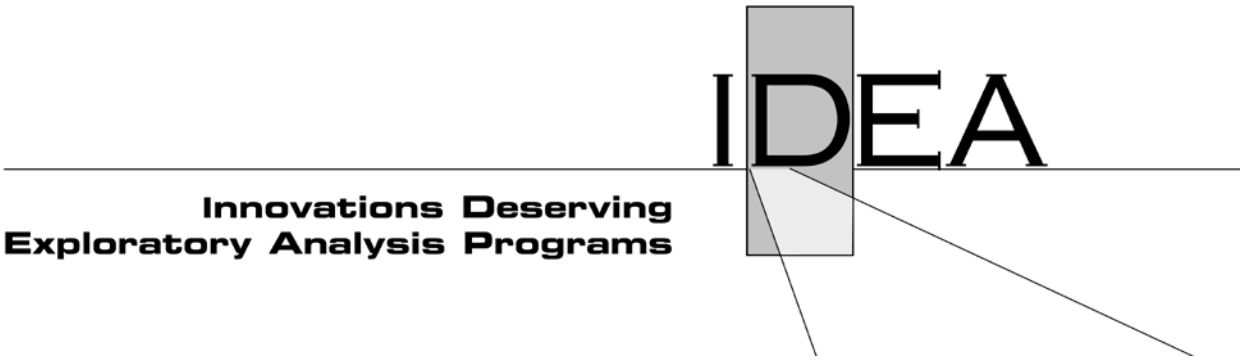


Transit IDEA Program

An Open Platform for Transit Agencies to Improve the Quality of Their Real-Time Data

Final Report for
Transit IDEA Project 93

Prepared by:
Drew Dara-Abrams, Ph.D.
Interline Technologies LLC

October 2020

TRB TRANSPORTATION RESEARCH BOARD
The National Academies of
SCIENCES · ENGINEERING · MEDICINE

Innovations Deserving Exploratory Analysis (IDEA) Programs Managed by the Transportation Research Board

This IDEA project was funded by the Transit IDEA Program.

The TRB currently manages the following three IDEA programs:

- The NCHRP IDEA Program, which focuses on advances in the design, construction, and maintenance of highway systems, is funded by American Association of State Highway and Transportation Officials (AASHTO) as part of the National Cooperative Highway Research Program (NCHRP).
- The Rail Safety IDEA Program currently focuses on innovative approaches for improving railroad safety or performance. The program is currently funded by the Federal Railroad Administration (FRA). The program was previously jointly funded by the Federal Motor Carrier Safety Administration (FMCSA) and the FRA.
- The Transit IDEA Program, which supports development and testing of innovative concepts and methods for advancing transit practice, is funded by the Federal Transit Administration (FTA) as part of the Transit Cooperative Research Program (TCRP).

Management of the three IDEA programs is coordinated to promote the development and testing of innovative concepts, methods, and technologies.

For information on the IDEA programs, check the IDEA website (www.trb.org/idea). For questions, contact the IDEA programs office by telephone at (202) 334-3310.

IDEA Programs
Transportation Research Board
500 Fifth Street, NW
Washington, DC 20001

An Open Platform for Transit Agencies to Improve the Quality of Their Real-Time Data

IDEA Program Final Report

For the period *January 2019* through *September 2020*

Contract Number TRANSIT-93

Prepared for the IDEA Program

Transportation Research Board

National Academies of Sciences, Engineering, and Medicine

Drew Dara-Abrams, Ph.D.
Interline Technologies LLC
October 14, 2020

Acknowledgments

Research team members include:

- Ian Rees, Ph.D.
  Interline Technologies LLC
  Principal
- Sean Barbeau, Ph.D.
  Center for Urban Transportation Research (CUTR), University of South Florida (USF)
  Principal Mobile Software Architect for R&D

Thanks to members of the T-93 expert review panel (ERP):

- Patricia Collette
- Stephen M. Stark
- Aaron Antrim
- Carol Schweiger
- Murat Omay
- Santosh Mishra

Thanks to Velvet Basemera-Fitzpatrick, Ph.D., Transit IDEA Program Manager.

Thanks to the following transit agencies and staff members for participating in user testing:

- Metropolitan Transportation Commission (San Francisco): Nisar Ahmed
- Metro Transit (Minneapolis): Laura Matson, Mark DiPasquale, Gary Nyberg, Joey Reid, Juan Villanueva
- Massachusetts Bay Transportation Authority (Boston): Paul Swartz, Jessie, Richards, Logan Nash
- TriMet (Portland, Oregon): Mike Gilligan, Guy Tinat
- Tompkins Consolidated Area Transit (Ithaca, New York): Tom Clavel, Matthew Yarrow
- Washington Metropolitan Area Transit Authority (Washington, D.C.): Stephanie Jones, Richard Carman
- Hillsborough Area Regional Transit (Tampa, Florida): Dexter Corbin, William Mozel, Tim Wictor
- Rogue Valley Transportation District (Medford, Oregon): Melissa Lowry

Glossary

API	Application Programing Interface - a software component that provides data, querying, and/or control capabilities to other, external software and users. Web platform APIs typically produce data in formats such as JSON or PBF. Web platform APIs typically have a series of URLs, which can also be referred to as API endpoints.
CAD/AVL	Computer Aided Dispatch and Automatic Vehicle Location - hardware and software systems that allow transit agency staff to manage and track the locations of their vehicle fleet. Often use radio frequency (RF) or cellular data connections to communicate between vehicles and control centers. CAD/AVL data is often used as an input to create GTFS Realtime feeds. CAD/AVL systems vary in how frequently they produce and send location data, from every few seconds to every few minutes.
CSV	Comma-separated values - a text file in which data fields are delimited by comma characters. Often begins with a header row that lists the names of each column (again separated by commas). These files typically end with a .csv file extension. However, when included in GTFS feed archives, they are given a .txt file extension. Microsoft Excel and other spreadsheet software can import and export from a CSV file to a spreadsheet.
DMFR	Distributed Mobility Feed Registry is a data schema defined by the Transitland project for cataloging GTFS and GTFS Realtime feeds. A DMFR file is a JSON file that includes references to one or more static GTFS URLs and/or GTFS Realtime API endpoints, along with metadata about the feed(s). DMFR files can be stored and shared by any means. Transitland 2.0, created during the course of this project, stores DMFR files in a Git repository on GitHub, for public use and contribution.
Endpoint	Endpoints refer to particular URLs that make up a web platform API. Each endpoint is particularly associated with one action or resource.
Git	A version control system typically used to store and track changes to software. Git was originally created as part of the

	Linux operating system effort, and has become widely used for many open-source software projects. GitHub, now owned by Microsoft, is a centralized service that hosts Git repositories (that is, projects) for many individuals and organizations. Note that the Git software can run without GitHub, although the two have become almost synonymous.
GPS	Global Position System - GPS is often used to refer to any system in which a device uses satellite signals to position itself. (GNSS, or global navigation satellite system, is the generic term that also includes satellite positioning systems deployed by other countries, but is less often used within the transit industry.) GPS receivers are now built into smartphones and many other consumer devices. CAD/AVL systems typically use antennas mounted on top of vehicles connected to industrial-grade GPS devices within the vehicle. This GPS device provides the vehicle location that is sent back to the control center and used to power a GTFS Realtime vehicle locations feed.
GTFS	General Transit Feed Specification - a data format used to represent public-transit agency services. Each GTFS feed is a ZIP archive of CSV files. Required CSV files describe agencies, stops, routes, trips, stop times, and calendars. Additional CSV files can be added to describe frequency-based route schedules, route alignments/shapes, transfers between routes, and so on. GTFS feeds are static, rather than real-time; that is, a GTFS feed represents transit service as scheduled, rather than as it may actually perform and be delivered to riders.
GTFS Realtime	GTFS Realtime extends the static GTFS data format to enable transit agencies to provide updates about transit service as it actually performs and is delivered to riders. GTFS Realtime feeds are typically served by a series of API endpoints for TripUpdates (predictions), VehiclePositions, and Alerts. All GTFS Realtime feeds must produce output using the protocol buffer format.
JSON	JavaScript Object Notation - a data format often used for web platform APIs. It can be readily consumed and produced by JavaScript, the scripting language that runs inside web browsers; most other scripting and programming languages have utilities for reading and writing JSON as well. JSON is a set of keys and values; it can represent values that are character strings or numbers. JSON is a text-

	based format, meaning it can easily be viewed in a web browser or a text editor. Many web platforms that previously used XML have switched to or added JSON support.
Open-source software	Open-source software is released under a license that allows reuse by other companies, governments, and independent developers. There are a wide variety of open-source license, which enable or disallow different types of reuse, modification, and commercial use.
Protocol Buffer	Protocol Buffer is a file format that can be used for transferring data from a web service API. It's often abbreviated as PB, PBF, or protobuf. In contrast with CSV and JSON (which are text-based formats), PB is a binary format. PB allows data typing (for example, some fields must be character strings while others must be numbers). PB can be a very efficient format for APIs that provide frequent and verbose updates. However, reading and writing PB data requires custom software tools. (A web browser will not display the contents of PB data, nor will Microsoft Excel.)
Transitland	Transitland is an open-source and open-data platform aggregating GTFS and GTFS Realtime feeds from around the world. It was started by Mapzen and is currently under the umbrella of the Linux Foundation and maintained by Interline Technologies.

Table of Contents

Investigator Profile

Interline Technologies LLC is a technical services and products company that helps organizations understand and improve transportation networks, digitally.

We are a team skilled in product management, software engineering, and system operations for modern web and mobile applications. We have deep domain expertise in public transit, transportation planning, travel behavior, and emerging mobility options. Our technical specialties include public-transit data (GTFS and GTFS Realtime), multi-modal trip planning, geographic analysis, open-source software, and open and public data.

Interline's principals come from years of experience building and deploying open-source software for transportation and geographic applications at Mapzen (the mapping R&D division of Samsung). Now, Interline provides technical services and know-how to a range of transportation providers and planners: Oregon Department of Transportation (ODOT), Vermont Agency of Transportation (VTrans), California Department of Transportation (Caltrans), Scoot Networks, The World Bank, Trillium Solutions, Los Angeles County Metropolitan Transportation Authority (LA Metro), the Metropolitan Transportation Commission (MTC), Replica Inc., San Francisco Municipal Transportation Agency (SFMTA), Cambridge Systematics, and the City of Palo Alto (California) among other clients.

Executive Summary

Real-time transit information has been shown to have many benefits for transit riders and agencies, including shorter perceived and actual wait times, a more welcoming experience for new riders, and an increased feeling of safety, and increased ridership. Real-time transit data is, in comparison with many other potential operational or capital improvements to bus or rail service, an affordable means of increasing ridership. In the last few years, a real-time complement to the General Transit Feed Specification (GTFS) format, GTFS Realtime, has emerged, which enables transit agencies to share real-time predictions, vehicle positions, and service alert data in a standardized format. Despite its promise, adoption of GTFS Realtime by transit agencies has been hampered by readily available validation tools.

In this project, Interline Technologies LLC and the Center for Urban Transportation Research (CUTR) at the University of South Florida (USF) have created a prototype platform that makes GTFS Realtime validation tools readily available to, potentially, all transit agencies in North America. Our team is building upon two open-source projects: the GTFS Realtime validator prototype and Transitland, an open transit data platform. This project applies the open-source and open-data community models to the challenges of creating and improving GTFS Realtime data.

Stage 1: Build and Test GTFS Realtime Data Platform The research team combined the Transitland open data platform, the GTFS Realtime Validator, and a list of 162 GTFS Realtime feed endpoints (provided by a partner organization). The combined platform collects GTFS Realtime data from each feed, runs the validator process, and produces a report on any detected errors. Each report shows the counts of data entities, the percentage with errors, and a brief text description of any errors. Links take users to additional documentation about each error type. Some errors also provide further contextual information in maps and tables to assist users as they try to determine root causes.

Stage 2: Testing and Expanding Catalogue: The project team has tested the platform by preparing validation reports for seven public-transit agencies and reviewing the results in the platform user interface with agency staff members over video calls. In these user-testing sessions, the project team collected information from agency staff about how GTFS and GTFS Realtime data are currently created at each agency, known issues, and any open goals. After being given a tour through the platform and its interface, agency staff reviewed the reports for their own GTFS Realtime feeds. Agency staff were asked to provide input on both the specific quality checks and the overall presentation and approach used by the platform with a standardized question list. The research team has summarized notes from the user-testing sessions and used these results to inform preliminary plans for expanding the platform for a wider range of users in the future.

Product Pay-Off Potential: By combining the open-source components of a GTFS Realtime validator with a catalog of GTFS Realtime feeds, hosted on Transitland's cloud servers, this project will make the process of validating real-time data simple and accessible to agency staff from any computer with a web browser. As a result, GTFS Realtime data will improve in quality and availability. Transit riders will have a better experience (which has been linked to higher ridership), agency staff will provide better service with less effort and cost, and system vendors will provide a higher quality product.

Product Transfer: To succeed, this platform will need to be usable by agency staff and to provide them with results that they can act upon, both within the context of their agencies and with their vendors. Therefore, the project team has involved agency stakeholders early and often in the project. The team recruited nine transit agencies to write letters of support as part of the Transit IDEA proposal, and seven agencies have participated in our user-testing process. These agencies represent a diverse range of rider population sizes, staff skill level, and location (urban and rural). The final report includes information on immediate plans to continue and expand Transitland's catalog GTFS Realtime feeds and longer term plans to develop a GTFS Realtime certification process supported by the platform.

IDEA Product

Real-time transit information has many benefits to transit riders and agencies, including shorter perceived and actual wait times (as riders are able to optimize their plans in advance of arriving at a stop or station), a more welcoming experience for new riders, and an increased feeling of safety, and increased ridership. Real-time transit data is, in comparison with many other potential operational or capital improvements to bus or rail service, an affordable means of increasing ridership. In the last few years, a real-time complement to the General Transit Feed Specification (GTFS) format, GTFS Realtime[1], has emerged. Despite

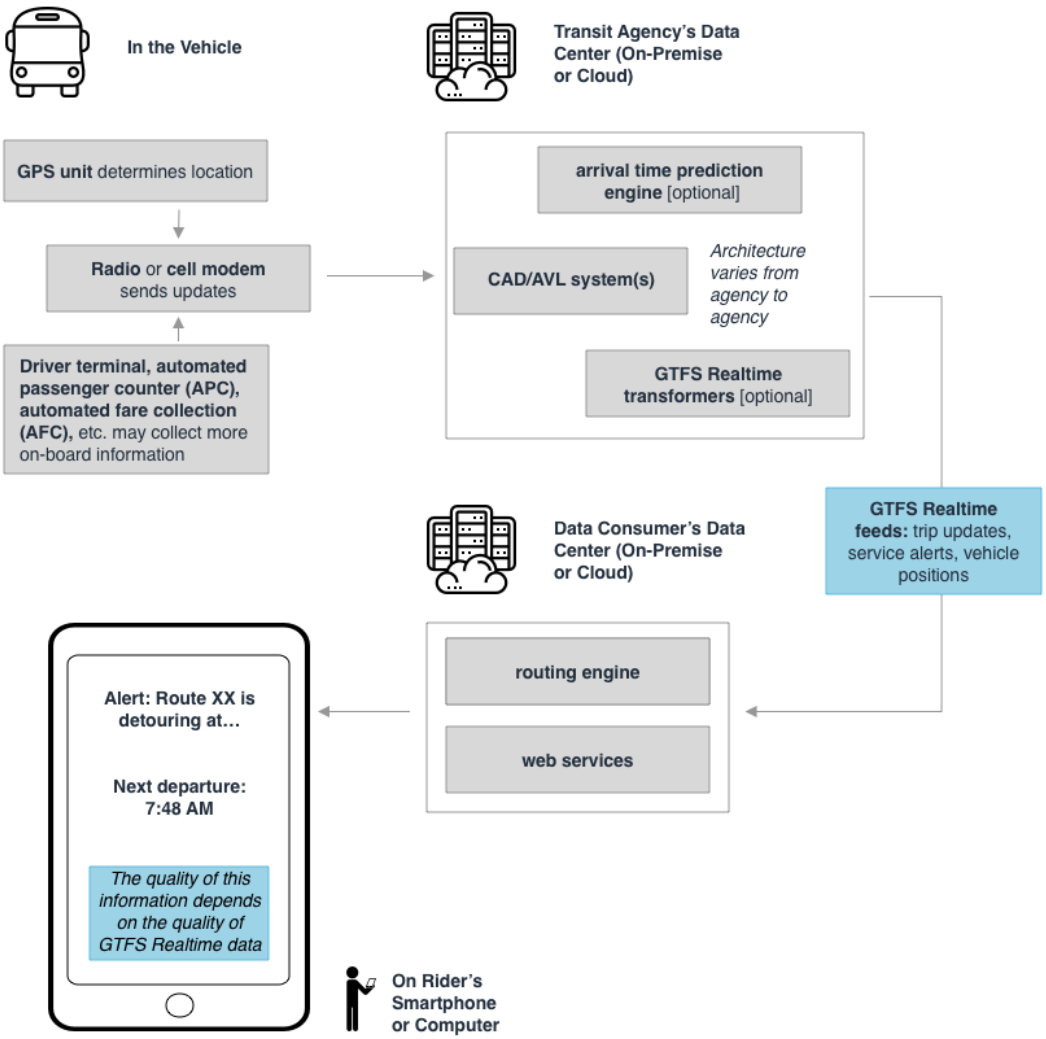


Figure 1. A diagram showing where GTFS Realtime feeds are typically produced and consumed (right) in an overall technical system architecture.

[1] https://developers.google.com/transit/gtfs-realtime/

its promise, adoption of GTFS Realtime by transit agencies has been constrained by documentation that requires expertise, a lack of readily available validation tools, and by complex overall system architectures.

Figure 1 illustrates how measurements from transit vehicles are turned into GTFS Realtime data and used to provide traveler information to transit riders.

GTFS Realtime data feeds (the arrow on the right side of the diagram) are the only connection between the internal complexity of each agency's transit data systems and the software systems of each data consumer. Data consumers like Google, Apple, and Interline's Transitland platform draw from thousands of agencies. Having dependable GTFS Realtime feeds from each agency is critical.

In this project, Interline Technologies LLC and the Center for Urban Transportation Research (CUTR) at the University of South Florida (USF) created a prototype platform that makes GTFS Realtime validation tools readily available to, potentially, all transit agencies in North America. We built upon two open-source projects: the GTFS Realtime validator prototype[2] and Transitland[3]. This project applied the open-source and open-data community models to the challenge of improving GTFS Realtime data.

The real-time validation platform will produce payoffs related to the following of the Transit IDEA panel's high-priority focus areas:

- *Increase transit ridership*. Realtime data that is available and of high-quality has been demonstrated to lead to increased ridership.
- *Improve transit capital and operating efficiency*. CAD/AVL systems involve many software and hardware components. Oftentimes, an agency will rely on multiple vendors. By helping agencies to measure and improve the quality of their GTFS Realtime output, the platform will provide agency staff with visibility into upstream systems. This insight may improve day-to-day operations and inform RFPs for future systems.
- *Quick delivery of timely information*. Agencies invest in CAD/AVL and systems that generate GTFS Realtime feeds with the goal of delivering timely information to their riders and to agency systems. However, they are not currently equipped to measure the success of their systems in meeting these goals. The platform will provide quantitative metrics on how well a GTFS Realtime feed is meeting basic standards.
- *Automated monitoring of transit vehicle locations and operations*. GTFS Realtime is useful for both rider-facing information and internal operations information. The platform evaluates feeds with both goals in mind.

[2] https://github.com/CUTR-at-USF/gtfs-realtime-validator
[3] https://transit.land

Concept and Innovation

Real-time transit information has many benefits to transit riders and agencies, including shorter perceived and actual wait times (1, 8, 9), a more welcoming experience for new riders (2), an increased feeling of safety (e.g., at night) (3, 4), and increased ridership (5, 6).[4] Real-time transit data is, in comparison with many other potential operational or capital improvements to bus or rail service, an affordable and efficient means of increasing ridership.[5] In the last few years, a real-time complement to the General Transit Feed Specification (GTFS) format, the GTFS Realtime extension, has emerged.[6] GTFS Realtime has the potential to standardize real-time data feeds and lead to widespread adoption for transit agencies and multimodal trip-planning apps.

Despite its promise, adoption of GTFS Realtime by transit agencies has been hampered by readily available validation tools. Hardware and software vendors for automatic vehicle location (AVL) systems also vary in the depth and quality of their support for the GTFS Realtime specification. As a result, transit agencies must invest significant time and effort to create and maintain high quality GTFS Realtime feeds. Furthermore, bad data have shown to have a negative effect on ridership, the rider's opinion of the agency, and the rider's satisfaction with multimodal trip-planning apps. For example, 74% of surveyed Puget Sound transit riders considered a difference between actual and estimated arrival times greater than 4 minutes as an "error." In addition, 9% of surveyed riders said that they took the bus less often due to errors they experienced (4). Thus, transit agencies must put even more effort toward ensuring that the GTFS Realtime data produced by their systems is of high quality.

An evaluation of 78 transit agencies' GTFS Realtime feeds using a prototype validator tool (discussed further below) showed integrity errors in 54 feeds and warnings in 58 feeds. That is, approximately 70% of these GTFS Realtime feeds have issues, indicating widespread problems with quality control. (10)

Further research into the availability and quality of GTFS Realtime data has been hampered by a lack of information on existing real-time data feeds and the systems used to create each feed. Transit agency staff regularly attend TRB and other academic and industry forums with stories of their own failures and successes with real-time data systems and vendors. However, the quality of these systems is not being assessed rigorously. As noted in multiple TCRP and TRB reports, "a limited number of [responding transit agencies] monitor the reliability and accuracy of the information provided on mobile devices" (14).

[4] For an overview of benefits of real-time arrival information, see (7, pp. 4-14).

[5] The APTA 2017 Fact Book notes: "The growth of automatic vehicle location systems, which improve the operation of bus fleets as well as the availability of information on bus arrival times, has made public transit systems more efficient and data more accessible."

[6] For more information on the General Transit Feed Specification (GTFS), see http://www.gtfs.org. For more information on the GTFS Realtime extension, see https://developers.google.com/transit/gtfs-real-time/.

A complicating factor is that many transit agencies' CAD/AVL systems do not "natively" produce GTFS Realtime. That is, these systems use proprietary interchange formats for their real-time information. To satisfy agencies' requests for GTFS Realtime output, vendors will add custom components that transform their proprietary format into the GTFS Realtime format (sometimes at additional expense to agencies). While there is nothing inherently wrong with this technical architecture, it often complicates the process of tracing errors in GTFS Realtime output. To debug errors in the public real-time data feeds requires rigorous assessment of the output GTFS Realtime data *and* full understanding of the "upstream" components in the technical architecture. In some cases, the upstream source is actual vehicle locations from an AVL system; in other cases, locations may come from prediction engines, which are in turn downstream from AVL systems. These many components can cut across departmental boundaries within a transit agency and may involve more than one vendor. Systems that only produce GTFS Realtime as a final, custom transformation step often have another complicating factor: the GTFS Realtime data is provided to public users but never used for any internal purposes for the transit agency. This means that staff are less aware of potential problems in their GTFS Realtime feeds, as data problems only affect outsiders, not directly themselves, their colleagues, or their internal initiatives that may be more closely monitored for accuracy and completeness.

In this project, a private-sector software firm (Interline Technologies) and a university transportation lab (the Center for Urban Transportation Research, CUTR) created a prototype platform that makes GTFS Realtime validation tools readily available to, potentially, all transit agencies in North America. We built upon two open-source projects: the GTFS Realtime validator prototype and Transitland, an open-transit data platform.

Transitland is an open data platform, originally created by Samsung's Mapzen division and now governed by the Linux Foundation and maintained by Interline Technologies LLC.[7] Transitland currently catalogs static GTFS schedule data for over 2,500 transit agencies around the world and provides application programming interfaces (APIs) and user interfaces (UIs) to users and developers of various skill levels. Users of the Transitland platform include Apple, the National Park Service, Sidewalk Labs, and a range of for-profit start-ups and civic advocacy groups. See Figure 2 for an example of one of its UIs.

The GTFS Realtime validator prototype was created by CUTR as part of research funded by the National Institute for Transportation and Communities (NITC) [16]. It is an open-source application which fetches and evaluates the contents of a GTFS Realtime feed. The prototype has been used to successfully estimate the scale of GTFS Realtime validation issues in a sample of feeds and to demonstrate the value for such a tool to agency staff and vendors.

[7] More information at https://transit.land See also: (11, 12, 13) For background on open data as created and used by transit agencies, see 15.

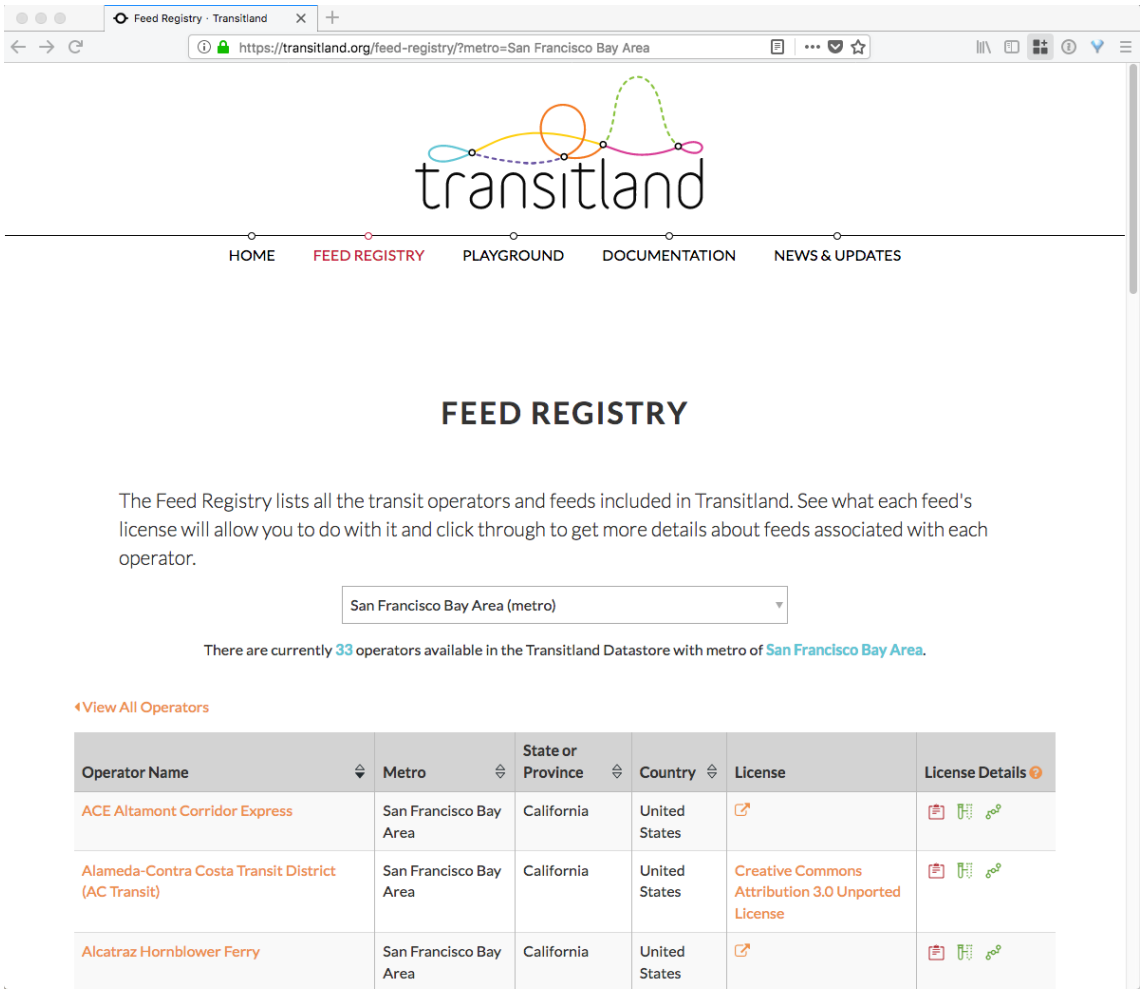


Figure 2. Screenshot of the Transitland Feed Registry.

In this project, the research team added a catalog of GTFS Realtime feeds to the Transitland API, to improve the "discoverability" of existing real-time data by researchers, vendors, agency staff, and application developers. Next, we integrated the GTFS Realtime validator and Transitland, so that all cataloged feeds can be validated on an ongoing or an on-demand basis. Finally, we tested the catalog and validation platform with eight transit agencies. Based on feedback and test results, we have planned future improvements to the platform to scale to support hundreds of GTFS Realtime feeds. We have also begun to plan a GTFS Realtime certification process with input from agency staff.

By combining the open-source components of a GTFS Realtime validator with a catalog of GTFS Realtime feeds, this project has demonstrated how validating real-time data can simple and accessible. As a result, GTFS Realtime data will improve in quality and availability. Transit riders will have a better experience (which has been linked to higher ridership), agency staff will provide better service with less effort and cost, and system vendors will provide higher quality product.

Investigation

The project plan was based on a thorough review of scholarly literature on the topics of information-communication technology (ICT) for public transit. In addition, we reviewed the current state of the art in the industry: open-source software available on GitHub and other online portals, as well as proprietary systems described in vendor marketing materials. Finally, the software development approach was informed by best practices from the IT industry for delivering high-quality software that meets user requirements, budgets, and schedules.

This project proceeded in two stages:

- Stage I: Build and Test GTFS Realtime Data Platform [Tasks 1 – 9]
- Stage II: Testing and Expanding Catalogue [Tasks 10 – 14]

The research team's progress and findings through the two stages follows as a narrative. Administrative tasks, such as meetings and reports, have been elided.

Tasks 2 and 3: Enable Transitland to catalog GTFS Realtime feeds

Ian Rees and Drew Dara-Abrams worked together to design the data model for this task. Dr. Rees completed the software development, including the writing of automated tests to ensure that the code works as intended.[8]

Task 4: Populate the directory of GTFS Realtime feeds and endpoints

Fortuitously, Dr. Dara-Abrams met an open-source developer working on his own GTFS Realtime feed cataloging project. He had already created a spreadsheet listing 162 GTFS Realtime URLs and associated them with Transitland records and Onestop ID identifiers.[9] He gave us permission to use a copy of his data. The research team transformed and imported the spreadsheet in to Transitland.

Task 5. Deploy the validator library to Transitland servers

The team is now able to run the validator on Transitland servers on an on-demand basis. Dr. Barbeau provided an introduction to the existing open-source software. Dr. Rees and Dr. Dara-Abrams designed a workflow to run the new process alongside the existing Transitland processes. When run against a feed, the validator library can produces errors and warning codes (see Table 1). The Transitland workflow aggregates this output and makes it available through an API for querying.

[8] Source code created during this task has been shared publicly on GitHub: https://github.com/transitland/transitland-datastore/pull/1275 and https://github.com/transitland/feed-registry/pull/403 and https://github.com/transitland/dispatcher/pull/284

[9] For more information on Transitland's Onestop ID scheme, see https://transit.land/documentation/onestop-id-scheme/

Table 1. GTFS Realtime Validator Errors and Warnings

ID Code (E=Error, W=Warning)	Error Title
E001	Not in POSIX time
E002	`stop_time_updates` not strictly sorted
E003	GTFS-rt `trip_id` does not exist in GTFS data
E004	GTFS-rt `route_id` does not exist in GTFS data
E006	Missing required trip field for frequency-based `exact_times` = 0
E009	GTFS-rt `stop_sequence` isn't provided for `trip` that visits same `stop_id` more than once
E010	`location_type` not `0` in `stops.txt` (Note that this is implemented but not executed because it's specific to GTFS - see issue #126)
E011	GTFS-rt `stop_id` does not exist in GTFS data
E012	Header `timestamp` should be greater than or equal to all other `timestamps`
E013	Frequency type 0 trip `schedule_relationship` should be UNSCHEDULED or empty
E015	All `stop_ids` referenced in GTFS-rt TripUpdates and VehiclePositions feeds must have the `location_type` = 0
E016	`trip_ids` with `schedule_relationship` ADDED must not be in GTFS data
E017	GTFS-rt content changed but has the same header `timestamp`
E018	GTFS-rt header `timestamp` decreased between two sequential iterations
E019	GTFS-rt frequency type 1 trip `start_time` must be a multiple of GTFS `headway_secs` later than GTFS `start_time`
E020	Invalid `start_time` format
E021	Invalid `start_date` format
E022	Sequential `stop_time_update` times are not increasing
E023	trip `start_time` does not match first GTFS `arrival_time`
E024	trip `direction_id` does not match GTFS data
E025	`stop_time_update` departure time is before arrival time
E026	Invalid vehicle `position`
E027	Invalid vehicle `bearing`
E028	Vehicle `position` outside agency coverage area
E029	Vehicle `position` far from trip shape
E030	GTFS-rt alert `trip_id` does not belong to GTFS-rt alert `route_id` in GTFS `trips.txt`
E031	Alert `informed_entity.route_id` does not match `informed_entity.trip.route_id`
E032	Alert does not have an `informed_entity`
E033	Alert `informed_entity` does not have any specifiers
E034	GTFS-rt `agency_id` does not exist in GTFS data
E035	GTFS-rt `trip.trip_id` does not belong to GTFS-rt `trip.route_id` in GTFS `trips.txt`
E036	Sequential `stop_time_updates` have the same `stop_sequence`

ID Code (E=Error, W=Warning)	Error Title
E037	Sequential `stop_time_updates` have the same `stop_id`
E038	Invalid `header.gtfs_realtime_version`
E039	`FULL_DATASET` feeds should not include `entity.is_deleted`
E040	`stop_time_update` doesn't contain `stop_id` or `stop_sequence`
E041	`trip` doesn't have any `stop_time_updates`
E042	`arrival` or `departure` provided for `NO_DATA` `stop_time_update`
E043	`stop_time_update` doesn't have `arrival` or `departure`
E044	`stop_time_update` `arrival/departure` doesn't have `delay` or `time`
E045	GTFS-rt `stop_time_update` `stop_sequence` and `stop_id` do not match GTFS
E046	GTFS-rt `stop_time_update` without `time` doesn't have arrival/departure time in GTFS
E047	`VehiclePosition` and `TripUpdate` ID pairing mismatch
E048	`header` `timestamp` not populated (GTFS-rt v2.0 and higher)
E049	`header` `incrementality` not populated (GTFS-rt v2.0 and higher)
E050	`timestamp` is in the future
E051	GTFS-rt `stop_sequence` not found in GTFS data
E052	`vehicle.id` is not unique
W001	`timestamps` not populated
W002	`vehicle_id` not populated
W003	ID in one feed missing from the other
W004	vehicle `speed` is unrealistic
W005	Missing `vehicle_id` in `trip_update` for frequency-based `exact_times` = 0
W006	`trip_update` missing `trip_id`
W007	Refresh interval is more than 35 seconds
W008	Header `timestamp` is older than 65 seconds
W009	`schedule_relationship` not populated

Note: For descriptions of each error or warning, see the full manual at https://github.com/CUTR-at-USF/gtfs-realtime-validator/blob/master/RULES.md

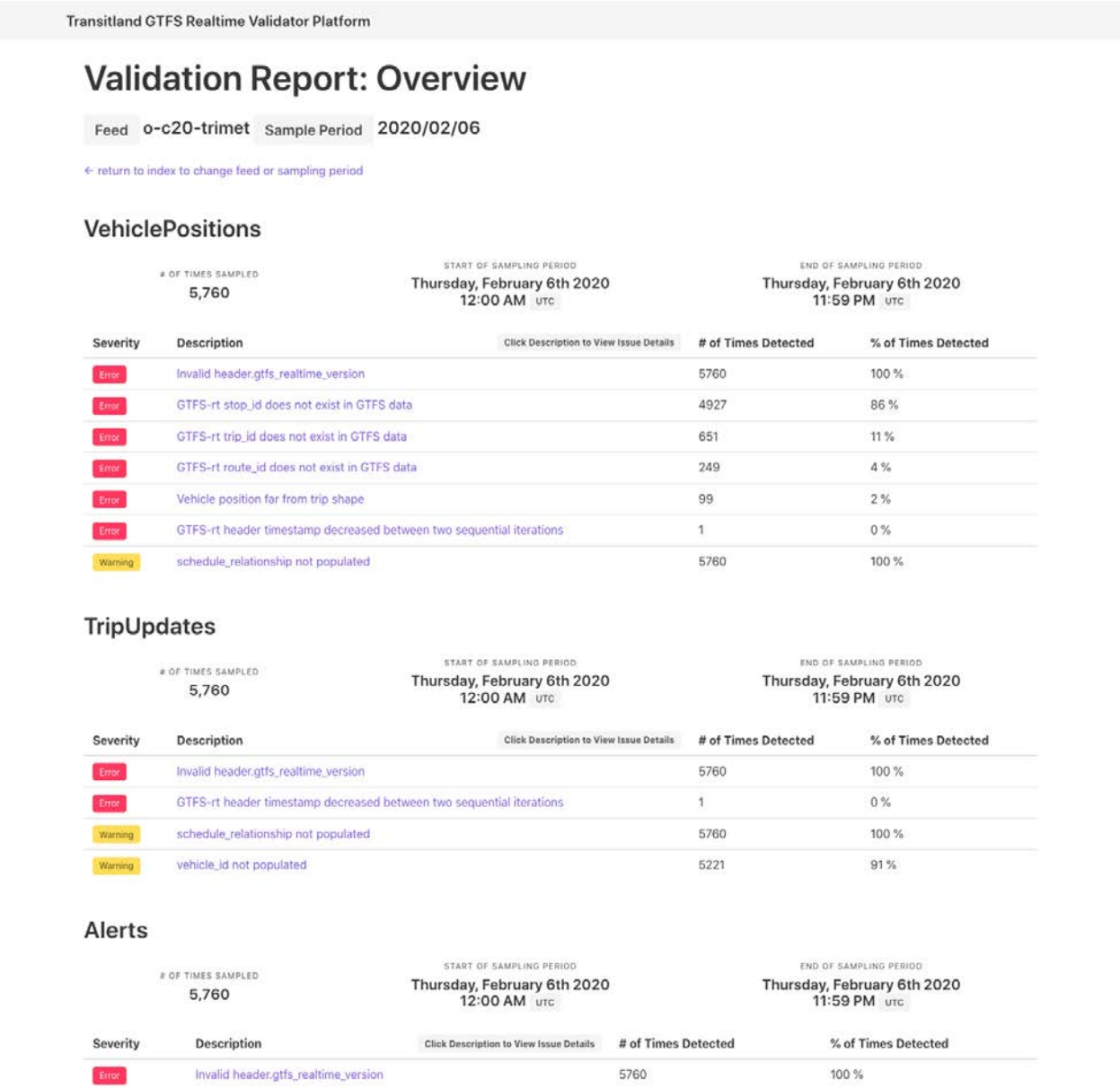
Task 6. Store results of the validator library to Transitland API

Dr. Rees designed a workflow that runs the GTFS Realtime validator library on demand against any GTFS Realtime data snapshots and stores the validation results on a public Transitland storage service.

One unexpected but positive finding as part of this task is that we've been able to make use of a large, existing archive of GTFS Realtime data snapshots. One of Interline's partners shared with us their archive that aggregates 190 different GTFS Realtime endpoints, snapshotted every 15 seconds. This is a high enough frequency to capture both dated AVL systems that update closer to every minute as well as newer systems that update more frequently (e.g., once every 15

seconds). Interline has provided access to Transitland cloud services to our partner, so they are now able to continue running the process that updates the archive and store the data in Transitland's Amazon Web Services account.

Task 7. Display validator results in Transitland user interfaces



Transitland GTFS Realtime Validator Platform

Validation Report: Overview

Feed o-c20-trimet Sample Period 2020/02/06

← return to index to change feed or sampling period

VehiclePositions

# OF TIMES SAMPLED	START OF SAMPLING PERIOD	END OF SAMPLING PERIOD
5,760	Thursday, February 6th 2020 12:00 AM UTC	Thursday, February 6th 2020 11:59 PM UTC

Severity	Description (Click Description to View Issue Details)	# of Times Detected	% of Times Detected
Error	Invalid header.gtfs_realtime_version	5760	100 %
Error	GTFS-rt stop_id does not exist in GTFS data	4927	86 %
Error	GTFS-rt trip_id does not exist in GTFS data	651	11 %
Error	GTFS-rt route_id does not exist in GTFS data	249	4 %
Error	Vehicle position far from trip shape	99	2 %
Error	GTFS-rt header timestamp decreased between two sequential iterations	1	0 %
Warning	schedule_relationship not populated	5760	100 %

TripUpdates

# OF TIMES SAMPLED	START OF SAMPLING PERIOD	END OF SAMPLING PERIOD
5,760	Thursday, February 6th 2020 12:00 AM UTC	Thursday, February 6th 2020 11:59 PM UTC

Severity	Description (Click Description to View Issue Details)	# of Times Detected	% of Times Detected
Error	Invalid header.gtfs_realtime_version	5760	100 %
Error	GTFS-rt header timestamp decreased between two sequential iterations	1	0 %
Warning	schedule_relationship not populated	5760	100 %
Warning	vehicle_id not populated	5221	91 %

Alerts

# OF TIMES SAMPLED	START OF SAMPLING PERIOD	END OF SAMPLING PERIOD
5,760	Thursday, February 6th 2020 12:00 AM UTC	Thursday, February 6th 2020 11:59 PM UTC

Severity	Description (Click Description to View Issue Details)	# of Times Detected	% of Times Detected
Error	Invalid header.gtfs_realtime_version	5760	100 %

Figure 3. The validator library run against a day's worth of Portland TriMet GTFS Realtime data, displaying with our new user interface in a web browser.

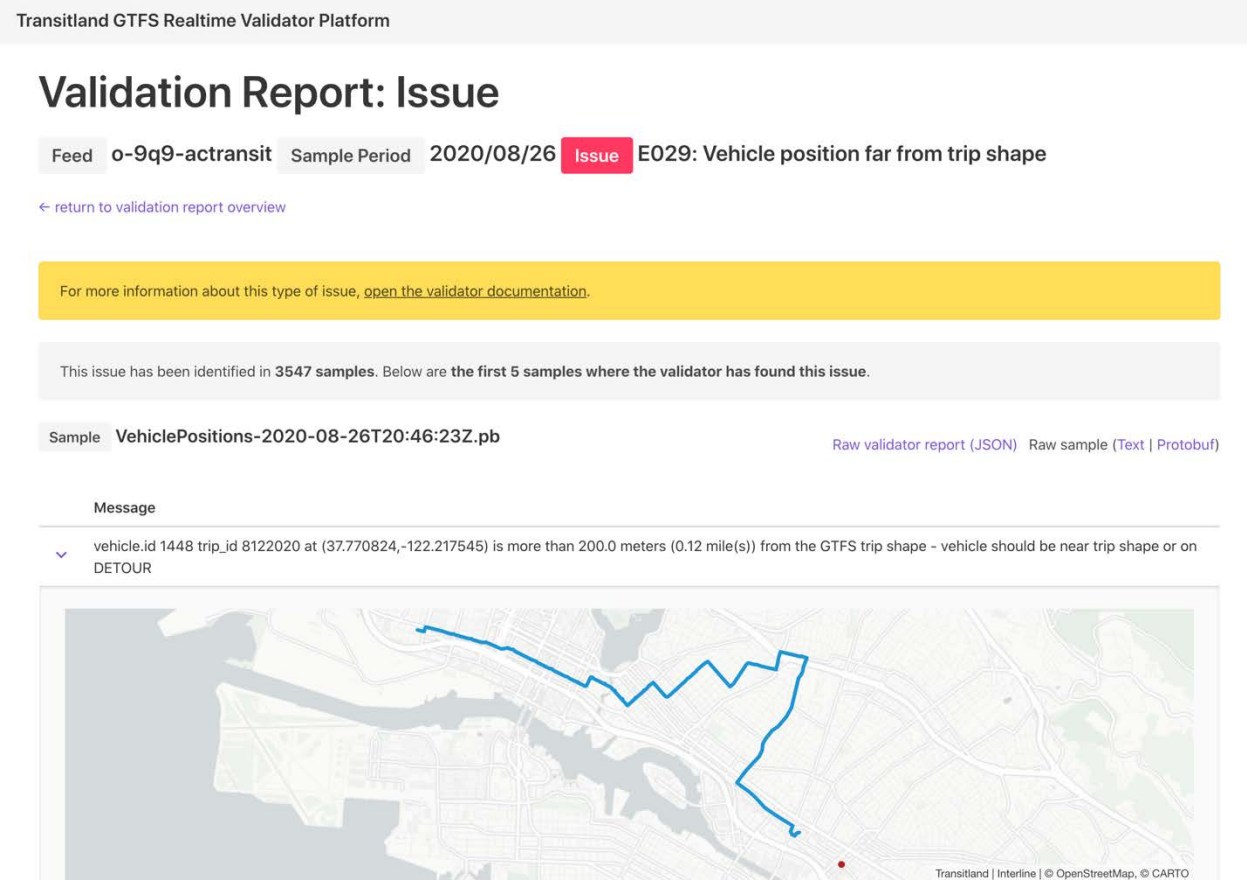


Figure 4. A screenshot of the validator platform showing contextual information from both a GTFS Realtime feed and its associated static GTFS feed. The error (E029) is that a vehicle position is too far from its associated trip ship. The map shows the vehicle (bus) position as a red dot and its scheduled route shape/alignment in blue.

The validator library takes GTFS Realtime protocol buffer files as input and produces its output in the JSON format. This file format is easy for a machine to read, straightforward for a developer to browse in a code editor, but not ideal for other potential users to browse. For the final task in this round of work, Dr. Rees and Dr. Dara-Abrams created a simple user interface that reads the JSON files and formats their contents in a web browser.

Figure 3 is an example screenshot, showing the validation results for a day of Portland TriMet GTFS Realtime data. Errors for the three endpoints are displayed in separate tables. For each validation error, the error code and description are given. (See Table 1 for a list of possible errors and warnings.)

By chance, another project at Interline led to work on GTFS Realtime data with the Metropolitan Transportation Commission (MTC). MTC aggregates and produces GTFS Realtime feeds for 17 transit agencies throughout the region. Interline deployed the validator to catch errors in MTC's GTFS Realtime feeds. This, in effect, turned into a first, informal user-testing session of the validator and helped us to refine our plans for formal user-testing sessions. When validating MTC's feeds, we found that some of the most important validation errors require

the most context to debug. For example, some errors in GTFS Realtime feeds also require the context of the associated static, or schedule, GTFS feed to fully understand. See Figure 4 for an example of how the research team added contextual information from static GTFS feeds to better explain a GTFS Realtime validation error.

Task 10. Share platform with transit agency staff members for feedback

User-Testing Participants

In order to evaluate our user interface and our overall approach to validating GTFS Realtime feeds using a platform, we conducted user-testing sessions with staff at transit agencies. Staff came from 7 different agencies of varying size, internal IT capacities, rider demographics, and level of customization within their IT systems for producing GTFS Realtime data (see Table 2).

Table 2. Transit Agencies Participating in User-Testing

Agency Name	Location	Service Area Population	Unlinked Passenger Trips (Per Year)
Metro Transit	Minneapolis/ St. Paul	1,837,223	80,653,405
Massachusetts Bay Transportation Authority (MBTA)	Boston	3,109,308	372,398,838
Tri-County Metropolitan Transportation District of Oregon (TriMet)	Portland, Oregon	1,551,531	97,033,281
Tompkins Consolidated Area Transit (TCAT)	Ithaca, New York	103,617	4,223,437
Washington Metropolitan Area Transit Authority (WMATA)	Washington, DC	3,719,567	351,298,962
Hillsborough Area Regional Transit (HART)	Tampa-St. Petersburg, Florida	807,015	12,182,690
Rogue Valley Transportation District (RVTD)	Medford, Oregon	132,022	1,210,070

Note: Figures are from US National Transit Database, as of 2018 fiscal year.

Preparation for User-Testing Sessions

Before each user-testing session, Dr. Dara-Abrams made arrangements with agency staff via email to ensure we were using their latest GTFS Realtime feeds and associated static GTFS feeds and to make sure the appropriate staff would be available to join the call. Agencies put responsibility for GTFS and GTFS Realtime feeds in a variety of different departments, and sometimes more than one department may be responsible for producing the feeds. The team tried to ensure that as many of these possible departments would be represented on the user-testing calls. (Vendors were not invited to participate, in order to allow agency

staff to speak freely about any known issues with their systems.) While scheduling the calls, the team also learned that some agencies produce multiple static GTFS feeds: one version of their static GTFS feed for public data-consumers and a separate version that is intended to be used together with their GTFS Realtime feeds—often in order to provide IDs for stops, routes, and trips, as defined within their CAD/AVL systems, rather than their static scheduling systems. To properly validate a GTFS Realtime feed, we have to make sure we're using the appropriately matched static GTFS feed.

After collecting relevant feed URLs and access information for each agency (sometimes registering for a user account or an authorization key), Dr. Rees processed a recent 24-hour period of GTFS Realtime samplings through the validation workflow. Dr. Rees and Dr. Barbeau reviewed the output together to catch any unexpected errors and to pre-select a handful errors/warnings, typically three, to discuss with agency staff on the video call.

User-Testing Script

Once these preparations were complete, each call was scheduled for 1 hour and was structured by the research team using the script in Table 3. After each call, we emailed agency staff a link to explore their validation report in their own web browser. We also shared the notes document, for agency staff to review, correct any errors, or to add additional detail.

Table 3. User-Testing Script

1. Preliminary questions (no screensharing):
    a. Who is responsible for GTFS Realtime? In what department do they work?
    b. What systems/vendors are used to produce GTFS Realtime?
    c. What office/department is responsible for static GTFS?
    d. What systems/vendors are used to produce static GTFS?
    e. Any current or past methods for checking/validating GTFS Realtime?
        i. If yes, what's your biggest obstacle to fixing problems?
        ii. What office/department is responsible for making sure GTFS and GTFS Realtime datasets match?
    f. Any currently known issues with GTFS Realtime? Ongoing and/or intermittent?
2. Research team member screenshares the user interface, showing a report for the agency's validation report. A brief tour to show functionality.
3. Research team steps the agency staff through three pre-selected warnings/errors, or lets the agency staff pick some warnings/errors to inspect together
    a. [Highlight and explain chosen error/warning]
    b. Does this user interface provide you sufficient information to understand this error/warning? (Do you have access to the full RT message, related static GTFS, and any other context?)
    c. Do you actually care about this error/warning? Do you think it's best called an "error" or "warning"?
    d. What are your next steps for debugging/fixing this error?

e. Would your agency or vendor be responsible for fixing this error? If vendor, would you expect to be charged to fix this error?

4. Wrap-up questions (no screensharing, unless agency staff ask to look at report again):
   a. Does this report help tell you which errors are important and which are "noise"?
   b. What other information would you find useful in this report?
   c. In terms of technical detail, is this report too much, too little, or just right?
   d. Would you share this report to explain the error to a colleague? A vendor? If so, would you share it all or just a portion?
   e. How often would you return to this report (assuming that it is regularly updated): weekly, monthly, quarterly, annually, never?

User-Testing Findings

In this section we will summarize findings from the user-testing sessions with agency staff. The following summaries are aggregated from across all interviews, so that we do not name agencies, individuals, or vendors.

Preliminary questions:

GTFS and GTFS Realtime are often the responsibility of separate groups/departments within an agency. For agencies that operate both bus and rail service, often separate groups/departments are responsible for the data systems associated with each mode. A few agencies have formed working groups with cross-cutting responsibilities for static and real-time data across bus and rail.

All agencies mentioned some amount of semi-manual process or custom scripts for preparing their static GTFS feeds for use with their GTFS Realtime feeds.

Most agencies validate their static GTFS feeds, but most have not validated the contents of their GTFS Realtime feeds. A few agencies checked the output of Google's GTFS Realtime validation tools when first adding their data to Google Maps, but haven't checked it since.

A few agencies monitor their GTFS Realtime feeds for uptime (that is, to ensure the API endpoints are available and responding to web requests).

Smaller and medium-sized agencies often outsource their GTFS Realtime systems fully to vendors and spoke of many frustrations:

- Vendor systems are "black boxes" with "no visibility into data generation."
- Instead of fixing GTFS Realtime feed issues, one vendor tells agency staff that its own proprietary app has no such issues.
- Improvements to GTFS Realtime feed generation must often happen at the same time as new CAD/AVL hardware is procured, so overall RFPs may be complex to write and evaluate. Agencies should also consider having on-going quality requirements, including compliance with validation tools, throughout the duration of their contract with CAD/AVL vendors.
- One vendor owns the data produced by the agency's CAD/AVL system, so the vendor insists that the agency purchase access to the data streams if it wishes to use them to produce alternative GTFS Realtime feeds.
- Several agencies voiced the opinion that vendors may be more responsive to fixing errors flagged by the validator rather than their own staff, as the

validator would be seen as an objective tool. Several agencies also stated that vendors may be more responsive to a grade being applied to the validation output (e.g. "A-F"), especially if it was publicly visible for several transit agencies using the same vendor.

Research team walks agency staff through pre-selected warnings/errors: All agencies found this experience of exploring the warnings/errors useful, to the point that most of these sessions ran long. Each of these sections turned into miniature consulting sessions, with the conversation between agency staff and the research team often covering the range from specific GTFS Realtime data fields up to system-level architecture concerns. On the one hand, this shows the power of the platform as a way to surface useful information for such wide-ranging investigations. Agency staff reported that some of this information was available through other sources but not aggregated in one place, while other information was previously unknown. On the other hand, these miniature consultations show how unique each GTFS Realtime system is and how wide a range of information and functionality is necessary for the platform to serve all agencies' potential needs. In many cases there can be several potential causes for errors, and troubleshooting the root cause can require in-depth knowledge of GTFS and GTFS Realtime. Future work could examine ways to include initial suggestions for troubleshooting common errors or peer support forums to allow agencies to exchange information with other agencies that have encountered the same problem with the same vendor.

Wrap-up questions:

<u>Does this report help tell you which errors are important and which are "noise"?</u>

"I like the error and warning tag."

"I think it's good. Severity indication is helpful. But 'warning' and 'error' could be defined a little more. Does 'error' mean it's non-functional?"

Errors in validation output are problems where the feed is outputting data that violates the GTFS Realtime specification, while warnings are suspicious values that the validator can't identify with 100% certainty (e.g., suspiciously high vehicle speeds), or items that aren't fully defined in the GTFS Realtime specification. The user interface could be enhanced to better explain this difference.

<u>What other information would you find useful in this report? In terms of technical detail, is this report too much, too little, or just right?</u>

Multiple agencies were interested in being able to do "regression testing" (to know if an error/warning was introduced by a new change to the system, or if it was present before the change). In the words of one agency staffer such functionality would let them say "well, we don't have any new problems" after making a system or data change.

Two agencies spoke about wanting to also know the trends in an error or warning: is it a one-off issue? Does it happen every day? Are the errors increasing or decreasing over time? One of these agencies said they use a similar technique for monitoring logs for software bugs, giving attention to the bug reports that are increasing in frequency.

Several agencies said that having consequence of errors better explained, and grouped by consequence, would be helpful. For example, errors that would typically result in no real-time information being shown in apps (e.g., mismatched static and real time trip IDs) could all be categorized together.

Several agencies stated that knowing the accuracy of predictions (e.g., if the feed said a bus was arriving in 5 minutes, did it actually arrive in 5 minutes?) would also be helpful. This is currently beyond the scope of the GTFS Realtime Validator, which focuses on specification compliance and feed integrity, but is something that could be investigated as part of the overall platform in the future.

Would you share this report to explain the error to a colleague? A vendor? If so, would you share it all or just a portion?

Yes, says one agency, it "helps to have tangible examples showing something is wrong."

One agency said a "strong yes" to sharing with their vendor, and encouraging other agencies that with the same vendor to do the same, to be a "strong voice for change."

Another agency said the "theoretical objective of this tool is useful" for sharing with their vendor, but their vendor regularly questions and doubts their agency staff, so they are not sure if it would be productive to share the report.

Another agency said yes, they would share this within their agency, because GTFS Realtime responsibility is split between bus and rail divisions.

Another agency would like to share but thinks others using the report would need additional information about the potential consequences of each error/warning. However, they don't want the addition of such high-level information to "dumb down" the rest of the report.

This is too verbose to share outside of one agency's customer IT department. Maybe they would share with some of their "savvier" vendors.

How often would you return to this report (assuming that it is regularly updated): weekly, monthly, quarterly, annually, never?

Daily: One agency would look at this on a daily basis and share access with operations to fix issues (e.g., canceled trips).

Weekly: One agency said they would like to return to this report on a weekly basis, but they would only share it with others within the agency on a less frequent basis: managers every 2 weeks, directors once a month.

Monthly: One agency said they'd prefer to check reports on a monthly basis when their systems are performing as expected (on a weekly basis when they have known issues). Another agency said they would want to check reports on a monthly basis, as they think they probably check static GTFS validation reports on a similar cadence.

Quarterly: Another agency said they would change their bus schedules on a quarterly basis and would be interested in checking GTFS Realtime validation results about a week after each quarterly bus schedule change. Another agency said their quarterly schedule changes (pick) sometimes lead to "everything going haywire" across all their IT systems, so they would rather investigate GTFS Realtime validation issues at different times, when they know the rest of the systems and data are calm and stable.

During a procurement process: "It can inform the conversation about what we need out of a new vendor." Vendors will skirt around some specifics during requirements definition; this platform could help enforce clarity at that step. Sharing the platform with vendors would help agencies, because the errors/warnings would be coming from an objective third party.

Task 11. Tune and improve system performance

When preparing validation reports in Task 6 and Task 10, the team repeatedly ran and tuned the software workflow. If the validation platform is like a telescope observing the night sky, we had to move and adjust the telescope multiple times in order to fully discern validation issues.

Repeated testing by Dr. Rees found that validating a 24-hour sample of one agency's GTFS Realtime feed typically took on the order of 10 minutes processing time. The process produced 50 to 500Mb of detailed report data per 24-hour sample (depending upon the size and complexity of the given agency's transit network) and approximately 10Mb of summary and validation data. It's only the latter files that are required to present to end-users, but we retained both the detailed- and summary-level data for use during the user-testing process. These tests were run on a virtual server with 26Gb of RAM and 4 CPU cores.

Task 12. Plan for expanding catalog of GTFS Realtime feeds

Since this project began, Interline began upgrading Transitland to a new version 2.0.[10] As part of the upgrade, the Transitland Feed Registry is switching to use the Distributed Mobility Feed Registry (DMFR) format, a standardized representation of transit data sources. [11] DMFR data can easily be shared as static JSON files and managed in a decentralized way (e.g. a Git repository).

Transitland's DMFR-based catalog of feeds is in a Git repository hosted on GitHub at https://github.com/transitland/transitland-atlas/ As a public repository, it is open to additions by anyone, although changes must be approved by Interline or partner staff before they are "merged" into the full repository.

See Figure 5 for an example of the DMFR file for Metro Transit. This file was originally created by Interline staff and updated directly by Metro Transit staff after they publicly released their GTFS Realtime feed.

[10] For more information on Transitland 2.0, see https://transit.land/news/2019/10/17/tlv2.html

[11] For more information on the Distributed Mobility Feed Registry format, see https://github.com/transitland/distributed-mobility-feed-registry/

```
{
  "$schema": "https://dmfr.transit.land/json-schema/dmfr.schema-v0.3.0.json",
  "feeds": [
    {
      "spec": "gtfs",
      "id": "f-9zv-twin~cities~minnesota",
      "urls": {
        "static_current": "https://svc.metrotransit.org/mtgtfs/gtfs.zip",
        "static_planned": ["https://svc.metrotransit.org/mtgtfs/next/gtfs.zip"]
      },
      "license": {
        "url": "https://svc.metrotransit.org",
        "use_without_attribution": "yes",
        "create_derived_product": "yes",
        "redistribute": "yes"
      }
    },
    {
      "spec": "gtfs-rt",
      "id": "f-twin~cities~minnesota~rt",
      "urls": {
        "realtime_vehicle_positions": "https://svc.metrotransit.org/mtgtfs/vehiclepositions.pb",
        "realtime_trip_updates": "https://svc.metrotransit.org/mtgtfs/tripupdates.pb",
        "realtime_alerts": "https://svc.metrotransit.org/mtgtfs/alerts.pb"
      },
      "feed_namespace_id": "o-9zvw-metro",
      "associated_feeds": [
        "f-9zv-twin~cities~minnesota"
      ],
      "license": {
        "url": "https://svc.metrotransit.org",
        "use_without_attribution": "yes",
        "create_derived_product": "yes",
        "redistribute": "yes"
      }
    }
  ],
  "license_spdx_identifier": "CDLA-Permissive-1.0"
}
```


Figure 5. A JSON file in the DMFR (Distributed Mobility Feed Format) that provides metadata on Metro Transit's public GTFS and GTFS Realtime feeds. This file in the Transitland Atlas repository on GitHub can be edited by the public. Once edits are approved, they are ingested and used to power public Transitland websites and APIs.

Task 13. Propose a certification process that allows GTFS Realtime providers to have their feeds validated on a regular basis

Throughout the project, we collected feedback from Expert Review Panel (ERP) members and user-testing participants about the potential for a GTFS Realtime certification process:

ISO and other certification processes from otherwise unrelated topics may be able to serve as useful examples of processes to follow for GTFS Realtime certification.

GTFS Realtime systems and data are dynamic and can change as conditions change. A CAD/AVL system or related software package for producing GTFS Realtime data may perform well given one agency's conditions but not another's. How can certification be an ongoing process of assessment and improvement, rather than a one-time "stamp of approval"?

Certification for GTFS Realtime data will also need to consider the quality of static GTFS data. If static GTFS is of poor quality (e.g., shape and stop data in wrong locations, incorrect trip IDs), then the rider experience will be poor even if the GTFS Realtime data is technically correct.

During the course of this project, the research team has started conversations with other organizations involved in related work on the state of transit data. Future work to establish a GTFS Realtime certification program should coordinate with these stakeholders to determine how the platform developed in this project could build upon these existing efforts:

The California Integrated Travel Program (Cal-ITP)[12] has created draft Minimum GTFS Guidelines[13]. These guidelines include a process for transit agencies to follow that is intended to increase the quality and availability of GTFS and GTFS Realtime data. Validation for both static and real-time GTFS data is referenced in the document as a good practice, although precise grading and certification based on validation output is not defined.

MobilityData[14] is in the process of drafting a GTFS Grading Scheme[15]. This grading scheme defines a process by which a static GTFS dataset is sampled and then manually compared to the "real world" to determine how well it represents what a traveler would encounter and assign a grade accordingly. This grading scheme currently does not consider any software validation output for static or real-time GTFS data. A similar grading scheme could be considered based on output from static and real-time GTFS validation.

CAD/AVL vendors are responsible for producing a large amount of GTFS Realtime, but most do not understand the GTFS Realtime specification itself and produce incomplete data. How can these vendors be engaged? Ultimately, for a certification process to have an impact on the industry, vendors must be incentivized to fix issues that are discovered during a certification process. Ideally

[12] https://dot.ca.gov/cal-itp
[13] https://dot.ca.gov/programs/rail-and-mass-transportation/cal-itp
[14] https://mobilitydata.org/
[15] https://bit.ly/gtfs-grading-scheme-v1

vendors that produce quality products should be highlighted and vendors that produce sub-standard products encouraged to improve their offerings.

Is it sufficient to validate public GTFS Realtime feeds, or do agencies and/or vendors need to be able to validate private feeds before they are released? Multiple agencies that participated in the user-testing process said they would find it useful to be able to validate private feeds prior to public release. Some agencies also spoke of using the GTFS Realtime validator as a component within their automated testing/release processes. These agencies were well-equipped with in-house technical capabilities, so they may be able to make use of the open-source software for their own purposes, and our hosted validation platform can focus on the needs for the many more agencies that have fewer technical capabilities of their own.

Based on this feedback and the user-testing findings, the research team distilled many options into two overall strategies: a strategy that focuses on specification-compliance and a strategy that focuses on user-application effects (see Figure 6).

Implementing a certification process with a *specification-compliance focus* would continue to focus on identifying as many specific issues in GTFS Realtime feed as possible and displaying that list to agency staff and vendors. The initial development of the GTFS Realtime Validator library prior to this project followed a systems- and data-driven, bottom-up approach, with many rules being implemented based on errors observed in real feeds. We would expand the list of errors/warnings (see Table 1) and add functionality to the platform to help diagnose these errors/warnings, with more context for each. We would work with agencies and vendors to create a formal process to certify their systems do not raise any critical errors, or that the number of errors they raise are under a reasonable threshold. This is similar to the approach taken so far in the user testing in this project. While agencies appreciated the detailed view of errors, they also expressed a desire for a simplified view of the information that communicated what the potential impacts of those errors would be on real-world applications.

Alternatively, to implement a certification process by way of a *user-application focus* would focus on identifying a short list of critical features for applications, focusing on errors/warnings relevant to those issues, and providing agency staff and vendors with a good deal of information about the potential causes and the likely side effects of these issues. The user-application approach would primarily focus its users' time on issues that are actionable and directly impact riders. We could begin by focusing on trip planning, real-time information app, and arrival-time predictions specifically—all applications that are relevant to the riders/customers of a transit agency (see Figure 6).

By narrowing down the use-cases of the platform, we can focus the primary user interface and its documentation on the information that users need to know to diagnose a small number of GTFS Realtime feed issues that can cause specific problems for these applications. Agency staff could use the platform on an ongoing but occasional basis (most user-testing participants said they would want to check on a monthly or quarterly basis) or at key moments, such as when selecting vendors. The platform could continue to measure and prepare reports for all

feeds, even at times when agencies and vendors do not need them, since user-application reports would be concise. Would a short top-down report focused on these problems provide enough information to be valuable? Based on the fact that none of the agencies that participated in our user-testing sessions are doing any GTFS Realtime validation at all at present, we can certainly say that a short report would be an improvement upon the status quo of zero.

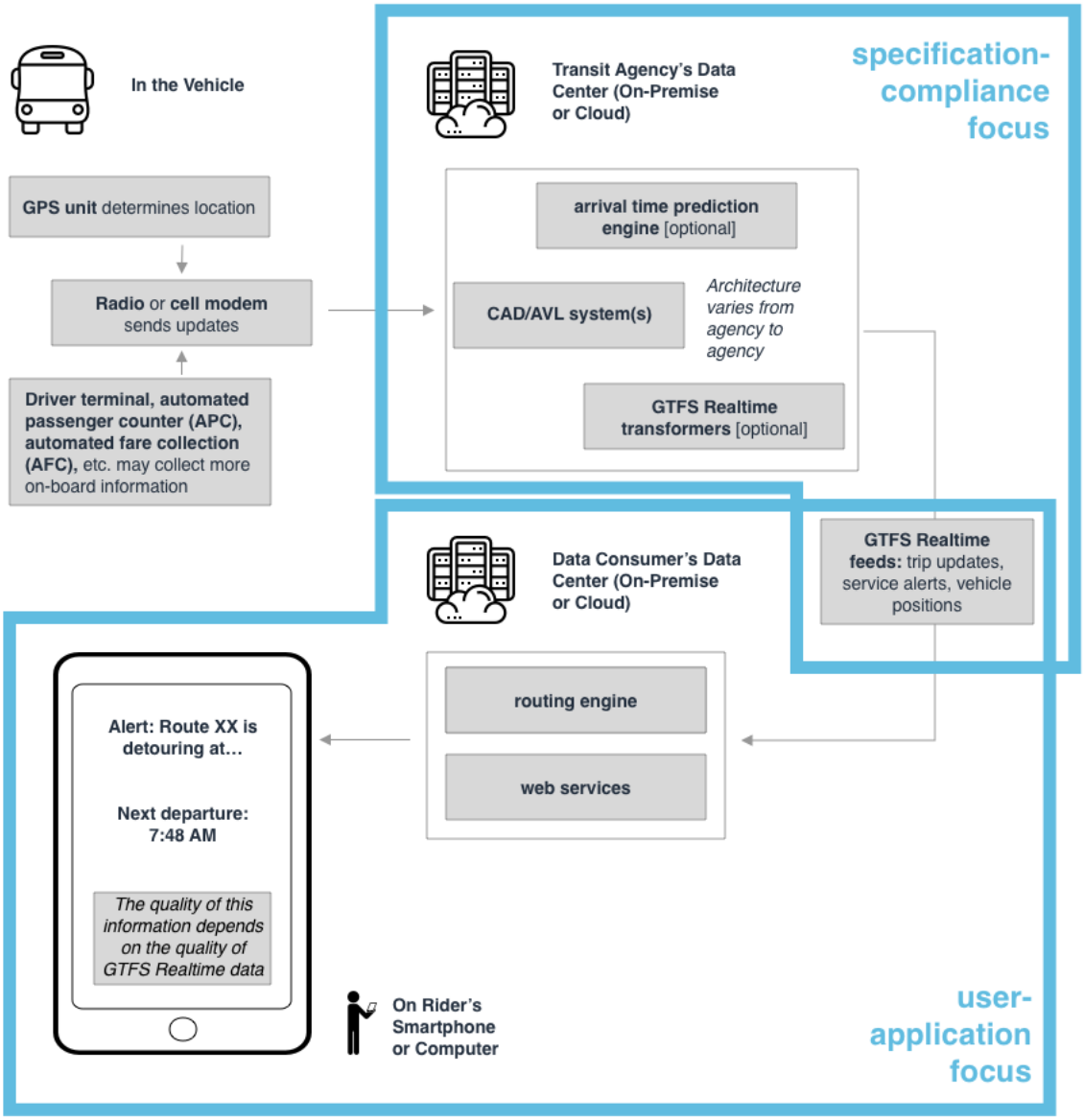


Figure 6. A revised version of the technical architecture diagram in Figure 1, now marking the portions of relevance to a specification-compliance focus (upper) and a user-application focus (lower). Note that both focus areas include the GTFS Realtime feeds; they differ in whether they include the "upstream" creation systems or "downstream" applications.

To consider which GTFS Realtime validation errors/warnings may be relevant to including on a shorter report that is focused on issues related to these types of rider-facing applications, the research team reviewed the full list of errors/warnings (Table 1) and considered potential side-effects each error/warning would generate downstream in the following applications:

- a trip planner: we used the example of the open-source OpenTripPlanner (OTP)[16]
- a real-time info app: we used the example of OneBusAway (OBA)[17]
- an arrival-time prediction algorithm: we used the example of TheTransitClock[18]

The research team has focused on open-source applications because of our familiarity with these, and also because it is straightforward to assess the specific impacts that a GTFS Realtime issue will have on the application's performance and output.

See Table 4 for an example list of issues that also contains potential impacts of that issue with a user-application focus. This explains how each error would impact transit riders using transit information. After discussion, the research team also added a handful of issues related to static GTFS and overall network connectivity (that is, Internet connectivity "uptime" for a GTFS Realtime feed)—we believe these additional issues would provide less technically sophisticated agencies to have a more holistic view of the most likely problems in the GTFS Realtime data. This shorter list focuses the validation platform and its users on a handful of key issues that affect riders and may be easier to understand by non-technical staff. The table also contains information about how each error may be translated into a grade that could be used in GTFS Realtime certification. We consider Table 4 to be a useful starting point for the next stage of planning and implementing a certification process for GTFS Realtime feeds.

The scope of this project focused on the internal validity of GTFS Realtime data, rather than its external accuracy, such as the accuracy of arrival-time predictions. Many ERP members and user-testing participants have suggested that as a future addition to the platform. After implementing a user-application focus and a short list of validation tests related to rider needs, the Transitland platform may serve as a foundation to add external accuracy checks in future phases of research.

[16] http://www.opentripplanner.org/
[17] https://onebusaway.org/
[18] https://thetransitclock.github.io/

Table 4. Initial List of GTFS Realtime Validation Issues that Directly Impact Rider Applications

Validator ID Code	Error Title	Likely effect in trip-planning app	Likely effect in real-time info app	Likely effect in arrival-time prediction engine	Fail Threshold	Impact on Letter Grade
	static GTFS stop error	Incorrect/missing trip itinerary	No real-time info	Inaccurate arrival-time estimate	fail on error in >5% trips	weighted reduction
	static GTFS trip error	Incorrect/missing trip itinerary	No real-time info	Inaccurate arrival-time estimate	fail on error in >5% trips	weighted reduction
	static GTFS shape problems	Incorrect display of trip	Incorrect display of trip	Inaccurate arrival-time estimate	fail on error in >5% trips	weighted reduction
E001	Not in POSIX time	No real-time info or Incorrect trip itinerary	Incorrect times	No arrival-time estimate	fail on 1 error	automatic F
E002	stop_time_updates not strictly sorted	No real-time info or Incorrect trip itinerary	Varies - times applied to wrong stops, partial or no RT data	N/A	fail on error in >5% trips	weighted reduction
E003 / E004	GTFS-rt trip_id or trip specified by (route_id,direction_id,start_time) does not exist in GTFS data	No real-time info or Incorrect trip itinerary	No real-time info	No arrival-time estimate	fail on error in >5% trips	weighted reduction
E009	GTFS-rt stop_sequence isn't provided for trip that visits same stop_id more than once	No real-time info or Incorrect trip itinerary	Varies - times applied to wrong stops, partial or no RT data	N/A	fail on error in >5% trips	weighted reduction
E011	GTFS-rt stop_id does not exist in GTFS data	No real-time info	No real-time info	No arrival-time estimate	fail on error in >5% trips	weighted reduction
E020 / E021	Invalid start_time or start_date format	No real-time info	No real-time info	No arrival-time estimate	fail on 1 error	automatic F
E022	Sequential stop_time_update times are not increasing	No real-time info or Incorrect trip itinerary	Incorrect times	N/A	fail on error in >5% trips	weighted reduction
E026	Invalid vehicle position	N/A	Missing vehicle from map	Inaccurate arrival-time estimate	fail on error in >1% trips	weighted reduction

Validator ID Code	Error Title	Likely effect in trip-planning app	Likely effect in real-time info app	Likely effect in arrival-time prediction engine	Fail Threshold	Impact on Letter Grade
E029	Vehicle position far from trip shape	N/A	Vehicle in wrong location on map	Inaccurate arrival-time estimate	fail on error in >10% trips	weighted reduction
E043	stop_time_update doesn't have arrival or departure	No real-time info	No real-time info	N/A	fail on error in >1% trips	weighted reduction
E044	stop_time_update arrival/departure doesn't have delay or time	No real-time info	No real-time info	N/A	fail on error in >1% trips	weighted reduction
E040	stop_time_update doesn't contain stop_id or stop_sequence	No real-time info	No real-time info	No arrival-time estimate	fail on error in >1% trips	weighted reduction
W007	Refresh interval is more than 35 seconds	Stale real-time data shown to riders	Stale real-time data shown to riders	Stale real-time data shown to riders	fail on warning >10% messages	weighted reduction
W008	Header timestamp is older than 65 seconds	Stale real-time data shown to riders	Stale real-time data shown to riders	Stale real-time data shown to riders	fail on warning >10% messages	weighted reduction
	Network connectivity failure / uptime	No real-time info	No real-time info	No arrival-time estimate	fail on >5% of requests	weighted reduction

Plans for Implementation

The next steps for deploying the technology developed in this project are to:

Focus the User Interface. Further refine the Transitland GTFS Realtime Validation Platform user interface to highlight the user-facing issues identified in Table 4. We may keep the option to view additional, detailed specification-compliance information, including many or all of the issues in Table 1. However, this detailed information would be deprioritized. Users (agency staffers and/or vendors) would first see the small subset of issues that concern rider applications.

Operations for Ongoing Validation. Reduce the frequency of GTFS Realtime sampling and the retention period of samples to support the user-facing issues in Table 4. It should still be possible to identify many of the other issues listed in the full Table 1, and the user interface will still allow agency staff and vendors to optionally "drill down" to view in full detail as many of these issues as are detected. This will manage the resources and costs required to maintain the platform. If the sampling frequency and retention period are not sufficient to meet all the needs of a particular agency, they may consider running their own copy of the GTFS Realtime Validator library, either temporarily or on an ongoing basis as part of their own systems architecture. (To use our metaphor, they can use their own telescope if the shared one does not regularly scan their portion of the sky as often as they require for their own needs.)

Outreach to More Agencies. The next phase of work should engage more agencies. Maintaining the Transitland Atlas feed catalog is underway (agency staff and third-party contributors are already helping Interline staff to maintain the Transitland Atlas repository on GitHub[19]). We should reach out to all of these agencies, and also have an open call to others, to use the next phase of the validator platform. This process will likely be self-serve, with a survey rather than a live consultation by video.

Collaborate with Relevant Organizations on Certification Process. The team will discuss this topic with potential sponsors at TRB and FTA; with MobilityData, a non-profit membership organization founded recently in Canada with members who represent transit agencies and relevant transit technology vendors from around the world; with the California Integrated Travel Program (Cal-ITP), who has established a draft Minimum GTFS Guidelines for California transit agencies, and also with Interline's consulting and services clients (some of whom may benefit from sponsoring and participating further in the creation of the platform and process).

[19] See https://github.com/transitland/transitland-atlas

Conclusions

GTFS Realtime feeds are an important component for public transit agencies of all sizes. These data feeds power many useful websites and mobile applications for riders. However, GTFS Realtime feeds sit in the middle of often complex technical architectures, bridging between an agency's internal systems and external data consumers. GTFS Realtime feeds rarely get direct attention via manual or automated means to ensure their contents meet the data specification, are internally consistent, and will not cause issues for "downstream" consuming applications.

This project has found clear benefits to the full, cloud-based GTFS Realtime validation platform. Agency staff who participated in testing the platform reported they currently have no equivalent means to inspect or test their real-time data feeds. Without prompting, agency staff member comments included:

- "I'm amazed and looking forward to what more you do with this tool."
- "What you guys are doing is really helpful. Even if we can't fix them it's good to know what the problems are."

Giving agencies actionable information about problems in real-time data could ultimately lead to improve real-time data quality, which could in turn lead to improved rider perception of the agency, increased ridership, and improved agency operations. One of the more immediate impacts could be by equipping agencies to better work with the vendors of their GTFS Realtime systems. This platform provides a rigorous view into the quality and performance of an agency's GTFS Realtime system, from a neutral third-party. By equipping agencies with this information, the platform can help them in their conversations with current vendors and/or plans to procure new systems.

Future work outlined in the "Plans for Implementation" section will expand the platform to handle more agencies in an ongoing and self-serve manner, while also focusing on the user interface and platform resources on the most important of issues for rider needs. These validation results will form the core of a GTFS Realtime certification process, to be further developed in coordination with other industry stakeholders. Together, these efforts will improve the quality of real-time information for transit agencies, their supporting vendors, and ultimately the riders that benefit from high-quality transit data.

References

[1] K.E. Watkins, B. Ferris, A. Borning, G.S. Rutherford, and D. Layton. Where Is My Bus? Impact of mobile real-time information on the perceived and actual wait time of transit riders. *Transportation Research Part A: Policy and Practice*, Vol. 45, 2011, pp. 839-848.

[2] Cluett, S. Bregman, and J. Richman (2003). *Customer Preferences for Transit ATIS*. Federal Transit Administration.

[3] Ferris, K. Watkins, and A. Borning, OneBusAway: results from providing real-time arrival information for public transit. Presented at the Proceedings of the 28th international conference on Human factors in computing systems, Atlanta, Georgia, USA, 2010.

[4] Gooze, K. Watkins, and A. Borning (2013), Benefits of Real-Time Information and the Impacts of Data Accuracy on the Rider Experience, in Transportation Research Board 92nd Annual Meeting, Washington, D.C., January 13, 2013.

[5] L. Tang and P. Thakuriah. Ridership effects of real-time bus information system: A case study in the City of Chicago. *Transportation Research Part C: Emerging Technologies*, Vol. 22, 2012, pp. 146-161.

[6] Brakewood, G. Macfarlane, and K. Watkins, The impact of real-time information on bus ridership in New York City, *Transportation Research Part C: Emerging Technologies*, Vol. 53, 2015, pp. 59-75.

[7] TCRP Transit Capacity and Quality of Service Manual, 3rd Edition

[8] K. Dziekan and A. Vermeulen. Psychological Effects of and Design Preferences for Real-Time Information Displays. *Journal of Public Transportation*, Vol. 9, No.1, 2006.

[9] K.F. Turnbull and R.H. Pratt. TCRP Report 95: Traveler Response to Transportation System Changes. Chapter 1 1 -Transit Information and Promotion. Transportation Research Board of the National Academies, Washington, D.C., 2003.

[10] S.J. Barbeau. Quality Control - Lessons Learned from Deployment and Evaluation of GTFS Realtime Feeds, 2018 Transportation Research Board 97th Annual Meeting, Washington, D.C., January 9, 2018.

[11] Dara-Abrams. Presentation in the Transformative Trends in Transit Data: General Transit Feed Specification (GTFS) Bonanza session, at the Transportation Research Board Annual Meeting, 2016.

[12] Dara-Abrams and I. Rees. Publishing, Consuming, and Improving GTFS with the Transitland Platform, URISA GIS in Transit conference, 2017.

[13] Dara-Abrams. Using Onestop IDs as a Crosswalk Between Transit Agencies and Data Sources, URISA GIS in Transit conference, 2015.

[14] C.L. Schweiger. TCRP Synthesis 91: Use and Deployment of Mobile Device Technology for Real-Time Transit Information. Washington, DC: The National Academies Press, 2001.

[15] C.L. Schweiger. TCRP Synthesis 115: Open Data: Challenges and Opportunities for Transit Agencies. Washington, DC: The National Academies Press, 2015.

[16] Barbeau, S. Overcoming Barriers for the Wide-scale Adoption of Standardized Real-time Transit Information. National Institute for Transportation and Communities, 2018.