

# Intelligent Pervasive Middleware for Context-Based and Localized Telematics Services

Chatchsik Bisdikian, Isaac Boamah, Paul Castro, Archan Misra, Jim Rubas, Nicolas Villoutreix, Danny Yeh  
IBM T.J. Watson Research Center, Hawthorne, NY 10532

Vladimir Rasin, Henry Huang, Craig Simonds  
Ford Motor Company, Scientific Research Laboratories, Dearborn MI 48121

## *Abstract*

Telematics is arguably the next-wave in mobile computing: with most cars of the future likely to be equipped with multiple embedded computing platforms, we shall witness the development of a variety of mobile services and applications with significant commercial potential. Telematics will only become a commercial reality when the underlying architecture is able to address significant concerns related to the security and privacy of telematics data, and is able to provide context information from and to a large number of mobile data sources in a scalable and device-independent manner. A telematics platform should utilize existing Internet components and technologies but cannot rely exclusively on these, especially since mobile commerce applications in the telematics environment impose specific requirements on the relationships between various services and data providers. In this paper we describe how we are developing an open standards telematics platform based on the ts-PWLAN wireless service environment and the Telematics Resource Manager middleware. Our design employs existing web service interfaces coupled with novel technology for connecting to these through a wireless gateway. Our middleware acts as a common substrate for building and deploying a wide range of telematics applications. We describe how several of these applications are currently being built on our infrastructure.

Key words: Mobile commerce, telematics, web services, wireless networks, TRM, ts-PWLAN

## 1 Introduction

Telematics is arguably the next wave in mobile computing -- cars in the near future will likely be equipped with many embedded computing platforms capable of running general purpose applications. This will spur the development of a variety of mobile services and applications with significant commercial potential. While serious business strategy and technological issues exist today, these problems will be solved given the enormous market potential for the telematics industry. In recent reports, proponents of the industry have predicted that telematics will be a \$41B industry within the next decade [12]. This view is bolstered by current industry trends such as the continuing decline in cost of rugged, small form-factor computer platforms, advances in embedded chip design, the steady deployment of broadband wireless networks, and the growing interest of government regulatory bodies in mandating a telematics platform for both public safety and revenue collecting [13].

Given the convergence of strong market forces and existing, exploitable technology, one would think that a widely available and successful mobile commerce services would exist in cars today. This is not the case. We believe two major technological reasons are:

- *Proprietary deployments* - Many telematics solutions, such as those for fleet management and emergency response, rely on vertical, proprietary networks and closed platforms, creating islands of non-interoperable technology. The value of a telematics solution lies not solely in the embedded platform but in the end-to-end infrastructure and services that support it, and this may be deployed by a variety of manufacturers and service vendors. Clearly, proprietary solutions both exclude third party developers from creating value-added services and force wasteful duplication of effort when creating applications. Mirroring the PC industry and the Internet, the wide acceptance of telematics can only occur after *de facto* standards emerge for a telematics infrastructure around which a critical mass of developers can build a variety of applications.
- *No consensus on connectivity and computational models for mobile platforms* – While standards will be needed, telematics does not require new standards to be developed, but it can reuse existing programming languages,

network protocols, and network-based services. However, in our experience the mere integration of existing technologies is not enough: telematics places its own requirements on existing Internet components and technologies, especially since mobile commerce applications in the telematics environment impose specific requirements on the relationship between various service and data providers. What we require is proper and widely accepted abstractions for application developers when dealing with mobile, highly dynamic clients on heterogeneous wireless networks. These clients will access a range of data providers and have different requirements for connectivity and quality of service.

In this paper, we present our on-going work in designing and building a telematics platform that can be used to deploy a variety of mobile commerce applications. Our platform, which utilizes open standards interfaces and existing technologies for providing web services, provides a flexible and extensible technological platform to build many telematics applications.

The remainder of this paper is as follows. In Section 2 we characterize the space of telematics applications we would like to address with our approach. In Section 3 we provide an overview of our component technology; specifically, we describe the connectivity and middleware components that make up our telematics infrastructure. In Section 4 we describe three scenarios we are currently implementing on our platform and report on the status of our implementation. We present related work in Section 5.

## 2 Categories of Telematics-based Commerce and Technical Implications

In this section, we describe the main categories and types of telematics applications, along with their commercial and technological implications. In particular, we wish to understand the various constraints that business or regulatory concerns impose on the architecture of telematics applications. We shall later see how the Telematics Resource Manager (TRM) middleware platform is designed to support the data-flow models of various telematics applications under these constraints.

### 2.1 *Architectural Characteristics*

The telematics platform on the car is likely to be closely controlled by the vehicle manufacturer and could differ across different manufacturers and brands. Each brand may be associated with a specific Telematics Service Provider (TSP), who determines the application suite and services installed on the vehicular computer and the type and format of vehicular data that can be exposed to external (Internet-based) applications. Retaining control over the application suite is important to ensure that user interfaces do not lead to dangerous forms of driver distraction (e.g., all Web applications may be constrained to be voice-based while a car is in motion). Providing configuration control over vehicular data exported to external applications is important to satisfy many critical concerns associated with data privacy and confidentiality<sup>1</sup>.

The communication infrastructure between the vehicular platform and the backbone TSP infrastructure is likely to utilize commercial wireless packet-networking technologies. Perhaps more significantly, users will benefit from a range of telematics applications only when the vehicle possesses technologies for both wide-area ubiquitous (e.g., cellular or satellite) coverage and also short-range localized hot-spot communications. As an illustration of this trend, we can witness the development of 3rd-generation (3G) packet-based cellular networks [15][16] aiming to offer about 100Kbps bandwidth at vehicular speeds. Localized, higher bandwidth hot-spot interactions via a vehicular computer are also likely to gain increasing traction with the recent allocation by the FCC of spectrum in the 5.9 GHz range for Dedicated Short-Range Communications (DSRC) [10] based on the IEEE 802.11a radio technology standards. The availability of multiple communication technologies will lead to multiple forms of telematics services and applications. Wide-area technologies are likely to be utilized for interaction with the designated TSP, which may provide an appropriate gateway to manage transport-layer connections and security between the TSP domain and the mobile on-board computing devices. To use these services, users must establish appropriate subscriber profiles with their TSP. Short-range services will often be based on public RF access technologies such as 802.11 or Bluetooth, which typically have no subscription requirements, and use unlicensed portion of the wireless spectrum.

### 2.2 *Application/ Service Categories*

The development of suitable telematics middleware depends not just on the architectural constraints, but also on the target applications. Broadly speaking, we define three distinct categories of telematics applications that present new challenges in

---

<sup>1</sup> In a well-reported case, a driver was fined hundreds of dollars by a rental car company that monitored his usage and charged him each time he exceeded the posted speed limit.

data management and mobile computing. Figure 1 shows the three categories of applications and their typical mode of interaction with the networking infrastructure.

### 2.2.1 Vehicle Diagnostic Services

Applications related to automatic retrieval and remote monitoring of vehicular diagnostic information appear to have the greatest promise in the short term. Preemptive monitoring of vehicular diagnostics is an interesting “win-win” telematics category that offers significant advantages to both the consumer and the vehicle manufacturer. Early detection of faults and failures not only improves consumer safety and reduces the incidence of unexpected problems, but can also provide significant savings, possibly billions of dollars, to manufacturers by avoiding the need for costly maintenance and part replacement (especially during the warranty period). Such applications are likely to have a high percentage of customized manufacturer and model-specific components. An easily understood example in this category uses periodic probing to monitor the health of a vehicle, notify the driver of any impending problems and possibly make necessary service appointments at an affiliated, close-by dealer.

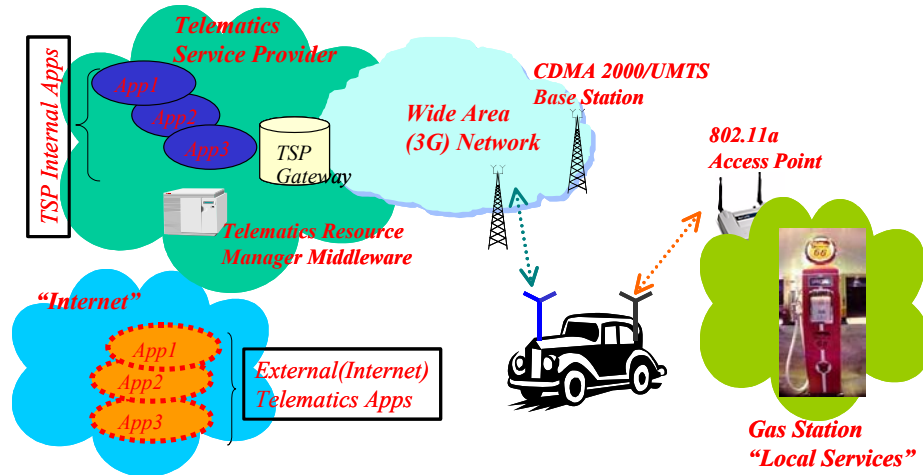


Figure 1. Service categories for telematics

### 2.2.2 Local Hot-Spot Services

We are likely to witness the deployment of a large class of local convenience services that simply treat the vehicular computer as an alternative mobile computing platform (logically equivalent to a PDA, laptop or a cellular phone). The increasing popularity of IEEE 802.11-based wireless LANs (WLANs) in public hot-spots, such as hotels, airports and shopping centers, will allow the car driver to interact with a host of local services. A simple user experience related to this scenario involves a visitor driving into the premises of a convention center equipped with a DSRC-based IEEE 802.11 WLAN. As the user enters the premises, she can discover the resources of the convention center, obtain directions to the designated parking lot, and make associated payments using the WLAN infrastructure. The driver can also navigate to the registration page of the convention hotel and perform self check-in even before stepping out of the car! Another attractive application would allow cars to pull up “near” a fast food outlet and place orders using the WLAN without needing to queue up at the drive-through lane. While such local services do offer significant benefits to the user, they do not appear to provide any direct revenue stream or savings to the vehicle manufacturer.

### 2.2.3 Context-Aware and Event-Based Services

Telematics offers significant new opportunities for the emergence of context-aware applications and services that use vehicular data (such as current location) as sources of context. For example, we have seen articles in the popular press on insurance applications that use actual driving statistics (e.g., highway vs. city, number of miles driven per day, average speed etc.) to specify or modify insurance rates, as well as intelligent yellow-page applications that push advertisements about local resources (such as restaurants, banks etc.) to the car console. It is important to realize that such applications can either be provided by the TSP (as part of a set of subscribed services) or by third-party service providers who can retrieve the appropriate contextual data. Accordingly, the underlying telematics infrastructure must not only allow TSPs to offer their own subscription-based services, but also permit the driver to access additional Internet-based services. While there has been much hype around these application scenarios, initial attempts in this space have often been service-centric and rely on closed architectures and proprietary protocols. We strongly believe that the successful adoption of context-aware pervasive services is possible only when they are built on a common (horizontal) and standards-based middleware infrastructure that allows application providers to develop context-aware services in a device-independent manner. Moreover, the development of such

a middleware framework is not simply a matter of integrating current Internet technologies, but also requires the development of additional protocols and algorithms for wide-area and dynamic data management and abstraction.

### **2.3 Technological Implications of Such Services and Applications**

From a computing standpoint, the telematics environment is another instance of a mobile computing framework with two interesting characteristics:

1. The data sources (and sinks) are not just mobile (at relatively high speeds) but also exhibit rapid variation in the content (e.g., car location, average speed or emissions rate) that is of interest to various Telematics applications. Current Internet applications typically do not deal with dynamic changes in context or are vertical solutions designed for very specific operating environments (e.g., fleet management).
2. The total number of data sources is very large (potentially millions of cars). To avoid an explosion in communication bandwidth, telematics applications must rapidly identify appropriate subsets of data sources based on relevant attributes.

We are primarily interested in developing middleware that eases the development and deployment of the telematics services mentioned above. On analyzing the constraints in the generic telematics architecture and the data requirements of various potential telematics services, we realized that any appropriate middleware platform must possess the following set of features.

#### *Maximum Use of Open and Standard Internet Protocols*

Local (hot-spot) services (e.g., at hotels, fast food outlets etc.) should be universally accessible from telematics consoles on all cars, irrespective of differences in the manufacturer or model. The on-board platform is likely to be installed by the individual car manufacturers and may thus exhibit significant heterogeneity. Since the wireless service provider cannot be expected to support the myriad manufacturer/model specific technologies, local services should be deployed using a basic set of *universally supported and platform-independent* technologies and protocols. In addition, the implementation of such local services should not assume the presence of service-specific software components or applications on the client device (car computer). Since the TSP is likely to exercise tight control on the installation of software components on the vehicle, the installation of such third-party components may indeed not be feasible. Our middleware platform is thus designed to enable individualized local services using only a standard HTTP-based browser on the client device; functions such as security and session management are provided using device-independent technologies.

#### *Intelligent Filtering and Abstraction of Heterogeneous Pervasive Data*

Many telematics applications are based on the retrieval and correlation of diagnostic, location and other information from the vehicle and the eventual composition of *compound events* from the raw data. The telematics applications space is, however, characterized by significant variation in the rate, frequency and quality of the vehicular data that is required by any specific application. For example, an insurance application offering rebates to drivers who drive less than 500 miles a week needs to retrieve odometer data only once a week. On the other hand, an application using vehicular speeds to detect highway congestion must monitor speedometer readings with much greater frequency (e.g., once every 30 seconds). To compound this diversity in application requirements, the actual rate of data dissemination from different cars may also exhibit significant variation, depending both on the manufacturer's chosen platform and the type of data. For example, our current telematics applications utilize GPS data from a moving vehicle that is streamed to the central gateway once every 8 seconds. Diagnostic data, such as tire pressure and emission readings, in contrast, are reported only once every few minutes. A middleware platform is needed to shield the individual applications from the platform-dependent variations in data sources and data rates and intelligently filter, transform, and format the platform-dependent data into a device-independent format that conforms to the specific needs of an individual application. An insurance application can then be written without an explicit knowledge of the different rates and precise formats in which different cars export their odometer information; the middleware is responsible for collecting and filtering the raw data to conform to the granularity, frequency and quality needed by a specific application.

#### *Dynamic Modification of Set Membership by Pervasive Sources*

Many intelligent telematics application scenarios are based on the ability to group and categorize vehicles and users based on certain attributes. For example, a service notification utility may issue reminders and notifications about upcoming scheduled service to vehicles whose odometer readings are approaching 30,000 miles. Such sets can be based on multiple attributes—thus, it is possible that the manufacturer of model A requires maintenance service at 30,000 miles, while model B's service schedule occurs at 40,000 miles. Tracking the subset of pervasive devices that satisfy such composite criteria is a significant challenge in the telematics environment. In the absence of an appropriate middleware, each application would need to be aware of platform-dependent attributes and would be responsible for continually monitoring the attribute list of potentially

millions of vehicles and dynamically adjusting their candidate set. This would lead to significant wastage of communication bandwidth, and also make the development of useful and novel telematics applications extremely difficult. The ability to compose and maintain dynamically varying sets based on wide-area and pervasive data sources is thus one of the most significant capabilities of our TRM middleware framework.

### Security and Privacy Management

Privacy concerns have often been articulated as one reason for the slow adoption of telematics scenarios by the driving population. Indeed, the notion of Big Brother gathering vehicular information, such as GPS location or car speeds, is anathema to the vast majority of drivers. Telematics applications must be built on top of a multi-layered security and privacy framework that allows individual users to retain control over the nature and destination of data that they wish to export. For example, a user may specify that his GPS location would be visible at maximum resolution (~5-10 meters) only to designated emergency services; he may also allow his GPS data to be available to insurance providers at much coarser granularity (e.g., 1-2 miles). Developing an effective middleware framework for user configurable exposure of telematics data is key to the adoption of commercial telematics applications, since such a configurable privacy framework effectively empowers consumers and service providers to negotiate commercial agreements based on mutual benefits, and also lowers the significant privacy concerns of the telematics consumer. As an example, a driver may then decide to permit a service providing local advertisements to access the vehicle’s GPS data in return for discounts at designated service stations.

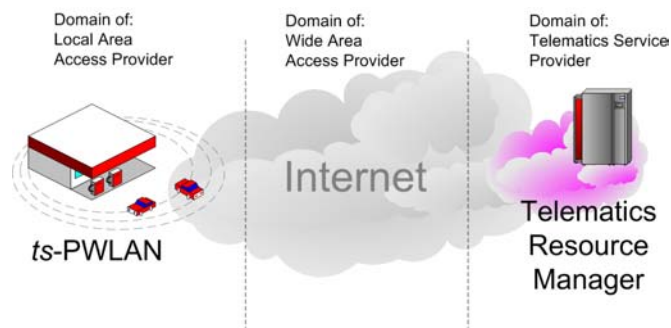


Figure 2. Component technologies: ts-PWLAN and TRM

## 3 Component Technologies

In this section, we describe the two principal technologies that we have developed or are currently developing. These core technologies are both based on publicly used Internet standards and do not have a narrow telematics focus, but are consistent with our belief that successful telematics applications should be based on a more generic framework whenever possible. The ts-PWLAN architecture is designed for service providers in public hot spots and allows telematics users to perform commercial transactions with such hot-spot services in a scalable manner using simple browsing capabilities on the on-board computer. The TRM architecture is a middleware platform that provides efficient indexing and management of mobile and pervasive data. This middleware is designed for use by current and emerging pervasive computing applications, which includes telematics, to retrieve appropriate data from pervasive sources by simply specifying the type of data, rather than having to worry about the specific identity of the potentially mobile devices that supply such data or the platform-specific protocols and technologies needed to interact with such devices. The relationship between these two components is shown in Figure 2.

### 3.1 ts-PWLAN

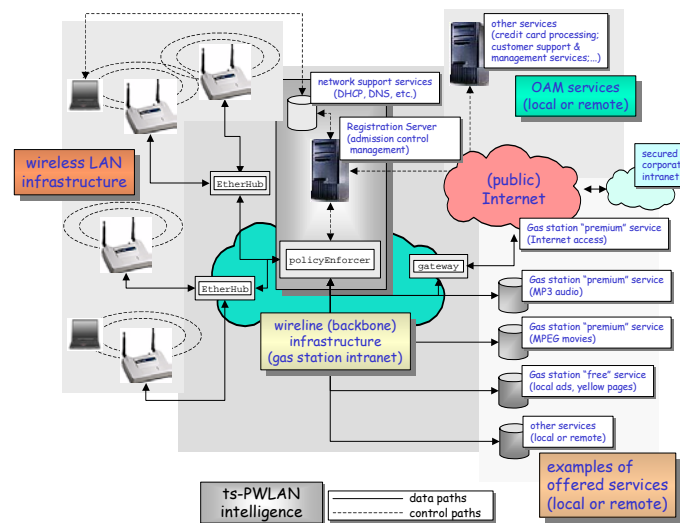
The ts-PWLAN<sup>2</sup> [1] is an initiative for dynamic access to tiered services at public-access hot spots. We have been witnessing the rapid deployment of IEEE 802.11 based wireless LANs in a variety of public *hot-spots*, such as airports, hotels, internet cafés, and so on. While the initial PWLAN deployment simply focuses on providing backbone (Internet) access, such LANs will eventually be used to offer differentiated sets of local services as well. Providers of such wireless services (called WISPs<sup>3</sup>) have a strong incentive to offer such differentiation, since this allows providers to introduce premium services and thereby charge differential amounts to different consumers. To offer commercial services at such hot-spots (e.g., cafes, airports etc.), the access architecture must possess the following capabilities:

<sup>2</sup> ts-PWLAN is an unwieldy acronym for “Tiered Services over Public Wireless LANS”.

<sup>3</sup> WISP stands for Wireless Internet Service Provider

1. *Rapid Service Discovery*: Users must be able to discover available resources and menus of offered resources whenever they roam into the coverage area of such a hot spot.
2. *Service Selection and Registration*: The architecture must allow different users to dynamically select their service tier and communicate this selection back to the WISP infrastructure. Users must also have the freedom to dynamically alter their selection anytime during the duration of their current selection.
3. *Profile Enforcement*: The WISP infrastructure must be capable of performing per-user access control to ensure that a particular user is able to gain access only to the set of network resources that she has registered for (and presumably paid for as well). The enforcement function must also be integrated with billing and accounting modules, since the user charges can be based on various parameters such as duration or actual resource consumption.

There are indeed several protocols and approaches for performing each of the above-mentioned functions. The real challenge, however, is in enabling such functions *using a minimum set of standard and universally deployed technologies on the client devices*. Since different users will use different computing platforms (e.g., laptops, PDAs, car computers), and quite possibly different operating systems, to access the hot-spot services, a solution based on specific device-characteristics or requiring service-specific client code will end up defeating the objective of universal access. Moreover, the architecture *must not assume the existence of pre-existing client profiles* at the specific WISP, but must allow the user to dynamically establish transient profiles (service selections) at their current hot spot. Of course, if such profiles do exist (we shall discuss this later in our VIP Gas Lane scenario), the architecture can leverage such information to automate the process of service tier selection.



**Figure 3.** Functional Elements of the *ts-PWLAN* Architecture

The *ts-PWLAN* architectural framework allows clients to use a simple standard HTTP-based browser to transact with local services. Additional functions are either performed using standard browser capabilities (e.g., service selection and registration, secure payments) or are based on signaling that is entirely confined to infrastructure devices within the WISP's administrative control. Moreover, the architecture uses standard Internet protocols and technologies, such as DHCP, DNS or HTTP cookies to provide registration, access control and intra-hot-spot roaming support. In contrast to alternative public access solutions (e.g., Microsoft's CHOICE network [2] or Cisco's BBSM [3]), the *ts-PWLAN* is unique in its support of client independence (no additional component on client device), on-the-fly service re-negotiation, and intra-hot-spot roaming (even across access technologies or IP subnets).

Figure 3 illustrates the logical layout of elements in the *ts-PWLAN* architectural framework. The wireless LAN infrastructure typically consists of a collection of access points (AP), which provide customers wireless connectivity to the *ts-PWLAN* access infrastructure. The infrastructure itself consists of certain networking configuration services (such as DHCP and DNS) that are not subject to any form of access control and are freely available to any device (such as the on-board computer) equipped with a wireless LAN card and an appropriate IP stack. The *ts-PWLAN* vision also includes a variety of *local* services, as well as *global* services such as Internet access, all of which lie behind an intelligent *gateway* that regulates access to these services. The access gateways are a key component of the *ts-PWLAN* architecture and perform per-packet and per-user filters to ensure that internal network resources are accessible or visible only to authorized (registered) users. We have implemented a prototype of the *ts-PWLAN* service framework using servlets and CGI web technologies to support user

selections, and automatic user configuration based on such selections, and user session management.

User access to the local services infrastructure is regulated by the *Registration Server (RS)*. The RS presents a *Web-based* registration menu to clients, who can then interact with the RS using a standard Web browser and select from a list of offered services. The RS registration menu is available at a locally unique URL, e.g., *http://www.public\_WLAN.com*-- this URL need not be the same in different hot spots. It is important to note that the registration-process is purely browser-based and does not require any modifications or other software or hardware utilities to be installed on the client device. As part of this registration process, commercial telematics service providers may require the client to provide authentication or payment credentials (such as a credit card) to the RS. Once a client selects a particular level of service, the RS will “tie” the IP address of the client device with the selected tier of services and then issue the appropriate remote configuration commands to one or more access control gateways, which then set up appropriate packet filters. The RS server is also responsible for coordinating the user de-registration process. The architecture also uses standard Web technologies to provide uninterrupted service access to registered users (by suitable IP address re-binding) even in the face of intermittent wireless connectivity (which is often experienced in real-life outdoor LAN environments).

In the telematics scenario, freely accessible local Web pages may offer functions such as menu listing (at fast food outlets), general parking directions and facilities listings (at convention centers) and hotel room availability (at hotels). Premium local services may include functions such as room service order placement (once the customer has registered as a hotel guest) or download of premium audio content (songs, audio books etc.) from a gas station repository once the customer has paid the appropriate charges. The original ts-PWLAN architecture was designed to include global network-centric services, such as Internet access or VPN connectivity to corporate Intranets, and was targeted towards traditional computing and communication devices, such as cellular phones, laptops and PDAs. The ts-PWLAN architecture can, however, be utilized by service providers interested in purely local services to support functions such as dynamic user registration, payment and infrastructure access control.

### *3.1.1 VIP Gas-Lane: A Sample Local Telematics Service based on the ts-PWLAN framework*

We are currently working on a sample telematics application that leverages the ts-PWLAN wireless solution. In this application, we assume that gas stations are equipped with a “VIP” lane that provides IEEE 802.11-based WLAN access to various local services. When a user drives within communication range, the Web browser on her vehicle displays the Welcome page of the gas station and offers unregistered users a range of services (including on-line payment for gas, and downloads of infotainment, such as music or video content) to the on-board device. Users who do not have prior subscriptions with the gas station brand may then select such services using ts-PWLAN’s dynamic pay-per-use mechanisms. A richer and more interesting model allows vehicles to have pre-established profiles with specific gas station brands (similar to the Mobil SpeedPass [8] solution). The ts-PWLAN solution then allows such vehicles to automatically provide the gas station server with their subscription credentials and thereby proactively perform functions such as infotainment download (without explicit human involvement) whenever the driver pulls up at the appropriate gas station brand. Of course, this feature requires the installation of additional code on the client device; this may however be performed off-line by the TSP (which may have an agreement with a specific gas station brand) itself.

## **3.2 TRM**

TRM is the middleware that powers backend, context-based services. Whereas ts-PWLAN provides client access to commercial services at public access hot spots, TRM is the common software substrate that facilitates the deployment of telematics applications and services that can be accessed at these hot spots. From the client point of view, services deployed under the TRM framework appear exactly like web services. Note, this does not imply that all services must be a web service; TRM provides this abstraction for the benefit of clients. This allows heterogeneous clients to access any number of services in TRM using standard web browsers. Clients use the HTTP protocol to send and receive XML formatted messages between themselves and applications. For simplicity, the current implementation of TRM uses the REST [7] model of interaction to pass messages. This model allows the exchange of data between clients and web services that mimic RPC functionality using only the mechanisms provided by the HTTP protocol.

Fundamentally, many context-based telematics services need to consult a wide range of network data providers. For example, fleet-management applications need to continuously monitor the locations of all vehicles in the fleet. A real-time traffic alerter needs to be aware of any traffic incident that is relevant its subscribers. Fast-food companies may want to push coupons to cars within range of a franchise location during mealtimes. Some of this data cannot be derived directly from a data provider but must be *composed* from many data providers. To support these functions, TRM provides four core services for all telematics application:

- A service environment that utilizes the web services model to easily build and deploy applications that require access to a dynamic set of data providers
- Computational framework for finding, retrieving, and transforming data

- Resource discovery mechanism that allows clients to locate and access a range of data providers without requiring *a priori* knowledge about the providers network locations.
- High-level data access abstractions to applications via protocol brokering between applications and heterogeneous network data providers

Each core function is encapsulated by an internal TRM component. Figure 3 shows the overall architecture. The Service Manager hosts web services that encode high-level application logic wrapped by a web services interface. Within the Service Manager is the Composition Engine that executes the program logic for high-level data composition. Data composition program logic is encoded in special modules called *composers*. For example, a composer would be responsible for detecting traffic accidents that are relevant to a user from various data sources. Abstractly, a composer is similar to widgets in the Context Toolkit that encapsulate specific logic for deriving context data [6]. In the current implementation, data composers are written in Java. However, from our experience, data composition has its own special requirements and it is not always convenient or possible to use an existing programming language to describe fundamental data composition operations. We are currently researching the applicability of a new data composition language for creating data composers called iQL [5].

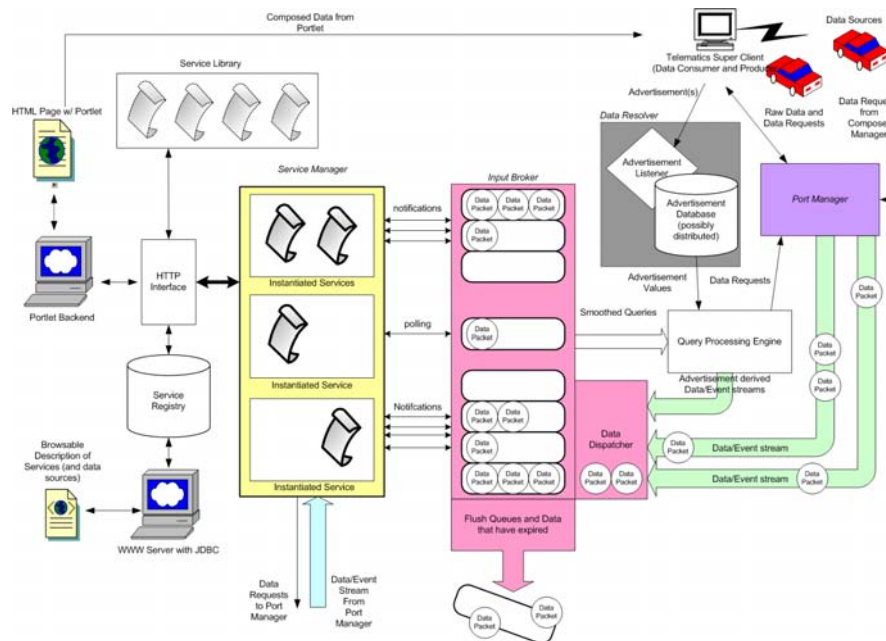


Figure 4. Overview of TRM middleware for Telematics.

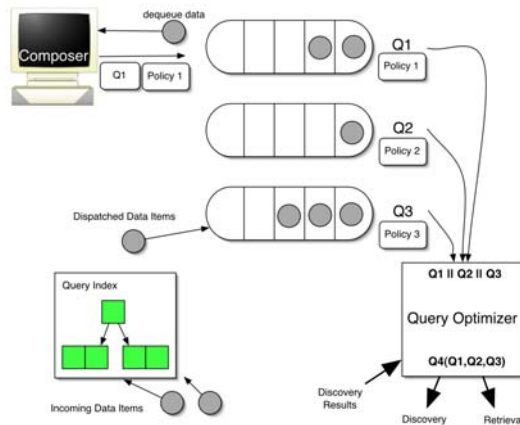
Data composers will often need to consult external data providers during their execution. To facilitate this process, composers send queries for data to the Input Broker component. The Input Broker is responsible for finding and retrieving data from data providers using the Data Resolver and Port Manager, respectively. The Input Broker is an optimization component and handles flow control issues, asynchronous messaging, and query processing and optimization.

The Input Broker processes queries from data composers in two steps: discovery and retrieval. Discovery involves finding data sources that can provide relevant data for a data composer. Retrieval involves contacting the data provider and retrieving the data. The Data Resolver is responsible for processing discovery queries. To facilitate this process, all data providers that wish to provide data to clients must register with TRM. Data providers register an *advertisement* with the Data Resolver. An advertisement is a message (in XML) that specifies the services a data provider can provide, its network address, and other important attributes. The Data Resolver is mainly responsible for indexing all available advertisements and processing queries from the Input Broker. It should be noted that queries and advertisements have a limited lifetime in the system. The Input Broker and Data Resolver will flush all stale items that are not actively maintained through a client leasing protocol. This allows the components to be resilient to failures in data providers since they may not be able to explicitly cancel an advertisement. If a TRM component fails, the client leasing protocol allows the component to repopulate itself with queries or advertisements.

The core components of TRM - the Service Manager, Data Composition Engine, Input Broker, Data Resolver, and Port Manager - form a *TRM node*. A TRM node is an entity that can provide core functionality for a set of applications to clients. To scale to a large number of clients and services, many TRM nodes can be deployed on the network to form an application-

level overlay network. In such a configuration, TRM nodes cooperate in executing data composition tasks as well as processing queries. Our goal is to design TRM nodes as autonomous entities that can self-organize into an overlay network that is resilient to failures. We have already investigated some aspects of this in designing a distributed resource discovery protocol for the Data Resolver [4] and are actively investigating several designs for the next version of TRM.

We expect that access control and privacy mechanisms can be implemented around the concept of a TRM node. Specifically, all data providers fall under the administrative control of the local TRM node. This node is responsible for authenticating and authorizing users who request data from their data providers.



**Figure 5.** Query processing overview

Clearly, the core services provided by TRM are useful beyond just telematics and it is our goal to build a general mechanism for deploying pervasive computing applications. Telematics is arguably the first pervasive computing application that has clear commercial significance, and we are using telematics to help drive the design of TRM.

To help understand TRM better, consider Figure 5, which depicts at a high level the process of retrieving data through TRM. Suppose a driver wants to continuously monitor all traffic incidents that are relevant to her commute. Using the platform in her car, she contacts a navigation advisor service over a wireless network. The navigation advisor employs a data composer that encodes the logic of how to detect abnormal traffic conditions from embedded loop sensors and correlate these with traffic incident reports available through static web pages. Thus, the composer will at least report incidents on the static web pages, but is much more likely to report the actual traffic condition on the road since it also consults the loop sensors.

This task uses the core functionality in TRM since there is no way to tell which particular loop sensors are available at any point in time. In general, data providers come and go intermittently. Instead, the data composer issues a query Q1 to the Input Broker for loop sensor data along the route of the client. Conceptually, this query represents a vector of relevant data and TRM will attempt to populate this vector with actual data items.

The Input Broker is responsible for setting aside computational resources to maintain this vector. In general, the Input Broker is responsible for many queries from different data composers, each requiring its own resources for maintenance. All queries are directed to a query optimizer. Outgoing queries from the Input Broker to the Data Resolver and Port Manager are optimized using traffic shaping algorithms to provide efficient use of shared resources. For examples, if  $M$  queries in the Input Broker want data from the same data provider, the query optimizer should only send out 1 query rather than  $M$ . Of course, this is not the only kind of optimization that should be done and optimizations are not limited to any single component in TRM. We plan to investigate optimization issues as part of our research.

The Input Broker first issues a query to the Data Resolver to “discover” relevant loop sensors. The Data Resolver returns a set of advertisements from relevant loop sensors (perhaps cooperating with remote Data Resolvers in a distributed deployment to process the query). Using the information in the advertisements, the Input Broker contacts the loop sensors through the Port Manager. Data from loop sensors arrives asynchronously and is processed by a dispatcher that consults a query index to efficiently place data items on the relevant sets maintained by the Input Broker. The efficient design of the query index is critical for providing reasonable throughput for the system and we plan to leverage existing research on this subject. The Input Broker can either notify the composer when new data arrives in response to its query, or the composer can poll the Input Broker to de-queue data items.

In our example, the data composer's request includes a policy on how to maintain the physical data set in the event of updates. For example, the policy might state that new data from a loop sensor with id=1 should always replace old data from the same loop sensor already in the set. Alternately, the policy might state that only N data items from loop sensors should be kept and the set should be managed as a FIFO queue. Policy specifications of this kind are an interesting area of research.

## 4 Scenarios and Implementation Status

In this section we describe some of the prototype telematics applications that we are building with our platform. While each of these applications can be built without using TRM, our intent is to demonstrate that such applications are more easily built and deployed using our middleware with no appreciable differences in performance. In addition we want to demonstrate that a wide variety of applications can be supported by our middleware and that benefits of interoperability between the system components allows the creation of more ambitious applications.

We currently have working prototypes for all the major components in TRM. We hope to provide a full report on a robust implementation in a future publication.

### 4.1 *Phone Number Lookup*

In this scenario, a client wishes to look up phone numbers that are not available on her phone. The simplest case is that the client uses her telematics platform to access a telephone directory where she can specify, through a voice interface, the name of the person she wants to call. In a more complex case, the client wishes to download many phone numbers into her telematics platform that are relevant to her current location. This *location-based* look up involves finding telephone numbers of places the client is likely to call.

### 4.2 *Service Reminder*

One benefit from the automobile manufacturer point of view in having a telematics platform in the car is the ability to extend and improve customer relations after the sale of the car. In the service reminder scenario, clients can opt to have diagnostic data and mileage of their car monitored by a service reminder application. When the car is operating out of normal parameters or is just in need of normal maintenance, the service reminder application can warn the client that service is necessary and also schedule an appointment with a mechanic on behalf of the client. This service is convenient for the client but also highlights a benefit to the manufacturer (and why this should be implemented in the infrastructure): through a telematics infrastructure a car manufacturer can get real-time data about their a car throughout its entire life cycle, potentially improving their core business.

### 4.3 *Navigation Assistant*

Driver distraction is an important safety concern for any telematics application. For navigation applications, the potential for driver distraction is great if the driver is continuously interrupted with turn-by-turn instructions or is forced to fiddle with application interface. In this scenario, we are implementing a navigation assistant that suggests routes to the driver and only interrupts if a real-time traffic incident will impact the driver's current route. This supports the most common form of navigation where the driver mostly knows the possible routes to take but does not know the optimal route given current traffic conditions.

As part of this implementation we are implementing a highly scalable navigation server that can assist many clients and keep track of a large number of traffic incidents. The navigation server has the dual role of being a TRM client (since it needs access to many data providers regarding traffic incidents) and a TRM data provider itself (since many drivers will consult it for route advice).

## 5 Related Work

Mobile commerce is a relatively new applied research area, e.g. Varshney reports on location management of mobile commerce applications [14]. Early deployments of telematics solutions are mostly in the industrial sector and fall into two general categories: call centers, optical disk based platforms. Examples of the former include OnStar [17] and ATX Technologies [17]. An example of the former is the turn-by-turn navigation system Magellan 750NAV [19]. The Smart Bus project at UCLA is equipping public buses with a telematics platform to collect data for data mining [11]. Work on wireless technologies for accessing commercial services is of growing interest. Much work has been done in context-aware services [6] as well as proposals for middleware for pervasive computing systems. However, to the best of our knowledge, our system is the first to be specifically designed to support telematics applications.

## References

- [1] A. Acharya, C. Bisdikian, Y. Ko, A. Misra. *ts-PWLAN: A Value-add System for Providing Tiered Wireless Services in Public Hot spots*, under submission, May 2002.
- [2] V. Bahl, W. Russel, Y-M. Wang, A. Balachandran, G. Voelker. *PAWNs: Satisfying the Need for Ubiquitous Secure Connectivity and Location Services*, IEEE Wireless Communications Magazine, Feb. 2002.
- [3] “Cisco Building Broadband Service Manager: Technical Overview”, *White paper*, Oct. 2001.
- [4] P. Castro, B. Greenstein, R. Muntz, C. Bisdikian, P. Kermani, M. Papadopouli, "Locating Application Data Across Service Discovery Domains ". 7th International Conference on Mobile Computing and Networking, Rome, Italy, July 2001.
- [5] N. Cohen, H. Lei, P. Castro, J.S. Davis II, A. Purakayastha, “Composing Pervasive Data Using iQL”, Proceedings of the 4<sup>th</sup> IEEE Workshop on Mobile Computing Systems and Applications, Jun 2002, to appear
- [6] A. Dey, “Providing Architectural Support for Building Context Aware Applications”, PhD Thesis, College of Computing, Georgia Institute of Technology, Dec 2000.
- [7] R. Fielding, “Architectural Styles and the Design of Network-based Software Architectures”, PhD Thesis, University of California, Irvine, 2000.
- [8] K. Hammonds, Pay as You Go, <http://www.fastcompany.com/online/52/speedpass.html>
- [9] IEEE Computer Society. *802.11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 1999. [Maples] J. Maples, “Will Telematics Drive the Industry to Distraction?”, Red Herring, April 1, 2001
- [10] ISO/TC204’s Working Group 15, Dedicated Short-Range Communication Standards, <http://www.iso.ch/iso/en/stddevelopment/tc/tclist/TechnicalCommitteeDetailPage.TechnicalCommitteeDetail?COMMID=4559>
- [11] C.J. Lai, R. Muntz, “UCLA Transit Services – Realtime Bus Tracking”, <http://www.cs.ucla.edu/~cjlai/bustrack/>
- [12] J. Maples, “Will Telematics Drive the Industry to Distraction?”, Red Herring, April 1, 2001
- [13] “Roadpricing”, <http://www.roadpricing.nl>
- [14] U. Varshney, “Location Management Support for Mobile Commerce Applications”, Proceedings of Mobile Commerce 2001.
- [15] 3<sup>rd</sup> Generation Partnership Project 2, “Cellular Radiotelecommunication Intersystem Operations Network Evolution to 3G”, <http://www.3gpp2.org>
- [16] 3<sup>rd</sup> Generation Partnership Project, <http://www.3gpp.org>
- [17] <http://www.atstechnologies.com>
- [18] <http://www.onstar.com>
- [19] <http://magellangps.com>