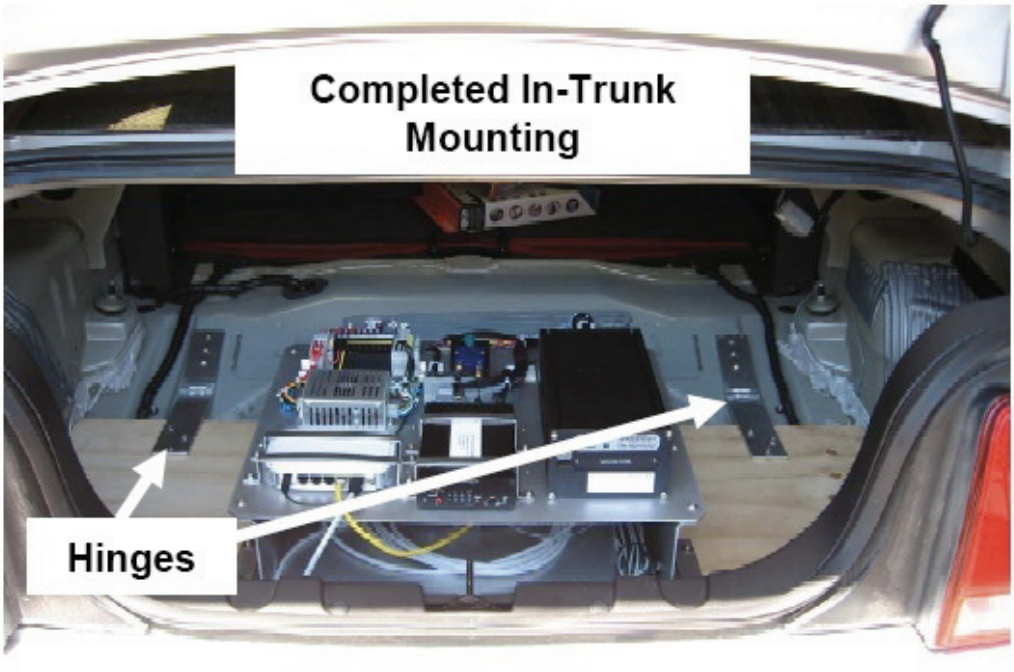


Figure 4-58 OBE Trunk Mounting

The HMI display also presented challenges. In some vehicles, it was possible to remove an existing screen or display subsystem and replace it with the HMI. In others, there was simply no room to do so.

Figures 4-59 and 4-60 show two typical HMI mounting approaches. Figure 4-59 is in a Ford Mustang, which required an external mounting approach. Figure 4-60 is a Ford Edge, which allowed a more fully integrated in-dash mount.

Figure 4-59 HMI External Dash Mount

Figure 4-60 HMI In-Dash Mount

The dual DSRC/GPS antenna was typically mounted to the roof by magnets embedded in the antenna housing. Figure 4-61 shows the antenna mounted on the center of the vehicle roof. This placement allows the antenna to be horizontal which results in symmetrical antenna coverage front-to-back. Other installations placed the antenna closer to the rear window (thereby eliminating the long cable run across the roof). This approach resulted in asymmetrical antenna coverage at times. The effects of these differences are discussed in Final Report --Volumes 3a, 4a and 5a.

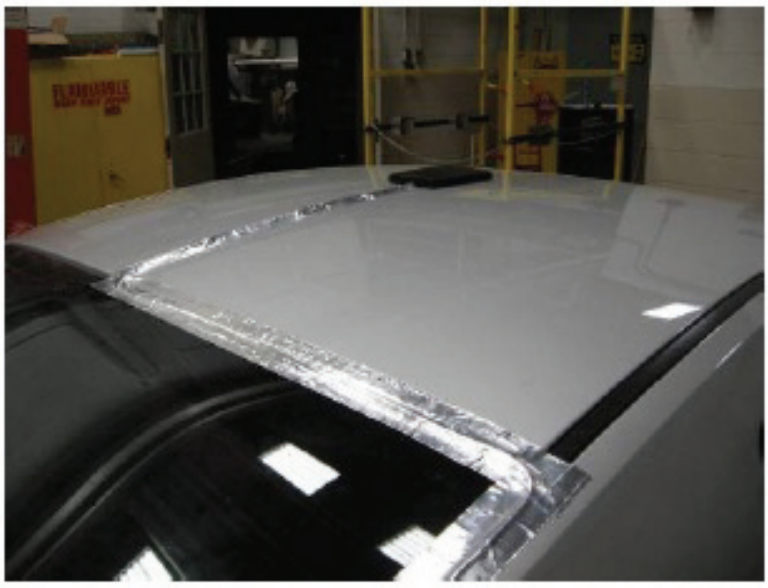


Figure 4-61 OBE Roof Mount DSRC/GPS Antenna

4.7 POC Applications Description

4.7.1 POC Applications Overview

As previously described, the VIIC POC effort developed seven applications that used and exercised the core system functions. These applications are:

In-Vehicle Signage receives electronic advisory messages from roadside units, and, based on location and timing information, presents the message content in graphical and audible form to the driver using the OBE HMI.

Probe Data Collection gathers vehicle operating data from the Vehicle Interface and position information from the Positioning Service, and compiles a "snapshot" of the vehicle state at that time and location. The application saves snapshots in a set and then uploads the snapshot set to the network-based PDCS when the vehicle encounters an RSE.

Electronic Payments-Toll sends out an announcement from local processor via an RSE. The announcement contains toll plaza location information. When the OBE application determines it is inside the toll plaza zone, it obtains toll payment information and toll payment zone information from the local toll processor. When the vehicle enters a payment zone, the OBE application notifies the payment service which sends a payment message to the local toll processor. All messaging relating to user identity and payments is encrypted, and transactions occur at vehicle road speed.

Electronic Payments-Parking operates on the same principles as tolling, but speeds are slower and payment and plaza zones are smaller and more complex.

Traveler Information / Off-Board Navigation sends a request for a route from the current OBE location to a pre-set destination. The request is forwarded by web services system to a navigation service provider, where the route is computed including turn-by-turn directions. Directions are sent back to the OBE at the same RSE as the request is received. If the delivery of route is interrupted, for example, by the vehicle leaving the RSE zone before the route download is complete then at the next RSE encounter the process starts where it left off. The route may also be updated based on real-time traffic data collected, for example, from the probe system.

Heartbeat compiles a regular vehicle status message containing speed and position data, sending messages out at regular intervals (typically every 100 ms). The OBE also receives the same type of message from other vehicles. Primary output is a log of sent and received messages (the current application does not do any safety processing on the message). This application is primarily used to assess high message rate generation and reception.

Traffic Signal Indication is a stub application. A traffic signal controller sends a Signal Phase and Timing (SPAT) message to a local RSE at regular intervals. The RSE transmits the message, and the OBE receives it. The Traffic Signal Indication Application decodes the message and presents the current signal state and the time remaining in that state using the HMI display. This application is used to test the effectiveness of the system in handling and prioritizing safety messages while supporting lower priority operations.

4.7.2 Tolling Payments Application

The POC Payment for Toll ("Tolling") Application allows the vehicle driver to securely make an automatic toll payment while passing through a defined tolling area.

Tolls are assessed and paid as the vehicle travels at high speed through a defined toll plaza and lane-specific charging zone. The system is operated by a Local Transaction Processor (LTP) that is in direct communication with a local RSE. The Tolling Application utilizes a network side debit account from which toll payments are deducted. The application uses digital certificates and digital signatures to encrypt all data transactions for privacy, and to authenticate all payment confirmations. These security measures guarantee that an undisputable toll amount is deducted from the correct financial account while preserving the confidentiality of the account information.

The Tolling Application is distributed over four major components: an In-Vehicle Component, a LTP Component, a VII system component and a Network Users Component (NUC).

The **In-Vehicle Component** contains the In-Vehicle Toll Processing (IVTP) Element and In-Vehicle Payment Service (IVPS) Element. The IVTP identifies when toll transactions occur, facilitates the transfer of account information between the driver and the vehicle, presents information to the driver, and gets selections from the driver via the vehicle HMI.

The **LTP Component** contains an LTP Toll Processing (LTTP) Element that defines the existence of toll zones, generates Toll-payment invoices and processes the corresponding acceptances of payment during the course of the toll payment transaction.

The **NUC** contains a Network Users Payment Service (NUPS) Element that verifies the toll payment.

The application operates through the VII network component that provides a Roadside Infrastructure Support Service and a Communications Service. Together, these two services support message routing and delivery, service announcements and communications support between the elements in the In-Vehicle, Local Transaction Processor and NUCs.

In addition, the Tolling Application makes use of the Communication Manager, the HMI Manager, the OBE VII Positioning Service and VII Security Service during the course of the Toll Payment Transaction.

The relationship of these components in the context of the Tolling Application is provided in Figure 4-62.

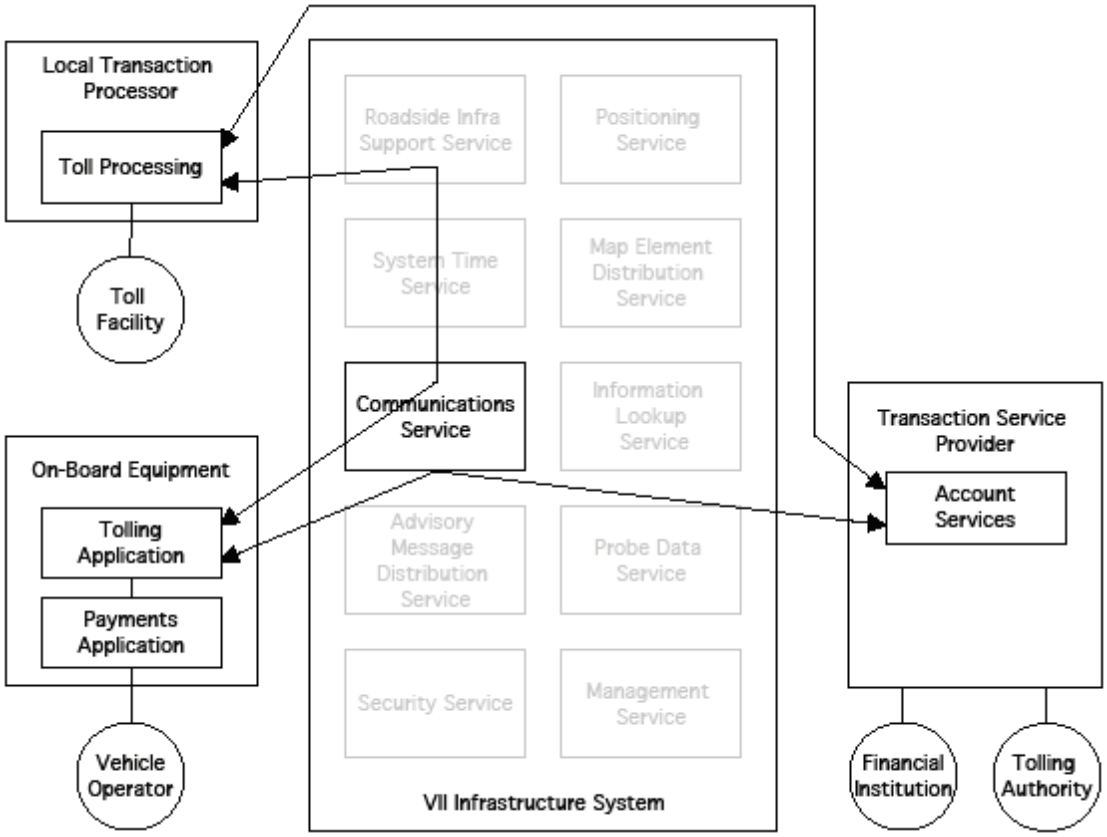


Figure 4-62 Payment for Toll Application System Overlay Diagram

4.7.2.1 POC Tolling Application Architecture

Figure 4-63 illustrates the elements of the Payment for Toll Application. The LTTP communicates to the IVTP via the RSE Radio Handler. The LTP communicates directly with the NUPS as shown, however the physical connection is routed through the Network Component for POC.

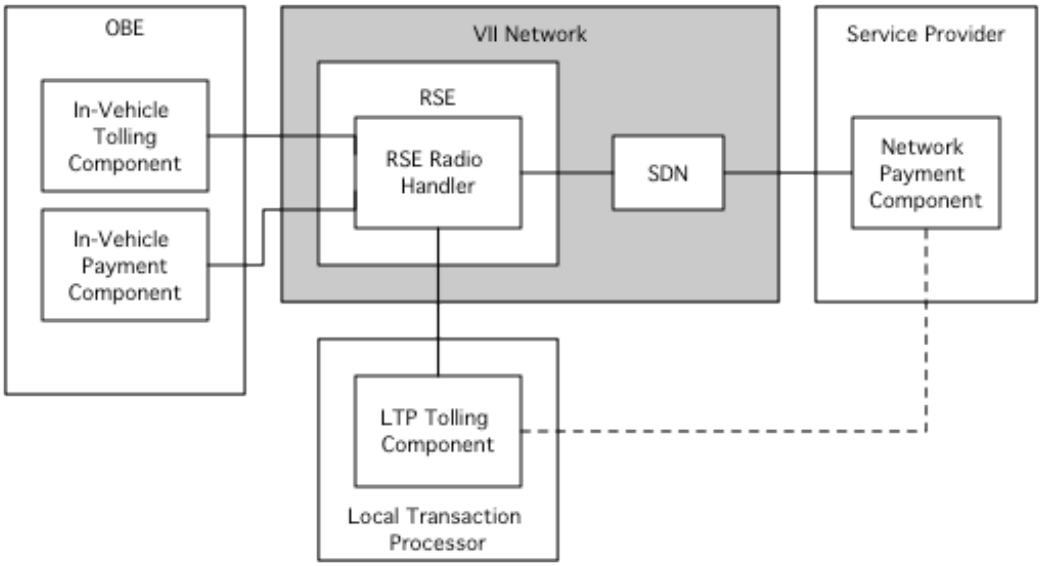


Figure 4-63 Payment for Toll Component Diagram

In-Vehicle Component

As shown in Figure 4-64, the Tolling Application is comprised of the IVPS Element and the IVTP Element. The IVTP Element uses the payment services provided by the IVPS Element during the course of a toll payment transaction.

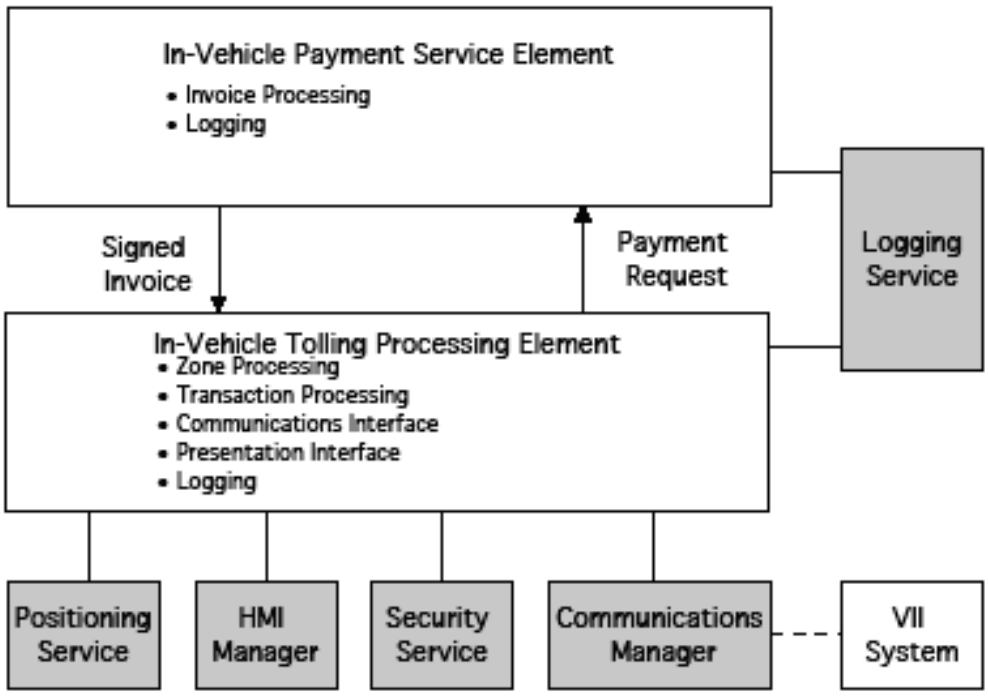


Figure 4-64 In-Vehicle Component Overview

In-Vehicle Payment Service Element
The IVPS contains account identification information for multiple accounts and provides an interface for the IVTP to either request account identification information, or exchange invoice and receipt data. The IVPS also has the ability to digitally sign invoices using the OBE Security Services.

In-Vehicle Toll Processing Element
The IVTP controls the toll transaction process inside the vehicle. To accomplish this, the IVTP determines when the vehicle is inside of the Toll Plaza Zone(s) and Toll Collection Zone(s) using the VII Positioning Service and the geographic information provided in the LTTP messages.

At each geometric crossing point, the IVTP initiates the next stage in the process by sending messages to the LTTP and by activating the IVPS (to start the actual payment process). The IVPS and IVTP are separate to allow for different types of payment methods without changing the tolling transaction logic.

In-Vehicle Toll Processing Presentation Management
The Presentation Management notifies the driver using visual displays and audible tones (Via the OBE HMI Manager) to indicate when a toll has been paid. The typical screen used to indicate a toll payment is shown in Figure 4-65.

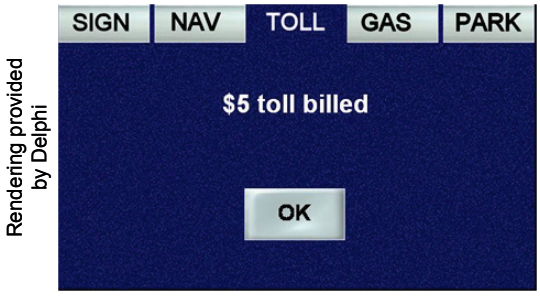


Figure 4-65 Toll Payment Screen

LTP Component
Shown in Figure 4-66, the LTP Component contains the LTTP. The LTTP provides service announcement and toll zone information to the IVTP(s) in vehicles that are in the coverage area of the RSE that the LTP is connected to. The LTTP generates invoices for vehicles passing through the toll zone, processes signed toll payment invoices, and provides transaction summary information to the NUPS.

The LTP is located adjacent to the RSE to minimize the effect of network latency. For low latency applications such as tolling, this approach avoids latency caused by the need to traverse the network to a remote site.

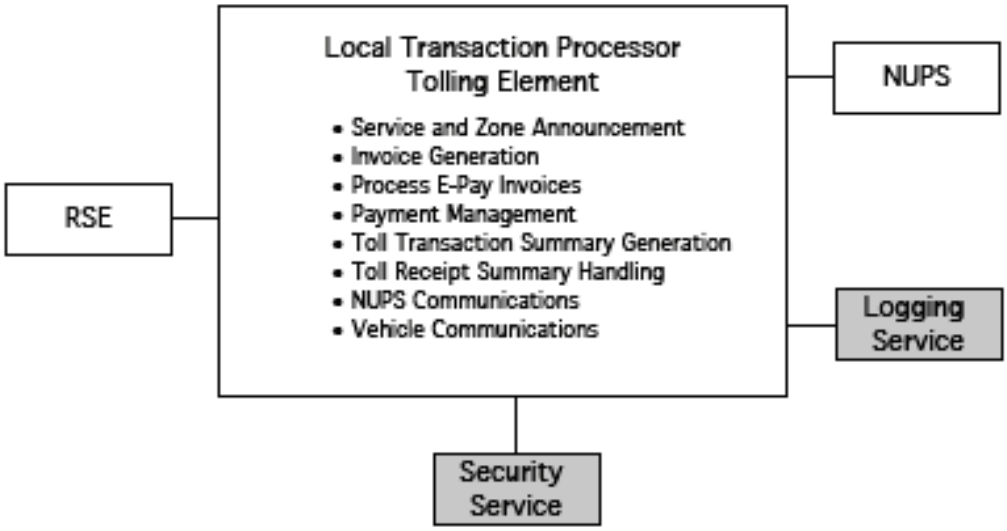


Figure 4-66 LTP Component Overview

Network User Component

The NUC of the Tolling Application, shown in Figure 4-67, is solely comprised of the NUPS Element. The NUPS authenticates secure connections from the LTTP, validates the digital signatures of signed toll payment invoices, creates toll payment receipts and summaries, and updates information in the User Account database.

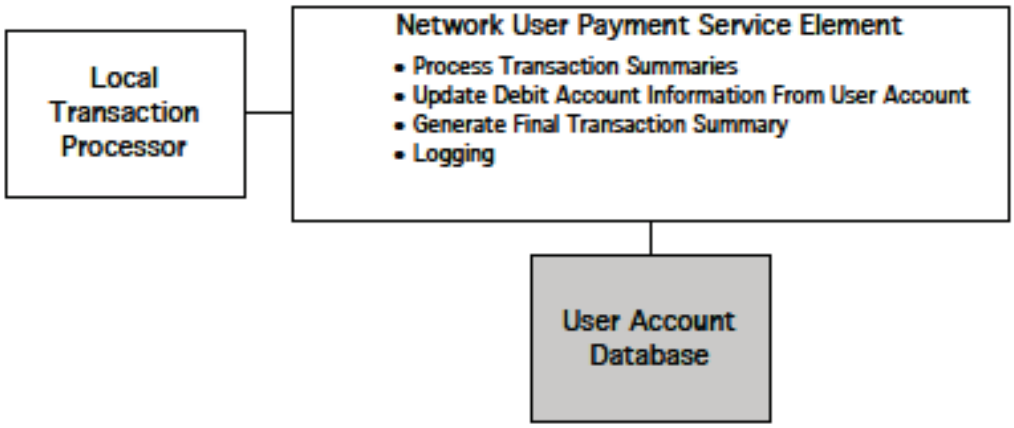


Figure 4-67 NUC Overview

4.7.2.2 Tolling Application Flow of Events

The general flow of events for the Tolling Application is briefly described in this section (for a detailed description, see the VIIC's Tolling System Functional and Performance Requirements APP 110-04).

Roadside Setup

1. The RSE Radio Handler announces the Toll Service in the WSA broadcast by the RSE. The PSC field of the WSA includes coordinates defining the "Toll Plaza Geometry" (the region inside which the vehicle should join the service and prepare to pay the toll).
2. The LTTP Element periodically creates a "Toll Zone Definition" message to be broadcast by the RSE on the SCH.
3. The LTTP element establishes a secure Transport Layer Security (TLS) session with the NUPS.

In Vehicle Setup

1. The IVTP Element registers for the Toll Service. Note that two registrations are required, one for the WSMP service and one for the IP service.

Operational Flow

1. The Vehicle approaches and enters the RSE coverage area, but has not yet entered the toll plaza area.
2. The IVTP Element receives the WSA, which includes the "Toll Plaza Geometry," indicating service is available and that the service has been joined.
3. The IVTP Element sends the Toll Plaza geometry to the OBE's Positioning Service.
4. The IVTP Element receives the Toll Zone Definition.
5. Time passes, vehicle moves…
6. The OBE Positioning Service notifies the IVTP Element when the Vehicle has entered the Toll Plaza geometry.
7. The IVTP Element sends a "Vehicle In-Plaza Zone" notification to the LTTP Element as an encrypted message.
8. The IVTP Element sends the Collection Zone(s) geometry from the Zone Definition message to the OBE's Positioning Service.
9. In response to the Request for Toll Invoice (Vehicle In-Plaza Zone) message, the LTTP Element sends a Session Acknowledgement to the IVTP Element.
10. Time passes, vehicle moves…
11. The vehicle reaches/enters a Toll Collection Zone (Defined in the original Zone Definition message).
12. The OBE Positioning Service notifies the IVTP Element that the Vehicle has reached a Toll Collection Zone.
13. The IVTP Element sends a "Vehicle In Collection Zone" notification containing the Vehicle's lane position and classification to the LTTP Element.
14. In response to the message, the LTTP Element sends a Toll Invoice to the IVTP Element.
15. The IVTP Element extracts the E-Payment Invoice from the Toll Invoice and passes the latter to the IVPS Element.
16. The IVPS Element signs the invoice using the WAVE Security service and returns it to the IVTP Element.
17. The IVTP Element incorporates the signed E-Payment Invoice into a Signed Toll Invoice message and sends it to the LTTP Element.
18. The LTTP Element sends an Invoice Confirmation message to the IVTP Element acknowledging the collection of the fee (for POC test purposes).
19. The IVTP Element causes a tone to be sounded on the Vehicle HMI and an indication to be displayed on the Vehicle HMI when the toll collection point is passed (for POC test purposes).
20. The IVTP Element removes all records of the transaction from the vehicle memory.
21. The Vehicle continues on its way.

22. The LTTP Element makes a record of the transaction and includes the signed E-Payment Invoice into a Toll Transaction Summary.
23. The LTTP Element regularly sends a Toll Transaction Summary to the NUPS Element for processing.
24. The NUPS Element extracts each signed E-Payment Invoice from the Toll Transaction Summary and verifies the signature using the WAVE Security Service.
25. The NUPS Element deducts the toll from the Driver's pre-paid debit account.
26. For each processed signed E-Payment Invoice, the NUPS creates an E-Payment Receipt and signs this using WAVE Security Service.
27. The NUPS includes the signed E-Payment Receipt into a Toll Receipt Summary message.
28. The NUPS regularly sends a Toll Receipt Summary to the LTTP Element.
29. The LTTP Element extracts each signed E-Payment Receipt from the Toll Receipt Summary and verifies the signature using the WAVE Security Service.
30. The LTTP Element correlates each E-payment Receipt with the Toll Invoice previously recorded.

4.7.3 Parking Payment Application

The Parking Payment Application allows the vehicle driver to securely make a payment for parking privileges. Parking fees associated with that vehicle within a parking lot are charged from the user's credit card account. In the POC implementation, the driver pays a fixed-rate event-style parking fee, charged at a standard rate on entry to the parking facility. The driver interacts with the application by way of the generic HMI developed for all POC applications.

The Parking Payment Application uses a pre-established credit card service from a VIIC surrogate financial institution to allow purchases made through the VII infrastructure. The application uses digital certificates and digital signatures to authorize payments and verify payment confirmations. All transaction data exchanged is encrypted. These security measures guarantee that an undisputable parking fee amount is charged to the correct credit card account while preserving the confidentiality of the user's account information.

The Parking Payment Application is constructed using essentially the same components used for the Tolling Payment Application. The substantive difference between the two applications is that the Parking Payment Application involves somewhat finer precision in the definition of the payment zones to allow for entry and exit regions located close to one another, and the transaction involves a few user interface screens to accept the payment. Architecturally however, the two applications are the same, and for brevity, we have not repeated the detailed description here.

The Parking Payment screens are shown in Figures 4-38 to 4-40.

4.7.4 Probe Data Collection Application

Figure 4-68 shows how the Vehicle Probe Data Generation Application fits into the VII POC architecture. In this figure the Probe Data Service (PDS), by way of a probe data proxy located at the RSE, announces that it is collecting probe data at that RSE. When the vehicle encounters an RSE making such an announcement, the PDC Application in the OBE then sends a package of "snapshots" of vehicle operation data (a suite of parameters) collected at regular intervals over a time preceding the encounter with the RSE to the Probe Data Proxy application in the RSE. The Probe Data Proxy application then passes this data on to the PDS at the SDN.

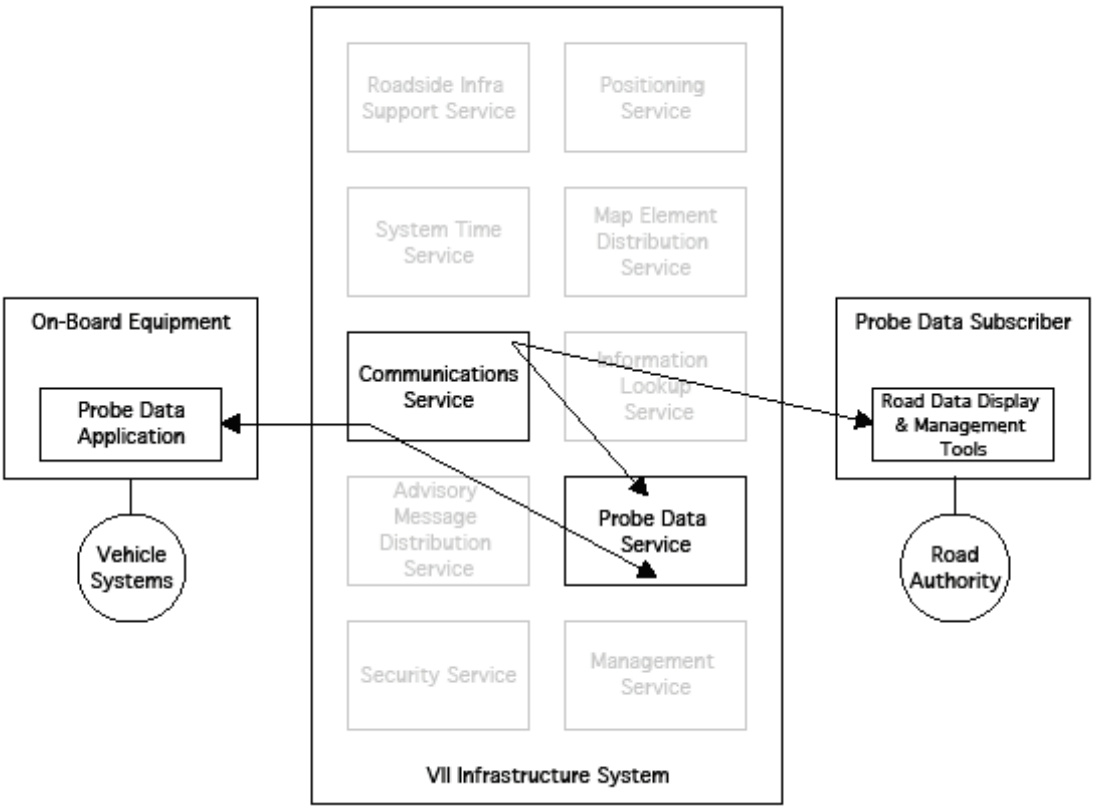


Figure 4-68 Vehicle Probe Data Generation Application System Overlay Diagram

The PDS at the SDN is composed of two components, the PDCS and the Probe Data Subscription Service (PDSS). The PDCS receives the data collected by the proxy applications at all RSEs and separates the various parameters into what are known as "topics" These topics are published using a conventional publish and subscribe architecture. The topics are essentially each type of collected parameter across the entire system. Not all vehicles will deliver all topics, but since the PDS aggregates data from many RSEs and many vehicles, most topics will be populated with data. The Probe Data Subscriber can then subscribe to any or all topics based on multiple specified criteria. For example, a subscriber might subscribe to speed at various geographic locations, or to windshield wiper state inside the boundary of a particular county. As data conforming to a subscriber's criteria arrive, at the PDS they are immediately forwarded to that subscriber.

The system operates by collecting "snapshots" of vehicle operating parameters. A number of these snapshots are typically assigned a Probe Sequence Number (PSN) so that they can be correlated as corresponding to a single vehicle when used by probe data subscribers. To avoid the obvious privacy concerns associated with this approach, the data collection policies include a required gap between PSN groups where no data is collected. This effectively separates snapshots of one PSN from those of another and makes it difficult to link the snapshots and thereby track the behavior of any single vehicle.

The system has no memory. This means that once data has been forwarded to all subscribers, it is deleted. This aids in scalability (as the system is expected to process enormous volumes of data) and it also avoids issues associated with public maintenance of data. The data collected is also anonymous. The messages are anonymously signed to assure that the sender is legitimate, and they are locally encrypted to avoid issues with radio eavesdropping, but no message contains any identifying information that might be used to link the data to a particular vehicle.

4.7.4.1 POC Probe Data Application Architecture

This section addresses the Probe Data Vehicle Component (PDVC) of the Probe Data Application. A detailed description of the PDS at the SDN is provided in Section 4.10.

The PDVC consists of five functional elements illustrated in Figure 4-69. The five functional elements are: Snapshot Generation, Buffer Management, Snapshot Transmission, Log Management and Probe Management Directives. These elements operate together to collect data from vehicle systems, compile messages and send the messages under specific conditions to a probe data proxy application at an RSE.

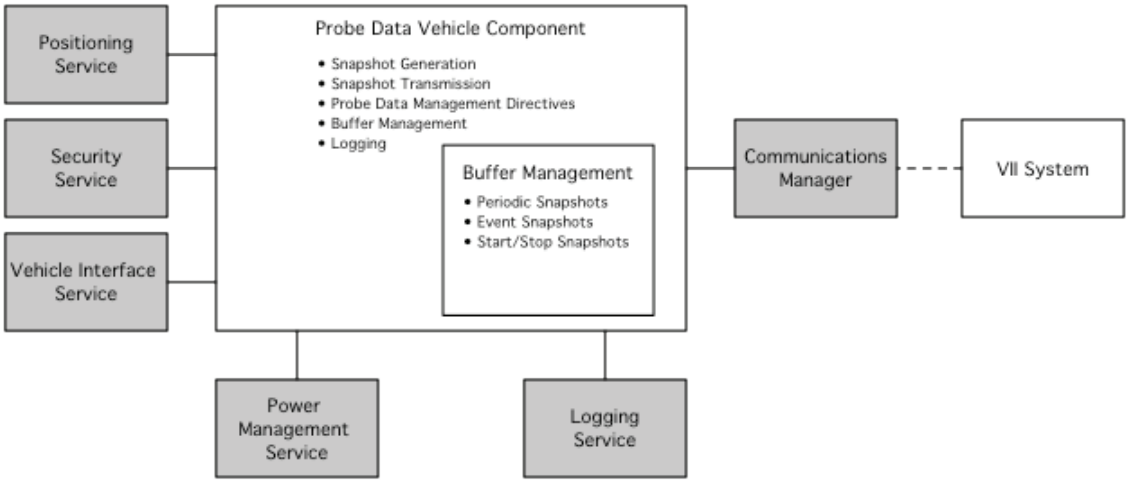


Figure 4-69 PDVC Functional Elements Overview

Snapshot Generation

Snapshot Generation combines data obtained from the VIS and positioning data from the Positioning Service with periodic, event-based or start/stop probe data snapshots based on a programmable data generation policy defining data collection rate, content, etc. The generation policy can be changed via directives from Probe Data Management Directives. The snapshots are then passed to the Buffer Management.

Under normal operation, Snapshot Generation compiles snapshots at intervals based on the vehicle's speed. Where defined by the policy, Snapshot Generation also compiles messages based on specific events in the vehicle such as the activation of traction control measures, braking threshold events, etc. This approach allows the collection of unique events that may have relevance for road maintenance, weather assessment and similar applications.

Buffer Management

Buffer Management receives the snapshots from Snapshot Generation and manages the data store of these snapshots via a configurable data replacement policy. This policy is used to define, for

example, how long a snapshot should remain unsent in the buffer before it is deleted, how long a gap between sets of snapshots taken under a given sequence number should be, and other criteria.

Buffer Management also supports key privacy policies that can have an important impact on the collection of probe data. For example, to avoid the ability to track a vehicle from one RSE to the next, the policy prohibits sending snapshots with the same PSN at two different RSEs. This protects the vehicle's privacy, but it also means that any data that is not sent in an RSE encounter is lost. For example, if the radio link is lost due to range or interference, the remaining data must be deleted.

VII-Datagram Transport Layer Security (V-DTLS)
Because Probe Data Snapshots contain information about the behavior of the vehicle at locations other than where the data is uploaded to the system, it is important to protect the data from local eavesdropping. This approach prevents, for example, a police officer from intercepting probe data indicating that the vehicle was speeding at some earlier location, linking the anonymous data to the vehicle through observation, and thereby issuing a citation. While this practice might prove to be illegal, encryption of the local radio link was seen as a sure way to avoid such issues from the start, so the concept of a locally encrypted link was introduced.

V-DTLS builds on the well-known Datagram Transport Layer Security (DTLS) system used in the Internet. Upon entering the radio coverage zone of an RSE, The Probe Data Application receives a unique identifier from the RSE, for example, the IP address of the RSE. The Probe Data Application then uses this identity to encrypt the Probe Data Message, and signs it using the OBE Application's (IEEE P1609.2) anonymous certificate. Since the IP address of the RSE is always sent in the WSA, this approach requires no additional steps in the setup of the secure link. The key to this approach is that the RSE has developed a private key that reverses the encryption done by the OBE using the RSE identity as a key. This method is not as secure as full blown asymmetric keys, but it is highly efficient as it avoids the time required for a complete secure key exchange, and it requires no compromise of the OBE's anonymity. For purposes of preventing local eavesdropping on data exchanges that will, eventually become public, this approach was seen as a good compromise between security and efficiency.

4.7.4.2 Probe Data Application Flow of Events

Preconditions
1. The PDVC has registered with the OCM to receive service notifications when PDCS are advertised from an RSE.
2. At least one RSE is set up with a Probe Data Proxy Application and is announcing the PDCS.
3. The PDS is active at the SDN.
4. A Network User has subscribed to one or more of the parameters collected in the vehicle through the PDS.

Flow of Events
1. As the vehicle travels over a distance of 2 km the PDVC Application collects various operating parameters from the vehicle via the VIS, and position and time from the Positioning Service, and compiles these into stored Probe Data Snapshots that share a common PSN.
2. The vehicle approaches the coverage zone of an RSE that is announcing PDCS.

3. The Communications Manager coordinates with the V-DTLS client on the RSE to set up a secure, anonymous communications session.
4. The Communications Manager notifies the PDVC Application that the PDCS is available.
5. The PDVC Application combines the stored snapshots sharing the common PSN into a series of Probe Data Messages.
6. The PDVC Application passes the Probe Data Messages to the V-DTLS element on the OBE for encryption.
7. The V-DTLS element encrypts the messages and transmits them to the RSE using the OBE DSRC Radio.
8. If the PDVC Application does not complete sending all of the stored snapshots with the same PSN, the remaining snapshots with that PSN are deleted.
9. The V-DTLS client on the RSE receives the Probe Data Messages, decrypts them and passes them to the PDC Proxy application.
10. The PDC Proxy application passes the Probe Data Messages through the network to the PDS at the SDN.
11. The PDS parses the probe data messages and passes each vehicle parameter to the Publish-And-Subscribe element of the PDS.
12. The Publish and Subscribe element of the PDS sends messages to the Network user containing only those parameters that they subscribed to.

4.7.5 In-Vehicle Signage Application

The POC In-Vehicle Signage Application is designed to provide broadcast advisory messages to the vehicle driver based upon location and situation relevant information. Messages are prioritized both for delivery and presentation based on the type of advisory. These messages may be in the form of text, graphics, or audio cues presented by the generic vehicle HMI developed for all POC applications.

As shown in Figure 4-70, the In-Vehicle Signage Application is composed of two major components: the Network User-based application component, referred to as the Signage NUC that generates advisory messages, and the Vehicle-based application component referred to as the Vehicle Signage Component (VSC) that presents advisory messages to the driver.

Two NUCs were used in the POC. The Traveler Information NUC that generates traffic and incident information and the Signage NUC that generates Next Exit and Work Zone advisory messages. These advisory messages are utilized by the VSC to inform the driver of current traffic conditions.

In addition to the two application components, the infrastructure system supports the Signage Application with three subsystems. The AMDS located at the SDN receives messages submitted by the NUCs and, based on the delivery instructions for the message, distributes the advisory messages to the appropriate RSEs. An AMDS Proxy running on each RSE accepts the messages from the AMDS and causes them to be broadcast by the RSE according to the delivery parameters for the message. The NUC also obtains location information about the RSEs from the ILS located at the SDN.

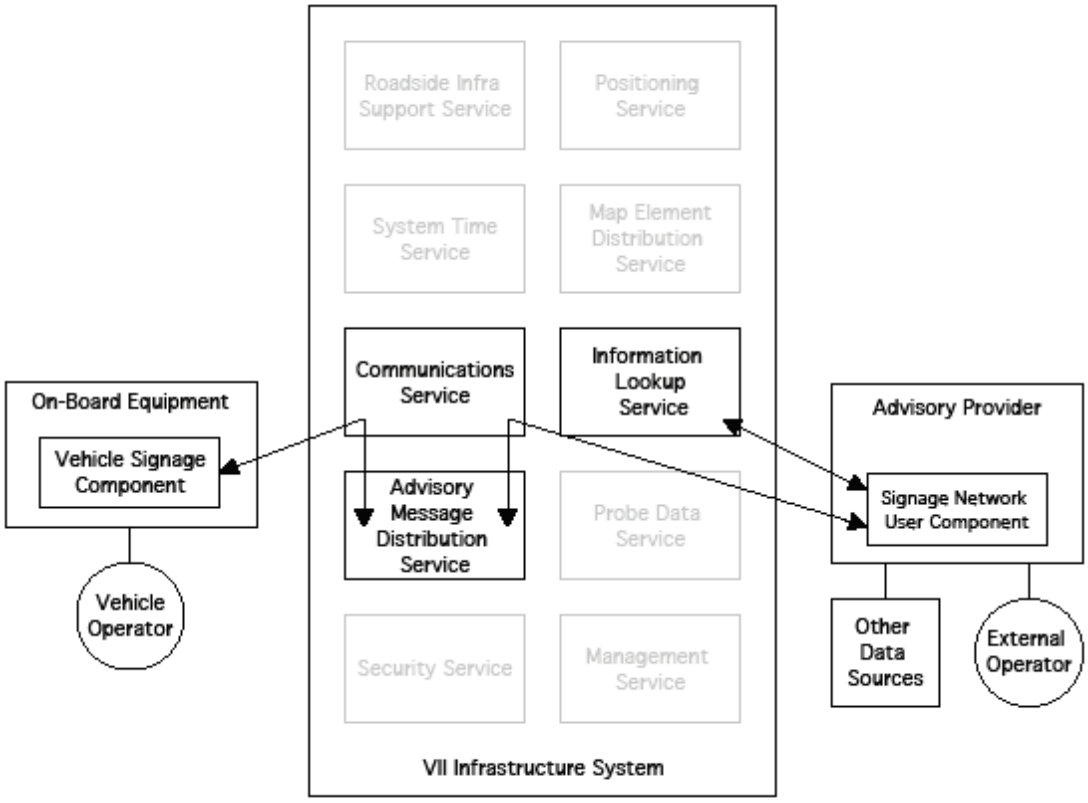


Figure 4-70 In-Vehicle Signage Application System Overlay Diagram

4.7.5.1 POC Signage Application Architecture

The overall POC architecture of the Signage Application is shown in Figure 4-71. Advisory messages (signs) originate with a Network user. These signs are generated from data derived from a variety of sources including the VII Probe Data system. They messages are compiled locally by the Network user using the Society of Automotive Engineers (SAE) J2735 standard format. These messages are then submitted to the AMDS along with delivery instructions. The delivery instructions indicate what RSEs the messages should be broadcast from, the priority of the message, details about the frequency of broadcasts and the duration for which the message should be broadcast. The AMDS then distributes the messages to the appropriate RSEs and the RSEs broadcast the messages in their local region according to the delivery instructions.

When the message is received by the VSC the specific display details (such as when and where the advisory message should be displayed) are extracted and the message is displayed when the vehicle situation meets the display conditions.

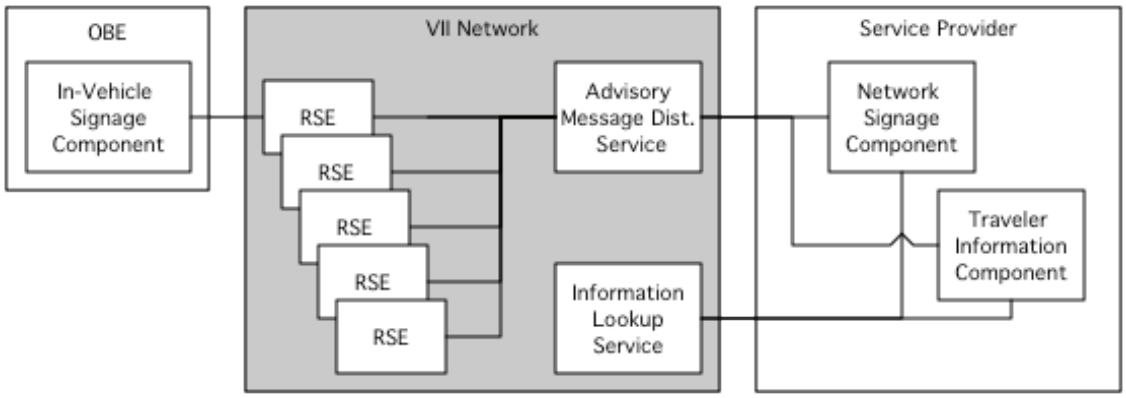


Figure 4-71 POC Signage Application Architecture

4.7.5.1.1 Signage Network Component

As illustrated in Figure 4-72, the Network Signage Component consists of 3 functional elements: Communications, Advisory Message Generation and Map Data Store. The remaining sections detail the functional and external interface requirements for each of the functional elements.

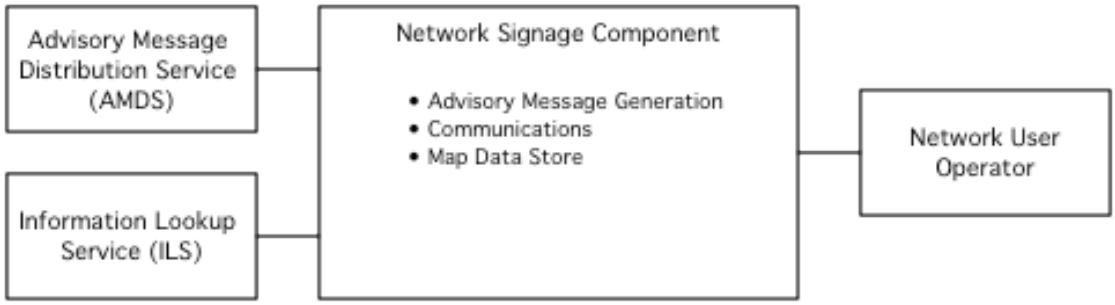


Figure 4-72 Network Signage Component Functional Elements Overview

Communications

Communications authenticates the Network Signage Component with the AMDS. It receives advisory messages from the Advisory Message Generator and forwards them to the AMDS.

Advisory Message Generation

Advisory Message Generation creates advisory messages and authenticates the Network Signage Component with the ILS. It uses RSE location information received from ILS queries to create the delivery instructions for the messages. The advisory messages and their delivery instructions are forwarded to Communications.

Map Data Store

The Map Data Store is a local table of information obtained from the ILS at the SDN. It contains the addresses for RSEs corresponding to specific geographic locations, and is used to allow the Network User Operator (signage provider) to geographically target messages simply by using network addresses.

4.7.5.1.2 Vehicle Signage Vehicle Component

The VSC consists of four functional elements as illustrated in Figure 4-73. The four functional elements are: Advisory Message Management, Next Advisory Message Determination, Presentation Management, and Log Management. The remaining sections detail the functional and external interface requirements for each of the functional elements.

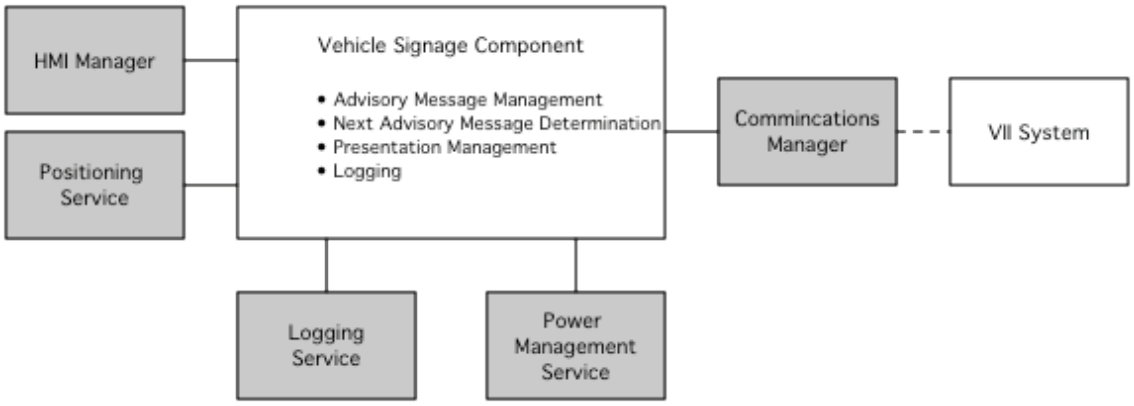


Figure 4-73 Vehicle Signage Component Functional Elements Overview

Advisory Message Management

The Advisory Message Management receives advisory messages from various NUCs, verifies that the advisory messages are not expired, duplicated or unsupported and stores valid advisory messages in the Advisory Message data store. It is also responsible for managing the Advisory Message data store by removing expired messages and determining what messages need to be replaced when the data store is full. Finally, Advisory Message Management will receive OBE shutdown notifications from the Power Management Service. This notification will be forwarded to the other functional elements to allow the VSC to shut down gracefully.

Next Advisory Message Determination

The Next Advisory Message Determination interfaces with the Positioning Service and uses the vehicle's positioning information in conjunction with the location parameters of the advisory messages to determine when an advisory message should be presented to the driver. The advisory messages that are ready for presentation are forwarded to the Presentation Manager. Next Advisory Message Determination is also responsible for notifying the Presentation Manager that an advisory message no longer needs to be presented due to a change in the vehicle's position.

Presentation Management

The Presentation Management manages the presentation of advisory messages to the driver via the HMI Manager. Typical Signage displays are shown in Figures 4-74 and 4-75.

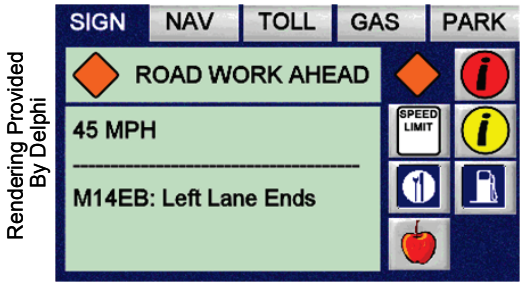


Figure 4-74 Example Road Advisory Display

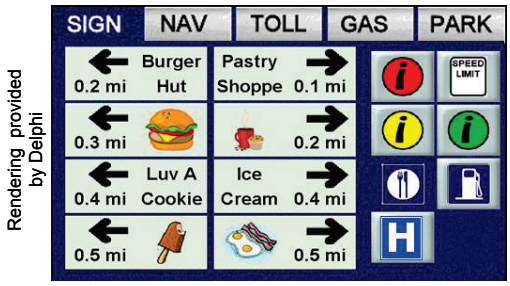


Figure 4-75 Example Next Exit Services Display

4.7.5.2 In-Vehicle Signage Application Flow of Events

This section describes the end-to-end flow of events of the application which consists of generating Signage and Advisory Messages, disseminating Geographically Focused Signage and Advisory Message Information, and presenting Signage Information.

Preconditions

1. NUC has registered with the VII system to broadcast messages. There are two NUCs available for POC: Network Signage and Traveler Information.
2. All available NUCs have been authenticated to use the ILS.
3. A VSC has registered with the OCM to receive advisory messages.
4. The VSC has received a Service Available event from the OCM.

Flow of Events

1. A Network user creates a new advisory message at a NUC.
2. The Network user provides delivery instructions for each advisory message based on identifiers for RSEs within the geographic area where the messages are to be disseminated.
3. The NUC sends the advisory message with delivery instructions to the AMDS at the SDN.
4. The NUC receives an acknowledgment as to the ability of the AMDS to forward the message to the identified RSEs.
5. Each RSE sets up its transmission playlist according to the advisory message delivery instructions.
6. Each RSE transmits announcements and messages on the appropriate DSRC channel according to the RSE operations and the advisory message delivery instructions.
7. The vehicle containing the VSC moves into range of an RSE.
8. The VSC receives the advisory messages from the RSE.
9. The VSC assesses the message for relevance.
    a. The VSC verifies that the advisory message is not a duplicate using the Packet ID and the issue time of the advisory message.
    b. Duplicate advisory messages are discarded.
    c. The VSC discards advisory messages with unknown or unsupported Packet IDs.
10. The VSC manages the data store of advisory messages.
    a. The VSC adds relevant advisory messages to its advisory message data store.
    b. The VSC updates relevant advisory messages in its Advisory Message data store based on updated information received from the NUCs.

c. The VSC deletes expired advisory messages from its data store.
d. The VSC determines what advisory messages are to be deleted and deletes them to make room for new messages. Messages are logged before they are deleted

11. The VSC uses the vehicle's positioning data and the advisory message presentation location information to determine when a message needs to be presented to the driver.
12. The VSC forwards advisory messages that need to be presented to the driver via the HMI Manager.
13. The HMI Manager presents the advisory message to the driver.
14. The VSC cancels advisory messages that are being presented to driver when they are no longer applicable.

4.7.6 Trip-Path Application

The Trip-Path Application is intended to collect information about how vehicles move and use the road network. By knowing where vehicles start and end their journeys and what roads they use to get from A to B, road management authorities can better understand the nature of road demand and better plan for changes, new additions and improvements. Because of privacy considerations, Trip-Path is an "opt-in" application, meaning that only those users who choose to participate have the application operating in their vehicle.

Figure 4-76 shows how the Vehicle TPGA fits into the VII POC architecture.

In a manner similar to PDC, the Trip-Path client in the OBE collects and saves location information at various intervals as the vehicle moves. When the vehicle completes a trip, the entire trip is saved, thereby capturing the origin, route and destination information. On the next run cycle, the trip data is uploaded to a Network user application that simply captures and stores the trip information. In the POC, no effort was made to analyze or use the Trip-Path data.

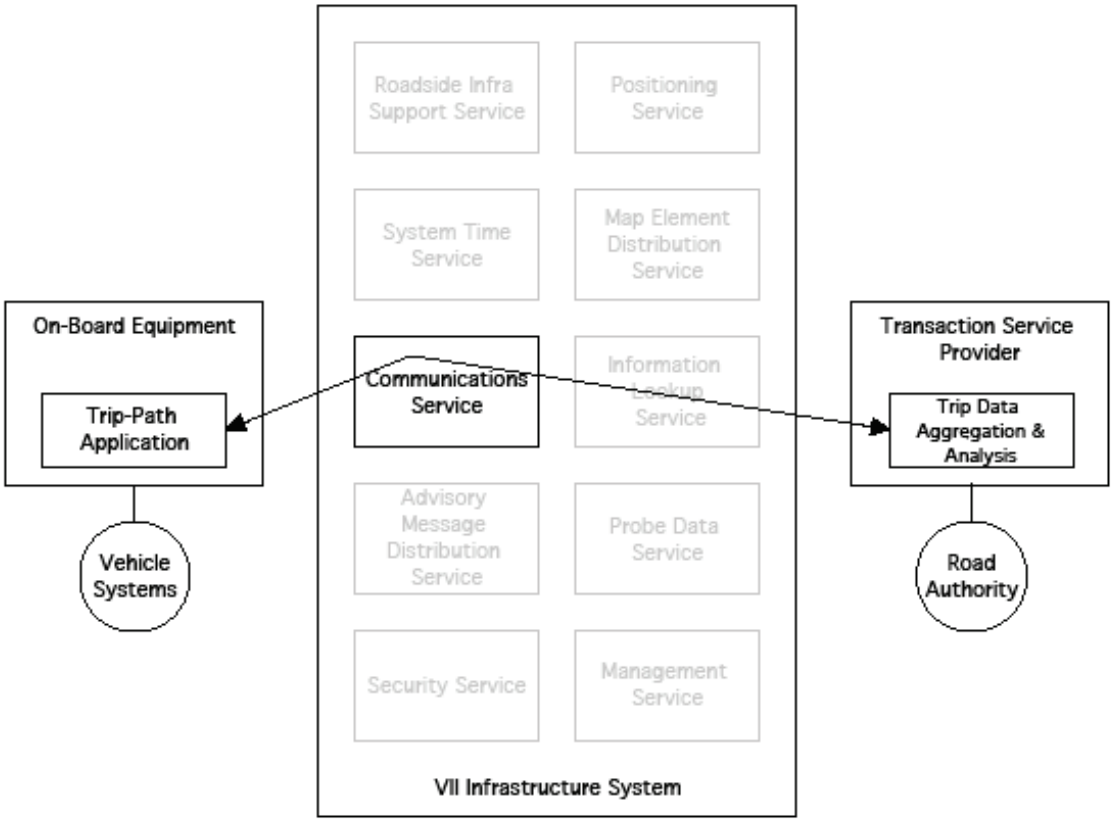


Figure 4-76 Trip-Path Application System Overlay Diagram

4.7.6.1 POC Trip-Path Application Architecture

As illustrated in Figure 4-77, the Trip-Path General Application (TPGA) consists of four functional elements: Trip-Path Collection, Buffer Management, Trip-Path Transmission (TPT), and Log Management.

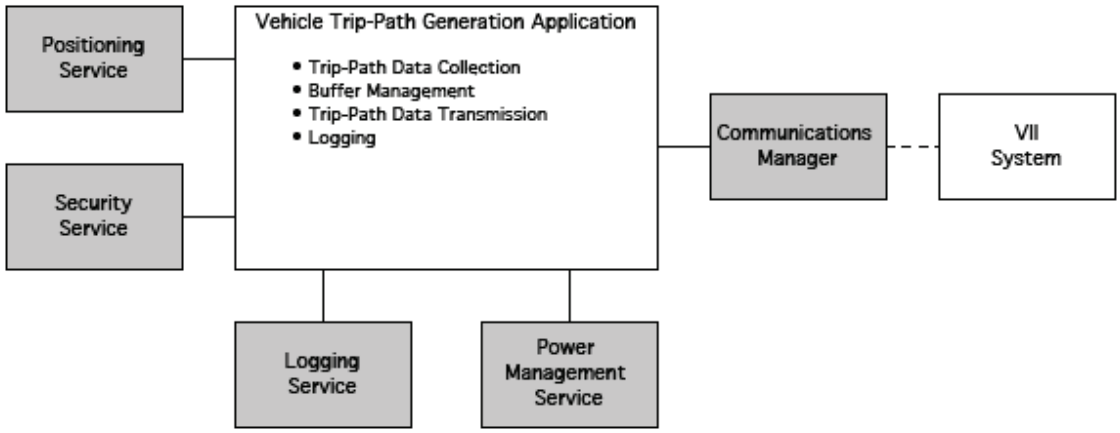


Figure 4-77 Vehicle Trip-Path Generation Functional Elements Overview

Trip-Path Collection
Trip-Path Collection records data associated with two different types of events:

- The path traveled from ignition on to ignition off with a duration of no more than six (6) hours, or
- A maximum of 4000 Trip-Path data points.

A new trip starts when either of these conditions has been met. In other words, the vehicle may collect multiple trips between ignition off and ignition on, and each trip can have no more than 4000 Trip-Path data points with the duration between the first and last Trip-Path data points not exceeding 6 hours.

The Trip-Path information is recorded as data points are generated using a configurable time and/or distance intervals. To preserve user privacy, Trip-Path Collection begins recording Trip-Path data points only after the vehicle has traversed a configurable distance from the vehicle's ignition on location. This eliminates the potential for the Trip-Path data to indicate a common location where the vehicle starts every trip (e.g. home).

Buffer Management
Buffer Management manages the data store of trips, and when it has a trip to deliver to the Trip-Path Data Accumulator Application during an RSE encounter, it then delivers the data to Trip-Path Transmission.

Trip-Path Transmission
Trip-Path Transmission (TPT) sends Trip-Path data to the Network Trip-Path Data Accumulator (NTPDA) Application via the OCM.

When TPT receives notification from the Communications Manager, it exchanges handshake messages with the NTPDA and sets up a secure session. It then sends the data in segments. To assure data transmission integrity, the NTPDA and TPT exchange acknowledgements before sending the next segment, so if data is lost or garbled due to the RF connection, the TPT resends it.

TPT deletes all information associated with a trip from the Buffer Management data store upon receipt of an acknowledgement of a successful transmission of the trip message from the NTPDA.

4.7.6.2 Trip-Path Application Flow of Events

This section describes the end-to-end flow of events of the application which consists of Generating Trip-Path records and uploading them to the Trip-Path Network Component.

Preconditions
1. The TPGA has registered with the OCM to receive service notifications when Trip-Path Collection services are advertised from an RSE.
2. At least one RSE is set up to announce Trip-Path Collection Services.
3. The conditions for a new trip have been met (path traveled from ignition off to ignition on with a duration of no more than six (6) hours, or a maximum of 4000 Trip-Path data points have been taken).

Flow of Events

1. When the vehicle starts, the TPGA begins collecting location points from the OBE Positioning Service.
2. The TPGA discards position data points until the distance from the start of the trip to the current position exceeds 2 km. Once this threshold is reached, Trip-Path Generation begins to record the position data points at regular intervals.
3. The vehicle travels over the course of the trip, and reaches its destination.
4. The vehicle ignition is turned off.
5. The vehicle ignition is turned on at some later time.
6. On key-on, the TPGA begins a new trip log. It examines the records of the prior trip log and discards records corresponding to the 2 km immediately preceding the end of the trip (arrival at the key-off destination).
7. The vehicle travels over the course of the new trip and encounters an RSE that is advertising Trip-Path Collection Services.
8. The Communications Manager notifies the TPGA that the Trip-Path Collection Service is available.
9. The Vehicle TPGA communicates with the Network Trip-Path Component to set up a secure data exchange.
10. The Vehicle TPGA sends a collected Trip-Path data record.
11. The Network Trip-Path Component acknowledges receipt of the data record
12. Steps 10 and 11 continue until all data has been sent.
13. The Vehicle TPGA deletes all sent Trip-Path records.

4.7.7 Off-Board Navigation Application

The POC OBNA provides turn-by-turn navigation cues to the vehicle driver with updates based upon location and situation-relevant information while the vehicle is en route. These cues are in the form of text or graphics such as driving instructions displayed on the HMI screen and/or audio cues from the Vehicle HMI.

The routes calculated by the OBNA can be enhanced by link travel time information collected by the VII Probe Data system and other external sources. One objective of the application is to determine the route with the shortest travel time for a designated vehicle. As the vehicle travels, updates to the route are provided through the VII system. If circumstances indicate that an alternate route will give the driver a shorter travel time, then the recommended route is changed to the newly determined route with shorter travel time.

The overall system context for the OBNA is shown in Figure 4-78. This figure illustrates that a route request originates at the OBE, and is communicated through the TSM (See Section 4.7.7.1) to the navigation service provider.

The OBNA is used to test, among other system functions, the ability of the system to route messages through the system to a remote Network user, and to maintain a service transaction session between the OBE and the Network user as the vehicle moves between RSEs. This allows the system to provide very long and detailed route information that might, under certain circumstances, be impossible to communicate in a single session as the vehicle moves through an RSE coverage zone.

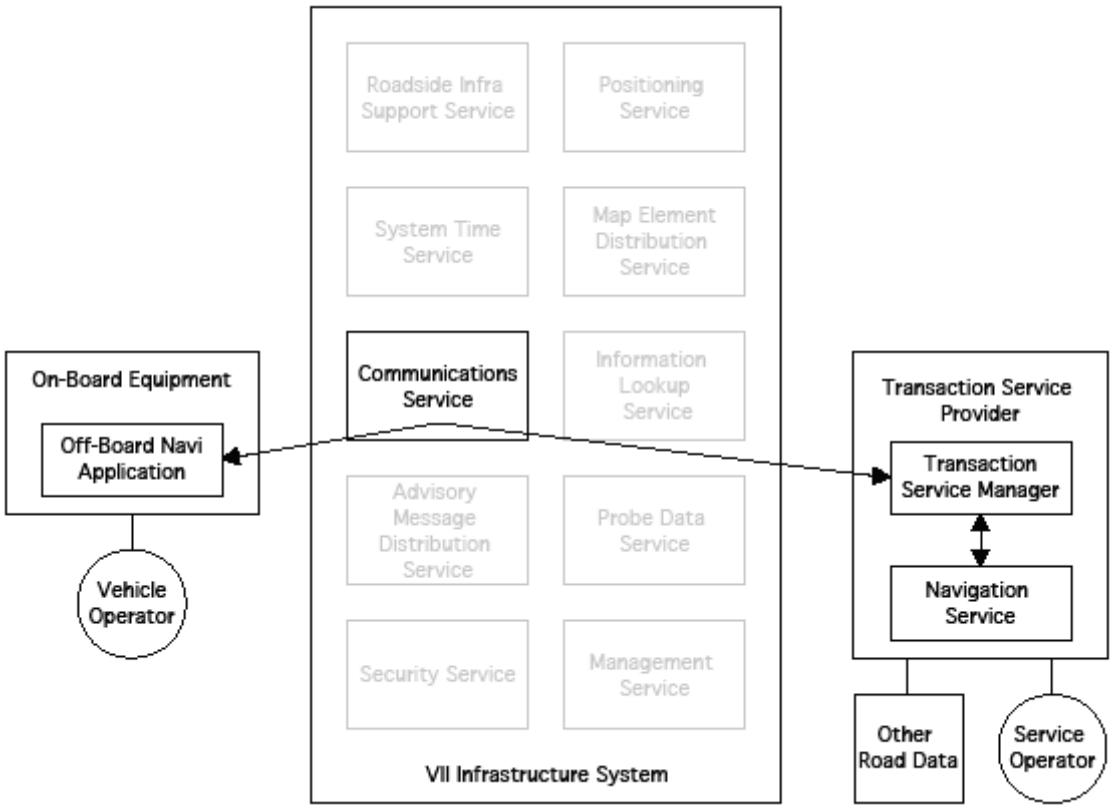


Figure 4-78 OBNA System Overlay Diagram

4.7.7.1 POC OBNA Architecture

As shown in Figure 4-79, the OBNA is composed of two major application components: a Network Component that manages the transactions and calculates navigation routes for all requesting vehicles, and a Vehicle Component that resides on the OBE to allow the driver to request a route and to display the results of the request (the route) to the driver.

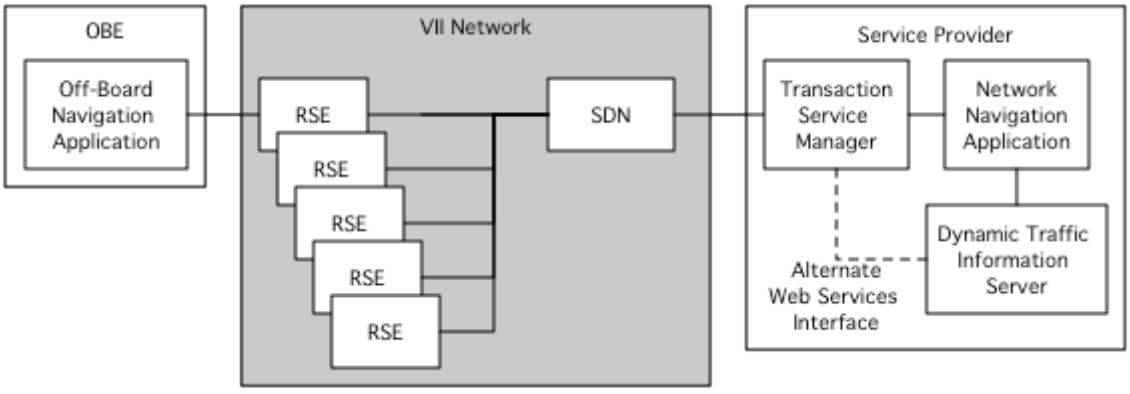


Figure 4-79 OBNA Functional Component Diagram

In the POC, the TSM was located at the SDN (See Section 4.10), and the Navigation Service components were located at the Navteq facility in Chicago. Communications from the TSM to the Navteq system were via the Internet.

The TSM was described in detail in Section 4.5.3.2.2. The following sections describe the Network Component and the Vehicle Component.

4.7.7.1.1 Off-Board Navigation Vehicle Component

As illustrated in Figure 4-80, the OBNA Vehicle Component consists of five (5) functional elements: Route Data Management, Next Maneuver Determination, Presentation Management, Off-Route Detection and Log Management; however, Off-Route Detection was not implemented in the POC.

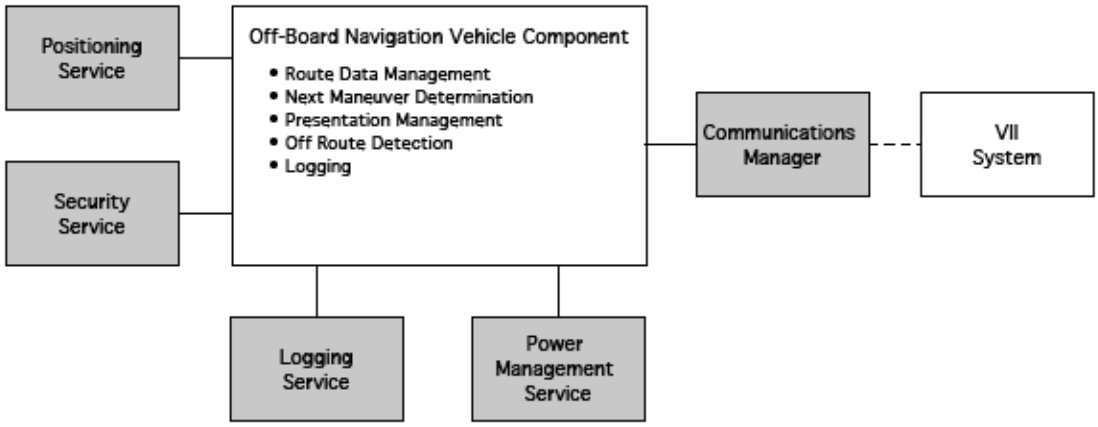


Figure 4-80 Vehicle Component Functional Elements

Presentation Management Element

Presentation Management provides the software interface between the OBNA Vehicle component and the HMI Manager. It is used to present available destination options to the driver and to obtain the driver's destination selection. It also provides data relating to the results of route search received from the Network Component. These results may be in the form of turn-by-turn directions or in the form of a route overview map. The Presentation Management screens (via the HMI Manager) also allow the user to scroll through multi-page route files.

Route Data Management Element

Route Data Management sends route requests to, and receives route responses from the OBNA Network Component. It extracts turn by turn maneuver information from the route responses and stores the instructions in the Route Data store. Because setting destinations represents a rather complex user interface problem (one that was not the focus of the POC program) the OBNA Vehicle Component is based on a set of pre-stored destinations. This eliminates the need for a specialized user interface to allow the user to enter a specific destination. It is assumed that a production implementation would address this issue in whatever way the developer saw fit.

Route Data Management thus supports a maximum of ten (10) predefined destinations that can be selected by the driver from a list presented on the HMI display. The predefined destinations are updated via the OBE interface to a USB External Memory Device.

In operation, Route Management sends predefined destination information to Presentation Management for display on the vehicle HMI. An example of the destination display is shown in Figure 4-81.

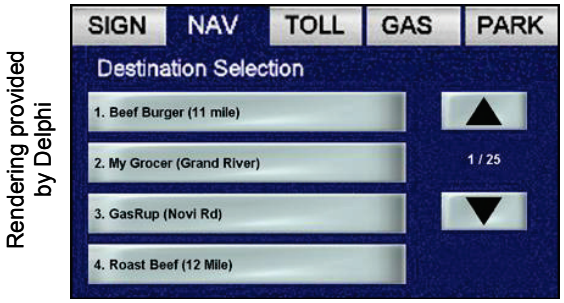


Figure 4-81 Example Destination List

When the driver selects a route from the list, Route Management receives the destination information from Presentation Management, and combines it with the vehicle's location at the time of request to form a Route Request. Route Management then sends this route request to the OBNA Network Component via the OCM at the first interaction with an RSE that supports communications with the TSM.

When Route Data Management receives Route Response Maneuvers from the OBNA Network Component via the OCM over one or more RSE encounters, it stores all route maps and maneuver information and assembles a complete route information set. At this time, it provides the Route Response Preview to the Presentation Management for display on the HMI. A typical Route Response Preview screen is shown in Figure 4-82.

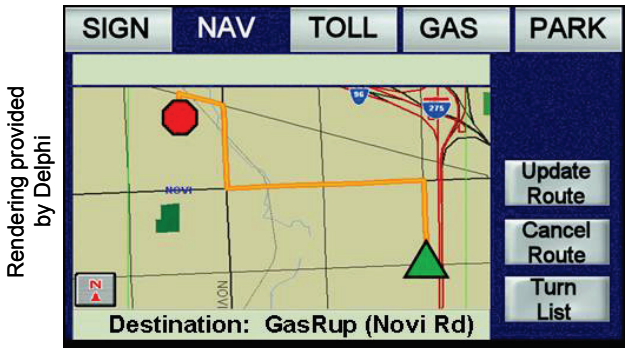


Figure 4-82 Route Overview Screen

Next Maneuver Determination Element

When the Presentation Manager is displaying the maneuver list, the Next Maneuver Determination uses the vehicle's positioning information to determine when the next maneuver on the route list should be highlighted. When the vehicle is close to the maneuver point, the Next Maneuver Determination passes a maneuver display to Presentation Management for display. So, as the vehicle moves along the route the next maneuver in the route list is presented to the driver. An example Maneuver diagram is provided in Figure 4-83.

Figure 4-83 Maneuver Diagram Display

Log Management Element
Log Management is used to log events and operations of the OBNA Vehicle Component for diagnostic and testing purposes.

4.7.7.1.2 Off-Board Navigation Network Component

As illustrated in Figure 4-84, the OBNA Network Component consists of seven (7) functional elements: Communications, Geo-Coding, Route Calculation, Direction Generation, Map Image Generation, Map Database Management and Log Management.

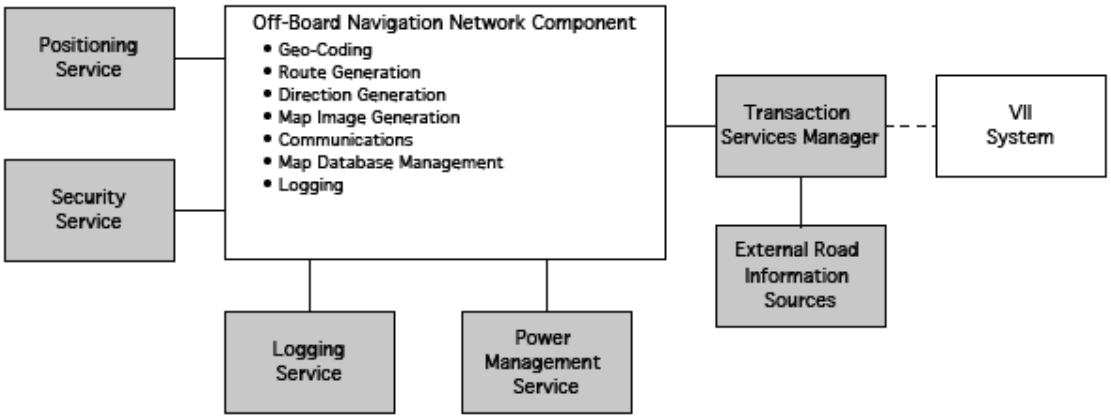


Figure 4-84 Off-Board Navigation Network Component Functional Elements

Communications Element
Communications receives route requests from the OBNA Vehicle Component, and sends a route response, using information provided by other Network Component functional elements, to the OBNA Vehicle Component in return. All communication with the Vehicle Component is done via the TSM. Route requests are forwarded to the Geo-Coding element for further processing.

Communications also receives dynamic traffic information from external sources. This information may come from other services orchestrated by the TSM, or it may come from a direct source. For the POC, the direct source was used. Dynamic traffic information is forwarded to Map Database Management for further processing.

Geo-Coding Element
Geo-Coding converts route requests containing destination addresses and the geographic (latitude/longitude) position of the vehicle into road network elements that correspond to the Map Database used by Map Database Management. This is a necessary step since the Map Database uses a special road segment format to create a road network that only indirectly corresponds to geographic positions or street addresses. Once the conversion is complete the Geo-Coded destination and route origin and destination are passed to the Route Calculation Element.

Route Calculation Element
Route Calculation computes a route with the shortest travel time, taking advantage of current dynamic link information.

Route Calculation receives Geo-Coded origin and destination information from Geo-Coding. It then interacts with Map Database Management to compute the route. This process uses well established algorithms to search the road network database to identify and compare various paths through the road network to get from the origin (the vehicle's current location) to the destination. Part of the comparison process involves using travel time for various road segments to determine the combination of roads that results in the shortest possible overall travel time. The computed route at this point consists of a sequence of road segment identifiers. These are passed to Direction Generation to create a human usable list of maneuvers.

Direction Generation Element
Direction Generation creates turn-by-turn maneuver information for a route based on the road segment list generated by Route Calculation. This element uses a combination of pre-stored maneuvers (for example "Bear Right," or "Turn Left", etc) and detailed information about the intersections of the road segments to create a textual sequence of maneuvers. These are then sent to the OBNA Vehicle Component via Communications and the TSM.

Direction Generation also passed the turn-by-turn direction list to the Map Image Generation element which creates the various map images used by the vehicle system.

Map Image Generation Element
Map Image Generation creates several map images corresponding to the various components of the route information package. Specifically, it creates a Route Overview map showing the entire route from start to finish as shown in Figure 4-85. It creates segment maps showing the route over smaller higher resolution portions of the route, and it generates individual diagrammatic maps images showing each specific maneuver in the turn list, as shown in Figure 4-86. Map Generation passes these map image files to Communications which forwards them to the OBNA Vehicle Component for display to the driver.

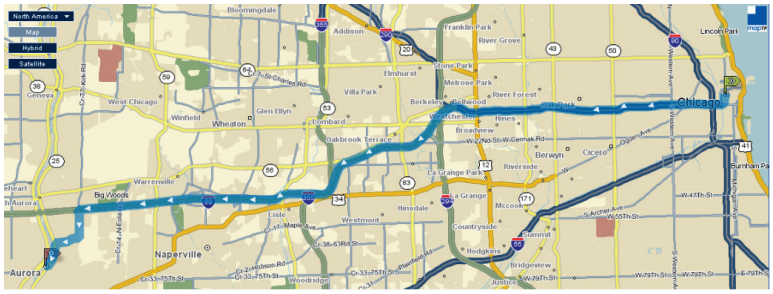


Figure 4-85 OBNA Overview Map

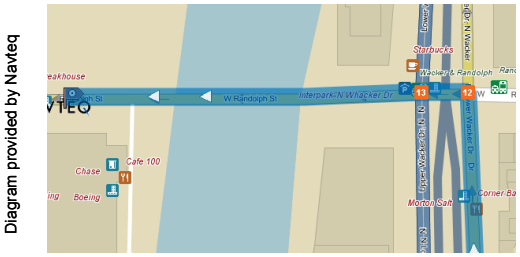


Figure 4-86 OBNA Maneuver Map

Map Database Management Functional Element
Map Database Management updates the map database with current dynamic link information received from the VII Traveler Information Application and Dynamic Link Data Provider(s) (e.g. MDOT's Data Use Analysis and Processing Service).

4.7.7.2 Off-Board Navigation Application Flow of Events

The Off-Board Navigation flow of events represents the nominal case operational flow for the VII OBNA.

Pre-conditions

1. The OBNA Vehicle Component has registered with the OCM for the VII system Communication Service.
2. The OBNA Vehicle Component has a subscription with the OBNA Network Component.
3. The Vehicle Component is preconfigured to know the URL of the Network Component.
4. A Dynamic Link Data Provider has up-to-date information in its store needed to generate dynamic traffic information.
5. The OBNA Network Component has subscribed to receive specific information from the Dynamic Link Data Provider(s) via the TSM.
6. The OBNA Vehicle Component has a list of pre-set destinations.

Flow of Events

1. Using the OBE HMI the driver activates the OBNA.
2. The OBNA Vehicle Component presents a destination list to the driver using the vehicle HMI.
3. The driver selects a destination from the displayed destination list.
4. Driver is notified that route guidance will not be available until the vehicle comes within communication range of an RSE that supports communications to the TSM.
5. The vehicle comes within communication range of an RSE that supports communications to the TSM.
6. The OBNA Vehicle Component sends a request for route guidance to the OBNA Network Component, including user identification, current position and selected destination.
7. The OBNA Network Component determines the route with the shortest travel time using current dynamic link information.
8. The OBNA network component sends route guidance information to the OBNA Vehicle component over one or more RSEs.
9. In the event that the vehicle leaves the RSE communications zone before the OBNA Network Component responds, the scenario proceeds as per event 10 through 14.
10. The TSM determines that the OBNA Vehicle Component is not responding, and saves the messages from the OBNA Network Component.

11. The Vehicle enters the communications zone of another RSE.
12. The OCM communicates with the TSM, and re-establishes the transaction session using the IP address of the new RSE.
13. The TSM re-sends the (remaining) route guidance information to the OBNA Vehicle Component via the new RSE.
14. The driver is presented with the route guidance information.

4.7.8 Heartbeat Application

The Heartbeat Application generates and sends Heartbeat messages at a configurable rate, and logs all Heartbeat messages received from other sources. The Heartbeat message is a WSM defined by the SAE J2735 standard. It contains vehicle position and speed and a few other vehicle related parameters.

The intent of the POC Heartbeat Application was not to test the utility of the Heartbeat message itself (as a safety element) but to test the ability of DSRC to support high-rate high-priority messaging in the presence of other DSRC uses. Figure 4-87 shows how the Vehicle Heartbeat Generation Application fits into the VII POC architecture.

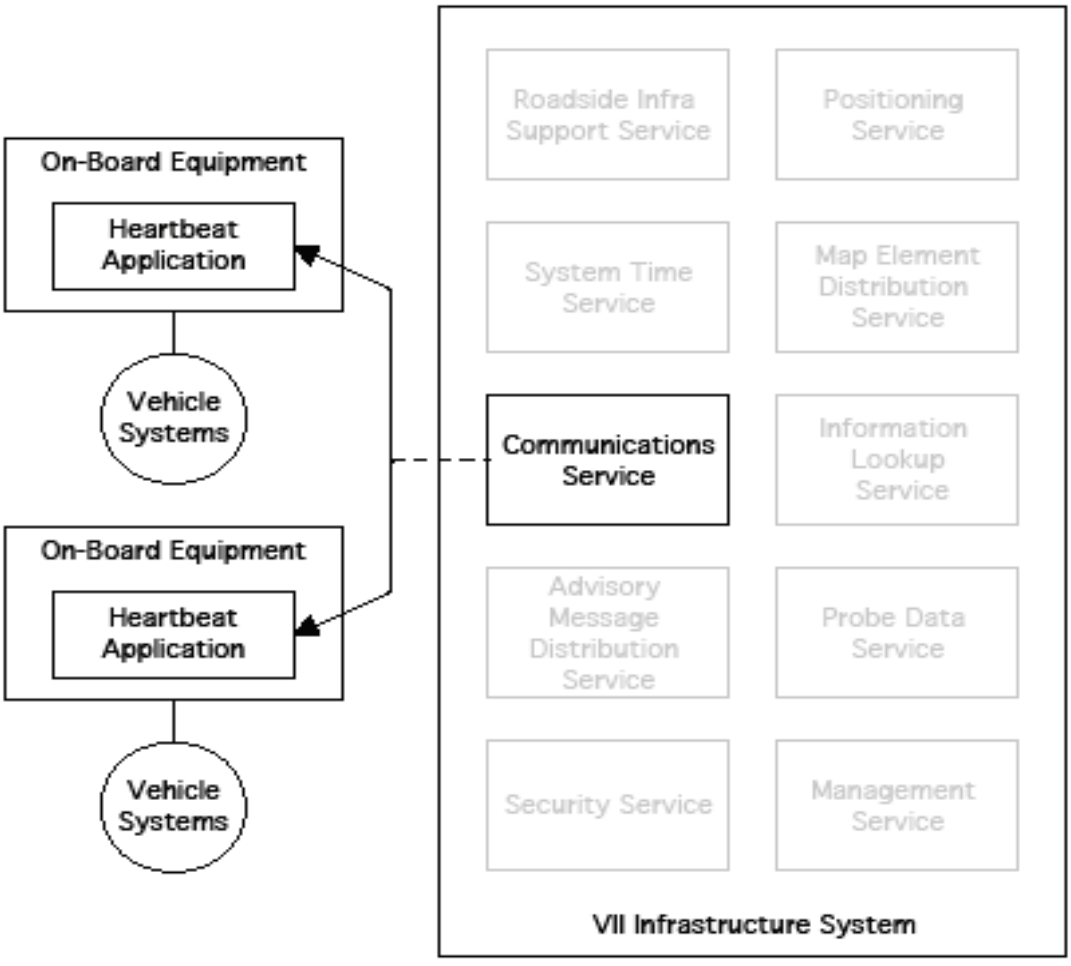


Figure 4-87 Vehicle Heartbeat Generation Application System Overlay Diagram

4.7.8.1 POC Heartbeat Application Architecture

The Heartbeat Vehicle Component (HBVC), as shown in Figure 4-88, consists of three elements, Heartbeat Generation, Heartbeat Transmission and Logging:

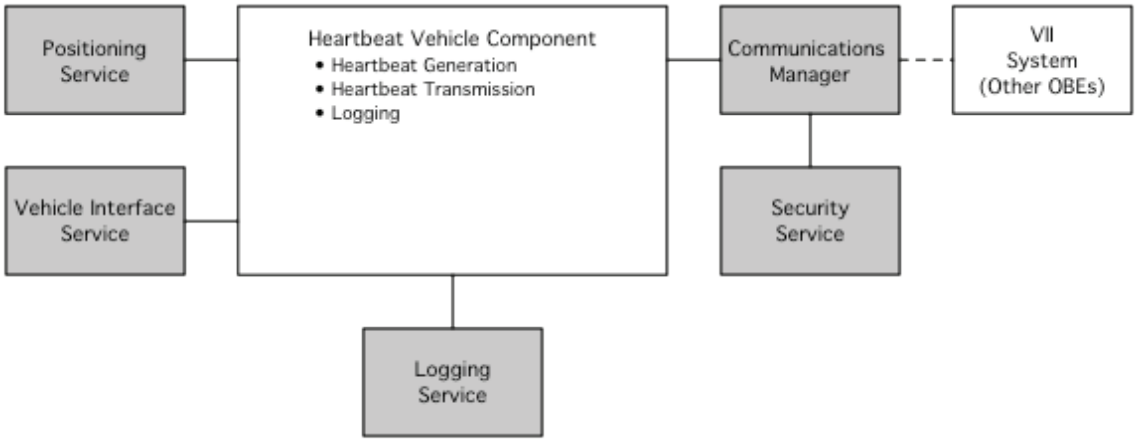


Figure 4-88 Heartbeat Vehicle Component Functional Elements Overview

Heartbeat Generation Element

Heartbeat Generation combines vehicle sensor data from the VIS and positioning data from the Positioning Service into the periodic Heartbeat message. The generation policy may be changed by changing configuration parameters.

This data is collected on a regular schedule and compiled into a Heartbeat WSM message in accordance with the content and format defined in SAE J2735. The snapshot will be generated with whatever data is provided by the API including null or zero values.

The compilation schedule is set by a configuration parameter and it may be set to any rate from zero up to 50 Hz (one message every 20 ms).

Heartbeat Transmission Element

Heartbeat Transmission passes generated Heartbeats to the Communications Manager for broadcast using the DSRC Radio.

Heartbeat Transmission logs all messages, sent and received, for the purposes of analysis, debugging and testing, and deletes the messages after they are logged.

4.7.8.2 Heartbeat Application Flow of Events

The following flow of events describes each step executed by the Heartbeat Application during normal operation.

Preconditions

1. The Heartbeat Application in Vehicle A has registered with the OCM to send and receive Heartbeat WSMs.
2. The Heartbeat Application in Vehicle B has registered with the OCM to send and receive Heartbeat WSMs. Optionally, additional vehicles running the Heartbeat Application may also be present.

3. Vehicles A and B are in DSRC Radio range of each other.

Flow of Events

1. The Heartbeat Application in Vehicle A collects data from the Positioning Service and Vehicle Interface and compiles a Heartbeat Message.
2. The Heartbeat Application in Vehicle A passes the Heartbeat message to the Communications Manager in Vehicle A.
3. The Communications Manager in Vehicle A optionally signs the message using the OBE Security Services, and submits the Heartbeat message to the DSRC Radio for transmission.
4. The DSRC Radio in Vehicle A transmits the message when the DSRC channel is clear.
5. The DSRC Radio in Vehicle B receives the Heartbeat message, and passes the message to the Communications Manager in Vehicle B.
6. The Communications Manager in Vehicle B optionally verifies the received message using the Vehicle B OBE Security Services.
7. The Communications Manager in Vehicle B passes the verified message to the Heartbeat application in Vehicle B.
8. The Vehicle B Heartbeat Application logs the receipt of the message and discards the message.
9. Vehicles A and B repeat this operation, each serving as both sender (Vehicle A above) and receiver (Vehicle B above) on a schedule set by an internal configuration parameter for each vehicle.

4.8 Network Description

The infrastructure network is shown schematically in Figure 4-89. This figure primarily illustrates the internal structure of the RSE, and the SDN. Many details have been omitted for clarity. A more detailed description is provided in Volume 2b.

The SDN is composed of interfaces to the Backbone (to other SDNs), the backhaul (to RSEs) and the Access Gateway (to Network Users), routing functions to properly direct messages traffic and a set of core services.

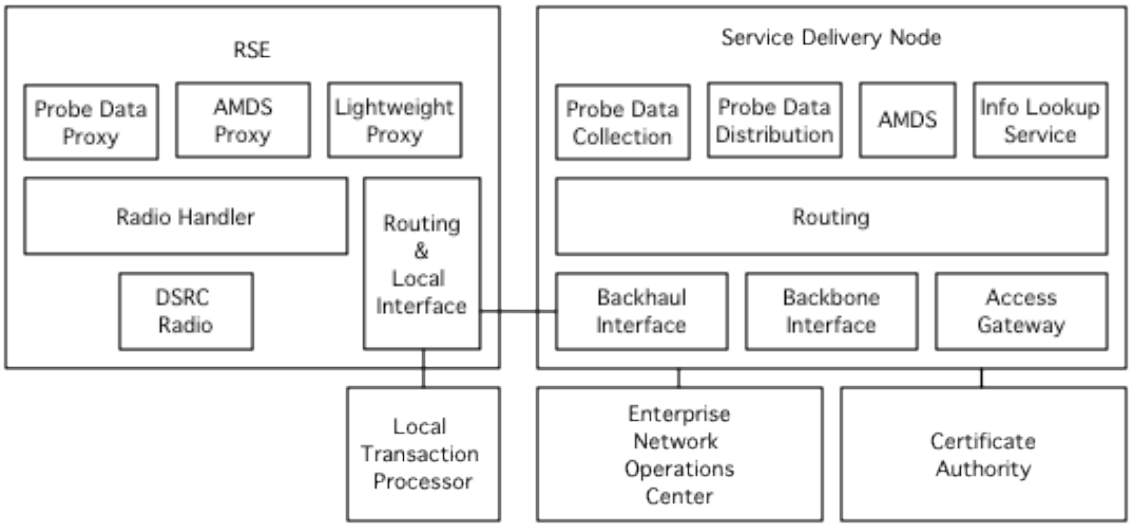


Figure 4-89 Infrastructure Side System

The POC network core services include:

Advisory Message Delivery Service (AMDS) accepts submitted messages from Network Users via the Access Gateway. These messages include delivery instructions such as RSE ID(s), repeat timing and message lifespan. The AMDS then passes these messages to the AMDS proxies resident in the appropriate RSEs for local broadcast to OBEs in the vicinity.

Probe Data Collection Service (PDC) interacts with the Probe Data Proxy in the RSE to accept a stream of probe data messages gathered as OBEs pass the RSE. The PDC then passes this data to the Probe Data Distribution Service (PDDS).

Probe Data Distribution Service (PDDS) accepts a stream of probe data messages from the PDC. It then parses these messages and places the various content elements (different probe data parameters such as speed, vehicle status, events, etc) into queues that are structured along these topical categories. The data in these topical queues is then sent via the Access Gateway to Network users that have established subscriptions on the basis of these topics. So, a network user that subscribes to Topic A at locations X, Y, and Z will receive any data associated with Topic A that is collected at any of the specified locations.

Information Lookup Service (ILS) is a support service used by Network users to determine information about the system. It is most often used to identify RSEs according to location so that a subscriber or provider can then properly reference the RSE.

Certificate Authority (CA) is a central point of trust in the system. The CA provides certificate to OBEs that attest to the authenticity and legitimacy of an OBE for use in singing both identified and anonymous messages, and provide certificates to other users to allow them to exchange signed and encrypted messages with OBE applications.

Other Services such as a Map Element Generator (MEG), a Map Element Distribution System (MEDS), Network Identity and Access Management Service, but these are outside the scope of this discussion.

The RSE is composed of the DSRC Radio subsystem, a routing function, and a set of proxy applications that extend the services residing at the SDN (described above) out to each RSE associated with that SDN. The proxies essentially pass messages to and from their counterpart SDN services and interface to the RSE radio subsystem. The radio subsystem includes a DSRC Radio, and a Radio Handler that accepts or sends messages from/to the various proxies. The radio handler also constructs or updates a play list that contains all broadcast messages to be transmitted.

Depending on the situation, an RSE may be connected to a LTP. This may be, for example, a local tolling system, or a traffic signal controller. In operation the LTP sends and receives messages to OBEs and to Network Users through the RSE functions. These messages usually have local relevance (as in tolling or signals) and thus need to originate local to the RSE.

The ENOC is used by system operators to control and manage the overall network and RSE suite.

The CA issues security credentials to elements of the system that require them. It also manages the overall security state of the system.

4.9 Roadside Equipment

The RSE is a self-contained unit installed at a given location along with the appropriate backhaul equipment. It acts as the gateway between the vehicle and the rest of the infrastructure. The RSE announces the services offered by the network and passes data between vehicles and network users.

The Radio portion of the RSE operates in the 5.9 GHz DSRC band with the IEEE 802.11p, IEEE 1609, and SAE J2735 communications stack. The following communications protocols are supported by the DSRC Radio:

- UDP
- WSMP
- IPv6

The backhaul connection from the RSE to the Michigan SDN is standard IPv4 and IPv6. The RSE is also equipped with GPS for self-positioning and providing Position Corrections to vehicles.

When deployed, security will enable the RSE to authenticate and validate vehicles, maintain a vehicle certificate revocation list and maintain its own certificates.

4.10 Service Delivery Node

The SDN houses the core service infrastructure of the VII system. The SDN contains server platforms, data stores, and software systems that support VII system data distribution and communications services. The SDN provides logical interfaces for RSE and network application connectivity. Traffic destined for or originating from RSEs will utilize multiple types of wired or wireless backhaul communications technologies interconnecting RSEs to their associated SDN. The SDN also provides connectivity for a number of public and private network user applications. These applications provide support for distribution of public probe data, generation of maps, dissemination of positioning correction information, advisory message delivery and dispatching, network management and Security Services. The VII network is made up of multiple Service Delivery Notes (SDNs), each of which represents a logical entity for the set of interfaces and routing functions that provide connectivity and ingress/egress traffic flow to the VII network. VII network traffic is distributed and flows from one SDN to another SDN across a backbone network.

4.11 Certificate Authority

The CA structure for the VII system is shown in Figure 4-90.

This structure identifies five types of CA:

1. Identified OBE Certifying Authority
2. OBE Authorizing Authority (OAA)
3. Anonymous OBE Certifying Authority (AOCA)
4. Infrastructure CA
5. Root CA

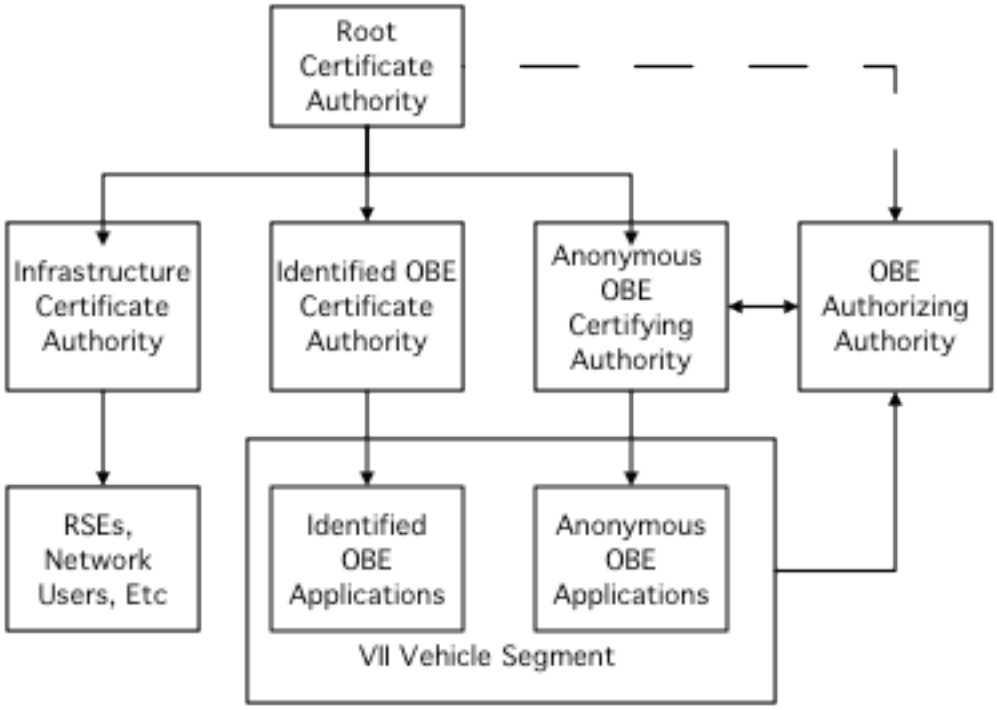


Figure 4-90 CA Structure

These CAs are shown as separate logical entities for clarity. It is assumed that they might be combined and/or regionally distributed to optimize system performance. The roles and responsibilities of identified CAs (Infrastructure, Identified OBE and Root) are well defined in many security standards, and these will not be discussed here other than to point out that the Identified OBE CA will issue certificates that are used by the OBE to encrypt and authenticate transactions that rely on identification as the key element of legitimacy and assurance. These types of transactions typically include purchases and/or transactions where the two parties have established a trusted identified relationship (e.g. a service provider and a vehicle with an established account with that provider). It is also important to point out that the various lower level CAs need to tier to a single root authority so that certificates from users, RSEs, etc., can be verified by the vehicle security systems, and vice versa.

The VII system architecture uses two different types of security credentials:

- **Identified credentials** are keys and their associated certificates that can be verified as belonging to a specific user. These keys are typically used for encrypted transactions where both parties know and trust each other (for example, making a purchase using a personal account). By properly nesting encryption processes, the identity of an individual user can be protected from eavesdroppers or from the system transporting the message between the parties. They are also used to sign messages from identified sources, for example, when a service provider identifies itself or an individual user signs a transaction.
- **Anonymous credentials** are keys and their associated certificates are used for signing messages that are broadcast to all local users. Since broadcast messages are intended for all receivers, the purpose of the security functions is to assure the receiving parties that

the message is legitimate. The VII system uses a pool of keys and corresponding certificates that are shared by many users. Each anonymous application will draw keys from an (application-specific) group pool. The process is designed to prevent any linkage between the identity of the user and the particular keys provided to that user/application. As a result, any application using these keys to sign messages will be impossible to differentiate from other users using that same key. To further protect identity, each anonymous user application will draw multiple keys from the pool and will rotate them randomly. The result is that anonymously signed messages will all appear to have no single traceable source (since many of them have exactly the same security credentials).

Since anonymous keys are used by many different users, the question arises as to what is actually being certified when the CA issues credentials to a user. Since the security credentials are only used by the security function, it is possible to prevent their use (e.g. by encrypting them or locking them) unless the vehicle system is able to prove that it has not been in some way tampered with or changed. Using this approach, an anonymous signature certifies that the message was sent by a vehicle system that was able to pass the tampering / legitimacy test. While this approach was developed conceptually during the POC program, there was insufficient time to implement or test this concept, and it remains a future task to determine what extent of test should be required, but the mechanism for this process is part of the current VII system design.

Since there is always the threat of key compromise, the certificates associated with keys (identified or anonymous) are designed to have a finite lifetime. As certificates expire, the security functions in the vehicle will replace them through secure transactions with the CA.

Of particular relevance to this document, are the roles of the OBE Certifying and Authorizing authorities. These entities are central to the means for managing anonymous certificates that are used when the receiving parties are not trusted, and when the assurance does not rely on identification.

To preserve anonymity, the VII system uses a shared pool of certificates and keys (credentials). Since these credentials need to be regularly updated (replaced as they expire or are revoked), the OBE must include a mechanism for requesting and managing credentials. However, since any credential provisioning transaction must be encrypted, it is difficult to prevent the entity providing these credentials from knowing the identity of the vehicle requesting them. The process and structure described in Figure 4-91 is intended to provide this desired anonymity.

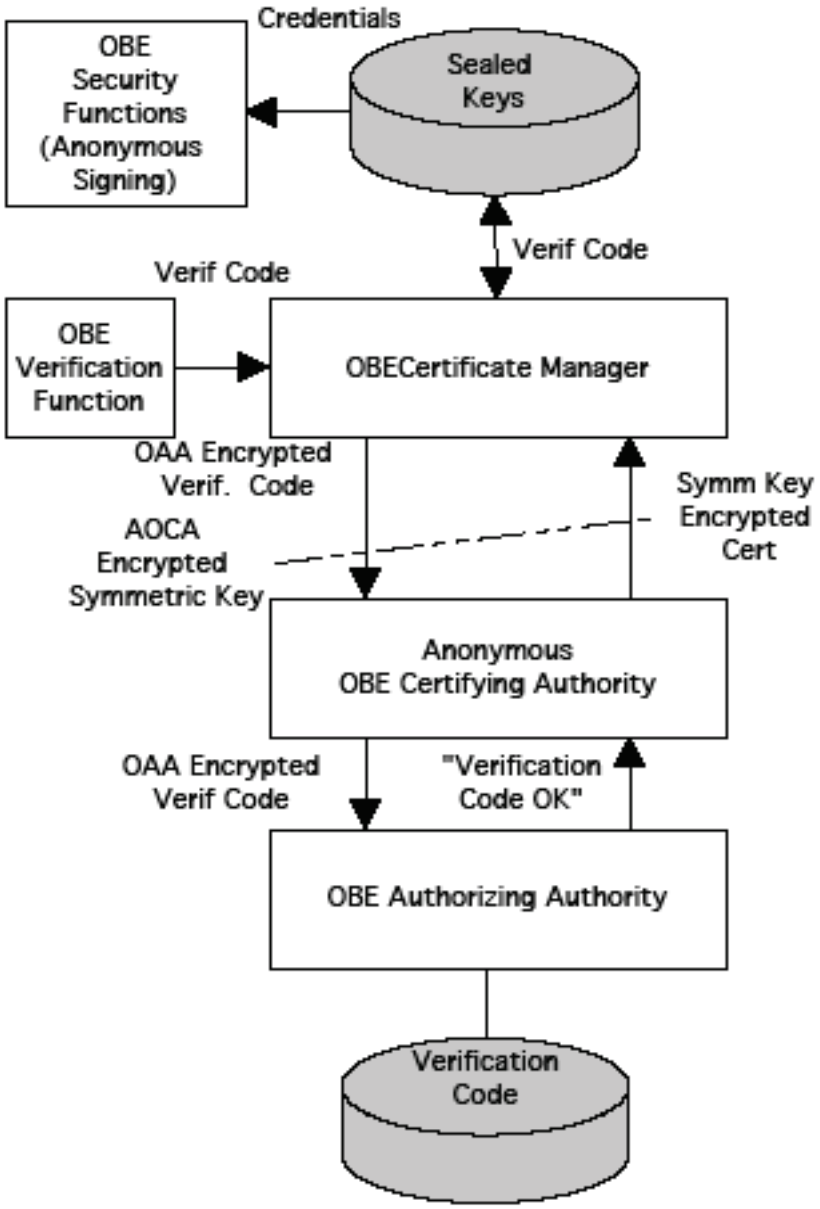


Figure 4-91 Anonymous Certificate Management

The anonymous certification process operates as follows:

The OBE CM sends a request to the AOCA. This message includes an OBE Verification Code representing the current physical and software state of the OBE. This is encrypted with the OAA's public key. The message also includes a symmetric key encrypted with the Anonymous OBE Certifying Authority's public key. This means that the AOCA is unable to determine any information about the OBE requesting the credentials.

The AOCA assigns a temporary ID to the request and passes it in its encrypted form to the OAA. The OAA decrypts the request and verifies that the OBE Verification Code is correct (i.e., that it

is entitled to request these credentials, and that it has not been somehow tampered with), and sends an authorization to the AOCA.

The AOCA then randomly selects the credentials from the anonymous pool, encrypts them using the symmetric key provided by the OBE, and sends them to the requesting OBE.

Using this system, the OAA knows that a particular identified OBE requested credentials, and it has determined that that OBE is legitimate (via its OBE Verification Code). However, while the OAA knows the identity of the OBE, it does not know which certificates were supplied to that OBE. Similarly, while the AOCA knows which certificates were issued, it does not know to which OBE they were issued (since the temporary ID is not maintained). Since the OAA and AOCA are separate entities, there is no linkage between the certificates provided to the vehicle by the AOCA and the OBE identity.

As a result, the OBE, upon proving that it is legitimate to the OAA can obtain security credentials that cannot be traced to its identity assuring the recipient that the sender was legitimate.

4.12 Test Track

Formal system testing was carried out under controlled conditions at a test facility provided by Chrysler. The Chrysler facility is shown in Figure 4-92. This track was set up with two RSEs situated so that the footprints would not overlap. The facility was used to refine many of the applications under real world conditions without the risks and inconvenience of operating in live traffic on an open road. The facility was also used extensively to test the system operation with vehicles running at high speed past RSEs, and also to test the positioning system dynamics.

Figure 4-92 Test Track Facility

4.13 Development Test Environment

Figure 4-93 provides an architectural overview of the entire DTE including components residing in Herndon, VA and in the Michigan DTE. The DTE setup includes 55 RSE's, 11 along freeways and 44 along arterials as shown in Figure 4-94. A typical RSE installation is shown in Figure 4-95.

The DTE also included a Michigan SDN and the Michigan Network Access Point (MINAP). The ENOC, located in Herndon VA, monitors and manages all components.

DTE RSE's are connected to the MI SDN via one of three backhaul communications technologies:

- WiMax
- Wireline
- 3G

The following services are provided by the Michigan DTE to vehicles and network users:

- Advisory Message Delivery Service (AMDS)
- Probe Data Service (PDS)
- Information Lookup Service (ILS)
- Communication Service
- Map Element Distribution Service (MEDS)
- Positioning Service (POS)

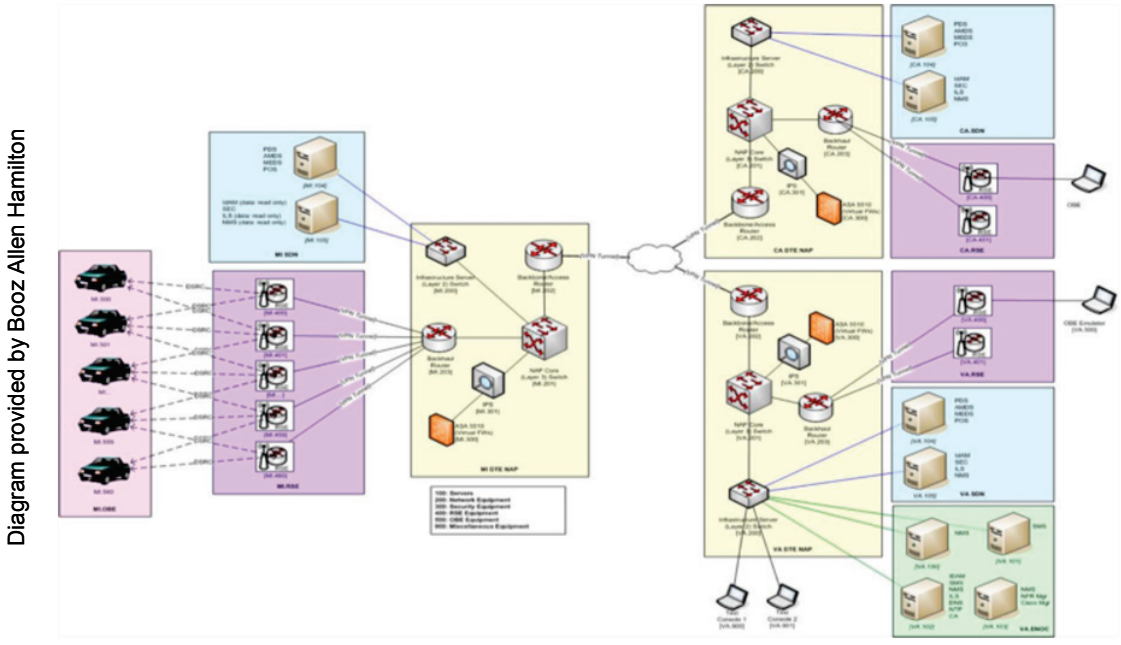
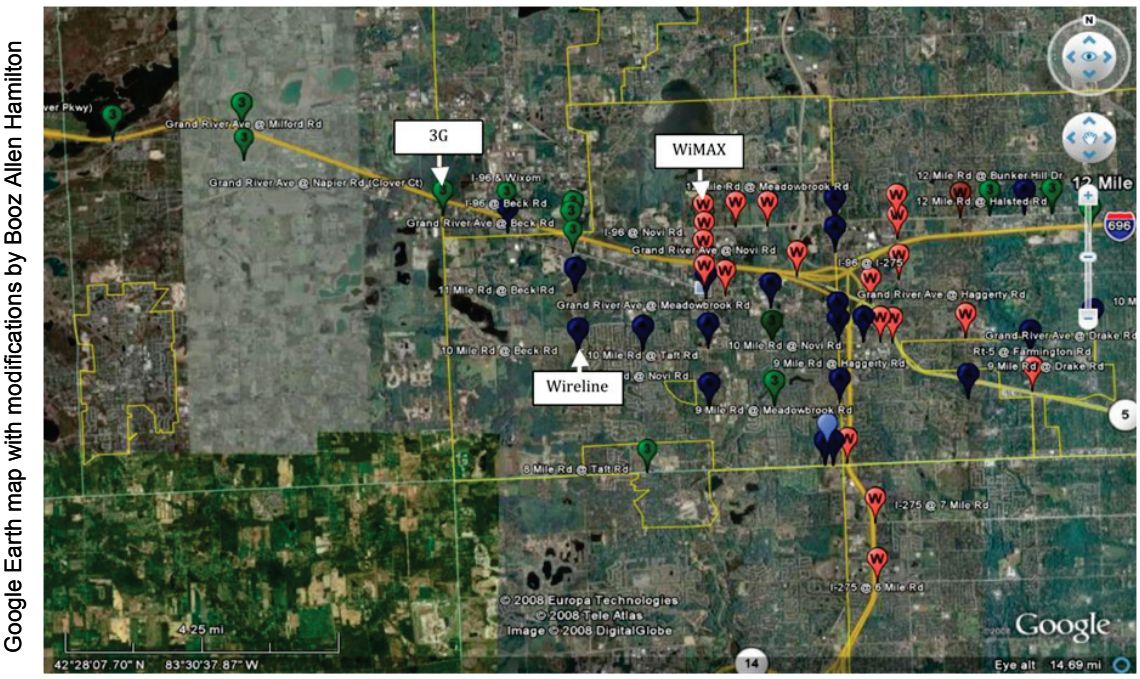


Diagram provided by Booz Allen Hamilton

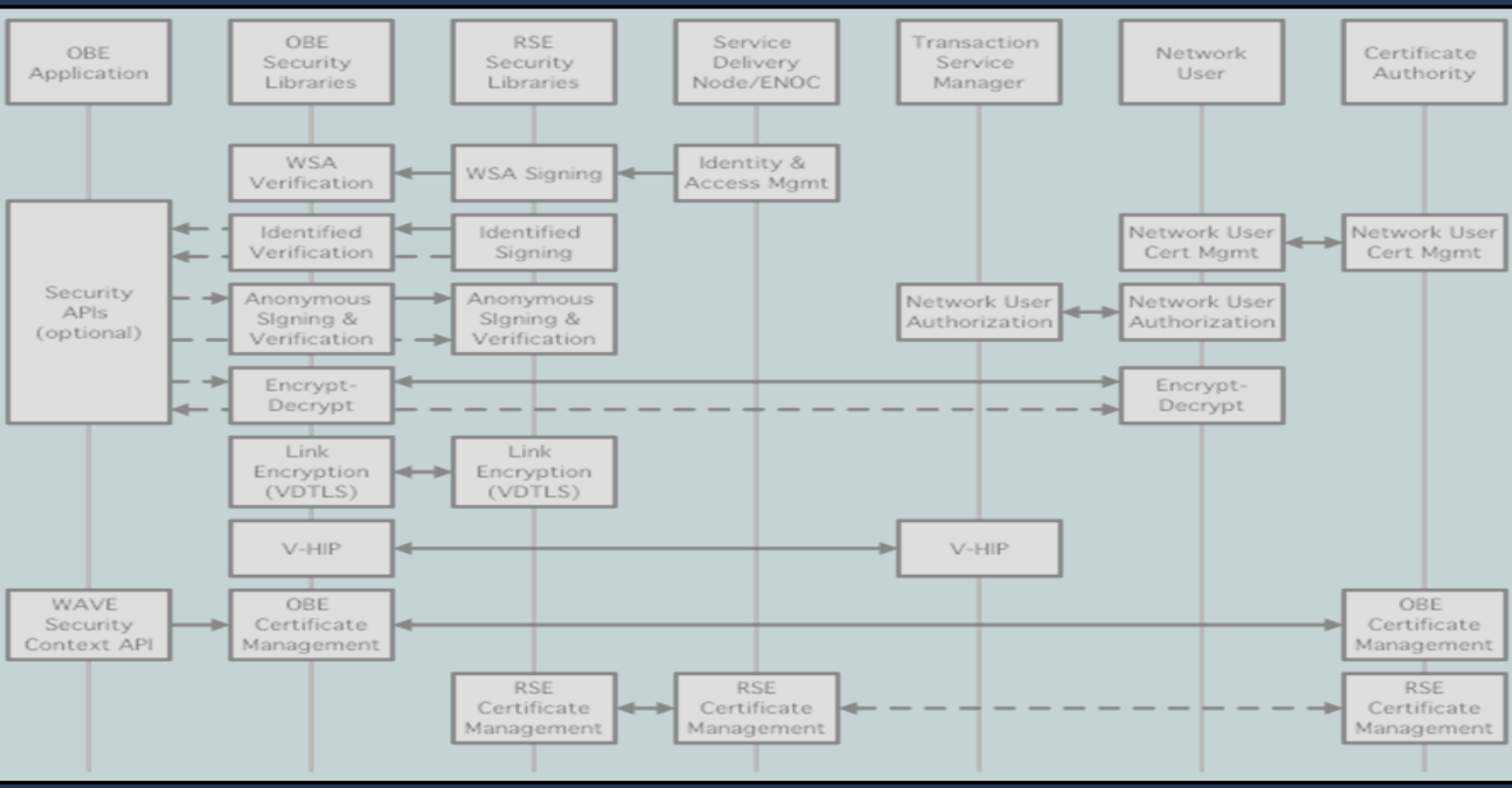
Figure 4-93 Overall VII Network System



Google Earth map with modifications by Booz Allen Hamilton

Figure 4-94 Demonstration Test Environment Map

Figure 4-95 Typical RSE Installation



EDL14458
FHWA - JPO- 09- 017