



Module 5: Transit Management Standards, Part 2 of 2

Table of Contents

Introduction/Purpose	2
Samples/Examples	2
References to Other Standards	6
Case Studies	20
Glossary	24
References	25
Study Questions	27



Module Description

This module (Transit Management Standards, Part 2 of 2) is the second of the two course modules on Transit Management Standards. It introduces hardware and software standards in addition to Transit Communications Interface Profiles (TCIP), and explains how to use them. The information in this module will help participants identify which standards are most applicable to a particular situation so that an agency can reduce the life-cycle cost of technologies that support Transit Management functions and facilitate the integration with legacy or future technology systems. The structure and use of data exchange standards for Transit Management systems are described along with how to apply standards to the development of procurement specifications. Case studies are incorporated to enhance participants' understanding of how to use the standards.

1. Introduction/Purpose

From a National Intelligent Transportation Systems (ITS) Architecture perspective, Transit Management covers technologies and systems that facilitate and automate operations, planning, management, maintenance, safety, security and data management functions of public transit systems. Transit Management systems are typically located on-board vehicles and at central dispatch or garage locations. Communications technologies provide the basis for receiving and transmitting the data and information generated by these systems. In some cases, information generated by several Transit Management systems provides the basis for generating customer-facing traveler information.

2. Samples/Examples

As described in Module 2, there are four categories of Transit Management systems/technologies. The Fleet Operations and Management category covers technologies that are implemented to facilitate transit operations and provide input to senior management in terms of overall system performance. The Safety and Security category covers those technologies that improve the safety and security of transit staff and passengers through on-board and facility technologies. The Maintenance category covers technologies that facilitate maintenance activities, such as engine and vehicle component monitoring and tracking of scheduled and unscheduled maintenance activities and inventory systems. The Data Management category covers public transit ITS components installed in vehicles, at central locations (e.g., control centers), or at other locations (e.g., stops and shelters). These components generate an enormous amount of data typically collected and archived in the individual databases of systems that generate the data. The extent of field data collected on-board vehicles depends on the configuration of the ITS and its subsystems on vehicles or at other locations (e.g., refresh rate of automatic vehicle location (AVL) subsystem, recording rate of on-board video surveillance subsystem and health diagnostics, refresh rate for off-site equipment). Once the ITS data are archived, they are used for a variety of "after-the-fact" analyses and reporting by different business units within a public transport organization (e.g., planning, operations, customer service). Each of the technologies within each category was briefly defined in Module 2.



The relationships among central Transit Management and other transit ITS technologies are shown in Figure 1. Figure 2 shows an example of the relationships on-board a transit vehicle.

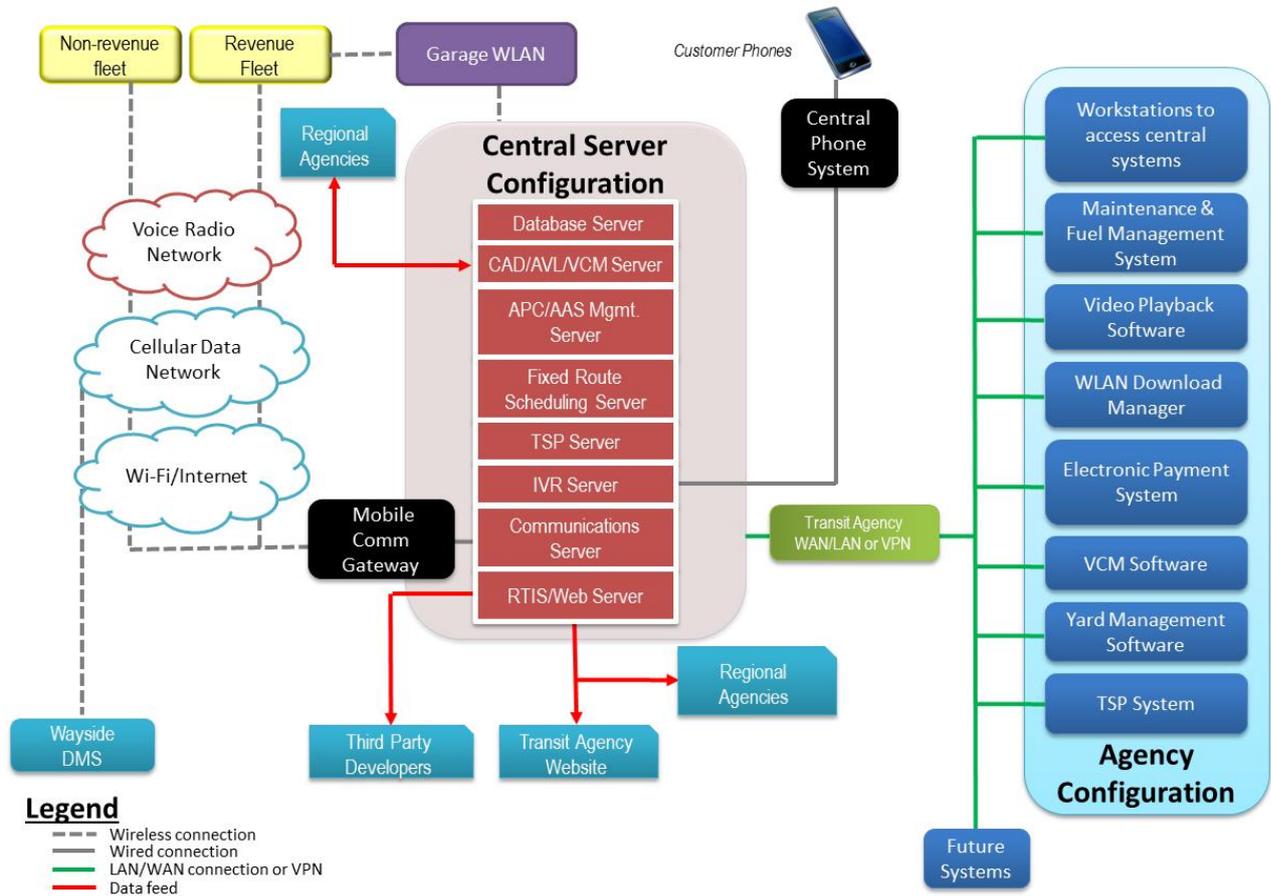


Figure 1. Example of Central System Technology Relationships (courtesy TranSystems Corporation)

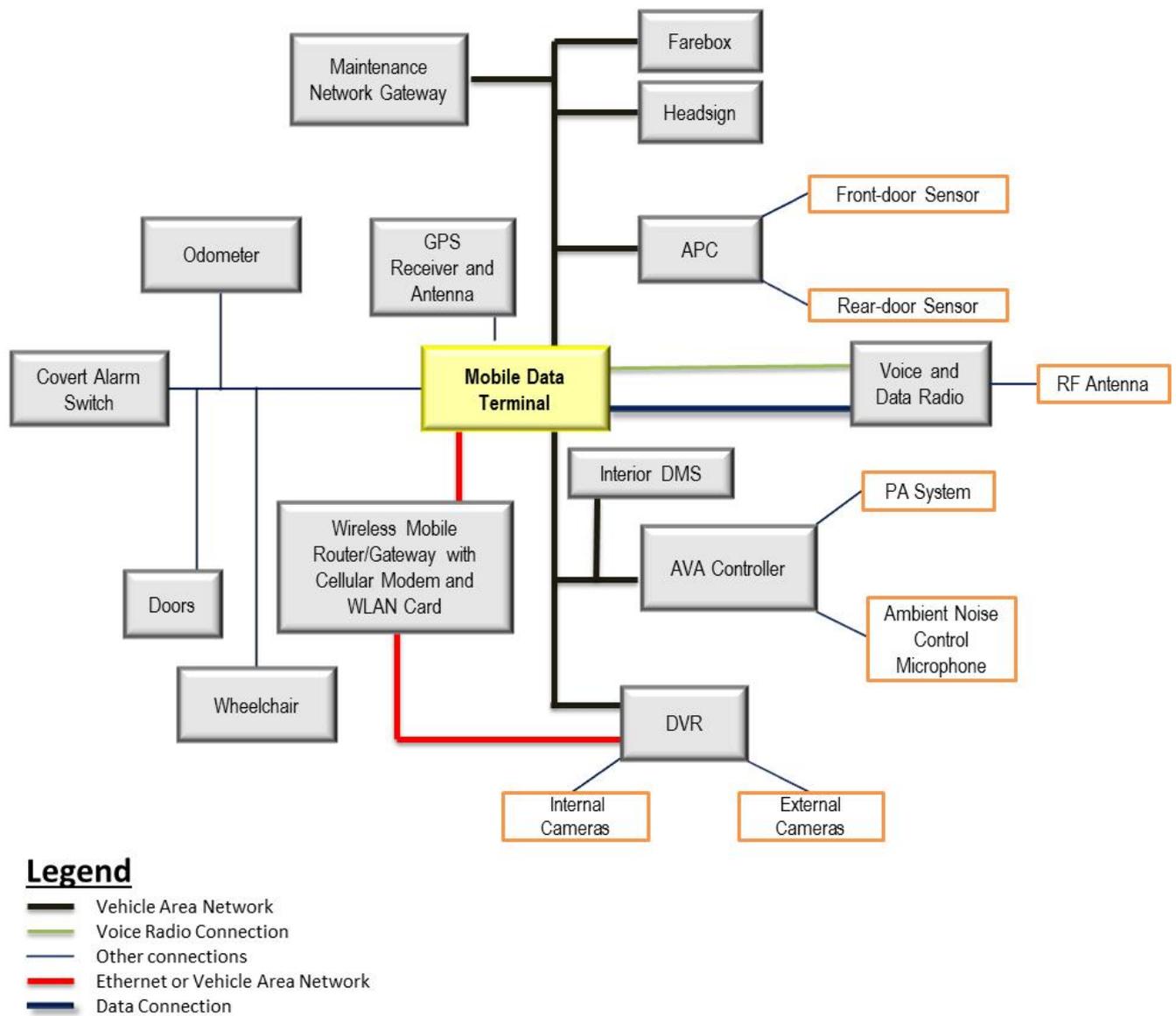


Figure 2. Example of On-Board Technology Relationships (courtesy TranSystems Corporation)



In order to provide a further, general understanding of how these technologies are related to each other, Table 1 shows the dependencies among the technologies.

Table 1. Dependencies Among Transit ITS Technologies

Category	System/Technology	Dependent on
Fleet Operations and Management	Communications technologies	Public/private voice and data communication backbones
	Computer-aided dispatch (CAD)	<ul style="list-style-type: none"> • Voice and data communications technologies • Automatic vehicle location (AVL) system • Route and vehicle schedule data
	Automatic vehicle location (AVL)	<ul style="list-style-type: none"> • Data communications technologies • Global positioning system (GPS) or other location enabling technologies, such as WiFi
	Automatic passenger counters (APCs)	<ul style="list-style-type: none"> • AVL system • Route and vehicle schedule data
	Scheduling (fixed-route and paratransit) systems	Stop database (contains data such as stop name, location, routes that stop at this stop, direction of travel from this stop, list of amenities available at this stop)
	Transfer connection protection (TCP)	<ul style="list-style-type: none"> • AVL system • CAD system
	Transit signal priority (TSP)	<ul style="list-style-type: none"> • AVL system • CAD system (when TSP used based on schedule adherence status) • Roadside signal infrastructure
	Yard management	Indoor positioning systems (e.g., radio frequency identification [RFID]-based, WiFi-based)
	Intelligent vehicle technologies (e.g., collision warning and precision docking)	Varies by technology application and deployment
	Lane control technologies	<ul style="list-style-type: none"> • AVL system • CAD • Virtual mirror • Lane guidance systems • Roadside signal infrastructure
Safety and Security	Fixed video surveillance	Data communications technologies
	Covert emergency alarm and covert live audio monitoring	<ul style="list-style-type: none"> • Voice and data communication technologies • CAD system • AVL system



Category	System/Technology	Dependent on
Maintenance	On-board digital video surveillance	No dependence on other systems
	G-force monitoring	AVL system
	Engine and drivetrain systems monitoring	OBD-II ¹ or Society of Automotive Engineers (SAE) J1708/J1939 compatibility of on-board computers within engine and drivetrain
	Maintenance software to schedule and track scheduled and unscheduled maintenance activities, and manage parts inventory	No dependence on other systems
Other	Fuel Management System	No dependence on other systems
	Enterprise database/ datawarehouse and reporting	<ul style="list-style-type: none"> • Open databases • Data dictionary
	Technology integration	Multiple dependencies
	Geographic information system (GIS) application	Spatial data recording and management systems
	Service coordination facilitated by technology	<ul style="list-style-type: none"> • CAD/AVL systems shared across participants • Voice and data communications technologies
Open data for third-party application development	Standard format for data such as General Transit Feed Specification (GTFS) and GTFS-real time	

3. Reference to Other Standards

3.1. Application Areas

Application areas, as shown in Table 2, provide a starting point for identifying the ITS standards and other resources (e.g., case studies, lessons learned) that may be relevant to a specific type of deployment. Application areas are deployment-oriented categories that focus on commonly deployed ITS services or systems. The National ITS Architecture is divided into interface classes, which are further subdivided into application areas. Interface classes are defined by the type of system at each end of the communications path: center, field, vehicle, and traveler.

Typically, public transit management interface classes are center to infrastructure (C2I) (interfaces between a management center and its field equipment (e.g., dynamic message signs at stops); center to vehicle/traveler (interfaces between a center and the devices used by transit drivers or travelers) (C2V); and center to center (interfaces between transportation management centers) (C2C).

¹ OBD-II is a standard that monitors engine, chassis, body, and accessory devices in a vehicle



An application area within the C2I interface class would include bidirectional communication between a center-type system (e.g., transit dispatch center) and a field-type system (e.g., dynamic message signs at a transit stop).

Table 2. ITS Standards Application Areas

National ITS Architecture Interface Class	Standards Application Areas
<p>Center to Center – This class of application areas includes interfaces between transportation management centers.</p>	<p>Data Archival Incident Management Rail Coordination Traffic Management Transit Management Traveler Information</p>
<p>Center to Field – This class of application areas includes interfaces between a management center and its field equipment (e.g., traffic monitoring, traffic control, environmental monitoring, driver information, security monitoring, and lighting control).</p>	<p>Data Collection/Monitoring Dynamic Message Signs Environmental Monitoring Lighting Management Ramp Metering Traffic Signals Vehicle Sensors Video Surveillance</p>
<p>Center to Vehicle/Traveler – This class of application areas includes interfaces between a center and the devices used by drivers or travelers. It includes interfaces with motorists and travelers for exchange of traveler and emergency information as well as interfaces between management centers and fleet vehicles to support vehicle fleet management.</p>	<p>Mayday Transit Vehicle Communications Traveler Information</p>
<p>Field to Field – This class of application areas includes interfaces between field equipment, such as between wayside equipment and signal equipment at a highway rail intersection.</p>	<p>Highway Rail Intersection (HRI)</p>
<p>Field to Vehicle – This class of application areas includes wireless communication interfaces between field equipment and vehicles on the road.</p>	<p>Probe Surveillance Signal Priority Toll/Fee Collection</p>



3.2. Transit Management Application Area

The Transit Management Application Area (center-to-center [C2C] application area) covers the interface between a transit management center and other centers and supports the following capabilities:

- Providing multimodal coordination between transit agencies and other types of public transportation (i.e. ferries, rail, airlines) at transfer points, including coordinating information between local/regional transit organizations including schedules, fare, on-time information, and ridership.
- Providing transit incident information, schedules, and fare and pricing information to a traveler information provider.
- Providing transit information suitable for media use.
- Providing emergency transit schedule information to traffic and emergency operations centers, especially during evacuations.
- Providing transit system information to traffic management centers.
- Supporting personalized transit routes requested by travelers via traveler information providers.
- Coordinating with financial institutions on the approval and status of electronic fare payments.
- Coordinating with law enforcement regarding the notification of violations.

The extent to which information and coordination are shared between centers is determined through working arrangements among agencies or jurisdictions.

Figure 3 shows the scope of the Center-to-Center Transit Management application area. This scope is described in the preceding text.

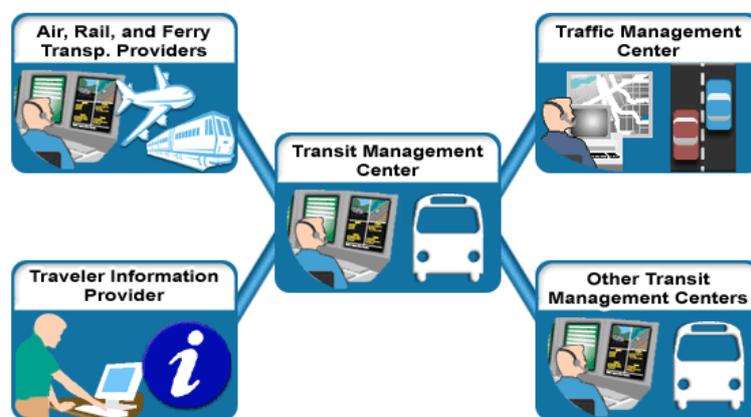


Figure 3. C2C Transit Management Application Area



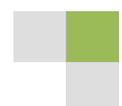
This application area focuses on the C2C interfaces with the Transit Management Subsystem (TRMS). The TRMS accepts requests for personalized transit routing, as well as general transit system information requests from the Information Service Provider. The TRMS interfaces to financial institutions to support electronic fare payment. It also coordinates with Other Transit Management systems, as well as Multimodal Transportation Service Providers (i.e., providers of other modes of transportation such as ferry or rail), for the exchange of information to allow schedule and routing coordination between systems. The TRMS disseminates transit incident information, schedules, and fare and price information to Information Service Providers and the Traffic Management Subsystems. The TRMS also distributes transit status and incident reports (appropriately tailored for external distribution) to the media. During emergency situations, the TRMS provides emergency transit schedule information to the Emergency Management (EM) and Traffic Management (TM) subsystems. Finally, the TRMS coordinates the Enforcement Agencies regarding the notification of violations. Note that managing and tracking of transit vehicles is not covered in this set of interfaces, but is included in the center-to-vehicle application areas. In addition, the management of transit incidents is covered in the incident management application area.

In general, the standards shown in Table 3 are applicable to Transit Management deployments. To determine which specific standards are applicable for a deployment you will need to determine which architecture flows will be needed for the Transit Management piece of your deployment.

Table 3. Standards Applicable to Transit Management

Standard	Development Status
APTA TCIP-S-001 3.0.0 Standard for Transit Communications Interface Profiles	Published
NTCIP 1102 Octet Encoding Rules (OER) Base Protocol	Published
NTCIP 1104 Center-to-Center Naming Convention Specification	Published
NTCIP 2104 Ethernet Subnetwork Profile	Published
NTCIP 2202 Internet (TCP/IP and UDP/IP) Transport Profile	Published
NTCIP 2303 File Transfer Protocol (FTP) Application Profile	Published
NTCIP 2304 Application Profile for DATEX-ASN (AP-DATEX)	Published
NTCIP 2306 Application Profile for XML Message Encoding and Transport in ITS Center-to-Center Communications (C2C XML)	Published
NTCIP 8003 Profile Framework	Published
NTCIP 9001 NTCIP Guide	Published
SAE J2266 Location Referencing Message Specification (LRMS)	Published
SAE J2354 Message Set for Advanced Traveler Information System (ATIS)	Published
SAE J2540 Messages for Handling Strings and Look-Up Tables in ATIS Standards	Published
SAE J2540/1 RDS (Radio Data System) Phrase Lists	Published
SAE J2540/2 ITIS (International Traveler Information Systems) Phrase Lists	Published
SAE J2540/3 National Names Phrase List	Published

In general, the standards shown in Table 3 are applicable to Transit Management deployments. To determine which specific standards are applicable for a deployment, you will need to determine which architecture flows will be needed for the Transit Management piece of your deployment.



3.3. Open System Interconnection

In terms of standards structures, Open System Interconnection (OSI) is an International Organization for Standardization (ISO) standard for worldwide communications that defines a networking framework for implementing protocols in seven layers. Control is passed from one layer to the next, starting at the application layer in one station, proceeding to the bottom layer, over the channel to the next station, and back up the hierarchy.

The seven layers are defined as follows:

- **Physical (Layer 1):** This layer conveys the bit stream – electrical impulse, light or radio signal – through the network at the electrical and mechanical level. It provides the hardware means of sending and receiving data on a carrier, including defining cables, cards, and physical aspects.
- **Data Link (Layer 2):** At this layer data packets are encoded and decoded into bits. It furnishes transmission protocol knowledge and management, and handles errors in the physical layer, flow control, and frame synchronization. The data link layer is divided into two sublayers: The Media Access Control (MAC) layer and the Logical Link Control (LLC) layer. The MAC sublayer controls how a computer on the network gains access to the data and permission to transmit it. The LLC sublayer controls frame synchronization, flow control, and error checking
- **Network (Layer 3):** This layer provides switching and routing technologies, creating logical paths, known as virtual circuits, for transmitting data from node to node. Routing and forwarding are functions of this layer, as well as addressing Internet working, error handling, congestion control, and packet sequencing.
- **Transport (Layer 4):** This layer provides transparent transfer of data between end systems, or hosts, and is responsible for end-to-end error recovery and flow control. It ensures complete data transfer.
- **Session (Layer 5):** This layer establishes, manages, and terminates connections between applications. The session layer sets up, coordinates, and terminates conversations, exchanges, and dialogues between the applications at each end. It deals with session and connection coordination.
- **Presentation (Layer 6):** This layer provides independence from differences in data representation (e.g., encryption) by translating from application to network format, and vice versa. The presentation layer works to transform data into the form that the application layer can accept. This layer formats and encrypts data to be sent across a network, providing freedom from compatibility problems. It is sometimes called the syntax layer.
- **Application (Layer 7):** This layer supports application and end-user processes. Communication partners are identified, quality of service is identified, user authentication and privacy are considered, and any constraints on data syntax are identified. Everything at



this layer is application-specific. This layer provides application services for file transfers, e-mail, and other network software services. Telnet and FTP are applications that exist entirely in the application level. Tiered application architectures are part of this layer.

3.4. Society of Automotive Engineers (SAE) J1708

J1708 is a Society of Automotive Engineers (SAE) specification developed especially for heavy duty vehicles including transit vehicles. It was considered the first standard for an on-board vehicle area network (VAN), which is analogous to a local area network (LAN) in an office environment. The standard is for serial communication between modules or components with microcontrollers. The standard means that data can be transferred between on-board devices in a more cost-effective way. Further advantages have been as follows:

- Minimizes the hardware cost.
- Offers flexibility and the possibility of further expansion of an existing system.
- Uses standard industry electronics, which gives more options when developing.

Since J1708 only describes the lower layers of the OSI model, it is always used with an overlaying application layer. An example of such a layer is J1587, which is used for data exchange between microcontrollers in heavy duty vehicles. Other key facts about J1708 are as follows:

- Describes the physical and data link layer according to OSI model
- Almost always used in conjunction with the application layer protocol SAE J1587

The J1587 protocol defines the format of J1708 messages sent between microprocessor devices in heavy duty vehicles. It also supports communication with external devices connected to the bus. J1587 is an application layer and is used together with J1708, which is the physical layer.

J1587 describes a message format and defines parameters. A J1587 message consists of MID (Message Identification), parameter identification (PID), data bytes, and a checksum. (A checksum is a count of the number of bits in a transmission unit that is included with the unit so that the receiver can check to see whether the same number of bits arrived.) The length of a J1587 message is limited to 21 bytes according to J1708. J1587 allows for sending messages longer than 21 bytes using a connection oriented transport service (COTS). The transmission rate is 9600 bps. A message contains of a one byte long MID, followed by a number of data bytes, and finally a checksum. A message can be up to 21 bytes long.

3.5. Society of Automotive Engineers (SAE) J1939

The SAE J1939 communications network is a high-speed ISO 11898-1 controller area network (CAN)²-based communications network that supports real-time closed loop control functions, simple information exchanges, and diagnostic data exchanges between on-board components, physically distributed throughout the vehicle. The SAE J1939 common communication architecture strives to

² CAN is a form of serial communications that was developed in 1985 for in-vehicle networks.



offer an open interconnect system that allows components associated with different manufacturers to communicate with each other.

The J1939 network was the next generation successor to the SAE J1708 and SAE J1587 low-speed networks. Those earlier standards provided simple information exchange, including diagnostic data, between on-board components. J1939 was capable of performing all of the functions of those earlier networks. It enhanced previous capabilities and added new ones to better support controls and multiplexing on a single network.

Technology had advanced to the point where a high-speed communication network was feasible. It was needed to secure higher bandwidth capabilities for more demanding control needs, so that component suppliers could integrate subsystems for improved performance, and to meet customer expectations and government regulations.

J1939 uses the CAN protocol, which permits any Electronic Control Unit (ECU) to transmit a message on the network when the bus is idle. Every message uses an identifier that defines:

- The message priority
- From whom it was sent
- The data contained within it

SAE J1939 is divided into several layers according to the OSI layer model, where each level is specified in a separate document. In a fashion similar to practically all fieldbus protocols, since layers 5 and 6 are not needed in SAE J1939, they are also not specified.

The functionality of SAE J1939 is divided into layers as follows:

- **Layer 1** (physical layer) describes, among other things, the electric interface with the physical medium.
- **Layer 2** (data link layer) describes the data communication via CAN based on the specification CAN 2.0B.
- **Layer 3** (network layer) essentially describes the functionality of a bridge for the transmission of messages between two network segments and is only relevant for the implementation of a J1939 bridge.
- **Layer 4** (transport layer) describes the various network services for message request mode, acknowledged transmission, and fragmented transmission of large data blocks.
- **Layer 7** (application layer) describes the actual data (parameters or network variables with value range, resolution, physical unit and the type of transmission). Each message is unambiguously referenced by a number (parameter group number).



Since the network management can be regarded as a separate unit that reaches through to the hardware (layer 1), this block in the layer model is shown as an independent function block on the right-hand side. The network management basically consists of the automatic allocation or determination of node addresses (plug & play principle). Node monitoring is not defined in SAE J1939 and must be implemented via cyclic messages at the application level.

The technical details of SAE J1939 are explained in an easy-to-understand overview in <http://www.simmasoftware.com/j1939-presentation.pdf>.

3.6. General Transit Feed Specification (GTFS)

The General Transit Feed Specification (GTFS), originally developed by Google, defines a common format for public transportation schedules and associated geographic information. GTFS "feeds" allow public transit agencies to publish their transit data and developers to write applications that consume that data in an interoperable way.

GTFS contains static schedule information for transit agencies including: stop locations, route geometries, and stop times. GTFS consists of a package of comma-delimited text files, each of which contains one aspect of the transit information and a set of rules on how to record it: six mandatory files (agency, stops, routes, trips, stops times, and calendar) and seven optional files (calendar dates, fare attributes, fare rules, shapes, frequencies, transfers, and feed info). The market success of GTFS has led to an unprecedented adoption rate by transit agencies as shown by total unlinked passenger trips for agencies with GTFS. For schedule data, GTFS adoption has substantially outpaced the TCIP and [Service Interface for Real Time Information] SIRI standards in North America due to its relative ease of use for transit agencies to describe, implement, and maintain data feeds.

GTFS is considered a de facto data standard that facilitates the process for agencies to integrate schedules and routes into Google Maps (now Google Transit), and for broader public disclosure of those same datasets.

The following (from <https://developers.google.com/transit/gtfs/examples/gtfs-feed>) is an example of a GTFS feed. This GTFS feed shows comma-delimited data samples for each file in a transit feed. The sample data files shown here are not all related to each other. You can also download a complete GTFS feed in final form to work with as well.

agency.txt:

```
agency_id,agency_name,agency_url,agency_timezone,agency_phone,agency_lang
```

```
FunBus,The Fun Bus,http://www.thefunbus.org,America/Los_Angeles,(310) 555-0222,en
```



stops.txt:

```

stop_id,stop_name,stop_desc,stop_lat,stop_lon,stop_url,location_type,parent_station

S1,Mission St. & Silver Ave.,The stop is located at the southwest corner of the
intersection.,37.728631,-122.431282,,,

S2,Mission St. & Cortland Ave.,The stop is located 20 feet south of Mission St.,37.74103,-
122.422482,,,

S3,Mission St. & 24th St.,The stop is located at the southwest corner of the intersection.,37.75223,-
122.418581,,,

S4,Mission St. & 21st St.,The stop is located at the northwest corner of the intersection.,37.75713,-
122.418982,,,

S5,Mission St. & 18th St.,The stop is located 25 feet west of 18th St.,37.761829,-122.419382,,,

S6,Mission St. & 15th St.,The stop is located 10 feet north of Mission St.,37.766629,-122.419782,,,

S7,24th St. Mission Station,,37.752240,-122.418450,,,S8

S8,24th St. Mission Station,,37.752240,-
122.418450,http://www.bart.gov/stations/stationguide/stationoverview\_24st.asp,1,

```

routes.txt:

```

route_id,route_short_name,route_long_name,route_desc,route_type

A,17,Mission,"The ""A"" route travels from lower Mission to Downtown.",3

```

trips.txt:

```

route_id,service_id,trip_id,trip_headsign,block_id

A,WE,AWE1,Downtown,1

A,WE,AWE2,Downtown,2

```



stop_times.txt:

```
trip_id,arrival_time,departure_time,stop_id,stop_sequence,pickup_type,drop_off_type
AWE1,0:06:10,0:06:10,S1,1,0,0,0
AWE1,,,S2,2,0,1,3
AWE1,0:06:20,0:06:30,S3,3,0,0,0
AWE1,,,S5,4,0,0,0
AWE1,0:06:45,0:06:45,S6,5,0,0,0
AWD1,0:06:10,0:06:10,S1,1,0,0,0
AWD1,,,S2,2,0,0,0
AWD1,0:06:20,0:06:20,S3,3,0,0,0
AWD1,,,S4,4,0,0,0
AWD1,,,S5,5,0,0,0
AWD1,0:06:45,0:06:45,S6,6,0,0,0
```

calendar.txt:

```
service_id,monday,tuesday,wednesday,thursday,friday,saturday,sunday,start_date,end_date
WE,0,0,0,0,0,1,1,20060701,20060731
WD,1,1,1,1,1,0,0,20060701,20060731
```

calendar_dates.txt:

This example shows service exceptions for the Independence Day holiday in 2006. On Monday, July 3, 2006, regular weekday service (service_id=WD) is interrupted (exception_type=2). Instead, weekend service (service_id=WE) runs on that date (exception_type=1). The same change applies on Tuesday, July 4, as well.



service_id,date,exception_type

WD,20060703,2

WE,20060703,1

WD,20060704,2

WE,20060704,1

fare_attributes.txt:

fare_id,price,currency_type,payment_method,transfers,transfer_duration

1,0.00,USD,0,0,0

2,0.50,USD,0,0,0

3,1.50,USD,0,0,0

4,2.00,USD,0,0,0

5,2.50,USD,0,0,0

fare_rules.txt:

fare_id,route_id,origin_id,destination_id,contains_id

a,TSW,1,1,

a,TSE,1,1,

a,GRT,1,1,

a,GRJ,1,1,

a,SVJ,1,1,

a,JSV,1,1,



```
a,GRT,2,4,
```

```
a,GRJ,4,2,
```

```
b,GRT,3,3,
```

```
c,GRT,,,6
```

shapes.txt:

```
shape_id,shape_pt_lat,shape_pt_lon,shape_pt_sequence,shape_dist_traveled
```

```
A_shp,37.61956,-122.48161,1,0
```

```
A_shp,37.64430,-122.41070,2,6.8310
```

```
A_shp,37.65863,-122.30839,3,15.8765
```

frequencies.txt:

```
trip_id,start_time,end_time,headway_secs
```

```
AWE1,05:30:00,06:30:00,300
```

```
AWE1,06:30:00,20:30:00,180
```

```
AWE1,20:30:00,28:00:00,420
```

transfers.txt:

```
from_stop_id,to_stop_id,transfer_type,min_transfer_time
```

```
S6,S7,2,300
```

```
S7,S6,3,
```

```
S23,S7,1,
```



Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#).

Last updated January 15, 2013.

3.7. GTFS-realtime

GTFS-realtime is a feed specification that allows public transportation agencies to provide real-time updates about their fleet to application developers. It is an extension to GTFS, which was discussed in the prior subsection. GTFS-realtime was designed around ease of implementation, good GTFS interoperability, and a focus on passenger information. It will be discussed more in Modules 6 and 7.

The specification was designed through a partnership among the initial Live Transit Updates partner agencies, a number of transit developers, and Google. The specification was introduced and released under the Creative Commons Attribution 3.0 license in August 2011.

The specification currently supports the following types of information:

- Trip updates – delays, cancellations, changed routes
- Service alerts – stop moved, unforeseen events affecting a station, route, or the entire network
- Vehicle positions – information about the vehicles including location and congestion level

Updates of each type are provided in a separate feed. Feeds are served via HTTP and updated frequently. There are no constraints on how frequently nor on the exact method of how the feed should be updated or retrieved. Because GTFS-realtime allows you to present the actual status of your fleet, the feed needs to be updated regularly – preferably whenever new data comes in from your Automatic Vehicle Location system.

The GTFS-realtime data exchange format is based on Protocol Buffers, which is a language- and platform-neutral mechanism for serializing structured data (think XML, but smaller, faster, and simpler). The data structure is defined in a `gtfs-realtime.proto` file, which then is used to generate source code to easily read and write your structured data from and to a variety of data streams, using a variety of languages (e.g. Java, C++ or Python).

3.8. Other Formats and Standards

There are several other formats that have become de facto standards and are used to exchange data in C2C, C2V and C2I application areas. These include, among others, JSON, Protocol Buffers, REST, SOAP, and XML.

JavaScript Object Notation (JSON) is a data-interchange and text format that is completely language independent, but uses conventions that are familiar to programmers of the C-family of languages. It



is human-readable, platform independent, and enjoys a wide availability of implementations. Data formatted according to the JSON standard is lightweight and can be parsed by JavaScript implementations with ease, making it an ideal data exchange format for certain types of web applications. Since it is primarily a data format, JSON is not limited to just certain types of web applications, and can be used in virtually any scenario where applications need to exchange or store structured information as text.

“Protocol buffers are Google's language-neutral, platform-neutral, extensible mechanism for serializing structured data – think XML, but smaller, faster, and simpler. You define how you want your data to be structured once, then you can use special generated source code to easily write and read your structured data to and from a variety of data streams and using a variety of languages.”³ The GTFS-realtime data exchange format is based on Protocol Buffers.

REST is an architecture style for designing networked applications. The idea is that rather than using complex mechanisms to connect between machines, simple HTTP is used to make calls between machines. RESTful applications use HTTP requests to post data (create and/or update), read data (e.g., make queries), and delete data. One of the case studies we use in this module discusses the use of REST.

The Simple Object Access Protocol (SOAP) facilitates interoperability among a wide range of programs and platforms, making existing applications accessible to a broader range of users. SOAP combines the proven web technology of HTTP with the flexibility and extensibility of XML. SOAP defines a way to move XML messages from point A to point B. It does this by providing an XML-based messaging framework that is 1) extensible, 2) usable over a variety of underlying networking protocols, and 3) independent of programming models.

Extensible Markup Language (XML) is a simple, very flexible text format. Originally designed to meet the challenges of large-scale electronic publishing, XML is playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere. It is used for defining data elements on a Web page and business-to-business documents. XML uses a similar tag structure as HTML; however, whereas HTML defines how elements are displayed, XML defines what those elements contain. While HTML uses predefined tags, XML allows tags to be defined by the developer of the page. Thus, virtually any data items can be identified, allowing web pages to function like database records. By providing a common method for identifying data, XML supports business-to-business transactions and has become "the" format for electronic data interchange and web services.

³ <https://developers.google.com/protocol-buffers/>



4. Case Studies

4.1. C2I Flows: Interurban Transit Partnership (ITP), Grand Rapids, MI

The Interurban Transit Partnership (ITP) deployed real-time information in 2013, using information generated by their CAD/AVL system. In this case, it was required that information be sent from their dispatch center to field infrastructure (dynamic message signs) located at their major transfer facility (called Central Station) and one other transfer location (Kentwood Station). This is a center-to-infrastructure case.

On the first deployment of wayside signs in Grand Rapids, the vendor developed their solution around the use of GTFS data. Each sign was equipped with an on-board PC that processed real-time and static GTFS, and parsed the applicable data relevant to a given sign location. This involved parsing up to 13 separate files that GTFS specifies. Because processors had to preload the GTFS files, it took a lot of central processing unit (CPU) power from the sign and was a slow process. It also meant that GTFS static files had to be up-to-date to display anything on the sign. Because of these limitations with GTFS, the vendor decided to take a step back and look at what data were really needed to be presented on a sign. The vendor then realized that a lot of unnecessary data were being sent down to the sign. From a product standpoint, the vendor also knew that they wanted to use cellular data connections in the future, and wanted to limit the amount of data being sent back and forth between the sign.

Ultimately, the vendor decided to use REST (Representational State Transfer). This provides the most flexibility moving forward and can be consumed by a wide variety of web clients. They are easily able to return JSON, XML, HTTP and other web standard formats. For ITP, using REST provided the ability to return both JSON and XML. Another benefit of REST is that it provides simple ways to filter data. For example, you can request all departures, departures for a specific sign by ID, departures for a specific sign by media access control (MAC) address, or departures for a single stop based on IDs. These are all built from the same data source, so there is consistency across different requests. You can easily add filtering to cut down on the volume of data or rest all the data based on how the application wants to use the information.

With minimal engineering effort, the vendor developed signage REST endpoints (modeled after existing ones for ITP's interactive voice response [IVR], website, and iPhone apps). The sign hits these endpoints, which feeds back JSON data.

Besides providing filtering, using REST can provide you with the specific data you want. You might want to do this to limit data transfers or only provide specific data to certain clients. The amount of effort to do this is minimal, which was another reason for the vendor to choose the REST architecture. For example, for a client that did not need arrival information, the vendor could easily use REST to leave the information out.

From a diagnostics perspective, the sign does an HTTP Post containing all the diagnostics from the sign. This includes number of LEDs out, current temperature, firmware version, errors, and light



sensors. This is consumed by the website, which can process the data and alert the user whenever certain conditions are met. This means that the user will know when the sign stops functioning or if LEDs have failed. This is a proactive approach so the sign is not sitting inoperable for days until it gets reported (usually by a customer).

4.2. C2V Flows: Chattanooga Area Regional Transportation Authority (ARTA)

Beginning in 2006, ARTA required the inclusion of a multiplex system on all buses purchased. This system connected to the SAE J1939 data bus; monitored common engine, transmission, and braking faults transmitted on the data bus (e.g., high engine oil temperature, low oil pressure, high transmission oil temperature); and logged the data for later retrieval. The main purpose of this system was for integration with other planned in-vehicle equipment to eventually provide ARTA with a full remote diagnostics maintenance system. The system is now operational. In 2009, ARTA implemented the daily upload via wireless local area networks (WLAN) of bus diagnostic information collected on-board to the Automatic Vehicle Monitoring (AVM) server, making these data available to maintenance staff. Prior to this, ARTA deployed on-board components for an AVM system on fixed route vehicles, including WLAN communications at both vehicle storage facilities to enable bulk data transfer with vehicles.

The 2009 rollout of the AVM system provided another example of ARTA's commitment to testing. The core infrastructure needed to support AVM – the on-board equipment, the WLAN for daily data uploads, and the AVM software – was in place early in 2008. However, it had been decided to focus 2008 ITS resources on implementing systems that would deliver the most direct and visible benefits most directly and visibly to riders, such as the next-stop announcements and the next-arrival-time predictions. After these systems came online in December 2008, ARTA shifted focus to rolling out AVM. By January 2009, the AVM system elements were integrated and the AVM system was receiving daily uploads of data from the buses. ARTA then conducted internal testing of the AVM system to confirm it was operating correctly before releasing it for use by the mechanics in March 2009.

ARTA included the requirements for the AVM system in the request for proposals (RFP) for their CAD/AVL system. In developing these specifications, they reviewed and selected the most appropriate standards that could facilitate the implementation of AVM. In their review, they assessed standards availability, applicability to their needs, maturity of the standards, and the vendors' use, experience, and acceptance of the standards. This led them to specify the use of either SAE J1708 or J1939 to facilitate the implementation of the AVM system.

After making this selection, they incorporated the requirement of SAE J1708 or J1939 into the specification in several places, as follows:

- Forming a vehicle area network (VAN) connecting the on-board mobile data terminal (MDT) with several other on-board devices (e.g., headsign)
- Integrating the automatic passenger counting (APC) controller with the on-board MDT



- Integrating the on-board MDT with interior DMS
- Integrating the on-board MDT with the VAN to provide Vehicle Component Monitoring (VCM) (collecting codes from Engine Control Module, Transmission Control Module, and Automatic Braking System)

4.3. Tri-Met Case Study

One of the largest and most common hurdles when developing ITS is to make them compatible with existing systems already deployed. There are several important factors that must be considered when integrating new systems with existing ones, and that can have significant impacts on the ITS costs and deployment schedules. These issues include integrating with existing legacy systems to save costs associated with implementing a new system, as well as complying with standards whenever possible.

- Comply with standards and select proven commercial off-the-shelf technology (hardware and software).
- Customized software may need to be developed in order to accommodate the partners' existing legacy systems.
- Integrating with existing legacy systems can save money associated with implementing a new system.
- Agencies may not be able to eliminate all technical issues encountered with integrating ITS components that may occur; however, planning for issues and developing solutions prior to project deployment may help to minimize issues.
- To identify and resolve system integration issues with existing legacy equipment, plan on adequate development time and thorough system testing to ensure systems are working properly after system integration.

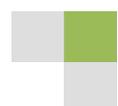
Experiences from the Tri-County Metropolitan Transportation District of Oregon (TriMet) show that following ITS standards and protocols helped ensure that ITS components being integrated can function together. Additionally, following ITS standards and protocols helps provide vendor and system flexibility. In the case of the TriMet, at the time of its procurement of light-emitting diode (LED) signs, no Transmission Control Protocol/Internet Protocol (TCP/IP) standards for the LED sign interface had been developed. Consequently, the agency was forced to consider sign vendors that had proprietary protocols. Even though no standards were available, TriMet knew it wanted the LED signs to interface with TCP/IP-compliant devices, so TriMet provided specifications that required the sign vendors to interface with the protocols. TriMet staff believed that there was an advantage to using TCP/IP and standard protocols that would enable the agency to use different communication methods, yet retain the same applications. Complying with ITS standards and protocols helps to ensure a modular and compatible infrastructure



Further, integrating with existing legacy systems can save money associated with implementing a new system. During procurement of a real-time bus arrival estimation system, TriMet encountered several technical issues that were addressed successfully during project deployment. The Transit Tracker system was built upon the same platform as TriMet's existing automated vehicle location bus dispatch and rail central control systems, saving software development time and system costs. A few minor changes needed to be addressed because of the different requirements necessary for reporting information to customers as opposed to reporting information to the dispatchers. For example, for the real-time Transit Tracker system, TriMet had to change the rate at which information was provided and expand the type of information provided by the system to respond to the needs of the customer.

4.4. Central Puget Sound Case Study

One successful strategy for procuring ITS technologies is to select commercial off-the-shelf technology (hardware and software) that is already proven. The experience of seven public transportation agencies in the Central Puget Sound region demonstrates that complying with standards and using commercial off-the-shelf technology can help save money, minimize risks, and make it easier to integrate existing systems with new ones, and modifying or customizing a particular technology entails greater risks. A modified or customized system has the advantage of closely meeting the specified needs of the regional partnership, along with the disadvantage of needing more development and testing to be sure it does what it is supposed to do. (In the case of the Puget Sound system, only the on-board driver display unit was significantly customized to accommodate emerging smart bus initiatives.) Customized software may need to be developed in order to accommodate the partners' existing legacy systems.



5. Glossary

Term	Definition
Application Area	Deployment-oriented category that focuses on commonly deployed ITS services or systems
Center-to-Center (C2C) Interface Class	This class of application areas includes interfaces between transportation management centers.
Center-to-Field (C2F) Interface Class	This class of application areas includes interfaces between a management center and its field equipment (e.g., traffic monitoring, traffic control, environmental monitoring, driver information, security monitoring, and lighting control).
Center to Vehicle/Traveler (C2V) Interface Class	This class of application areas includes interfaces between a center and the devices used by drivers or travelers. It includes interfaces with motorists and travelers for exchange of traveler and emergency information as well as interfaces between management centers and fleet vehicles to support vehicle fleet management.
Field to Field (F2F) Interface Class	This class of application areas includes interfaces between field equipment, such as between wayside equipment and signal equipment at a highway rail intersection.
Field to Vehicle (F2V) Interface Class	This class of application areas includes wireless communication interfaces between field equipment and vehicles on the road.
Service Package	Service Packages represent slices of the Physical Architecture that address specific services like surface street control. A service package collects together several different subsystems, equipment packages, terminators, and architecture flows that provide the desired service.
Open System Interconnection (OSI)	Open System Interconnection is an ISO standard for worldwide communications that defines a networking framework for implementing protocols in seven layers. Control is passed from one layer to the next, starting at the application layer in one station, proceeding to the bottom layer, over the channel to the next station, and back up the hierarchy.



Term	Definition
General Transit Feed Specification (GTFS)	The General Transit Feed Specification (GTFS), originally developed by Google, defines a common format for public transportation schedules and associated geographic information. GTFS "feeds" allow public transit agencies to publish their transit data and developers to write applications that consume that data in an interoperable way.

6. References

- A Case Study on Applying the Systems Engineering Approach: Best Practices and Lessons Learned from the Chattanooga SmartBus Project, http://ntl.bts.gov/lib/32000/32600/32672/61027_se.pdf
- A Synthesis of ITS Lessons: TECHNICAL INTEGRATION, [http://www.itslessons.its.dot.gov/its/benecost.nsf/files/LessonSynthesis/\\$File/technical%20integration%20synthesis-final.doc](http://www.itslessons.its.dot.gov/its/benecost.nsf/files/LessonSynthesis/$File/technical%20integration%20synthesis-final.doc)
- A202 ITS Standards Training: <http://www.pcb.its.dot.gov/standardstraining/mod07/sup/m07sup.htm> and <http://www.pcb.its.dot.gov/standardstraining/mod07/ppt/m07ppt.htm>
- About ITS Standards, Application Areas, <http://www.standards.its.dot.gov/LearnAboutStandards/ApplicationAreas>
- An Introduction to JavaScript Object Notation (JSON) in JavaScript and .NET, <http://msdn.microsoft.com/en-us/library/bb299886.aspx>
- Carol L. Schweiger, Open Data: Challenges and Opportunities for Transit Agencies, TCRP Synthesis 115, Transportation Research Board, 2015, http://onlinepubs.trb.org/Onlinepubs/tcrp/tcrp_syn_115.pdf
- Carol L. Schweiger and Santosh Mishra, "Utilizing Archived ITS Data: Opportunities for Public Transport," 19th ITS World Congress, Vienna, Austria, 22/26 October 2012, Paper Number AM-00078
- Controller Area Network (CAN) Overview, <http://www.ni.com/white-paper/2732/en/>
- Definition of XML, <http://www.pcmag.com/encyclopedia/term/55048/xml>
- Department for Transport, TransXChange: An XML Standard for the Data Exchange of Bus Schedules and Related Information, TransXChange Schema Guide, <http://81.17.70.199/transxchange/training/EBSR/EBSR%20Training%20Toolkit%20v1.0/3%20Resources/Guides/TransXChange%20Schema%20Guide-2.1-v-44.pdf>
- Example GTFS Feed, <https://developers.google.com/transit/gtfs/examples/gtfs-feed>
- Feed Files, <https://developers.google.com/transit/gtfs/reference#FeedFiles>
- General Transit Feed Specification (GTFS): <https://developers.google.com/transit/gtfs/>



- General Transit Feed Specification (GTFS)-realtime: <https://developers.google.com/transit/gtfs-real-time/>
- GOV.UK, Collection TransXChange, <https://www.gov.uk/government/collections/transxchange>
- GOV.UK, Guidance National public transport access nodes, <https://www.gov.uk/government/publications/national-public-transport-access-node-schema>
- Guide on the use of standards for travel information and retailing, https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/4360/standards-guide.pdf
- <http://rest.elkstein.org/>, <http://rest.elkstein.org/>
- International Standardization in Public Transport: European Committee for Standardization (CEN)/Technical Committee (TC) 278/Working Group (WG)3 – Public Transport http://www.itsstandards.eu/index.php?option=com_content&view=article&id=141&Itemid=87
- International Organization for Standardization (ISO)/Technical Committee (TC) 204/Working Group (WG) 8: http://www.iso.org/iso/home/standards_development/list_of_iso_technical_committees/iso_technical_committee.htm?commid=54706
- Introducing JSON, <http://json.org/>
- Introduction to SAE J1708, <http://www.kvaser.com/about-can/can-standards/j1708>
- ITS ePrimer: Module 7: Public Transportation, <http://www.pcb.its.dot.gov/eprimer/documents/module7p.pdf>
- J1939 Introduction, <http://www.kvaser.com/about-can/higher-layer-protocols/j1939-introduction/>
- Module 6 - A201, Details On Acquiring Standards-Based ITS Systems, <http://www.pcb.its.dot.gov/standardstraining/mod06/ppt/m06ppt.pdf>
- National ITS Architecture 7.0, Transportation Layer: <http://www.iteris.com/itsarch/html/archlayers/transportationlayer.htm>
- National ITS Architecture 7.0, Transit Management Application Area: <http://www.iteris.com/itsarch/html/appareas/ifclass1apparea9.htm>
- National ITS Architecture 7.0, Service Packages, <http://www.iteris.com/itsarch/html/mp/mpindex.htm>
- NIST Information Technology Laboratory, Conformance Testing, <http://www.nist.gov/itl/ssd/is/conformancetesting.cfm>
- NTI course on Systems Engineering for Transit ITS Projects
- Review of standards for travel information and retailing, https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/4361/standards-review.pdf
- SAE J1939 Introduction, http://www.ixxat.com/introduction_sae_j1939_en.html
- SAE J1939 Standards Collection on the Web: Content, <http://www.sae.org/standardsdev/groundvehicle/j1939a.htm>



- Serial Control and Communications Heavy Duty Vehicle Network: http://standards.sae.org/j1939_201308/. Related standards:
 - SAE J1708
 - SAE J1587
 - SAE J1128
 - SAE J1944
- Standards, Intellectual Property Rights (IPRs) and Standards-setting Process, http://www.wipo.int/export/sites/www/sme/en/documents/pdf/ip_standards.pdf
- Systems Engineering for Intelligent Transportation Systems: <http://ops.fhwa.dot.gov/publications/seitsguide/section3.htm>
- The SAE J1939 Communications Network: An overview of the J1939 family of standards and how they are used, <http://www.sae.org/misc/pdfs/J1939.pdf>
- TransXChange: <https://www.gov.uk/government/collections/transxchange>
- Transit Communications Interface Profiles (TCIP): <http://www.aptatcip.com/>
- Understanding SOAP, <http://msdn.microsoft.com/en-us/library/ms995800.aspx>
- Understanding SAE J1939, <http://www.simmasoftware.com/j1939-presentation.pdf>
- User Service Bundles and User Services: <http://www.iteris.com/itsarch/html/user/userserv.htm>
- Webopedia definition of OSI - Open System Interconnection, <http://www.webopedia.com/TERM/O/OSI.html>
- Webopedia definition of The 7 Layers of the OSI Model, http://www.webopedia.com/quick_ref/OSI_Layers.asp
- What is GTFS? <https://developers.google.com/transit/gtfs/>
- What is GTFS-realtime? <https://developers.google.com/transit/gtfs-realtime/>
- What Are Protocol Buffers? <https://developers.google.com/protocol-buffers/>

7. Study Questions

1. How many Public Transportation Service Packages are there in the National ITS Architecture?
 - a) 4
 - b) 10
 - c) 11
 - d) 15
2. Which of these is not a Public Transportation Service Package?
 - a) Transit Vehicle Tracking



- b) Multimodal Connection Protection
 - c) Multimodal Coordination
 - d) Broadcast Traveler Information
3. Which one of these is not a layer within the Open System Interconnection (OSI) model?
- a) Application
 - b) Data
 - c) Service
 - d) Physical
4. Which of these standards are on-board vehicle area network (VAN) standards?
- a) SAE J1939
 - b) ISO 11898
 - c) SAE J1708
 - d) All of the above
5. Which of these issues typically drive the costs associated with using standards?
- a) Adaptation
 - b) Abstraction
 - c) Testing
 - d) All of the above
6. What de facto standard did ITP's vendor use to successfully exchange data between the dispatch center and dynamic message signs in the field?
- a) General Transit Feed Specification (GTFS)
 - b) eXtensible Markup Language (XML)
 - c) JavaScript Object Notation (JSON)
 - d) Representational State Transfer (REST)
7. Does complying with standards and using commercial off-the-shelf technology help make it easier to integrate existing systems with new ones?
- a) Yes
 - b) No



8. A good approach to defining the functions of your system is to select items from manufacturers' datasheets.
 - a) True
 - b) False

9. Which of the following elements are included in a conformance testing program?
 - a) Standard specification
 - b) Procedures for testing
 - c) Organization(s) to do testing, issue certificates of validation, and arbitrate disputes
 - d) All of the above

