Module 7: Traveler Information Standards, Part 2 of 2

Table of Contents

Module Description

This module (Traveler Information Standards, Part 2 of 2) is the second of the two course modules on traveler information using standards. Using the background knowledge provided in Module 6 (Traveler Information, Part 1 of 2), this module begins the process of applying the standards that facilitate the implementation of traveler information technologies. It will provide participants with the knowledge needed to identify and select the most appropriate standard for their particular situation, to use software and hardware standards introduced in the previous module, and to incorporate standards into procurements. The structure and use of data exchange standards for traveler information 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.

By obtaining this detailed knowledge, transit agency staff will assist in reducing the life-cycle cost of these technologies and facilitating the integration with legacy or future technology systems. In addition, it will simplify defining and implementing data exchange among regional or local agencies, and disseminating Traveler Information to customers and the public via a wide variety of media.

1. Introduction/Purpose

Traveler information covers customer-facing technologies that provide the public with information regarding trip planning and real-time operational information. It is generated by on-board and central systems that are used typically to monitor operations (discussed in the Transit Management modules). It can be provided to the public via a variety of dissemination media including electronic signage located at stops/stations, mobile devices, the Internet, 511 systems and interactive voice response (IVR) systems. Agencies can provide traveler information directly to the public and indirectly by making the underlying data open. Open data can be used by developers to create traveler information applications that can be used by the public on mobile devices and the Internet.

2. Samples/Examples

As described in Module 6, traveler information systems cover customer-facing technologies that provide the public with information regarding trip planning and real-time operational information. Traveler information can be generated by on-board and central systems that are used typically to monitor and manage operations (discussed in the Transit Management modules [2 and 5]), as well as those systems that facilitate providing traveler information (e.g., itinerary planning software).

Figure 1 provides context for Traveler Information data exchanges. On the far left of the diagram, CAD/AVL systems that monitor the operations of each mode provide real-time information to the Information Reconciliation process, which takes input from other databases as shown: scheduling, planning, marketing and customer databases. Once all of the information is compiled and reconciled, information is sent through the Data Feed Layer to a Real-time Customer Information Server, which interfaces with the Monitoring/Feedback Layer, which interfaces with Performance Standards and Surveys/Feedback. Then, information is sent through the Dissemination Layer to various end user media, including travelers' devices, information service providers, third-party developers, and within the transit agency to various media (e.g., interactive voice response, dynamic message signs).

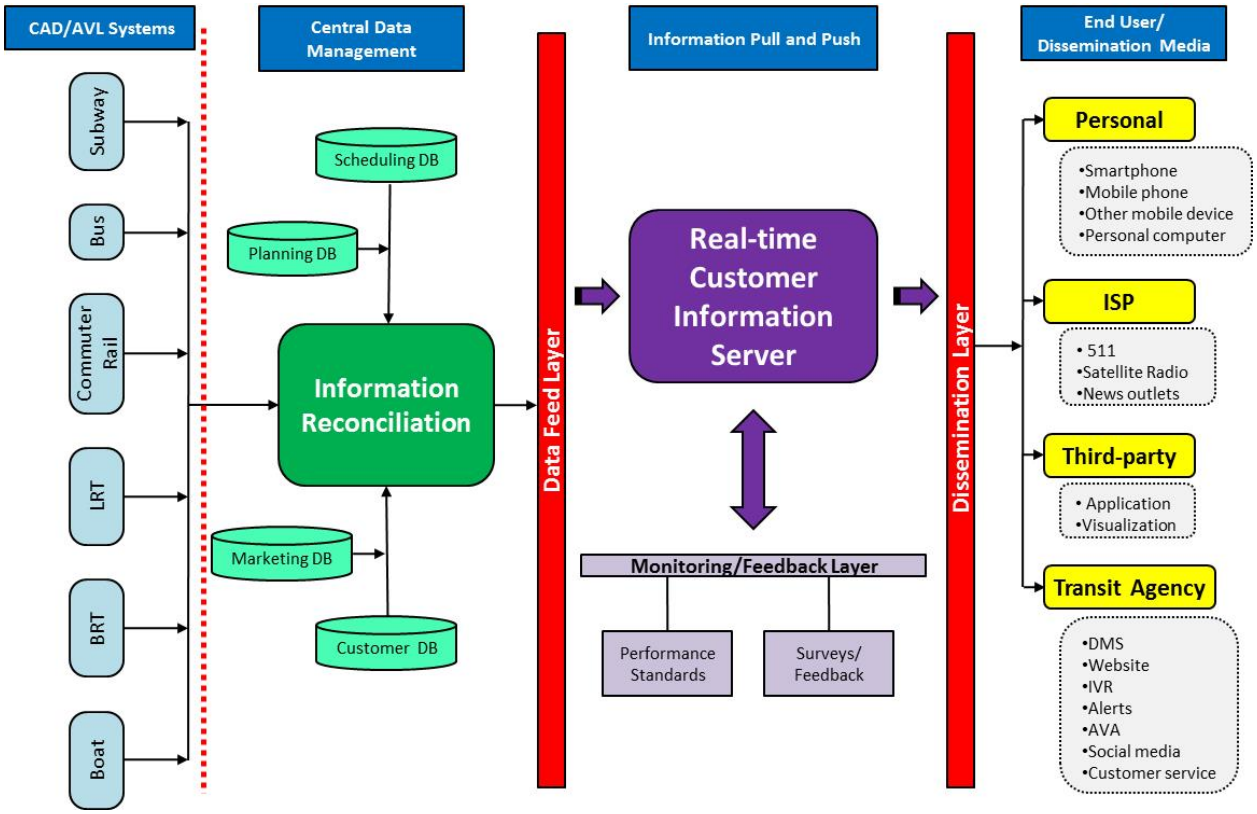

Figure 1. Traveler Information Data Exchanges

Figure 2 shows the relationships between central traveler information, and transit management and other transit ITS technologies. The area within the dark dotted lines represents traveler information-specific elements of the central systems within an agency. Figure 3 shows the relationships among on-board components. The area within the dark dotted lines represents traveler information-specific elements of on-board systems, namely interior dynamic message signs (DMS) and the automatic voice announcement (AVA) system.

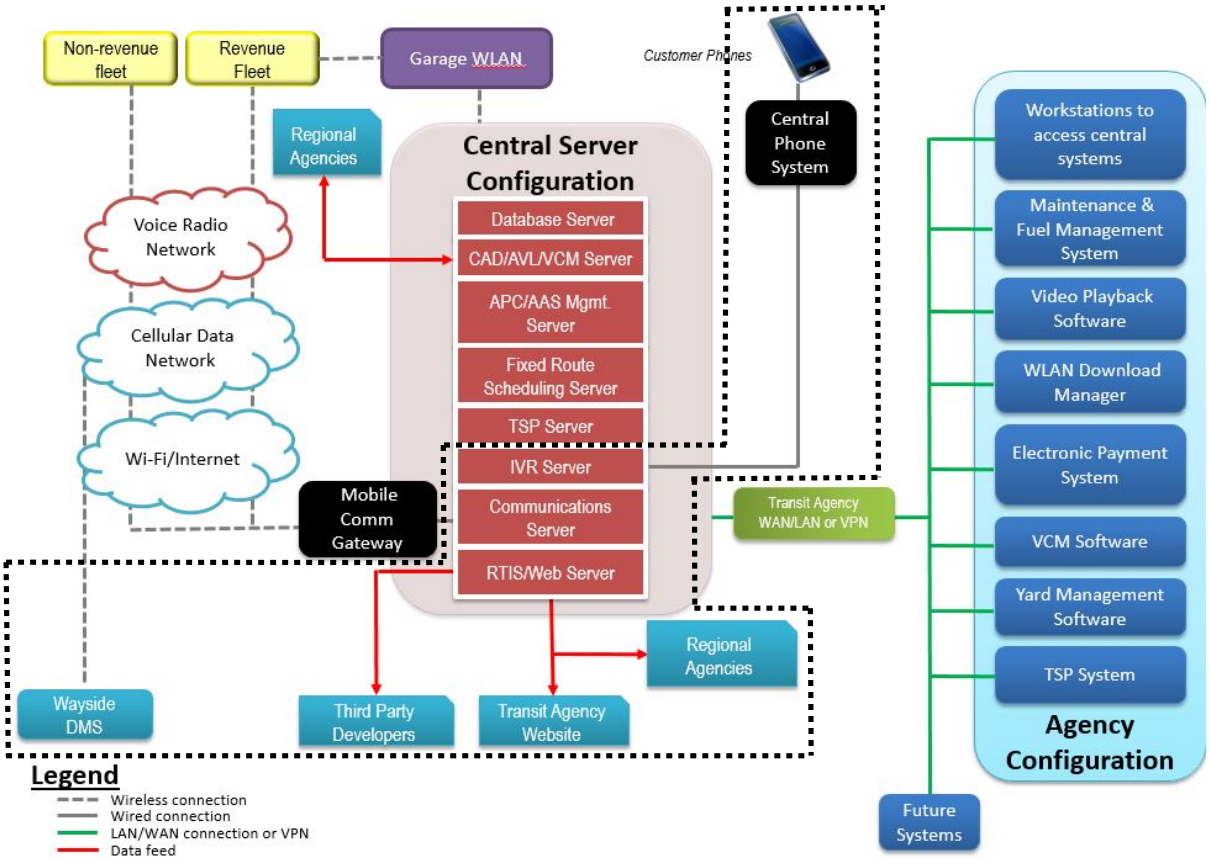

Figure 2. Traveler Information and Transit Management Technology Relationships

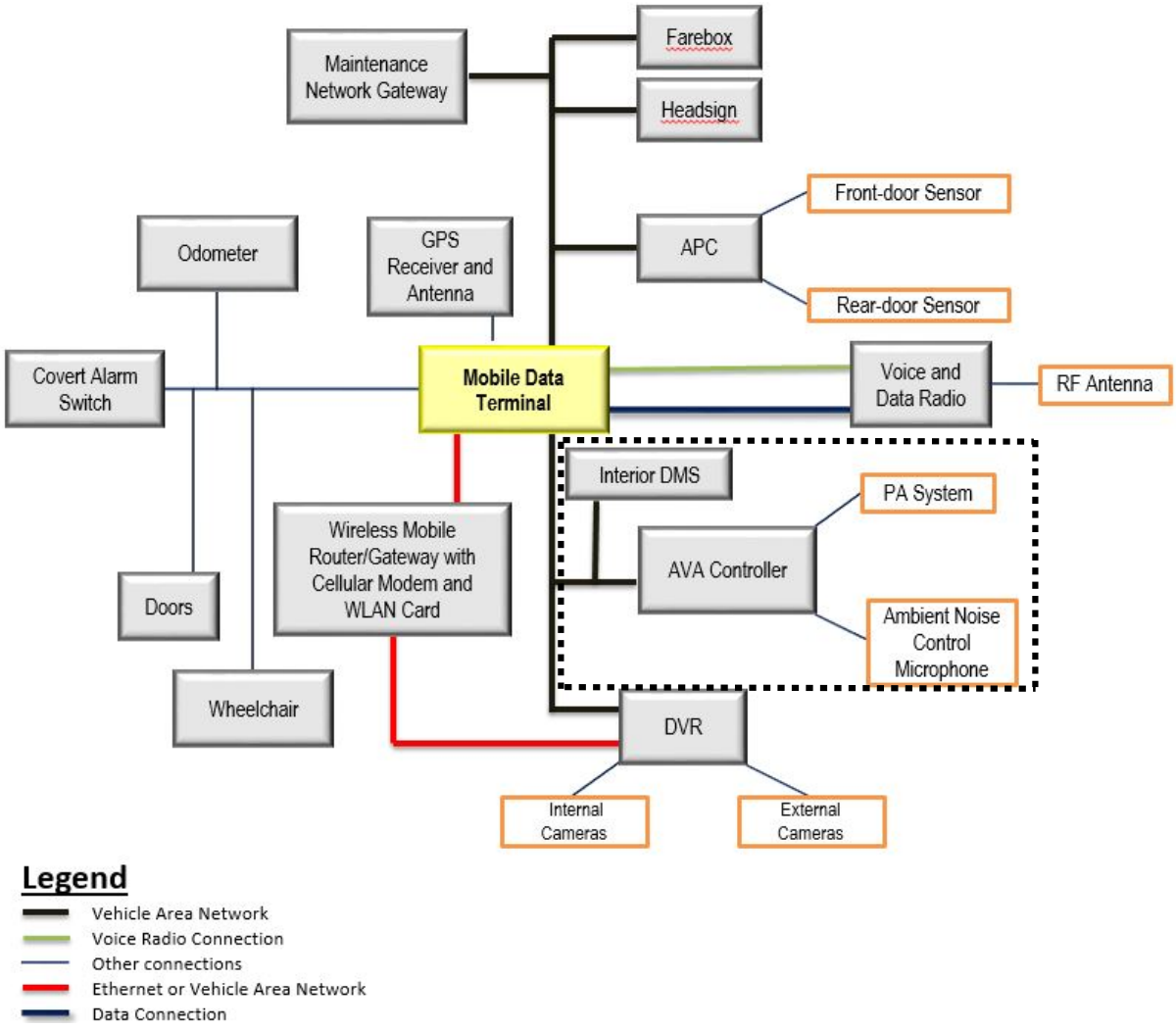

Figure 3. On-Board Technology Relationships

3. Reference to Other Standards

3.1. Application Areas

Application areas, as shown in Table 1, 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.

Table 1. ITS Standards Application Areas

National ITS Architecture Interface Class	Standards Application Areas
Center-to-Center – This class of application areas includes interfaces between transportation management centers.	Data Archival Incident Management Rail Coordination Traffic Management Transit Management Traveler Information
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).	Data Collection/Monitoring Dynamic Message Signs Environmental Monitoring Lighting Management Ramp Metering Traffic Signals Vehicle Sensors Video Surveillance
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.	Mayday Transit Vehicle Communications Traveler Information
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.	Highway Rail Intersection (HRI)
Field-to-Vehicle – This class of application areas includes wireless communication interfaces between field equipment and vehicles on the road.	Probe Surveillance Signal Priority Toll/Fee Collection

Typically, traveler information interface classes are center-to-center (interfaces between transportation management centers) (C2C), center-to-infrastructure (C2I) (interfaces between a management center and its field equipment (e.g., dynamic message signs at stops), and center-to-vehicle/traveler (interfaces between a center and the devices used by transit drivers or travelers) (C2V). An application area within the C2C interface class would include bidirectional communication between a center-type system (e.g., transit dispatch center) and a traveler information provider (called an information service provider or ISP).

3.2. Traveler Information Application Areas

3.2.1. Center-to-Center Application Area

This center-to-center application area covers the interfaces between a traveler information provider (called an information service provider or ISP) and other centers that provide transportation data to the ISP or support traveler services. ISPs typically collect transportation data from a variety of sources, integrate the data, and disseminate the data through many types of distribution channels (i.e., Internet, personal data assistants, kiosks, radio, and television). The application area supports a number of capabilities including:

- Collecting traveler information to provide to subscribers, including air quality data, toll data, event schedules, yellow pages information (e.g., restaurants, lodging, service stations), alternate modes information (e.g., airline, rail, ferry), parking information, and care facility information for emergency situations.
- Integrating traveler data into traveler information, including coordination with other traveler information systems. The traveler information created could support broadcast capability, the creation of personalized travel data, or the creation of route-guidance data.
- Disseminating traveler information to the media.
- Coordinating with providers of yellow pages services for reservations. (e.g., hotels and restaurants).
- Providing route guidance and real-time traffic information to organizations (e.g., fleets of commercial vehicles).
- Providing notification to traffic management centers of anticipated routes for guided vehicles, oversized vehicles or groups of vehicles (e.g., motorcade) that may require changes in the traffic control strategy.
- Supporting requests for traveler information from voice-based traveler information providers, such as 511.
- Coordinating monetary transactions with financial institutions particularly for subscription-based information services.

Figure 4 shows the scope of the center-to-center traveler information application area. This scope is described in the preceding text.

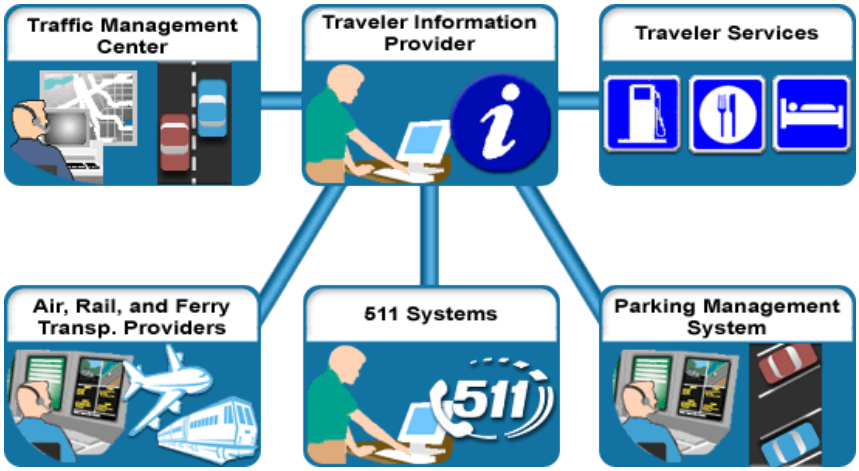

Figure 4. C2C Traveler Information Application Area

In general, the following standards are applicable to traveler information deployments. To determine which specific standards are applicable for a deployment you will need to determine which architecture flows will be needed for the traveler information piece of your deployment.

- ATIS General Use: Advanced Traveler Information Systems (ATIS) General Use Standards Group
- ATIS Low Bandwidth: Advanced Traveler Information Systems (ATIS) Bandwidth Limited Standards Group
- Mayday: On-board Vehicle Mayday Standards Group
- NTCIP C2C: NTCIP Center-to-Center Standards Group
- NTCIP C2F: NTCIP Center-to-Field Standards Group
- ITE TMDD: Traffic Management Data Dictionary (TMDD) and Message Sets for External Traffic Management Center Communications (MS/ETMCC)

3.2.2. Center-to-Traveler Application Area

This center-to-vehicle/traveler application area covers the interfaces between traveler information providers and the devices used by the traveling public. Travelers may access the information either pre-trip or en route using equipment within a vehicle, a personal computer (or personal handheld device), a public kiosk, or equipment at a transit stop or transit facility. Information exchanges, whether personalized, or broadcast to the general public, support the following:

- General traveler information, including traffic information, transit information (fares, real-time schedules, and transactions), incident information, event information, and parking information
- Emergency traveler information, including alerts and advisories, and evacuation information.
- Traveler services information (e.g., dining, lodging, etc.)
- Trip planning, using various modes of surface transportation and route options
- Route guidance

Figure 5 shows the scope of the center-to-center traveler information application area. This scope is described in the preceding text.

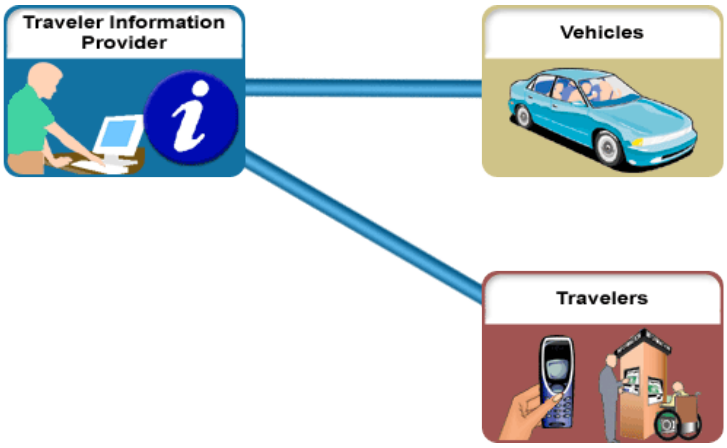

Figure 5. C2V Traveler Information Application Area

In general, the following standards are applicable to Traveler Information deployments. To determine which specific standards are applicable for a deployment you will need to determine which architecture flows will be needed for the traveler information piece of your deployment.

- ATIS General Use: Advanced Traveler Information Systems (ATIS) General Use Standards Group
- ATIS Low Bandwidth: Advanced Traveler Information Systems (ATIS) Bandwidth Limited Standards Group
- CEN SIRI: Intelligent transport systems
- Mayday: On-board Vehicle Mayday Standards Group
- APTA TCIP-S-001 3.0.4: Standard for Transit Communications Interface Profiles

3.3. 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 (see Figure 6). 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.

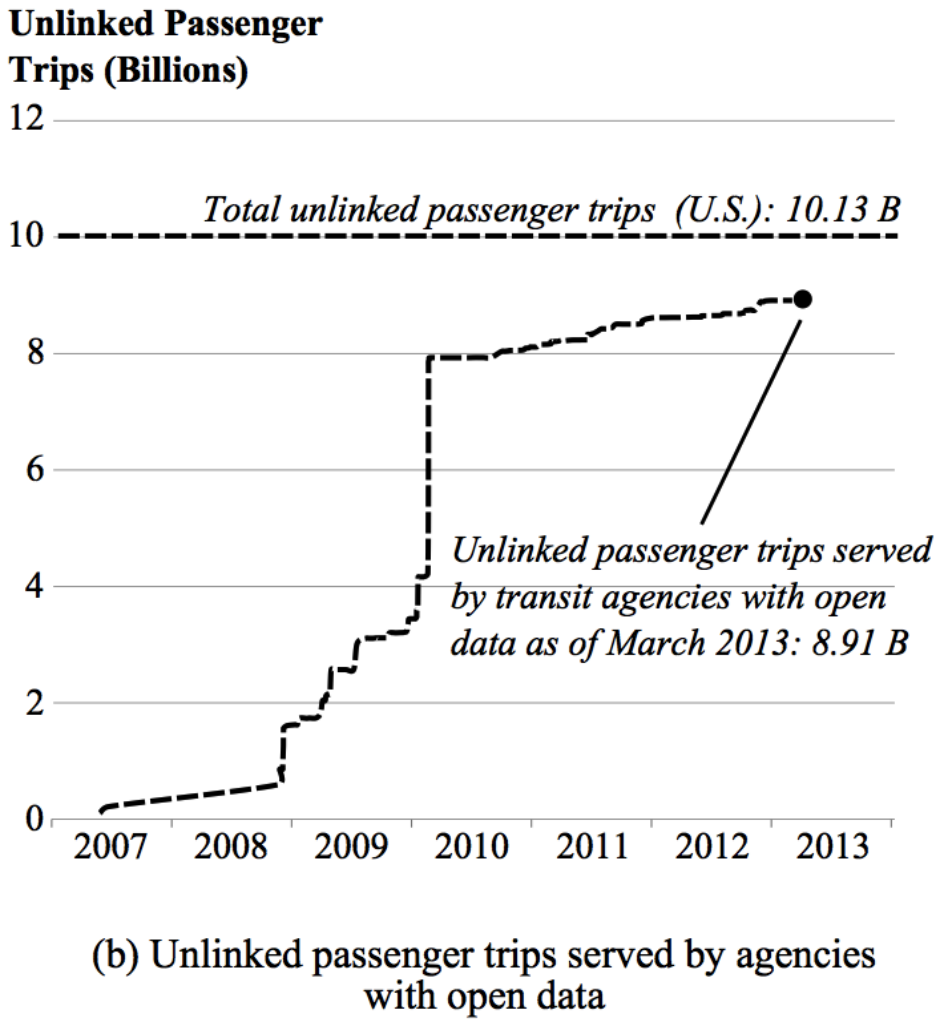

Figure 6. Growth of Transit Agencies with Open Data by Passenger Miles Served (from https://smartech.gatech.edu/bitstream/handle/1853/50218/REED-THESIS-2013.pdf?sequence=1, page 2)

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 is an example of a GTFS feed (from https://developers.google.com/transit/gtfs/examples/gtfs-feed). This example 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,
```


3.4. 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, discussed in the prior subsection. GTFS-realtime was designed around ease of implementation, good GTFS interoperability and a focus on passenger information.

The specification was designed through a partnership of 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 or 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 (AVL) 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).

3.5. Service Interface for Real-time Information (SIRI)

”Service interface for real-time information [SIRI] relating to public transport operations” is a XML protocol for exchange of public transport real-time information. The Comité Européen de Normalisation (French for European Committee for Standardization) (CEN) standard was developed with representatives from France, Germany, Scandinavia and UK and is based on Transmodel. SIRI was established as a European standard in October 2006.

SIRI consist of five parts:

1. CEN TS 15531-1:2006, Part 1: Context and framework
2. CEN TS 15531-2:2006, Part 2: Communications infrastructure
3. CEN TS 15531-3:2006, Part 3: Functional service interfaces
4. CEN/TS 15531-4:2010, Part 4: Functional service interfaces: Facility Monitoring
5. CEN/TS 15531-5:2010, Part 5: Functional service interfaces: Situation Exchange

Each system wishing to exchange information implements the SIRI protocols as XML services. SIRI comprises a general communications architecture, and a number of specific services that operate within that architecture. The communications architecture supports two different patterns of interchange.

- A synchronous request/response protocol: Each exchange of data consists of a request message from a client consumer, and a response message from a producer server.
- An asynchronous subscribe/publish protocol: The client subscribes to information by sending a message to the server containing both request information, and sensitivity criteria with which to filter messages. The producer server establishes a subscription for the consumer and will send messages back to the consumer whenever the criteria are satisfied, until the subscription ends. This pattern is "stateful", that is to say, both parties in the interaction must manage the use of subscriptions that persist over time through successive interactions.

In both cases, the messages consist primarily of XML documents, whose tags and content are exactly specified by the SIRI XML Schemas.

SIRI allows implementers to make different tradeoffs as to message size, use of bandwidth, frequency of update, verbosity of data, sensitivity to change, etc. This is reflected in particular in support for two different patterns of message exchange to return data from a producer server to a consumer client.

- A direct delivery protocol: The data returned in response to a request or subscription consists of a single delivery message, sent directly from the producer server to a client consumer.
- A fetched delivery protocol: The data returned in response to a request or subscription is delivered through an exchange of messages, comprising a notification from the producer server to a client consumer, followed by a request / response interaction from the consumer to the client to fetch the actual payload data. This is also a stateful pattern of communication.

SIRI also has protocols to monitor the system status.

An example of a StopMonitoringRequest consists of a standard header, which is similar for all SIRI requests, and then topics and policies that are specific to the SIRI-SM functional service. The request asks for the departures for a stop HLTST011. The response is to contain up to seven vehicle journeys at the stop, as set by the policy. If there are more than seven available, then only the first two for each line will be shown.

Figure 7 is an example of a StopMonitoringRequest. The request consists of a standard header, which is similar for all SIRI requests, and then topics and policies that are specific to the SIRI-SM functional service. The request asks for the departures for a stop HLTST011. The response is to contain up to seven vehicle journeys at the stop, as set by the policy. If there are more than seven available, then only the first two for each line will be shown.

```
<ServiceRequest>
        <RequestorRef>NADER</RequestorRef>
        <RequestTimestamp>2004-12-17T09:30:47-05:00</RequestTimestamp>
        <StopMonitoringRequest version="1.0">
        <!-- All LINE77services from stop EH00001to destination PLACE457 in the next 30 minutes-->
                <RequestTimestamp>2004-12-17T09:30:47-05:00</RequestTimestamp>
                <MessageIdentifier>NDR06756</MessageIdentifier>
                <!--=======TOPIC ==================================== -->
                <PreviewInterval>P30M</PreviewInterval>
                <MonitoringRef> HLTST011</MonitoringRef>
                <!--=======POLICY=========================================-->
                <MaximumStopVisits>7</MaximumStopVisits>
                <MinimumStopVisitsPerLine>2</MinimumStopVisitsPerLine>
                <StopMonitoringDetailLevel>normal</StopMonitoringDetailLevel>
        </StopMonitoringRequest>
</ServiceRequest>
```

Figure 7. Example of StopMonitoringRequest

3.6. TransXChange

TransXChange provides a means to exchange bus routes and timetables between different computer systems, together with related operational data. Stops are identified using the National Public Transport Access Node Standard (NaPTAN). It is used in particular for the Electronic Submission of Bus Registrations to Vehicle and Operator Services Agency (VOSA) (now called Driver and Vehicle Standards Agency). TransXChange comprises the following main components:

1. A registration and a general version of the TransXChange XML Schema.
2. The TransXChange Publisher: A tool for publishing TransXChange XML documents from TransXChange format into tabular PDF files.
3. Documentation on the TransXChange schema and the processes to use it.
4. Examples of schedules encoded as TransXChange XML documents.

There are two different TransXChange XML schemas, identical except for a few constraints as what fields are required:

- A registration schema for registering a bus timetable with VOSA. This includes mandatory elements for statutory registration.
- A general schema for exchanging bus timetable and ancillary data between systems of all sorts.

The TransXChange schemas are modularized into functional packages and share a common set of base modules with NaPTAN. TransXChange schemas can be used to exchange the following information:

- Bus schedules including stops, routes, departures times / frequencies, operational notes, and maps. Routes may have complex topologies such as circular routes, cloverleaf and lollipops, and complex workings such as short working and express patterns. Connections with other services can also be described.
- The days on which the services run, including availability on public holidays and other exceptions.
- Term times and holidays of Schools, Local Educational Authorities, and other organizations serviced by a bus service.
- Details of the statutory Registration of the Service with a Traffic Area Office, including any Short Notice Registration details.
- Information about the Bus Operators providing the service.
- Additional operational information, including fare stages, positioning runs, garages, layovers, duty crews, useful for AVL and on-board ticketing systems.
- Information about accessibility of stops and services for wheelchair and other users.

The TransXChange model) has seven basic concepts: Service, Registration, Operator, Route, StopPoint, JourneyPattern, and VehicleJourney:

- A Service brings together the information about a registered bus service, and may contain two types of component service: Standard or Flexible; a mix of both types is allowed within a single Service.
- A normal bus schedule is described by a StandardService and a Route. A Route describes the physical path taken by buses on the service as a set of route links.
- A FlexibleService describes a bus service that does not have a fixed route, but only a catchment area or a few variable stops with no prescribed pattern of use.
- A StandardService has one or more JourneyPattern elements to describe the common logical path of traversal of the stops of the Route as a sequence of timing links (see later), and one or

more VehicleJourney elements, which describe individual scheduled journeys by buses over the Route and JourneyPattern at a specific time.

- Both types of service have a registered Operator, who runs the service. Other associated operator roles can also be specified.
- Route, JourneyPattern and VehicleJourney follow a sequence of NaPTAN StopPoints. A Route specifies in effect an ordered list of StopPoints. A JourneyPattern specifies an ordered list of links between these points, giving relative times between each stop; a VehicleJourney follows the same list of stops at specific absolute passing times. (The detailed timing Link and elements that connect VehicleJourneys, JourneyPatterns etc., to StopPoints are not shown in Figure 3-1). StopPoints may be grouped within StopAreas.
- The StopPoints used in a JourneyPattern or Route are either declared locally or by referenced to an external definition using an AnnotatedStopRef.
- A Registration specifies the registration details for a service. It is mandatory in the registration schema.

Figure 8 introduces, in UML class diagram notation, the core elements of the TransXChange schema. Reusable elements with a global scope are organized beneath the root TransXChange.

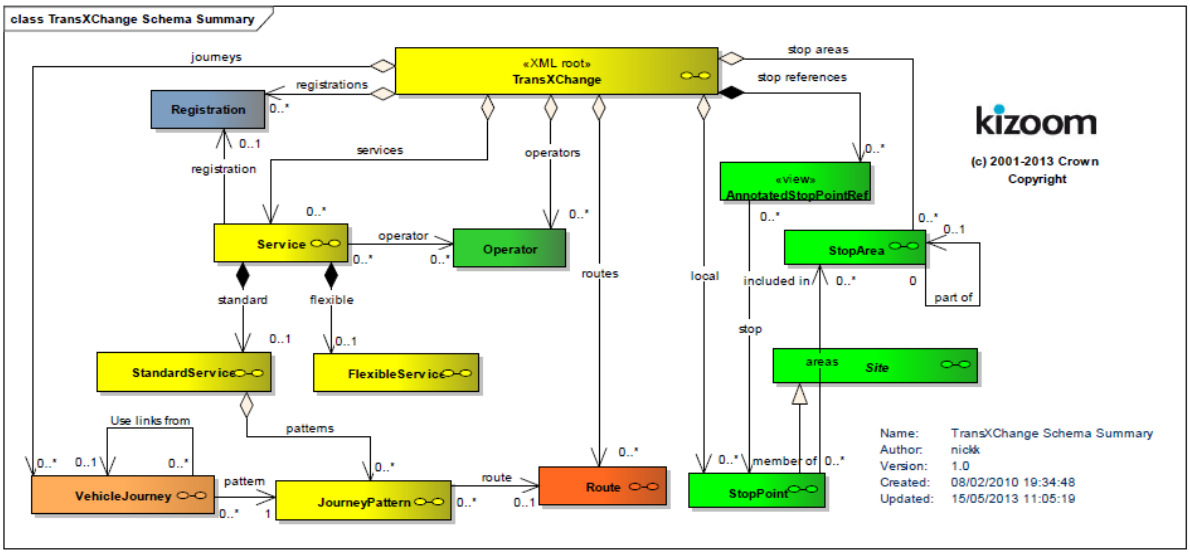

Figure 8. TransXChange Model

3.7. Other Formats and Standards

Several other formats have become defacto 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, and 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.” 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.

4. Case Study: Example of Relating Requirements to Specific Standards

4.1. Selection of Standards: Massachusetts Bay Transportation Authority (MBTA), Boston, MA

In 2009, the Massachusetts Bay Transportation Authority (MBTA) opened their data to the public. At that time, only schedule data was being provided using GTFS. Beginning in 2010, real-time information was added to the open data program. In 2012, the MBTA participated in the pilot of GTFS-realtime. Given the rapid development of the open data program, the underlying systems providing the data were Independent and the data feeds were incompatible due to the use of a variety of standards and formats (see next slide).

The choice of standards used for the open data was based on what was needed within an Application Programming Interface (API) and what was available in the marketplace.

When the MBTA reviewed potential standards, agency personnel wanted to support GTFS-realtime and wanted to be on Google. (The MBTA was one of the first transit agencies in the United States to provide real-time information on Google Transit.) They looked at SIRI but found it to be verbose and somewhat complicated, so they decided against using it. However, if SIRI does become more prevalent among developers, they could support it. TCIP was well-suited for communications within an agency, but not for communications with developers, so it was not selected for use with the open data. The MBTA developed an API in order to retrieve smaller sets of information than what is contained in GTFS-realtime (GTFS-RT) and include some information that is not in GTFS-RT. The MBTA selected XML format for its API because it is an industry standard for APIs.

Figure 9 shows the use of multiple standards when the MBTA first provided open data.

In 2013, the MBTA began to re-organize the feeds and standards (as shown on the next slide) by developing new MBTA-realtime.

Software. This reorganization consisted of the following steps:

- C# application with Microsoft SQL Server back-end on two Amazon cloud servers
- Based around foundation of GTFS data
- Feeds, MBTA website, subscription service
- Email and SMS sent by GovDelivery

- Phase 1 launched 2013, Phase 2 launched Spring 2014

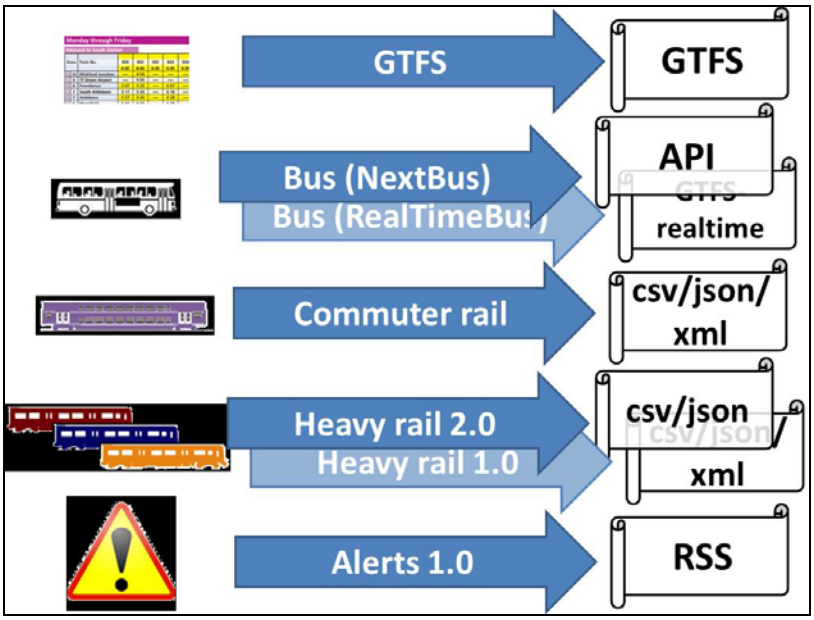

Figure 9. Multiple Standards when MBTA Open Data First Available

The MBTA's experience with standards for their open data program is that GTFS, GTFS-realtime, and their API are popular with developers. There have been no requests for TCIP or SIRI to date.

Figure 10 shows the reorganization of the open data program and use of standards.

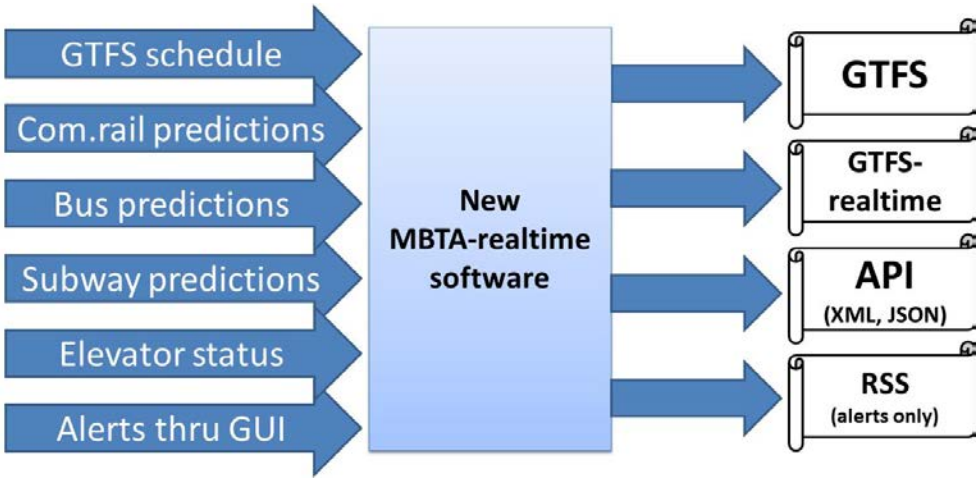

Figure 10. Reorganization of MBTA Open Data and Standards

4.2. Changing Traveler Information Standards: Metropolitan Transportation Authority (MTA), New York, NY

The Metropolitan Transportation Authority (MTA) in New York City provides free real-time developer API's to the MTA Bus Time system. After some research and consultation with the local developer community, MTA identified the SIRI standard and are using it. The open nature of this standard and its growing adoption were the two primary selling points to MTA in the context of the open-architecture, standards-based MTA Bus Time project. Based on their research, this project represented the first SIRI implementation that is public-facing and free to access.

One thing SIRI did not cover is the "distance away" (rather than time-based predictions) concept that MTA introduced in the Bus Time project. SIRI is an extensible standard so MTA was able to make it work for this project. As well, while SIRI requests are typically SOAP requests sent via HTTP POST, in this case MTA implemented a slimmed-down RESTful interface using HTTP GET requests.

The SIRI web service calls implemented by MTA Bus Time are:

- VehicleMonitoring: Real-time information about one, many, or all vehicles tracked by the system
- StopMonitoring: Real-time information about vehicles serving a particular stop

While the SIRI standard specifies XML as an interchange format, it incurs a substantial penalty in data-transfer and processing requirements. In contrast, JSON (Javascript Object Notation) is a lightweight data-interchange format. Human-readable, lightweight, and easy to parse, JSON offers an alternative. Thus, MTA Bus Time provides both XML and JSON versions of its API.

The two SIRI calls use the MTA's GTFS data as a reference, for example for stop ID's and names. The OneBusAway software that powers MTA Bus Time comes with a free utility to strip down a GTFS file to only the data relevant to a given route. Developers can use this utility to generate the GTFS for only the routes that are served by Bus Time.

The body/committee which manages the SIRI standard has adopted many of the changes/extension MTA made for this project into the forthcoming SIRI 2.0 specification, including: inclusion of "distance-away" values, RESTful request-response, and non-XML encodings.

MTA Bus Time uses SIRI to expose real-time information about buses serving the routes and stops covered by the system. MTA Bus Time also exposes a modified version of the OneBusAway RESTful API for so-called "discovery" services. This API allows the discovery of static/baseline information about the bus services covered under MTA Bus Time. It is, effectively, a set of web services that gives the user convenient HTTP-based access to selected subsets of information otherwise available from the GTFS files.

OneBusAway is an open source platform for real-time transit information. Ten criteria define open source software:

- Free redistribution
- Program must include source code
- Allow modifications and derived works
- Permit distribution of software built from modified source code
- No discrimination against persons or groups
- No discrimination against fields of endeavor
- Rights must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties
- License must not be specific to a product
- License must not restrict other software
- License must be technology-neutral

There is a difference between open source software/platform and open standards and open data. An open standard is a standard made available to the general public, and is developed (or approved) and maintained via a collaborative and consensus driven process. Open data is information that is available for anyone to use, for any purpose, at no cost. Open data has to have a license that says it is open data.

5. Glossary

Term	Definition
Comma-separated values (CSV)	A file format used as a portable representation of a database. Each line is one entry or record and the fields in a record are separated by commas. Agencies using GTFS have committed to producing and maintaining their schedule data in standardized CSV tables to display their system on Google Transit's trip planner and, increasingly, opening this data to other third-party application developers.
eXtensible Markup Language (XML)	The XML format is more robust than GTFS in its abilities to represent large complex models, but the approach is more common in Europe and raises standardization challenges in the face of hyper flexibility.

Term	Definition
General Transit Feed Specification (GTFS)	The General Transit Feed Specification, originally developed by Google, 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)" (8, page 1).
Geo JavaScript Object Notation (GeoJSON)	GeoJSON is a format for encoding a variety of geographic data structures. Geospatial data interchange format based on JavaScript Object Notation (JSON)
GTFS-realtime	GTFS-realtime contains real-time information related to vehicle positions, service alerts and trip updates (delays, cancellations, etc.)
Identification of Fixed Objects in Public Transport (IFOPT)	IFOPT defines a model and identification principles for the main fixed objects related to public access to Public Transport (e.g. stop points, stop areas, stations, connection links, entrances, etc.). IFOPT Standard builds on the TransModel Standard to define four related sub models.
JavaScript Object Notation (JSON)	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. (http://json.org/)
Network Exchange (NeTEx)	NeTEx is intended to be a general purpose format capable of exchanging timetables for Rail, Bus, Coach, Ferry, Air, or any other mode of public transport.
Protocol Buffers	GTFS-realtime data exchange format is based on Protocol Buffers. Protocol buffers are 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 is then 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.
Really Simple Syndication or Rich Site Summary (RSS)	RSS is a format for delivering regularly changing web content.
Representational state transfer (REST)	REST is a distributed system framework that uses Web protocols and technologies. (40)
Resource Description Framework (RDF)	RDF is a standard model for data interchange on the Web. (39)

Term	Definition
Service Interface for Real Time Information (SIRI)	SIRI is a real-time data standard predominant in Europe, but making significant inroads into the US market, notably at [Metropolitan Transportation Authority] MTA in New York. Recent change proposals to the SIRI standard include the definition of a structure for SIRI web services. The SIRI standard includes a component for schedule data, but is designed for real time and is therefore more complex than some other standards.
Service Package	The service packages, formerly known as market packages, provide an accessible, service-oriented perspective to the National ITS Architecture. They are tailored to fit, separately or in combination, real world transportation problems and needs. Service packages collect together one or more equipment packages that must work together to deliver a given ITS service and the architecture flows that connect them and other important external systems. In other words, they identify the pieces of the physical architecture that are required to implement a particular ITS service. Service packages are implemented through projects (or groups of projects, aka programs) and in transportation planning, are directly related to ITS strategies used to meet regional goals and objectives.
Simple Object Access Protocol (SOAP)	SOAP is a method of transferring messages, or small amounts of information, over the Internet. SOAP messages are formatted in XML and are typically sent using HTTP (hypertext transfer protocol).
Systems Engineering	An interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem. Systems engineering integrates all the disciplines and specialty groups into a team effort forming a structured development process that proceeds from concept to production to operation. Systems engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs.
TransModel	TransModel is the European Reference Data Model for Public Transport; it provides an abstract model of common public transport concepts and structures that can be used to build many different kinds of public transport information system, including for timetabling, fares, operational management, real-time data, etc. (43)

Term	Definition
TransXChange (TxC)	TxC is the UK nationwide standard for exchanging bus schedules and related data. TxC provides a means to exchange bus routes and timetables between different computer systems, together with related operational data.

6. References

- The National ITS Architecture 7.0, ITS Standards, http://www.iteris.com/itsarch/html/standard/standard.htm
- The National ITS Architecture 7.0, Service Packages, http://www.iteris.com/itsarch/html/mp/mpindex.htm
- The National ITS Architecture 7.0, APTS08-Transit Traveler Information, http://www.iteris.com/itsarch/html/mp/mpapts08.htm
- Dr. Ioanna Spyropoulou, "Pre-Trip Impacts of Advanced Traveller Information Systems," http://www.noehumanist.org/documents/DesignGuidelines-useracceptance-and-Impact-of-ITS/User-acceptance-and-impact-of-ITS/IoI2_ispyropoulou.pdf
- Carol L. Schweiger, Strategies for Improved Traveler Information, TCRP 92, 2003, http://onlinepubs.trb.org/onlinepubs/tcrp/tcrp_rpt_92.pdf
- Carol L. Schweiger, Use and Deployment of Mobile Device Technology for Real-Time Transit Information, TCRP Synthesis 91, 2011, http://onlinepubs.trb.org/onlinepubs/tcrp/tcrp_syn_91.pdf
- Susan Bregman, Uses of Social Media in Public Transportation, TCRP Synthesis 99, 2012, http://onlinepubs.trb.org/onlinepubs/tcrp/tcrp_syn_99.pdf
- Carol L. Schweiger, ITS ePrimer Presentation, Module 7: Public Transportation, http://www.pcb.its.dot.gov/eprimer/module7p.aspx
- Carol L. Schweiger, Use of Electronic Passenger Information Signage in Transit, TCRP Synthesis 104, 2013, http://onlinepubs.trb.org/onlinepubs/tcrp/tcrp_syn_104.pdf
- Live Transit Updates, https://developers.google.com/transit/google-transit#LiveTransitUpdates
- DATEX II V2.0 USER GUIDE, Document version: 1.0, 01 July 2009, European Commission Directorate General for Transport and Energy, http://www.datex2.eu/sites/www.datex2.eu/files/sites/test.datex2.eu/files/DATEXIIv2.0-UserGuide_v1.0.pdf
- General Transit Feed Specification Reference, https://developers.google.com/transit/gtfs/reference

- What is GTFS-realtime? https://developers.google.com/transit/gtfs-realtime/
- NextBus, Public XML Feed Revision 1.22 April 4, 2013, http://www.nextbus.com/xmlFeedDocs/NextBusXMLFeed.pdf
- TransXChange, https://www.gov.uk/government/collections/transxchange
- Systems Engineering Role in the Acquisition Process, http://sebokwiki.org/wiki/Systems_Engineering_and_Procurement/Acquisition#Systems_Engineering_Role_in_the_Acquisition_Process
- Systems Engineering for Intelligent Transportation Systems, Chapter 3 - What is Systems Engineering? http://ops.fhwa.dot.gov/publications/seitsguide/section3.htm
- National Transit Institute Course "Systems Engineering for Technology Projects"
- Systems Engineering for Intelligent Transportation Systems, Chapter 6 - Applying Systems Engineering, http://ops.fhwa.dot.gov/publications/seitsguide/section6.htm
- Guide to Contracting ITS Projects, NCHRP 560, 2006, http://onlinepubs.trb.org/onlinepubs/nchrp/nchrp_rpt_560.pdf
- NCHRP Project 03-77: Guide to Contracting ITS, http://www.citeconsortium.org/Model/index.htm
- 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
- 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
- Transport Direct, "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
- Transport Direct, "Review of standards for travel information and retailing," https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/4361/standards-review.pdf
- About ITS Standards, Application Areas, http://www.standards.its.dot.gov/LearnAboutStandards/ApplicationAreas

- Dynamic Message Signs Application Area, http://www.iteris.com/itsarch/html/appareas/ifclass2apparea2.htm
- CEN TS 15531 Service Interface for Real Time Information (SIRI), https://www.vdv.de/siri.aspx
- Service Interface for Real Time Information, CEN/TS 15531 (prCEN/TS-OO278181), http://user47094.vs.easily.co.uk/siri/index.htm
- Department for Transport, TransXChange – An XML Standard for the Data Exchange of Bus Schedules and Related Information, TransXChange Schema Guide 2.5, http://www.transxchange.org.uk/schema/2.5/doc/TransXChangeSchemaGuide-2.5-v-59.pdf
- Introducing JSON, http://json.org/
- An Introduction to JavaScript Object Notation (JSON) in JavaScript and .NET, http://msdn.microsoft.com/en-us/library/bb299886.aspx
- Protocol Buffers, https://developers.google.com/protocol-buffers/
- Learn REST: A Tutorial, http://rest.elkstein.org/
- Understanding SOAP, http://msdn.microsoft.com/en-us/library/ms995800.aspx
- Definition of XML, http://www.pcmag.com/encyclopedia/term/55048/xml
- Application Areas, Transit Management, http://www.standards.its.dot.gov/ApplicationArea/2
- Carol L. Schweiger, "Open Data: Challenges and Opportunities for Transit Agencies," TCRP Synthesis 115, http://onlinepubs.trb.org/Onlinepubs/tcrp/tcrp_syn_115.pdf
- MTA Bus Time, Introduction to SIRI, http://bustime.mta.info/wiki/Developers/SIRIIntro
- A Synthesis of ITS Lessons, Technical Integration, http://www.itslessons.its.dot.gov/its/benecost.nsf/files/LessonSynthesis/$File/technical%20integration%20synthesis-final.doc
- TransXChange, https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/313471/1-1_Overview.pdf
- Conformance Testing, http://www.nist.gov/itl/ssd/is/conformancetesting.cfm
- Standards, Intellectual Property Rights (IPRs) and Standards-setting Process, http://www.wipo.int/export/sites/www/sme/en/documents/pdf/ip_standards.pdf

7. Study Questions

1. On-board automated voice announcements (AVA) are dependent upon which of these?
    a) AVL system
    b) APC system
    c) Route and vehicle schedule data
    d) All of the above
    e) A and C

2. Which one of these standard is not a traveler information standard?
    a) SIRI
    b) SAE J1939
    c) TransXChange
    d) GTFS

3. Which standard is not being used by the MBTA for their open data program?
    a) General Transit Feed Specification (GTFS)
    b) eXtensible Markup Language (XML)
    c) JavaScript Object Notation (JSON)
    d) Service Interface for Real Time Information (SIRI)

4. Which of the following elements are included in a conformance testing program?
    a) Standard or specification
    b) Procedures for testing
    c) Organization(s) to test/issue certificates of validation
    d) All of the above