NCTR
NATIONAL CENTER for
TRANSIT RESEARCH

Improving Access to Transit Through Crowdsourced Information

Final Report

November 2017

PROJECT NO.
BDV26-977-05

PREPARED FOR
Florida Department of Transportation

NCTR at CUTR
National Center for Transit Research

National Center for Transit Research
Center for Urban Transportation Research
University of South Florida
www.nctr.usf.edu

Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the Department of Transportation University Transportation Centers Program and the Florida Department of Transportation, in the interest of information exchange. The U.S. Government and the Florida Department of Transportation assume no liability for the contents or use thereof.

The opinions, findings, and conclusions expressed in this publication are those of the authors and not necessarily those of the State of Florida Department of Transportation.

Metric Conversion

SI* Modern Metric Conversion Factors as provided by the Department of Transportation, Federal Highway Administration http://www.fhwa.dot.gov/aaa/metricp.htm

Length

SYMBOL	WHEN YOU	MULTIPLY BY	TO FIND	SYMBOL
in	inches	25.4	millimeters	mm
ft	feet	0.305	meters	m
yd	yards	0.914	meters	m
mi	miles	1.61	kilometers	km

Area

SYMBOL	WHEN YOU	MULTIPLY BY	TO FIND	SYMBOL
in^2	square inches	645.2	square	mm^2
ft^2	square feet	0.093	square meters	m2
yd^2	square yard	0.836	square meters	m2
ac	acres	0.405	hectares	ha
mi^2	square miles	2.59	square	km^2

Length

SYMBOL	WHEN YOU	MULTIPLY	TO FIND	SYMBOL
mm	millimeters	0.039	inches	in
m	meters	3.28	feet	ft
m	meters	1.09	yards	yd
km	kilometers	0.621	miles	mi

Area

SYMBOL	WHEN YOU	MULTIPLY BY	TO FIND	SYMBOL
mm^2	square	0.0016	square inches	in^2
m2	square meters	10.764	square feet	ft^2
m2	square meters	1.195	square yards	yd^2
ha	hectares	2.47	acres	ac
km^2	square	0.386	square miles	mi^2

*SI is the symbol for the International System of Units. Appropriate rounding should be made to comply with Section 4 of ASTM E380.

Technical Report Documentation Page

1. Report No.

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle:
Improving Access to Transit through Crowdsourced Information

5. Report Date
November 2017

6. Performing Organization Code

7. Author(s)
Sean J. Barbeau, Ph.D.

8. Performing Organization Report No.
NCTR 77969/ BDV26 977-05
U.S. DOT DTRS98-G-0032

9. Performing Organization Name and Address
National Center for Transit Research
Center for Urban Transportation Research
University of South Florida
4202 E Fowler Avenue, CUT 100, Tampa, FL 33620-5375

10. Work Unit No. (TRAIS)

11. Contract or Grant No.
BDV26-977-05

12. Sponsoring Agency Name and Address
Research and Innovative Technology Administration
U.S. Department of Transportation, Mail Code RDT-30
1200 New Jersey Ave, SE, Room E33, Washington, DC 20590-0001

Florida Department of Transportation Research Center
605 Suwannee Street, MS 30, Tallahassee, FL 32399

13. Type of Report and Period Covered
Final Report
04/02/14-11/14/17

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

The purpose of this research was to facilitate the ongoing collection of information from the public about potential areas of multimodal service and infrastructure improvements and easily share these problems with transit agencies, departments of transportation, and city and county governments. The research team implemented open-source software that leveraged the Open311 issue-reporting standard to capture various types of data from actual users of public transportation via the OneBusAway mobile app, a real-time transit information system. Lessons learned from regional collaboration surrounding issue reporting and infrastructure improvements are discussed, as are the technical design and challenges behind implementing such a system. The results of six months of system deployment in Hillsborough and Pinellas Counties are presented, including specific examples of cross-jurisdictional and multimodal issues reported by the public. Using this crowd-sourced data and issue management tools, transit agencies, departments of transportation, and city and county government will be able to better target improvements to bike, pedestrian, and transit infrastructure.

17. Key Words
Mobile app, issue, problem, transit, public transportation, intermodal, open311, GTFS, feedback

18. Distribution Statement

19. Security Classification (of this report)	20. Security Classification (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	68	

Acknowledgments

This report was prepared by the National Center for Transit Research, at the Center for Urban Transportation Research, University of South Florida, through the sponsorship of the Florida Department of Transportation. The author would like to sincerely thank the Florida Department of Transportation's Elba Lopez for her oversight, management, and support of this research effort. The author would also like to thank Hillsborough Area Regional Transit (HART) and Pinellas Suncoast Transit Authority (PSTA) for their participation in the project, as well as Cambridge Systematics that supports OneBusAway for HART, as well as Clever Devices, PSTA's Automatic Vehicle Location vendor, and SeeClickFix for their implementation of the Open311 API. Thanks also to Cagri Cetin for his invaluable work implementing many software components critical to the success of this project and troubleshooting and documenting data issues.

FDOT Project Manager
Elba Lopez, Regional Transit/Intermodal Systems Planning, District 7

Executive Summary

Offering real-time arrival information to riders via mobile applications has been shown to improve the rider's perception of transit and increase ridership. This direct connection to riders also offers the agency an opportunity to collect feedback on how transit service and infrastructure can be improved, including pedestrian and bicycle access to transit. Feedback from these travelers could also be key in reducing the risk for bicyclists and pedestrians. According to the National Highway Traffic Safety Administration, in 2015 the largest number of pedalcyclist fatalities occurred in Florida (150), followed by California (129) [1]. Every other State had 50 or fewer pedalcyclist fatalities [1]. Additionally, Florida also had the highest pedalcyclist fatality rate per million population (7.4 fatalities per million residents), compared to the national rate of 2.5 [1]. Similarly, Florida had the second highest number of pedestrian fatalities (628) as well as the second highest pedestrian fatality rate per 100,000 at 3.10 [2]. Hillsborough County in the Tampa Bay, Florida area has more traffic deaths per resident than any other large county in the country [3]. Rider feedback could be critical to providing safe and accessible transit service and planning improvements for supporting pedestrian and bicycle infrastructure.

While mobile apps have opened a direct line of feedback from the rider to the agency, managing the sheer volume of this rider feedback remains challenging. This is especially true when determining where to assign the responsibility for addressing rider-identified problems where various departments and agencies (e.g., city/county governments) are involved (e.g., does this broken bench belong to the transit agency or the county?).

The purpose of this research was to facilitate the ongoing collection of information from the public about potential areas of multimodal service and infrastructure improvements and easily share these problems with transit agencies, departments of transportation, and city and county governments. The research team implemented open-source software that leveraged the Open311 issue-reporting standard to capture of various types of data from actual users of public transportation via the OneBusAway mobile app, a real-time transit information system. Agencies can now use any Open311-compatible issue report application along with the OneBusAway app, enabling them to effectively triage, manage, and respond to feedback from the public.

Deployment of the system for six months in the Tampa Bay area resulted in a total of 677 issue reports captured via the OneBusAway app for Hillsborough County, with approximately 87% of the issues being related to arrival times and schedule problems and 13% related to other categories that included public safety issues. Examples of reported cross-jurisdictional safety issues include an open electrical box near a bus stop, as well as a broken storm drain with debris that impeded a pedestrian and bike travel path. Hillsborough Area Regional Transit (HART) was able to relay these reports, including a photo of the issue, to Frontier Communications and the City of Tampa, respectively, who quickly resolved the problems. In Pinellas County, of the 4780 issues reported only 54

(1.13%) were related to public transportation. This is because PSTA hadn't widely advertised the availability of their information in OneBusAway at the time of the deployment, and Pinellas County has been using the SeeClickFix issue reporting tool for several years. HART and PSTA reported that a key advantage to using a dedicated issue management platform was the additional transit metadata that OneBusAway was able to attach to the issues as well as the ability to create data-driven customer service and maintenance performance goals.

The research team identified potential areas of improvement in the Open311 specification, including standardizing a discovery protocol for Open311-compliant systems, the direct submission of image data in place of URL, and the addition of "hidden fields" for metadata that shouldn't be shown to the public but is useful to the agency for troubleshooting problems.

General guidance for issue reporting categories related to public transportation would assist the industry in tracking common metrics across agencies as well as communication between agencies about issues.

Future improvements to OneBusAway should examine the ability to capture the time when the user arrives at a bus stop to assist with troubleshooting arrival and departure time predictions problems, as well as the ability to collect travel path information (with the user's permission). This feature would need to be balanced with privacy protections and transparency for the user.

For new agencies implementing issue management systems, participating agencies suggest that the agency have dedicated personnel ready and trained prior to making the system available to the public, be prepared to manage two systems during a transition period, put thought into secondary questions for users to limit back-and-forth communication, and ensure that public and private comments are channeled to the correct venues.

Based on the research team's experience troubleshooting GTFS (General Transit Feed Specification)-realtime data in this project, the creation of an open-source GTFS-realtime validation tool is strongly recommended. Such a tool would save the industry significant transit agency, AVL vendor, and application developer time and resources that are currently being spent on identifying and debugging errors in feeds.

Table of Contents

List of Figures

Abbreviations and Acronyms

API	Application Programming Interface
AVL	Automatic Vehicle Location
CUTR	Center for Urban Transportation Research
FDOT	Florida Department of Transportation
GTFS	General Transit Feed Specification
GTFS-realtime	General Transit Feed Specification-realtime
HART	Hillsborough Area Regional Transit
ID	Identifier
OBA	OneBusAway
PSTA	Pinellas Suncoast Transit Authority
TCRP	Transit Cooperative Research Program
URL	Uniform Resource Locator
USF	University of South Florida

Chapter 1 - Introduction

Real-time transit information has been shown to have many benefits to transit riders, including shorter perceived wait time [4], shorter actual wait time [4], a lowered learning curve for new riders [5], and increased feeling of safety (e.g., at night) [6, 7].   Transit agencies who have deployed real-time information have also benefitted from increased ridership [8, 9], as well as a better perception of the agency and its transit service, even if its service has not actually changed [10].

This direct connection to riders also offers the agency an opportunity to collect feedback on how transit service and infrastructure can be improved, including pedestrian and bicycle access to transit. Feedback from these travelers could also be key in better understanding bicycle and pedestrian fatalities. According to the National Highway Traffic Safety Administration, in 2015 the largest number of pedalcyclist fatalities occurred in Florida (150), followed by California (129) [1]. Every other state had 50 or fewer pedalcyclist fatalities [1]. Additionally, Florida also had the highest pedalcyclist fatality rate per million population (7.4 fatalities per million residents), compared to the national rate of 2.5 [1]. Similarly, Florida had the second highest number of pedestrian fatalities (628) as well as the second highest pedestrian fatality rate per 100,000 at 3.10 [2]. Hillsborough County in the Tampa Bay, Florida, area has more traffic deaths per resident than any other large county in the country [3]. Rider feedback is critical to providing safe and accessible transit service and for planning improvements to pedestrian and bicycle infrastructure.

While mobile apps have opened a direct line of feedback from the rider to the agency, managing the sheer volume of this rider feedback remains challenging. This is especially true when determining where to assign the responsibility for addressing rider identified problems when various departments and agencies (e.g., city/county government) are involved (e.g., does this broken bench belong to the transit agency or the county?).

The purpose of this research was to facilitate the ongoing collection of information from the public about potential areas of multimodal service and infrastructure improvements and easily share these problems with transit agencies, departments of transportation, and city and county governments. As part of this research, the team examined existing issue management systems, as well as the Open311 standard for issue reporting, to determine how existing challenges with the OneBusAway mobile app user feedback could be addressed, which are discussed in Chapter 2. Chapter 3 discusses the creation of the Open311-based issue reporting solution for the OneBusAway mobile app, Chapter 4 presents the results of the six-month deployment as well as lessons learned, and Chapter 5 presents the project's conclusions. To demonstrate an improved issue reporting system within the context of a regional system with multiple transit agencies, the research team added real-time information for a second transit agency to the OneBusAway Tampa Bay region in the OneBusAway mobile app, which is detailed in Appendix A.

Chapter 2 - Challenges with Managing User Feedback

OneBusAway (OBA) is a mobile application for real-time multimodal information that allows transit riders to see predicted arrival and departure times for each bus stop [11]. Unlike other mobile transit apps, OneBusAway is open-source, which means that anyone can download and deploy the software source code in their region. OneBusAway started as a project created by graduate students at the University of Washington, and has since spread to over 10 cities with over 325,000 active users (users that have opened the app in the last 30 days).

OneBusAway was officially launched in Tampa, Florida in August 2013 [12], and continues to be the main source of real-time public transportation information for Hillsborough Area Regional Transit (HART) for over 35,000 active users in Tampa. However, one of the significant challenges in deploying such a popular app is the large amount of feedback provided by users. The initial design of OneBusAway provided a single regional email contact point for feedback – tapping on "Contact Us" in the OneBusAway app would open an email directed to the regional contact point (initially HART).

However, this feedback design creates several challenges:

1. Managing the issue lifecycle via email is inefficient – Email is not a good medium for tracking individual issues, including who they are assigned to and when a specific issue is resolved.
2. OneBusAway regions can contain multiple agencies – If there is only a single email contact point for a region, multiple transit agencies must share that email address. This causes additional confusion and issue lifecycle management difficulties.
3. Intermodal issues are difficult to manage – Because the transit agency is the lead organization responding to issues, any issues related to sidewalks, bicycle lanes, parking, right of way, roads, or other infrastructure not managed by the transit agency must be shared with other organizations.

In 2011, Ferris et al. implemented a proof-of-concept crowd-sourcing issue reporting system within OneBusAway that focused on arrival times (i.e., trip) problem reports, and stop information problem reports [13]. These improvements gave users a way to categorize problems falling into these two categories and submit them to a OneBusAway server database, although additional challenges were encountered:

1. Processing and analyzing the volume of reports is burdensome – The issue management interface within the OneBusAway administrative web site does not allow the transit agency to search, analyze, and aggregate issues. As a result, dealing with a large volume of reports is inefficient and costly leading to concerns of delays in response.

2. Arrival times and stop information does not sufficient cover problems reported by users – HART wanted to add more categories for users to report problems for (e.g., Wi-Fi on bus not working, broken stop benches)
3. The problem reporting categories were rigid and hard-coded into the mobile app - HART needed the ongoing ability to add and update problem-reporting categories within the app without requiring software code changes.
4. The administrative interface within OneBusAway to manage the issue lifecycle was very limiting – All reports were anonymous, which did not give HART the ability to respond to users to let them know that an issue was resolved. This also creates the rider perception that the transit agency is not receiving or reacting to any of the feedback. Additionally, multiple tiers of user permissions were not supported in the administrative interface, which did not give the ability to assign restricted permissions to customer service staff that should be able to manage reported issues, but not have access to other OneBusAway administrative tools (e.g., updating system schedule data).

Based on the above challenges encountered with the existing email and arrival time and stop information reporting mechanisms, the project team developed the following criteria in coordination with the participating agencies for creating a new issue reporting solution within OneBusAway:

- **Interoperability** – Issue reporting management systems need to allow interoperability with external systems to be successfully integrated with OneBusAway. This type of integration is typically accomplished via an open Application Programming Interface (API) that would allow OBA to interact and submit issue reports to the system. The new data collection system should be easy to implement for multiple platforms (e.g., iOS, Android).
- **Powerful interface to manage issues –** The issue management administrative interface should support the ability to search, triage, aggregate, summarize, assign, and track the progress towards resolution of issues. The administrative interface for managing issues should also be separated from the other administrative features of OneBusAway.
- **User feedback and interaction** - After issue submission, the data collection system should return useful feedback to the application, and allow additional communication between the user and managing agency if needed. Anonymous reporting of issues is also desired.
- **Multi-jurisdictional issue notification and management** – Transit service often occurs at the boundary of service jurisdictions where infrastructure may be owned and maintained by several different parties. For example, a traveler walking to a bus stop may encounter a broken sidewalk that is maintained by the county. When they reach their bus stop, if the bench is broken, it may be maintained by the transit agency. If a bus consistently gets stuck at a red traffic light, that traffic signal may be owned by the city. In addition, a pothole on the road, which the traveler must step around to board the bus, may be owned by the department of

transportation. The public should be shielded from these complexities. Travelers should be able to simply report a problem to a system, and that system should be able to automatically assign the problem based on exactly where it was reported and the issue category, and then notify the appropriate jurisdiction that would own that problem. This process would cut down on frustration from both the public and from jurisdictions that get many issues that they themselves cannot fix, and instead must forward to another jurisdiction. The research team worked closely with the Florida Department of Transportation (FDOT) District 7 as well as HART, Pinellas Suncoast Transit Authority (PSTA), and Pinellas County to ensure that a solution could meet the needs of each of these agencies in the greater Tampa Bay area.

- **Standardization of data elements** – If possible, standardized reporting protocols and data elements should be used so that a common set of information is reported across all OBA deployments.

2.1 Review of existing systems

The research team reviewed several civic issue management tools and programs, which are discussed in the following sections.

2.1.1 Mystery Riders Program

Some public transportation agencies use a "mystery rider" program to help collect data about how the transit system is working. These programs have an employee go "undercover" on the transit system and impersonate a normal rider so the agency can see what the riders experience is like. One such program by TransDev [14] monitors the bus routes and measure drivers' compliance with expected behaviors. Mystery riders are specially trained to create accurate reports from agencies. They randomly check the routes and verify driver behavior on particular routes.

The following items are examples of information that are monitored by the mystery riders:

- ➢ Public timetable availability
- ➢ Operator customer service acumen
- ➢ Cleanliness of bus
- ➢ Timeliness of trip
- ➢ Comfort of the HVAC and the ride
- ➢ Operational safety

2.1.2 Street Bump

Street Bump is a mobile crowdsourcing app that helps residents improve their neighborhood by automatically reporting potholes in roads while they drive [15]. The user opens the street bump mobile app while they are driving and the mobile app uses the embedded sensors (e.g., accelerometer, gyroscope) in the phone to detect bumps and report this information to local governments.

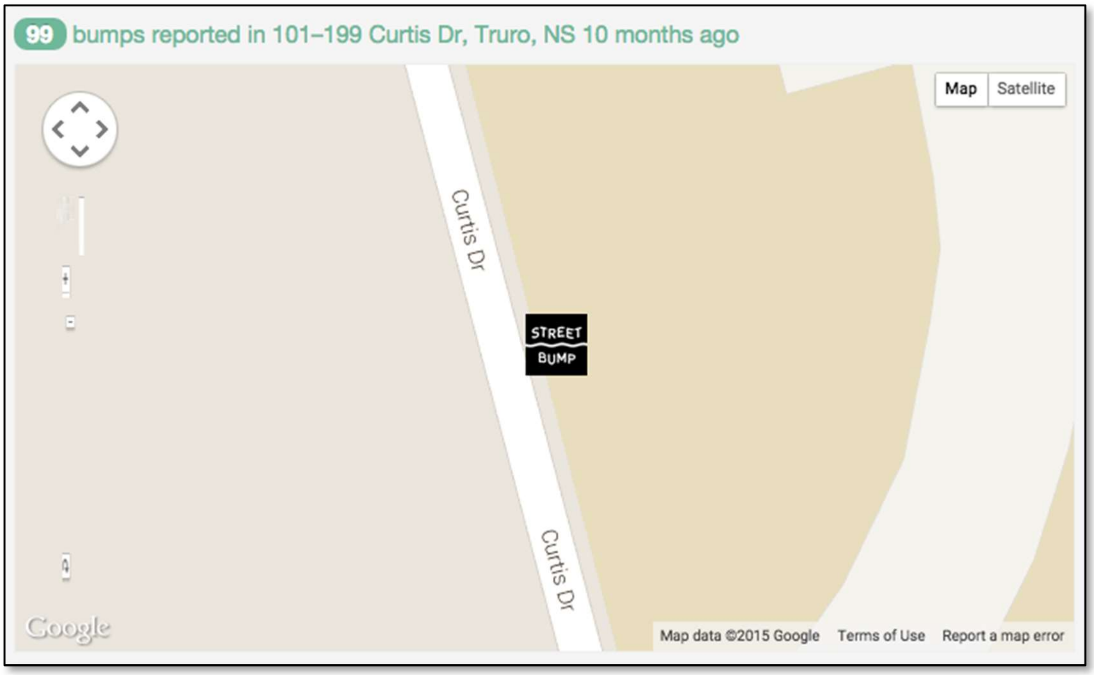


Figure 1 - Reported bump from a mobile device

2.1.3 *SeeClickFix*

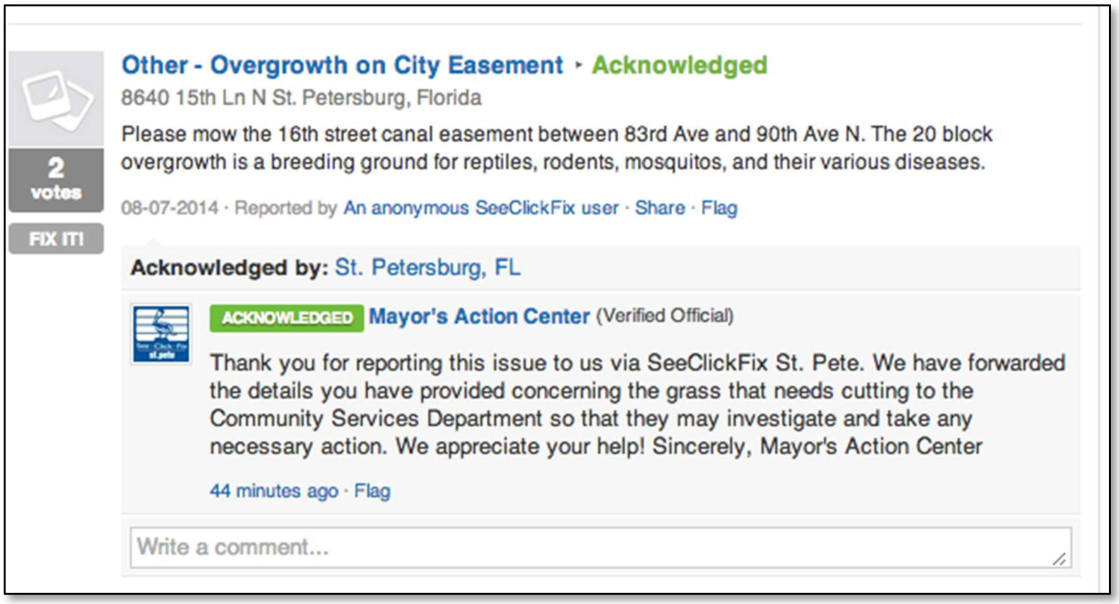
SeeClickFix is an interactive platform for citizens to report non-emergency issues, and governments to manage, track and reply to reported issues [16].

SeeClickFix is used as a crowdsourced issue reporting platform by many municipalities, including Pinellas County in the Tampa Bay area. Pinellas County coordinates with the following agencies for resolving issues reported by the public:

- PSTA
- Duke Energy
- FDOT District 7
- St. Petersburg
- Belleair
- Clearwater
- Dunedin
- Gulfport
- Largo
- Oldsmar
- Pinellas Park

- Safety Harbor
- Seminole
- St. Pete Beach
- Tarpon Springs
- Treasure Island

SeeClickFix has a web and mobile applications for users to report issues. Figure 2 shows an issue displayed on the SeeClickFix web page that was reported by a member of the public to Pinellas County, and Figure 3 shows issue reporting categories that Pinellas County has defined for their jurisdiction. Jurisdictions can define their own set of categories for which they allow the public to report issues.



Other - Overgrowth on City Easement ‣ Acknowledged

8640 15th Ln N St. Petersburg, Florida

Please mow the 16th street canal easement between 83rd Ave and 90th Ave N. The 20 block overgrowth is a breeding ground for reptiles, rodents, mosquitos, and their various diseases.

08-07-2014 · Reported by An anonymous SeeClickFix user · Share · Flag

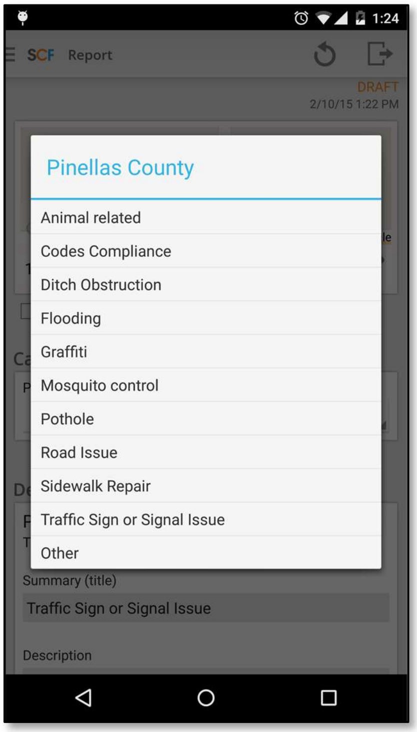
Acknowledged by: St. Petersburg, FL

ACKNOWLEDGED Mayor's Action Center (Verified Official)

Thank you for reporting this issue to us via SeeClickFix St. Pete. We have forwarded the details you have provided concerning the grass that needs cutting to the Community Services Department so that they may investigate and take any necessary action. We appreciate your help! Sincerely, Mayor's Action Center

44 minutes ago · Flag

Figure 2 - Issue submitted by the public in SeeClickFix



Pinellas County

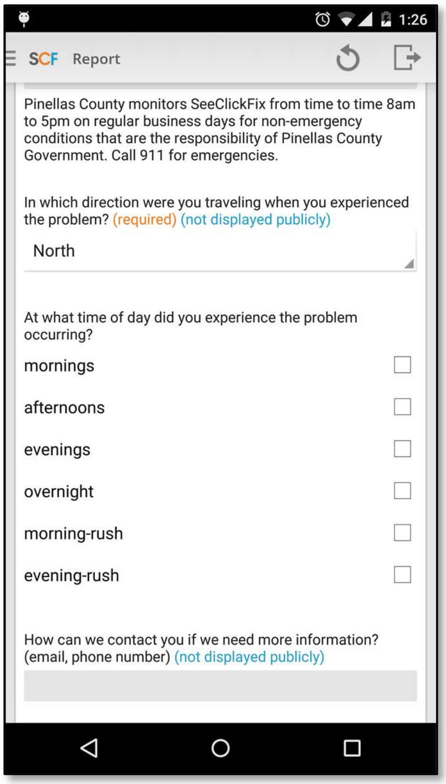
- Animal related
- Codes Compliance
- Ditch Obstruction
- Flooding
- Graffiti
- Mosquito control
- Pothole
- Road Issue
- Sidewalk Repair
- Traffic Sign or Signal Issue
- Other

Summary (title)

Traffic Sign or Signal Issue

Description

Figure 3 - Issue categories defined by Pinellas County



Pinellas County monitors SeeClickFix from time to time 8am to 5pm on regular business days for non-emergency conditions that are the responsibility of Pinellas County Government. Call 911 for emergencies.

In which direction were you traveling when you experienced the problem? (required) (not displayed publicly)

North

At what time of day did you experience the problem occurring?

- mornings □
- afternoons □
- evenings □
- overnight □
- morning-rush □
- evening-rush □

How can we contact you if we need more information? (email, phone number) (not displayed publicly)

Figure 4 - Interactive follow-up questions in the SeeClickFix mobile app

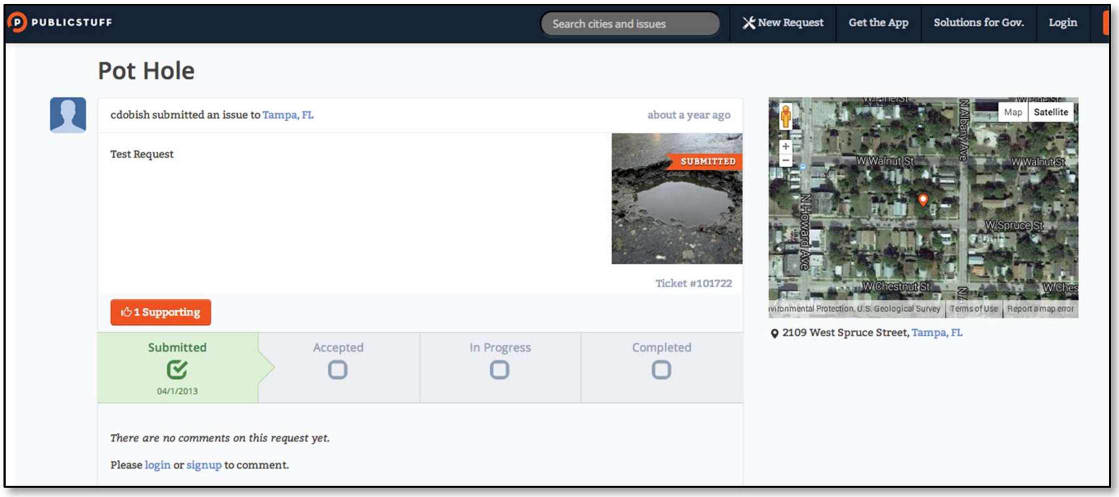
Jurisdictions can also define precise reporting boundaries using Geographic Information System (GIS) shape files so that they are only notified when an issue is reported within their precise jurisdiction. If desired, agencies can also subscribe to be notified for issues with specific categories that may be within their own jurisdiction but not their responsibility. For example, if the City of St. Petersburg is responsible for trash pickup within Pinellas County, they can choose to be notified whenever a trash issue is reported within the City of St. Petersburg, or they can elect not to be notified for trash issues within the City of St. Petersburg. Since jurisdictional boundaries and responsibilities for each type of issue can be complex, having this specificity of controlled reporting is very important to avoid flooding jurisdictions with issues that they are not interested in or do not have control over. In SeeClickFix, GIS files defining jurisdictional boundaries can be complex, including support for scenarios where a department of transportation may want to be notified if an issue is reported within 30 feet of their road network.

SeeClickFix also offers the ability to define "follow-up" questions for particular issue categories. For example, as shown in Figure 4, when a member of the public chooses the "Traffic Signal" category, Pinellas County requires them to answer several follow-up questions prior to submitting the issue, including the direction the person was traveling and time of day that they experienced the problem. This feature is important to provide the jurisdiction with all the information they need to take action on a reported problem, and reduces the overhead of back-and-forth communication with the person reporting the problem if the user did not provide the additional required information. A jurisdiction can still follow-up with a user if needed to collect extra information via a Facebook-like comments system.

SeeClickFix offers a subscription model for jurisdictions to pay for access to the service. SeeClickFix also includes custom-branded iOS and Android mobile apps for the jurisdiction as part of one of their packages.

2.1.4 PublicStuff

Similar to SeeClickFix, another crowdsourced reporting platform for non-emergency issues is PublicStuff [17] (Figure 5).



Pot Hole

cdobish submitted an issue to Tampa, FL

about a year ago

Test Request

Ticket #101722

Submitted ☑ 04/1/2013 | Accepted □ | In Progress □ | Completed □

2109 West Spruce Street, Tampa, FL

There are no comments on this request yet.

Please login or signup to comment.

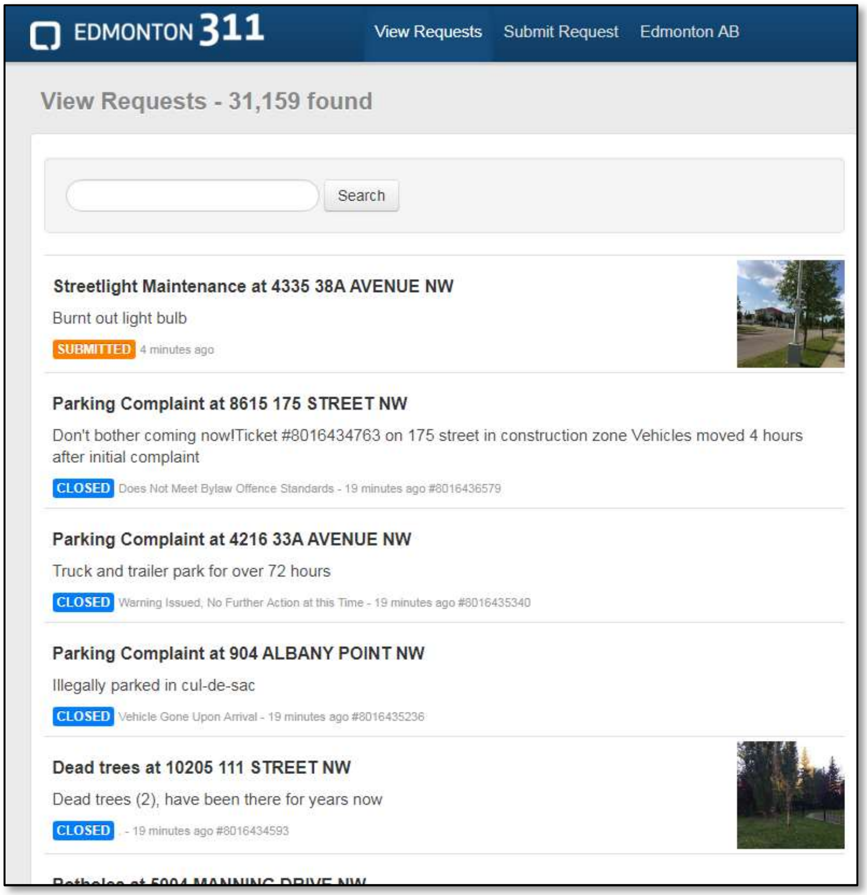
Figure 5 - Issue submitted to PublicStuff

PublicStuff also allows jurisdictions to define custom issue reporting categories as well as required follow-up questions that users must answer before submitting the issue.

PublicStuff is also subscription-based with annual plans, and also includes custom-branded iOS and Android mobile apps for the jurisdiction as part of one of their packages.

2.1.5 Connected Bits

Connected Bits [18] provides a similar issue reporting platform to SeeClickFix and PublicStuff (Figure 6).



View Requests - 31,159 found

Streetlight Maintenance at 4335 38A AVENUE NW

Burnt out light bulb

SUBMITTED 4 minutes ago

Parking Complaint at 8615 175 STREET NW

Don't bother coming now!Ticket #8016434763 on 175 street in construction zone Vehicles moved 4 hours after initial complaint

CLOSED Does Not Meet Bylaw Offence Standards - 19 minutes ago #8016436579

Parking Complaint at 4216 33A AVENUE NW

Truck and trailer park for over 72 hours

CLOSED Warning Issued, No Further Action at this Time - 19 minutes ago #8016435340

Parking Complaint at 904 ALBANY POINT NW

Illegally parked in cul-de-sac

CLOSED Vehicle Gone Upon Arrival - 19 minutes ago #8016435236

Dead trees at 10205 111 STREET NW

Dead trees (2), have been there for years now

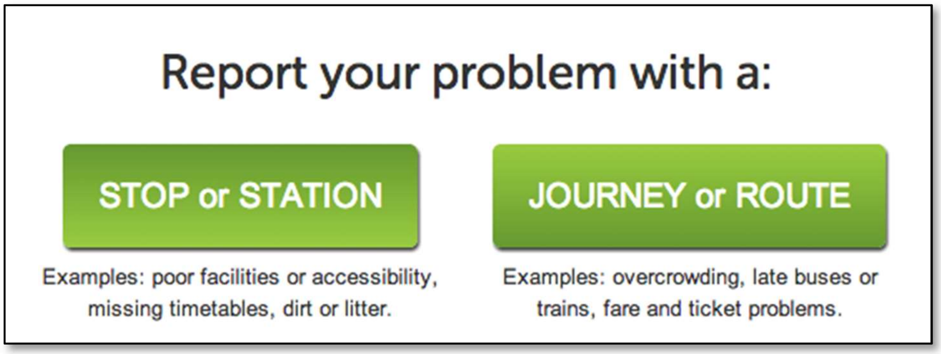
CLOSED - 19 minutes ago #8016434593

Potholes at 5004 MANNING DRIVE NW

Figure 6 - Edmonton 311 system by Connected Bits

2.1.6 FixMyTransport

FixMyTransport [19] was the only crowdsourced reporting system that has focused exclusively on public transport. FixMyTransport was deployed exclusively in the United Kingdom, but in January 2015 the organization behind FixMyTransport decided to stop offering the service. It had a proprietary API, which makes the platform less flexible. However, since it focuses on public transportation, it has more transit-specific issue categorization. Figure 7 and Figure 8 show the issue reporting flow that was used with FixMyTransport.

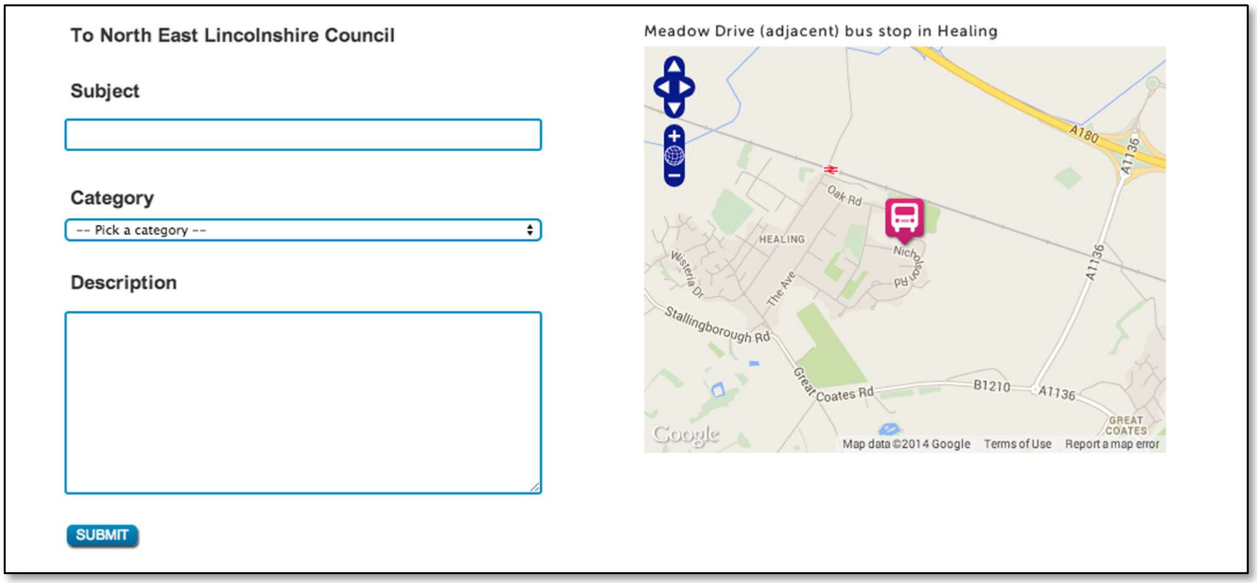


Report your problem with a:

Examples: poor facilities or accessibility, missing timetables, dirt or litter.

Examples: overcrowding, late buses or trains, fare and ticket problems.

Figure 7 - Issue categorization with FixMyTransport



To North East Lincolnshire Council

Subject

Category

-- Pick a category --

Description

Meadow Drive (adjacent) bus stop in Healing

Figure 8 - Issue submission with FixMyTransport

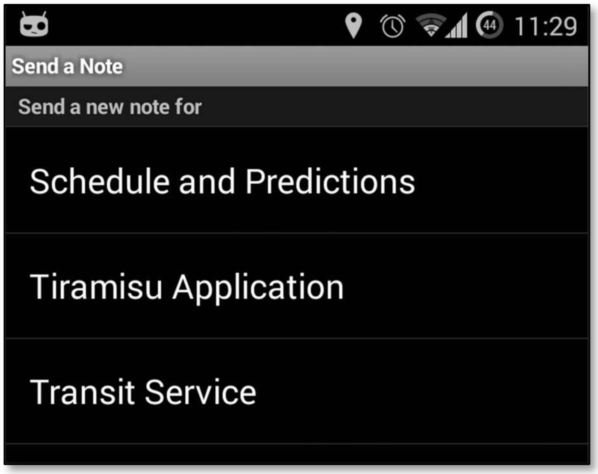
Different transportation sources are listed as the category and the users can specify the issues as free text during the issue submission (i.e., the issue itself does not need to be defined as a specific problem type).

The following shows the issue category list that was used by FixMyTransport:

- Stop or Station
    - Bus stop
    - Train station
    - Tram, metro or tube stop
    - Ferry terminal
- Journey of Route
    - Bus/Coach
    - Train
    - Tram/Metro underground
    - Ferry

2.1.7 Tiramisu Transit

Tiramisu Transit [20] is a mobile app from Carnegie Mellon University that focuses on providing transit riders real-time transit information in Pittsburgh, PA, Syracuse, NY, and New York City, NY. However, instead of requiring a vehicle tracking system, Tiramisu uses locations provided from user's devices to track where a bus is located. Thus, it crowdsources arrival predictions and occupancy level of buses from its users. Tiramisu transit has very limited issue reporting categorization as shown in Figure 9 and Figure 10.



11:29

Send a Note

Send a new note for

Schedule and Predictions

Tiramisu Application

Transit Service

Figure 9 - Issue categorization with Tiramisu Transit

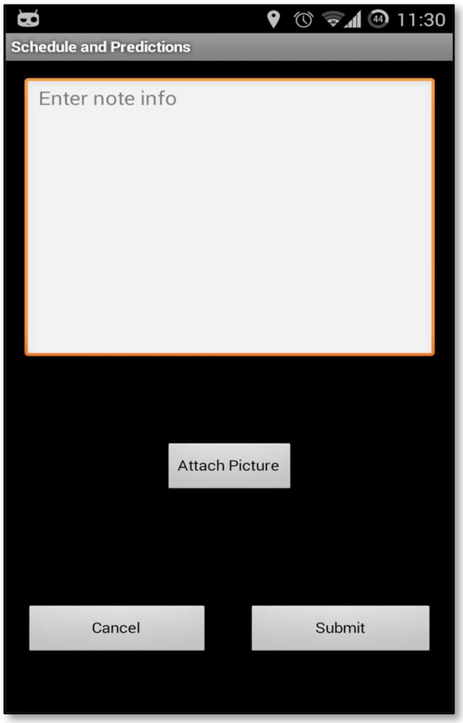


Figure 10 - Issue submission with Tiramisu Transit

The following is the full category list of the Tiramisu Transit:

- Schedule and Predictions
- Tiramisu Application
- Transit Service
  - Route problem
  - Vehicle problem
  - Driver problem
  - Bus stop problem

Tiramisu Transit does not have API that allows external applications to report issues to the system, and therefore cannot be easily integrated with applications such as OneBusAway.

2.2 Open311 Standard

During the review of other systems, the research team found that several major civic issue management products (e.g., SeeClickFix [16], Connected Bits [18], Public Stuff [17]) supported the Open311 standard [21]. Open311 is an open API specification for reporting non-emergency issues deployed in over 35 cities around the world. When an issue management product supports the Open311 protocol, it means that third-party applications can connect to the product, discover available issue categories that have been entered by local agencies, and then submit issues from users to that product. In addition, if the third-party application supports the Open311 protocol, then it can connect to any product that supports the Open311 standard.

Because of the above, Open311 was selected by the research team as the issue reporting protocol for OneBusAway. To the research team's knowledge, this is the first time the Open311 format has been used for issue reporting for public transportation. Open311 support in OneBusAway allows a region to choose the issue reporting tool/service that they want to use, and if that tool/service supports the Open311 format, OneBusAway and the tool should work together.

Given that SeeClickFix (which is Open311-compatible) was already in use by Pinellas County, FL and the City of St. Petersburg, and that OneBusAway was deployed in Tampa with HART, the greater Tampa Bay area was an excellent model region for a pilot of enhanced issue reporting within OneBusAway. FDOT District 7 also participated in the project to give a perspective on intermodal issue management and streamline issue-related communications with HART and PSTA. Appendix A discusses the effort to add Pinellas County Transit Authority (PSTA) to the OneBusAway Tampa region in detail, including analysis and troubleshooting PSTA's transit data in the General Transit Feed Specification (GTFS) and GTFS-realtime formats as the data exchange format between PSTA's Automatic Vehicle Location (AVL) system and OneBusAway.

The following section discusses the addition of Open311 support to the OneBusAway system architecture and mobile app.

Chapter 3 - Improving the Issue Reporting Experience in OneBusAway

Because Open311 has been broadly adopted across major civic issue management products, the research team worked on adding support for Open311 in the OneBusAway mobile app. The following sections discuss the system architecture for OneBusAway and the addition of Open311 support, the improved issue reporting interface within the OneBusAway app, and the platform used to view issue reports within the transit agency.

3.1 System architecture

OneBusAway uses a regional model, where the mobile applications contact a centralized Regions API to discover the available local OneBusAway deployments in each city (Figure 1). The Regions API contains information for each region, including the region name (e.g., "Tampa"), region OneBusAway server URL (e.g., http://api.tampa.onebusaway.org), and the geographic bounds for each region.

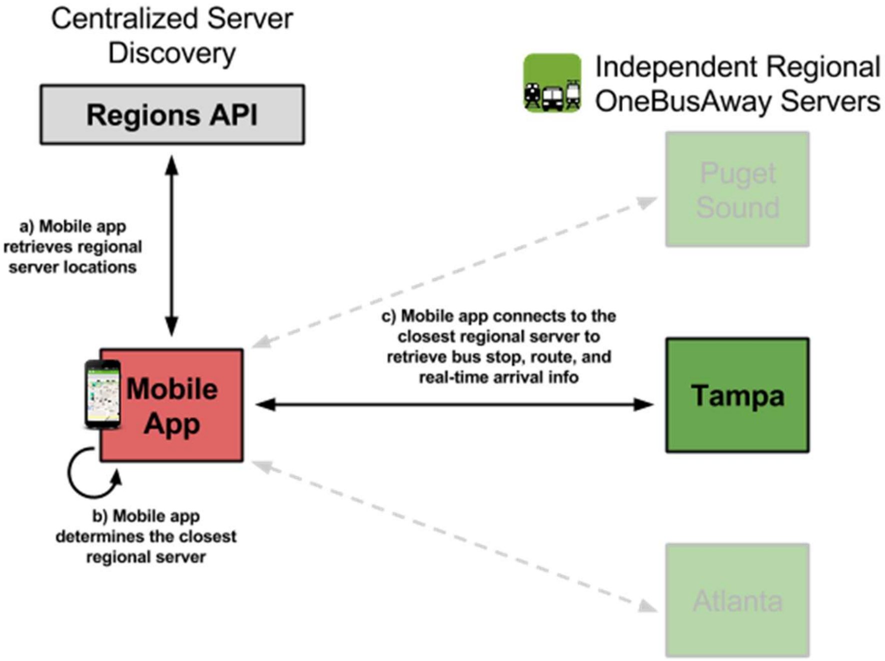
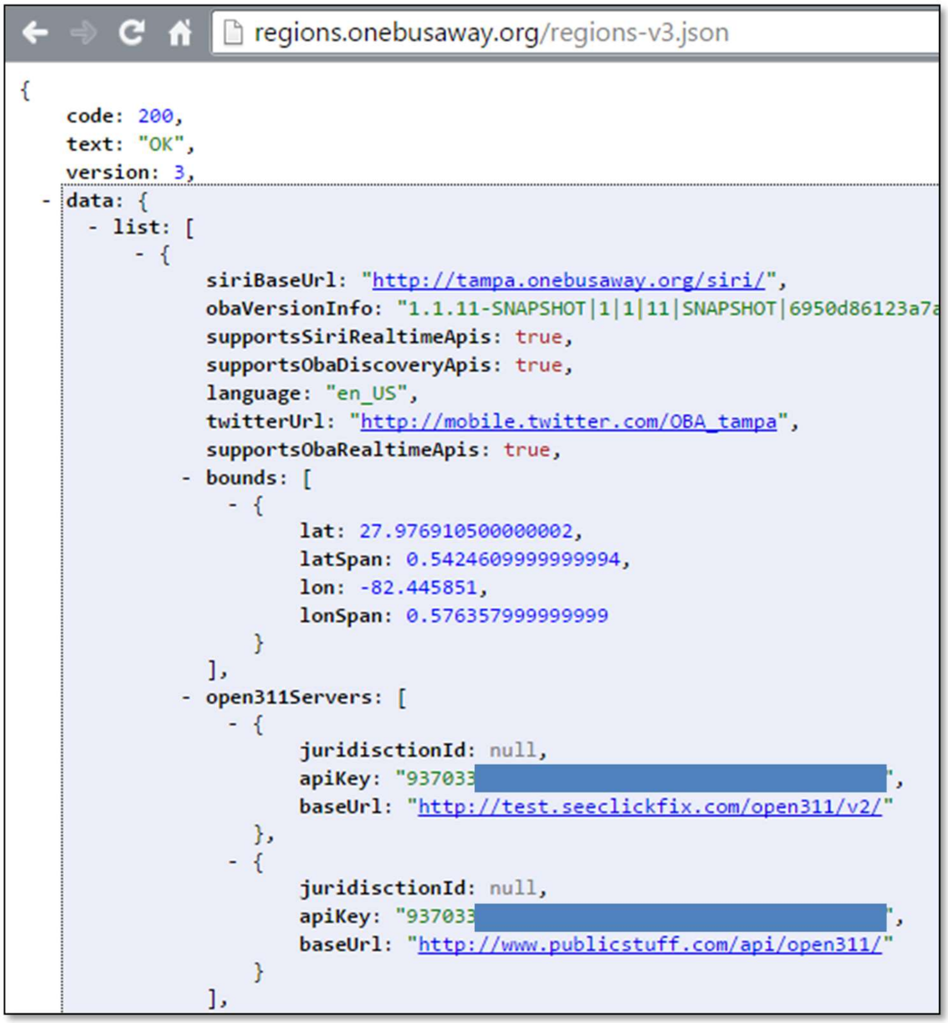

Centralized Server Discovery
Regions API
Independent Regional OneBusAway Servers
Puget Sound
a) Mobile app retrieves regional server locations
c) Mobile app connects to the closest regional server to retrieve bus stop, route, and real-time arrival info
Mobile App
Tampa
b) Mobile app determines the closest regional server
Atlanta

Figure 11 - OneBusAway uses a regional model where servers are maintained by each city

To add support for enhanced issue reporting, a set of new fields for each region was created in the OneBusAway Server Directory to allow the designation of Open311-compliant servers for each region.

The new fields in the Regions API for each region are:

- **Open311_Base_URLs** – A pipe (|) separated list of Open311 endpoints in the region (e.g., http://api.seeclickfix.com/|http://api.myserver.com/). If no Open311 server is available for a region, this field should be left blank.
- **Open311_API_Keys** - A pipe (|) separated list of API keys for the Open311 endpoints in the region. If no Open311 server is available, this field should be left blank.
- **Open311_Jurisdiction_ID** - The default jurisdiction ID for the Open311 endpoint in the region (e.g., tampa-fl). No spaces should be included. Note that OBA can currently only support one jurisdiction per region. Therefore, if this field is populated, there should only be one value in the Open311_Base_URLs and Open311_API_Keys fields for this region. If no Open311 server is available, this field should be left blank.



```
{
    code: 200,
    text: "OK",
    version: 3,
  - data: {
      - list: [
          - {
                siriBaseUrl: "http://tampa.onebusaway.org/siri/",
                obaVersionInfo: "1.1.11-SNAPSHOT|1|1|11|SNAPSHOT|6950d86123a7a
                supportsSiriRealtimeApis: true,
                supportsObaDiscoveryApis: true,
                language: "en_US",
                twitterUrl: "http://mobile.twitter.com/OBA_tampa",
                supportsObaRealtimeApis: true,
              - bounds: [
                  - {
                        lat: 27.976910500000002,
                        latSpan: 0.5424609999999994,
                        lon: -82.445851,
                        lonSpan: 0.576357999999999
                    }
                ],
              - open311Servers: [
                  - {
                        juridisctionId: null,
                        apiKey: "937033[redacted]",
                        baseUrl: "http://test.seeclickfix.com/open311/v2/"
                    },
                  - {
                        juridisctionId: null,
                        apiKey: "937033[redacted]",
                        baseUrl: "http://www.publicstuff.com/api/open311/"
                    }
                ],
```

Figure 12 - The Regions API response now includes the Open311 server information for each region

This information will allow the mobile apps to discover the URLs of the local Open311-compliant issue reporting servers so the Open311 client software knows where to submit issue report data.

For regions without Open311 servers, the same issue reporting process that is currently supported within the mobile apps (i.e., directly sending feedback to the OneBusAway servers or via a single regional email contact point) is used. If the region provides an Open311 server, then the enhanced issue reporting processes created as part of this project are made available to that region. Therefore, the results of this research project are instantly available to any OneBusAway region that wants to implement an Open311-based issue management system.

The project team reviewed the Open311 GeoReport v2 specification [21] and designed and implemented the software necessary to submit issue reports to an Open311-compliant server. This software has been implemented as a library [22] so it could be used in any application, including OneBusAway. The library can also be retrieved from the Jcenter repository[1], and can be added to any application using the Maven or Gradle build systems. A diagram showing the protocol used to communicate between an application using the Open311 client library and an Open311 server is shown in Figure 13.

[1] https://bintray.com/cutr-at-usf/cutr-mvn-repo/open311client/view

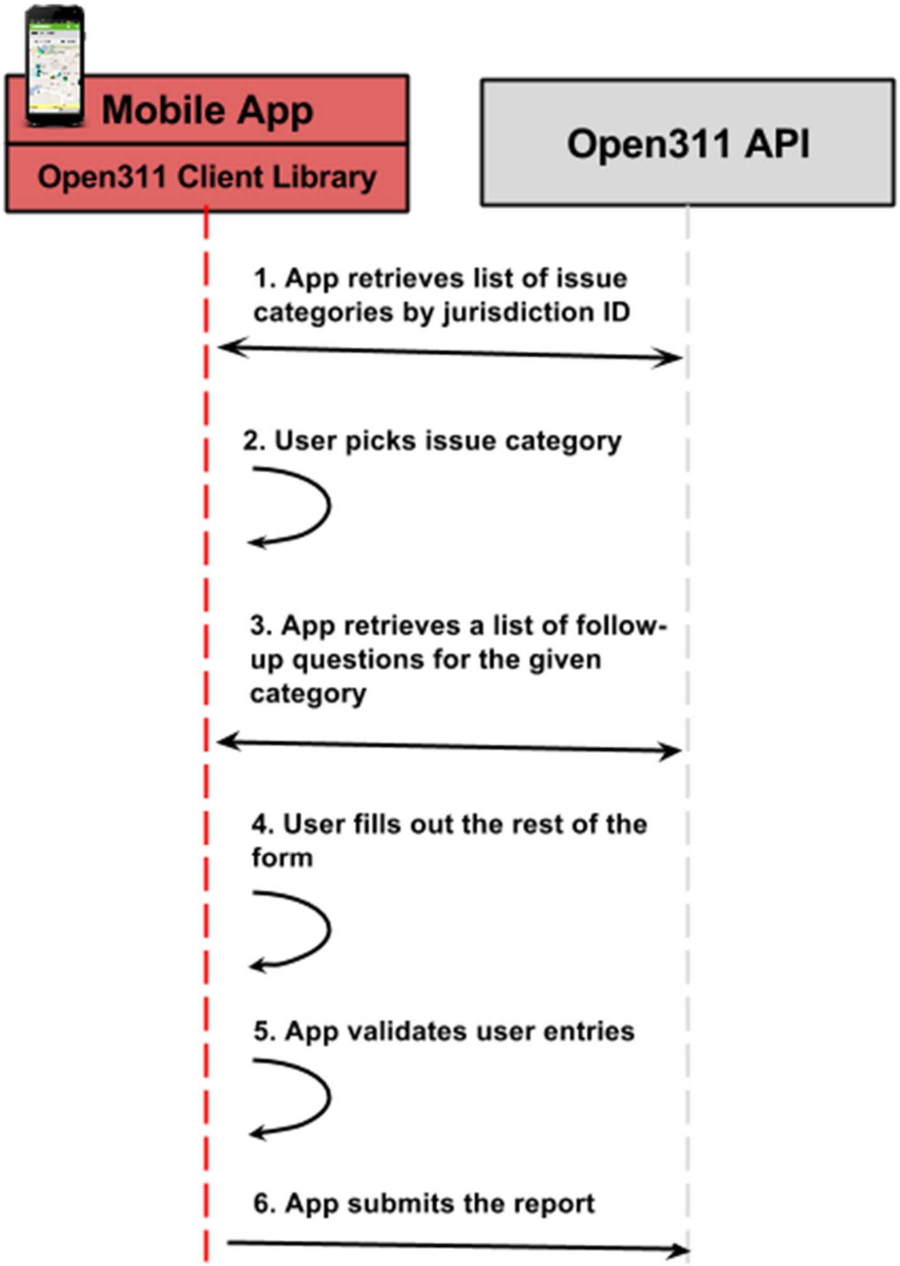
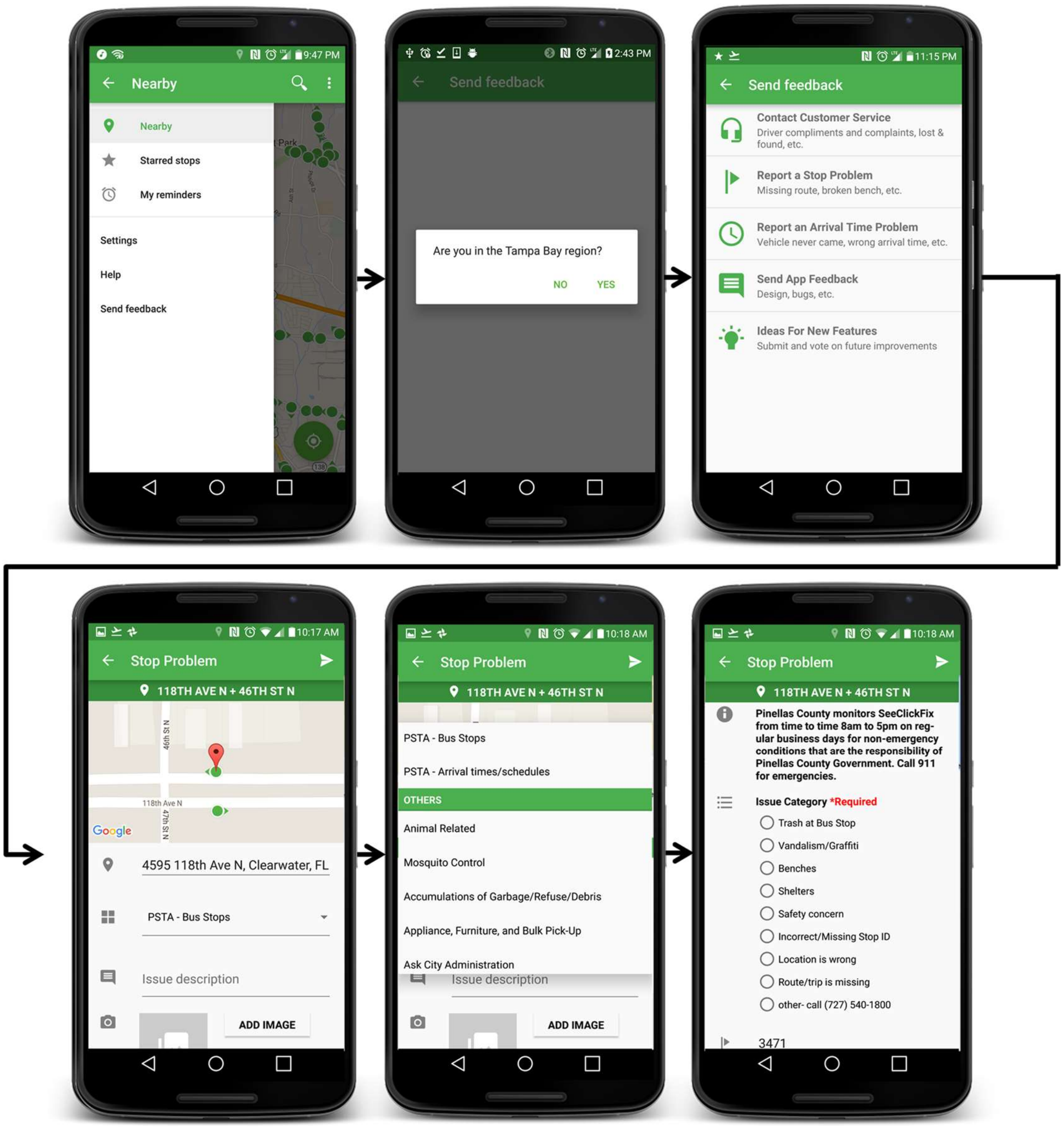

Mobile App
Open311 Client Library
Open311 API
1. App retrieves list of issue categories by jurisdiction ID
2. User picks issue category
3. App retrieves a list of follow-up questions for the given category
4. User fills out the rest of the form
5. App validates user entries
6. App submits the report

Figure 13 - The protocol for the communication between an app using the Open311 Client Library and an Open311 API Server

3.2 User Interface

A new issue reporting user interface for OneBusAway was also designed and implemented in the OneBusAway Android app (Figure 14). The source code created by the project team for both the new feedback user interface and the integration of the Open311 Client library is available on Github [22]. As discussed shortly, this design evolved throughout the project as the research team received feedback from agencies – the final design is shown in Figure 14.



Are you in the Tampa Bay region?

4595 118th Ave N, Clearwater, FL

PSTA - Bus Stops

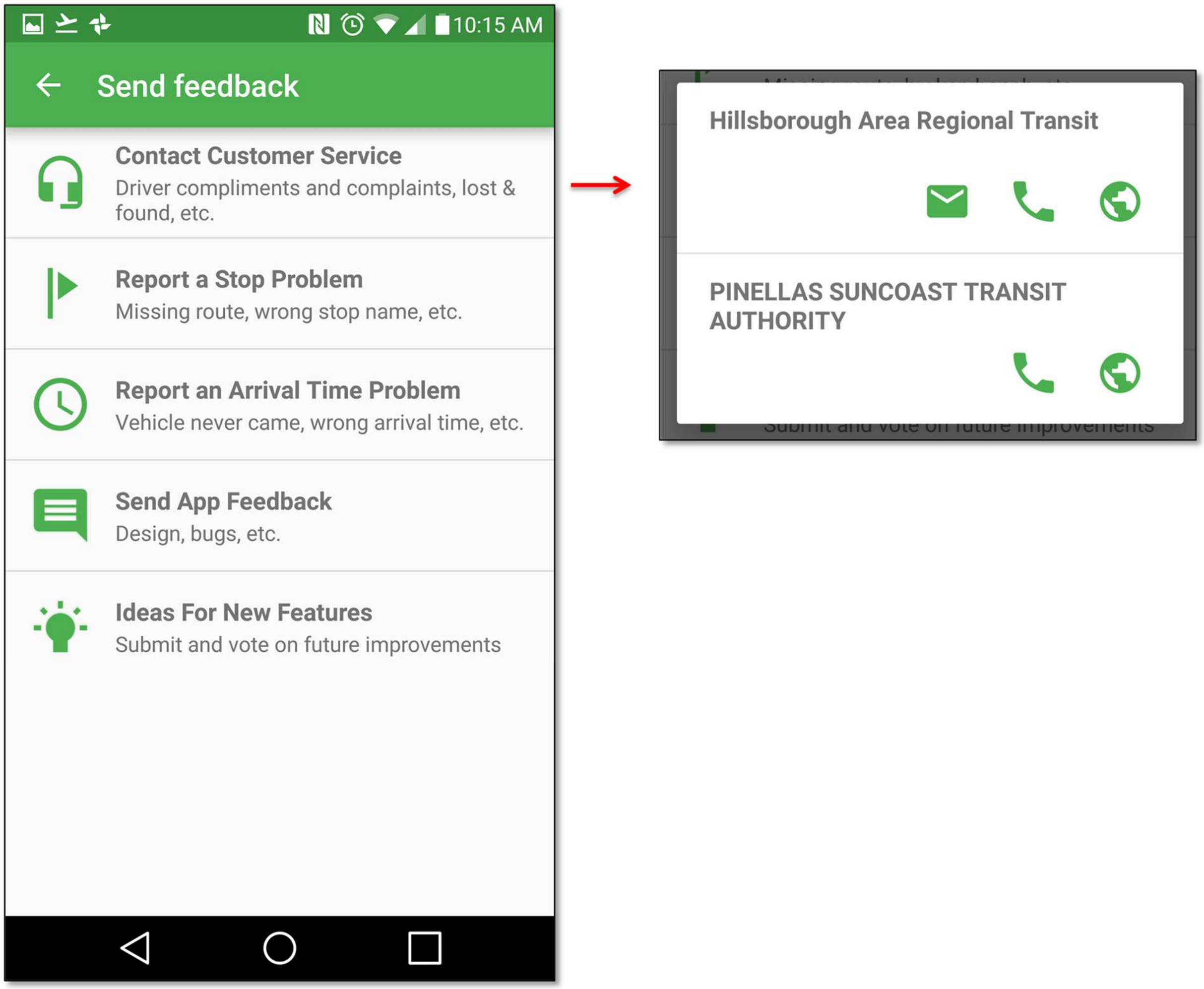
Pinellas County monitors SeeClickFix from time to time 8am to 5pm on regular business days for non-emergency conditions that are the responsibility of Pinellas County Government. Call 911 for emergencies.

3471

Figure 14 - The new issue reporting process for the OneBusAway mobile app, including directing issues to Open311-compliant servers

After tapping on the new “Send feedback” link in the main navigation drawer (Figure 14 - first screen) for the first time, the user is prompted to confirm that they are connected to the proper regional OneBusAway server (Figure 14 - second screen). If the user indicates they are not in the region the app has auto-selected (e.g., if there is a GPS error with their device), they are prompted to enter the correct region. This screen helps users resolve location errors from their device without needing to contact customer service.

Based on discussions with participating agencies, early beta users of the feedback system, and the OneBusAway community, the organization of the general feedback categories visible to the user within the app (Figure 14 - third screen, as well as Figure 15 left) evolved to facilitate the flow of issues to the correct contact point at the agency. The original set of feedback categories did not include “Contact Customer Service” or “Ideas for New Features.”



Send feedback

Contact Customer Service
Driver compliments and complaints, lost & found, etc.

Report a Stop Problem
Missing route, wrong stop name, etc.

Report an Arrival Time Problem
Vehicle never came, wrong arrival time, etc.

Send App Feedback
Design, bugs, etc.

Ideas For New Features
Submit and vote on future improvements

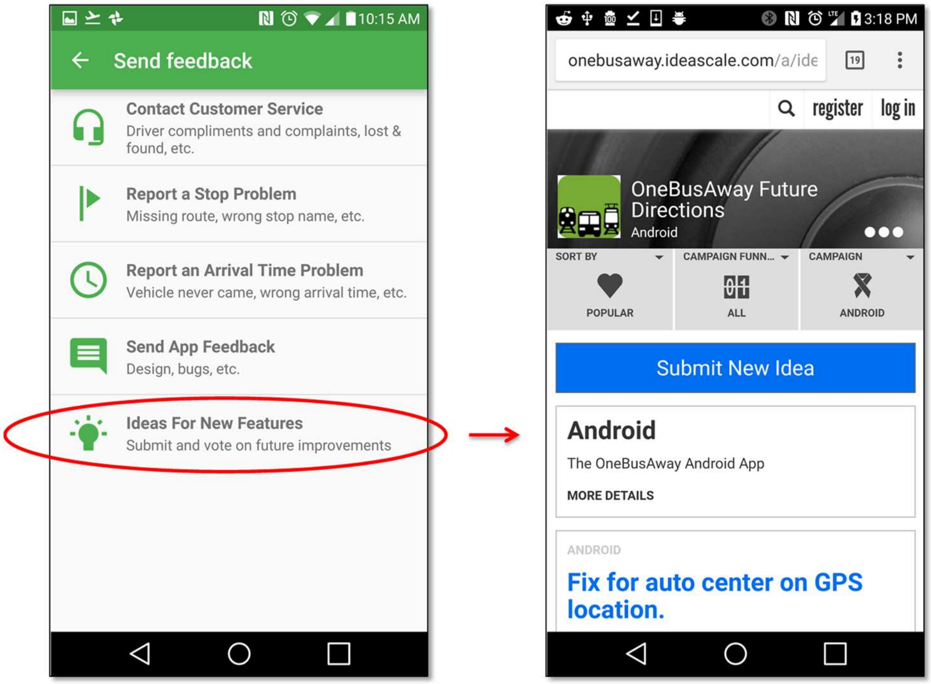
Hillsborough Area Regional Transit

PINELLAS SUNCOAST TRANSIT AUTHORITY

Figure 15 - A new "Contact Customer Service" option allows users to reach out to each regional agency directly via phone, email, or website

The “Contact Customer Service” category was added to capture issues that should be directed straight to the agency’s customer service department, which are not typically handled through an geographic-based issue reporting system via Open311 (e.g., SeeClickFix). Examples include bus driver compliments and complaints, and lost and found requests that don’t have a specific geographic component (the user doesn’t know where they left something). Figure 15 shows an example of how each agency can define contact options specific to their customer service departments. In this example, HART has allowed users to contact them via phone, email, or website, while PSTA has chosen to only allow contact via phone or website (not email).

The control of contact point options is provided directly to the transit agency via their General Transit Feed Specification (GTFS) data, which is used by OneBusAway to represent transit stops, schedules, and routes for each agency in the mobile apps. One file in the GTFS dataset, agency.txt, currently allows the agency to specify a agency_phone and agency_website for phone and website contact points, respectively, which is the data that OneBusAway then uses to direct their feedback in Figure 15. The research team successfully proposed the addition of the agency_email field to the official GTFS format via the GTFS community process in order to support feedback to agencies via email – the proposal can be viewed online[2].



10:15 AM

Send feedback

Contact Customer Service
Driver compliments and complaints, lost & found, etc.

Report a Stop Problem
Missing route, wrong stop name, etc.

Report an Arrival Time Problem
Vehicle never came, wrong arrival time, etc.

Send App Feedback
Design, bugs, etc.

Ideas For New Features
Submit and vote on future improvements

3:18 PM

onebusaway.ideascale.com/a/ide

register log in

OneBusAway Future Directions
Android

SORT BY | CAMPAIGN FUNN... | CAMPAIGN
POPULAR | ALL | ANDROID

Submit New Idea

Android
The OneBusAway Android App
MORE DETAILS

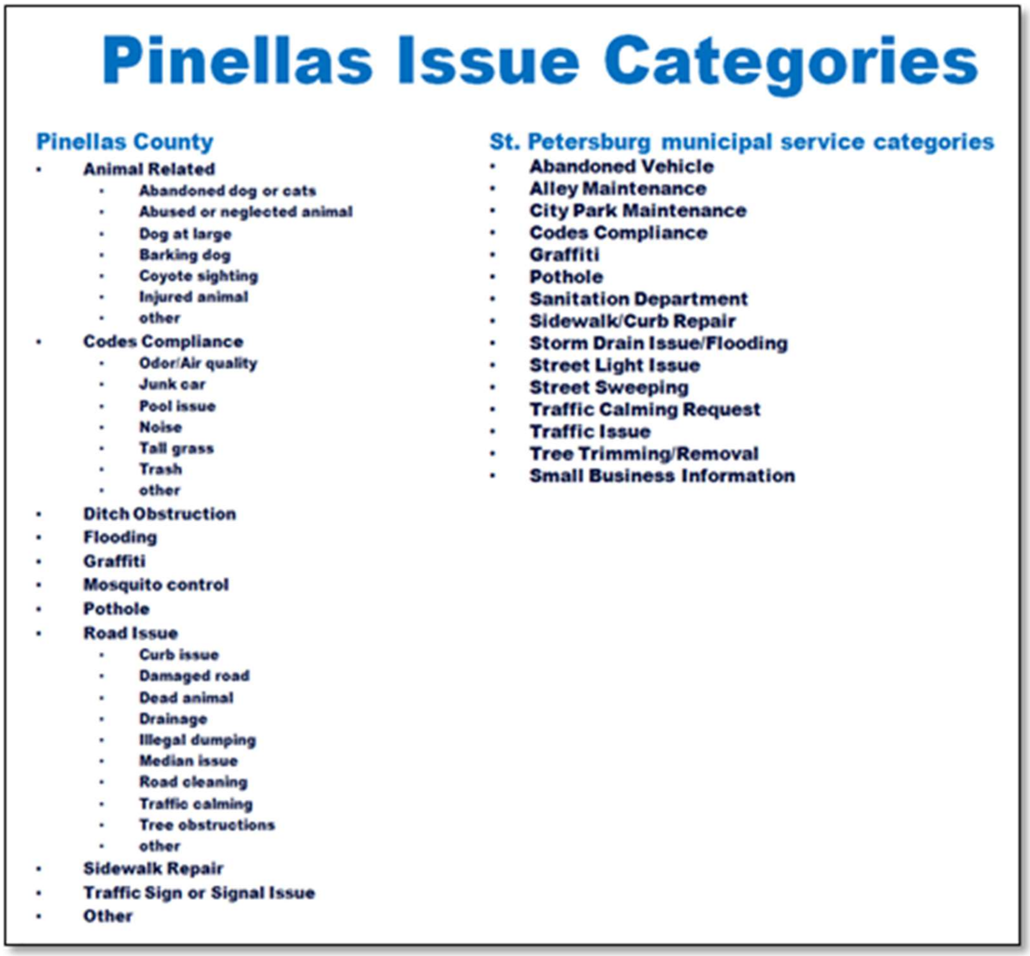
ANDROID
Fix for auto center on GPS location.

Figure 16 - Users can submit and vote on ideas for new features at the OneBusAway IdeaScale site

[2] https://groups.google.com/forum/#!topic/gtfs-changes/aezjQsriLYA

The “Ideas For New Features” category (Figure 16) was added to direct community-generated ideas directly to the main OneBusAway project IdeaScale site (http://onebusaway.ideascale.com/). The OneBusAway IdeaScale site allows users from all regions to submit and vote on ideas for new features in the OneBusAway open-source project, which will help transit agencies from around the world to prioritize and collaboratively fund enhancements to the OneBusAway open-source project. Agencies participating in the OneBusAway project are already showing these cross-regional benefits to agencies, with improvements from NYC being deployed in Tampa, and improvements from Tampa being deployed in Atlanta and Seattle.

The fourth screen in Figure 14 shows the available categories for which an issue can be reported for the given geographic location in the app. One advantage of the Open311 standard is that it allows each local jurisdiction to define the issue categories they want to share with the public. The Open311 client can retrieve a list of these issues from the local jurisdiction, and display these issues to the user. The issue categories used by Pinellas County and the City of St. Petersburg in SeeClickFix are shown in Figure 17.



Pinellas Issue Categories

Pinellas County

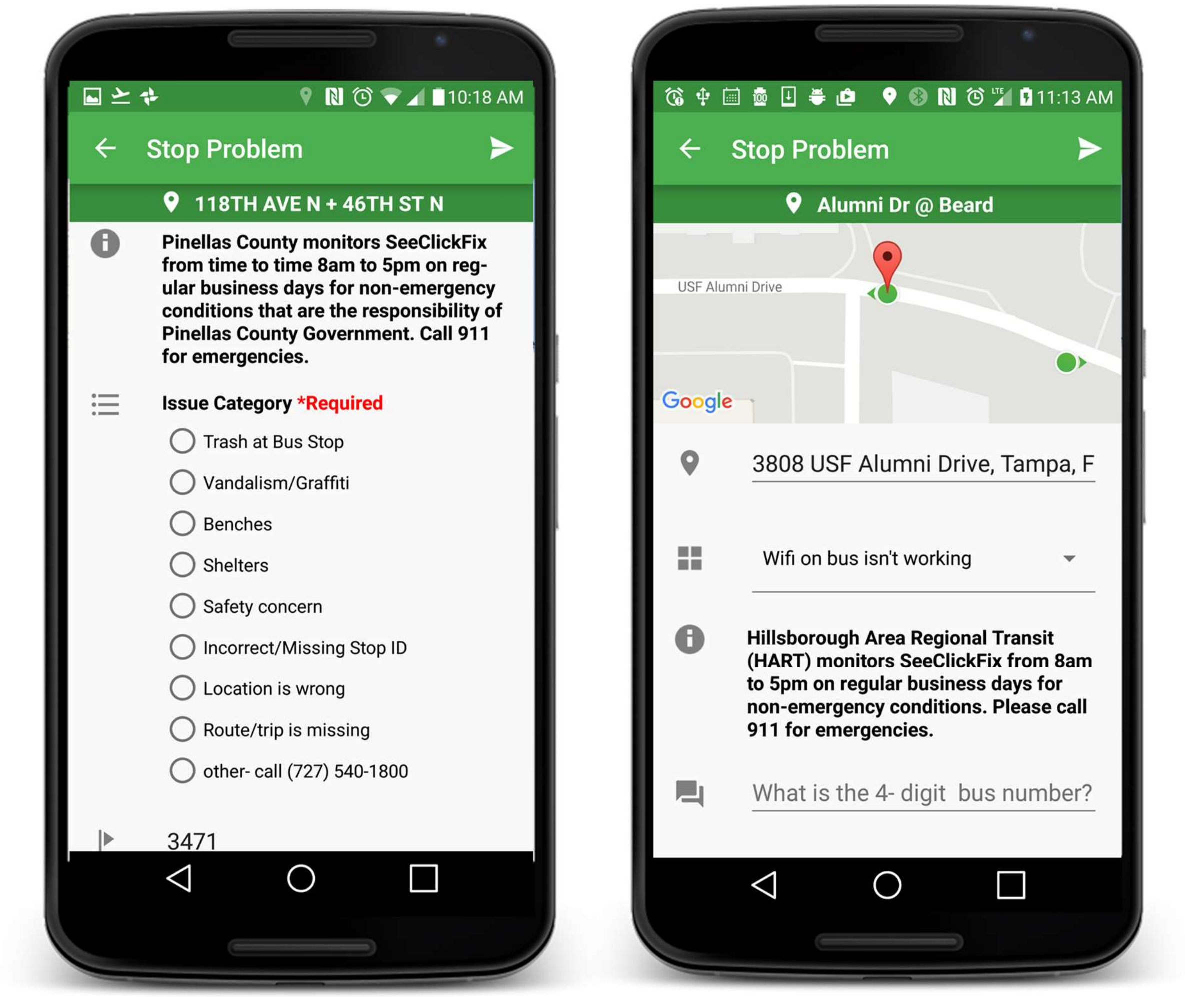
- Animal Related
  - Abandoned dog or cats
  - Abused or neglected animal
  - Dog at large
  - Barking dog
  - Coyote sighting
  - Injured animal
  - other
- Codes Compliance
  - Odor/Air quality
  - Junk car
  - Pool issue
  - Noise
  - Tall grass
  - Trash
  - other
- Ditch Obstruction
- Flooding
- Graffiti
- Mosquito control
- Pothole
- Road Issue
  - Curb issue
  - Damaged road
  - Dead animal
  - Drainage
  - Illegal dumping
  - Median issue
  - Road cleaning
  - Traffic calming
  - Tree obstructions
  - other
- Sidewalk Repair
- Traffic Sign or Signal Issue
- Other

St. Petersburg municipal service categories

- Abandoned Vehicle
- Alley Maintenance
- City Park Maintenance
- Codes Compliance
- Graffiti
- Pothole
- Sanitation Department
- Sidewalk/Curb Repair
- Storm Drain Issue/Flooding
- Street Light Issue
- Street Sweeping
- Traffic Calming Request
- Traffic Issue
- Tree Trimming/Removal
- Small Business Information

Figure 17 - Each jurisdiction can define issue categories specific to their area

The issue categories used by Pinellas County and the City of St. Petersburg in SeeClickFix included a variety of non-transit categories, including animal-related (barking dog), code compliance (noise, trash), ditch obstruction, flooding, graffiti, and mosquito control. PSTA added the categories of “Bus Stops” and “Arrival times” to this list, and HART defined many issue categories specific to their system, including “Arrival Times”, “Benches”, “Environmental Issues”, “Safety Concern”, “Shelters”, and more. Additionally, the jurisdiction can define follow-up questions for each category that the user must answer before submitting the issue report, which are also retrievable by OneBusAway via the Open311 API. This enables the jurisdiction to ensure they have all information needed for the report to be actionable. The follow up questions to a “Bus Stop” category defined by PSTA in Pinellas County’s SeeClickFix account are shown in the OneBusAway app on the left in Figure 18, while a HART-specific category and follow-up question for Wi-Fi on buses is shown on the right.



Pinellas County monitors SeeClickFix from time to time 8am to 5pm on regular business days for non-emergency conditions that are the responsibility of Pinellas County Government. Call 911 for emergencies.

Issue Category *Required

○ Trash at Bus Stop
○ Vandalism/Graffiti
○ Benches
○ Shelters
○ Safety concern
○ Incorrect/Missing Stop ID
○ Location is wrong
○ Route/trip is missing
○ other- call (727) 540-1800

3471

3808 USF Alumni Drive, Tampa, F

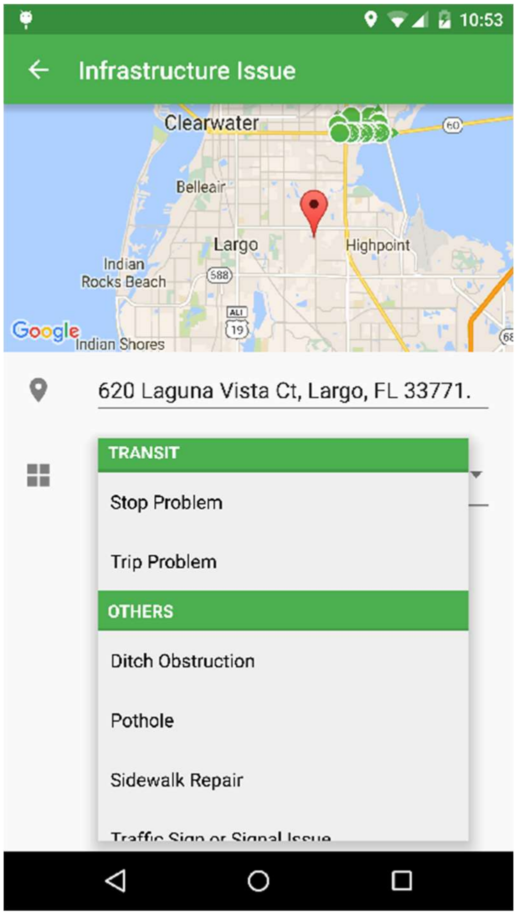
Wifi on bus isn't working

Hillsborough Area Regional Transit (HART) monitors SeeClickFix from 8am to 5pm on regular business days for non-emergency conditions. Please call 911 for emergencies.

What is the 4- digit bus number?

Figure 18 – Each jurisdictions/agency (PSTA on left, HART on right) defines their own issue categories and follow-up questions, which are automatically shown in the OneBusAway app

Following discussions with participating agencies, it was determined that the OneBusAway application should prioritize transit-related issue categories when they are displayed to the user, if possible. The research team implemented software to use the server-side “keywords” and “categories” Open311 elements to label certain issues as “transit-related”. When defined server-side by an agency, these categories show up at the top of the list of issue categories when presented within the OneBusAway app, as shown in Figure 19.



620 Laguna Vista Ct, Largo, FL 33771.

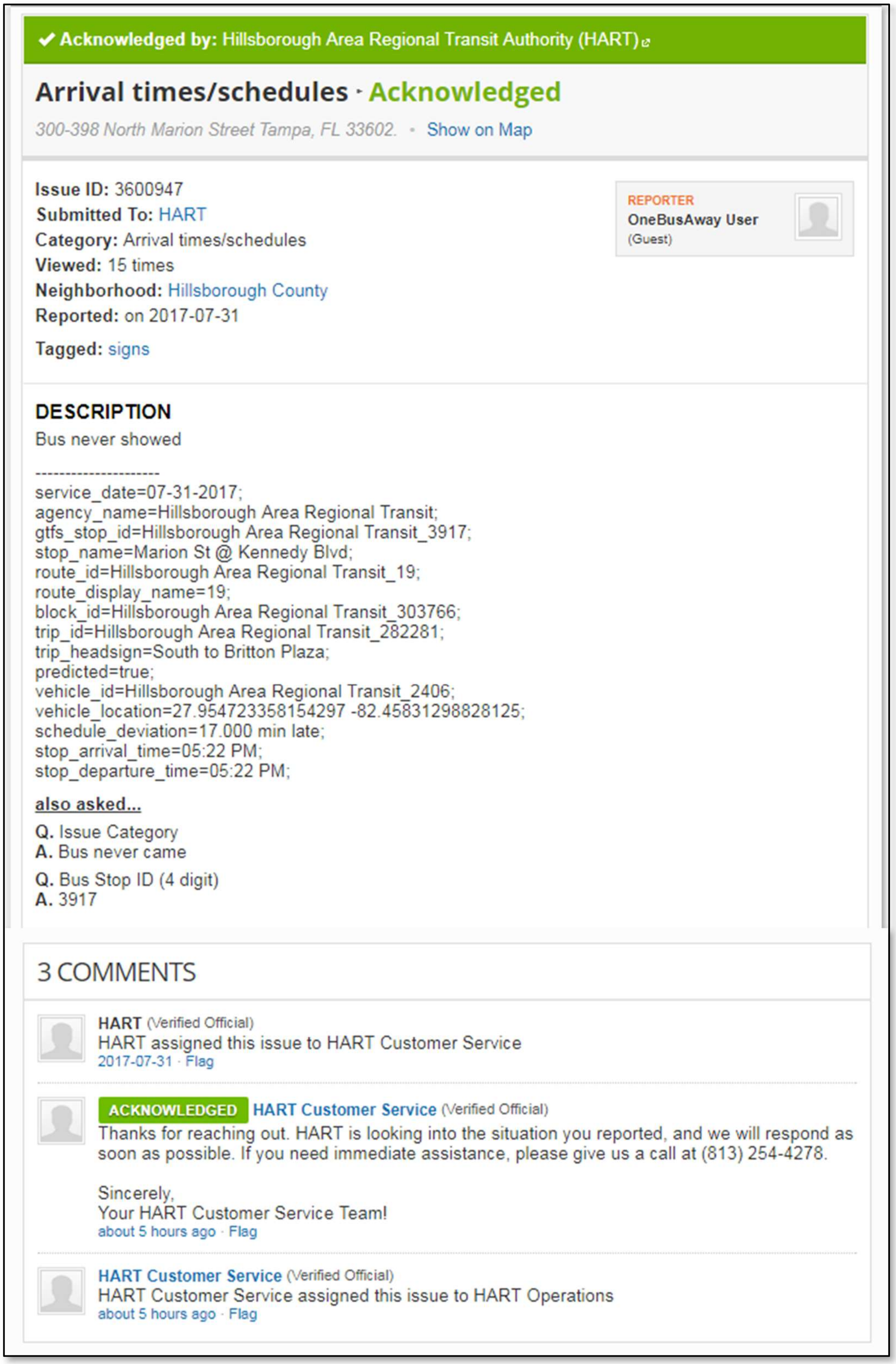
Figure 19 - Transit-related issue report categories now appear at the top of the list

The source code for the implementation of the enhanced client/server communication, including the new Open311 server discovery protocol, image submissions, and transit keyword/category labeling is available on Github[3].

An example issue reported by a OneBusAway user via the Open311 protocol to the SeeClickFix system is shown in Figure 20. In this example, a user is saying the app showed a predicted arrival for a bus, but no bus arrived (i.e., a “ghost” bus). When the user taps on the arrival time in the app to report a problem for that arrival, OneBusAway

[3] https://github.com/OneBusAway/onebusaway-android/pull/392

automatically captures a large amount of metadata for that vehicle (e.g., user location based on GPS and/or user tapping on map, stop_id, trip_id, route_id, block_id, trip_headsign, predicted and scheduled arrival and departure times) and attaches this to the issue report. This information, visible in the "Description" section of Figure 20, enables the transit agency to quickly triage the issue report and respond with preset answers. In this case, HART Customer Service has assigned the issue to HART Operations so that Operations can evaluate if a problem with the automatic vehicle location (AVL) system is causing these "ghost" buses. After Operations reviews the issue and takes action, HART can close the issue and provide a resolution to the user. Issue reporting products such as SeeClickFix tracks these actions and can provide detailed reports to the agency so they can track how long it takes various departments within the agency to acknowledge, assign, and close out issues.



✔ Acknowledged by: Hillsborough Area Regional Transit Authority (HART)

Arrival times/schedules · Acknowledged

300-398 North Marion Street Tampa, FL 33602. • Show on Map

Issue ID: 3600947
Submitted To: HART
Category: Arrival times/schedules
Viewed: 15 times
Neighborhood: Hillsborough County
Reported: on 2017-07-31

Tagged: signs

REPORTER
OneBusAway User
(Guest)

DESCRIPTION
Bus never showed

```
---------------------
service_date=07-31-2017;
agency_name=Hillsborough Area Regional Transit;
gtfs_stop_id=Hillsborough Area Regional Transit_3917;
stop_name=Marion St @ Kennedy Blvd;
route_id=Hillsborough Area Regional Transit_19;
route_display_name=19;
block_id=Hillsborough Area Regional Transit_303766;
trip_id=Hillsborough Area Regional Transit_282281;
trip_headsign=South to Britton Plaza;
predicted=true;
vehicle_id=Hillsborough Area Regional Transit_2406;
vehicle_location=27.954723358154297 -82.45831298828125;
schedule_deviation=17.000 min late;
stop_arrival_time=05:22 PM;
stop_departure_time=05:22 PM;
```

also asked...

Q. Issue Category
A. Bus never came

Q. Bus Stop ID (4 digit)
A. 3917

3 COMMENTS

HART (Verified Official)
HART assigned this issue to HART Customer Service
2017-07-31 · Flag

ACKNOWLEDGED **HART Customer Service** (Verified Official)
Thanks for reaching out. HART is looking into the situation you reported, and we will respond as soon as possible. If you need immediate assistance, please give us a call at (813) 254-4278.

Sincerely,
Your HART Customer Service Team!
about 5 hours ago · Flag

HART Customer Service (Verified Official)
HART Customer Service assigned this issue to HART Operations
about 5 hours ago · Flag

Figure 20 – Metadata for the vehicle, including predicted and scheduled arrival and departure times, is automatically attached to issue report

More details and screenshots of the improvements to the issue reporting process within OneBusAway based on agency feedback is detailed in a presentation that has been shared with all OneBusAway regional agencies, available online[4].

3.3 Analytics

To help agencies better understand how transit users are using the OneBusAway application, the research team implemented software to collect analytics data in the OneBusAway mobile application via Google Analytics, which is free to use. These data show which screens of the mobile app are being used most frequently, as well as which bus stops are being viewed most frequently and approximately how far the user is from the bus stop when they view the information. All data is reported anonymously and does not include any user-identifiable data. On March 30, 2016 the enhanced analytics feature was released to the general public and can be downloaded online[5]. Technical details on the analytics software implementation are available on Github[6]. The mobile app software source code with the embedded analytics source code can be downloaded from Github[7].

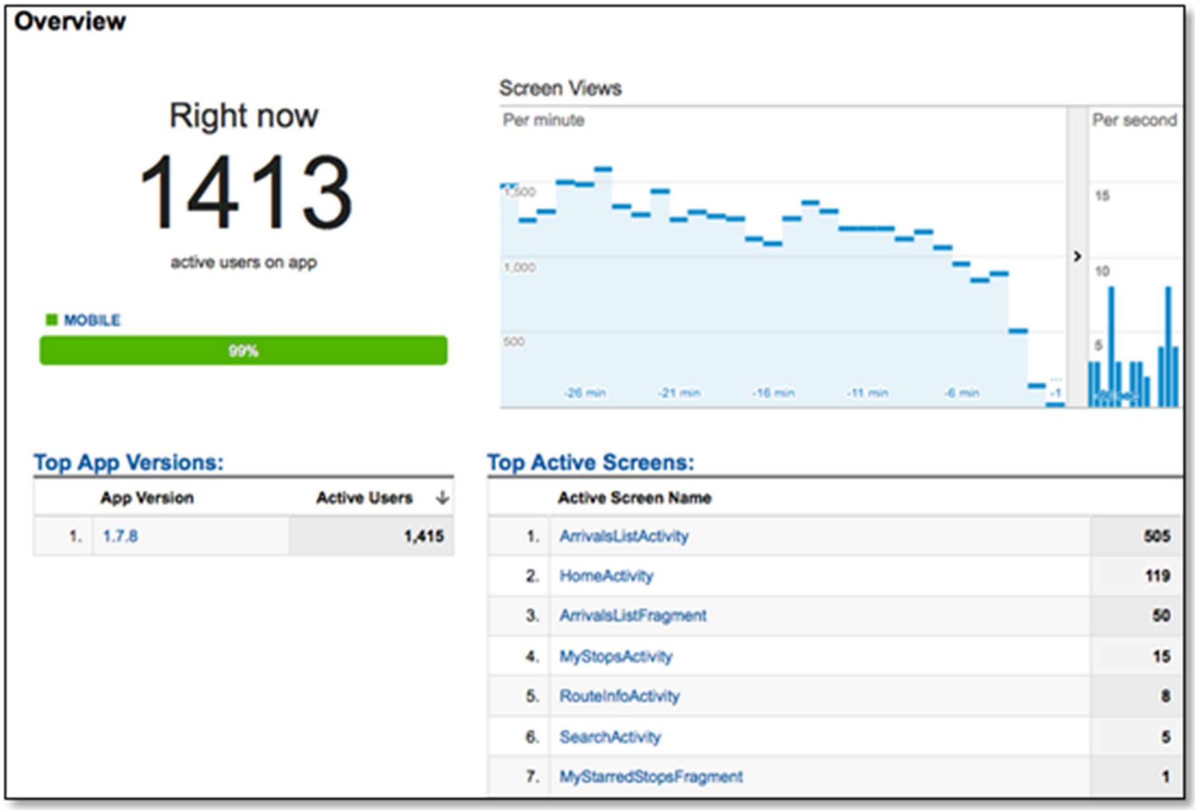
A dashboard from the analytics tool can be seen in Figure 21, which shows which type of information riders are accessing and how many users are currently using the system. Figure 22 shows an interesting metric of how far from a bus stop a transit rider is when they request information about that stop, which will help agencies better understand how rider's adjust their travel behavior based on real-time transit data.

[4] http://www.slideshare.net/sjbarbeau/onebusaway-issue-reporting-challenges-and-improvements

[5] https://play.google.com/store/apps/details?id=com.joulespersecond.seattlebusbot

[6] https://github.com/OneBusAway/onebusaway-android/issues/105

[7] https://github.com/OneBusAway/onebusaway-android



Overview

Right now

1413

active users on app

🟩 MOBILE

99%

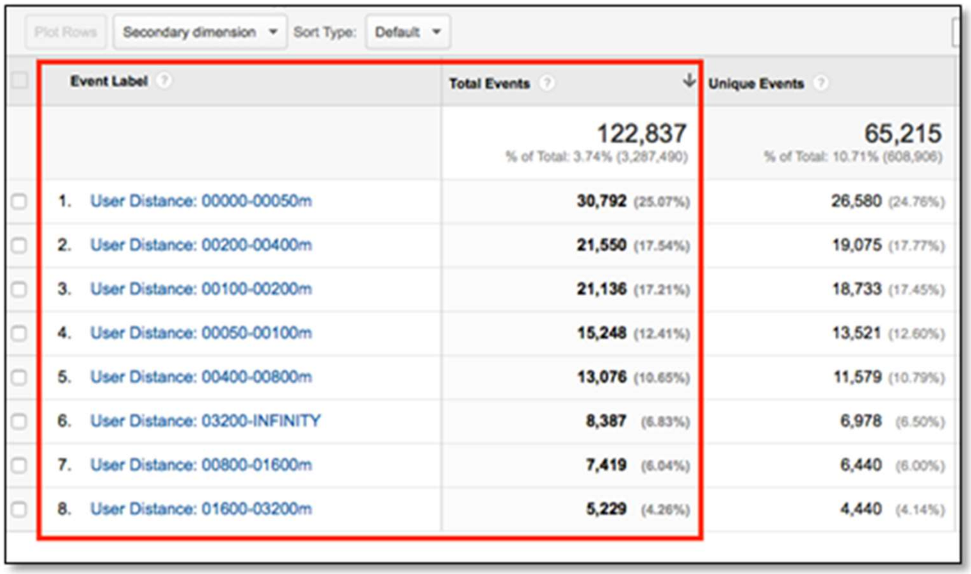
Top App Versions:

	App Version	Active Users ↓
1.	1.7.8	1,415

Top Active Screens:

	Active Screen Name	
1.	ArrivalsListActivity	505
2.	HomeActivity	119
3.	ArrivalsListFragment	50
4.	MyStopsActivity	15
5.	RouteInfoActivity	8
6.	SearchActivity	5
7.	MyStarredStopsFragment	1

Figure 21 - The analytics data being collected from the application in real-time shows how the application being used by transit riders



Plot Rows | Secondary dimension ▾ | Sort Type: | Default ▾

□	Event Label	Total Events ↓	Unique Events
		122,837 % of Total: 3.74% (3,287,490)	65,215 % of Total: 10.71% (608,906)
□	1. User Distance: 00000-00050m	30,792 (25.07%)	26,580 (24.76%)
□	2. User Distance: 00200-00400m	21,550 (17.54%)	19,075 (17.77%)
□	3. User Distance: 00100-00200m	21,136 (17.21%)	18,733 (17.45%)
□	4. User Distance: 00050-00100m	15,248 (12.41%)	13,521 (12.60%)
□	5. User Distance: 00400-00800m	13,076 (10.65%)	11,579 (10.79%)
□	6. User Distance: 03200-INFINITY	8,387 (6.83%)	6,978 (6.50%)
□	7. User Distance: 00800-01600m	7,419 (6.04%)	6,440 (6.00%)
□	8. User Distance: 01600-03200m	5,229 (4.26%)	4,440 (4.14%)

Figure 22 - Transit agencies can now see how far from the bus stop a transit rider is when accessing information about when the bus will arrive

Chapter 4 - Deployment and Lessons Learned

The enhanced version of OneBusAway with Open311 support was deployed throughout 2016 with as part of beta testing with HART and PSTA transit riders in Tampa Bay, using SeeClickFix as the Open311-compliant issue management system. HART announced an official launch of SeeClickFix as their issue reporting management tool on January 30th, 2017. The following sections discussed lessons learned from the first six months of deployment through August 1st, 2017.

4.1 Cross jurisdictional issue reports

One goal of using an Open311-compliant issue reporting platform was to enable the app to capture diverse types of issues beyond simply reporting errors in arrival times. Figure 23 shows that of the 677 issue reports in Hillsborough County, approximately 87% were related to arrival times and schedules, which means that approximately 13% were related to other categories.

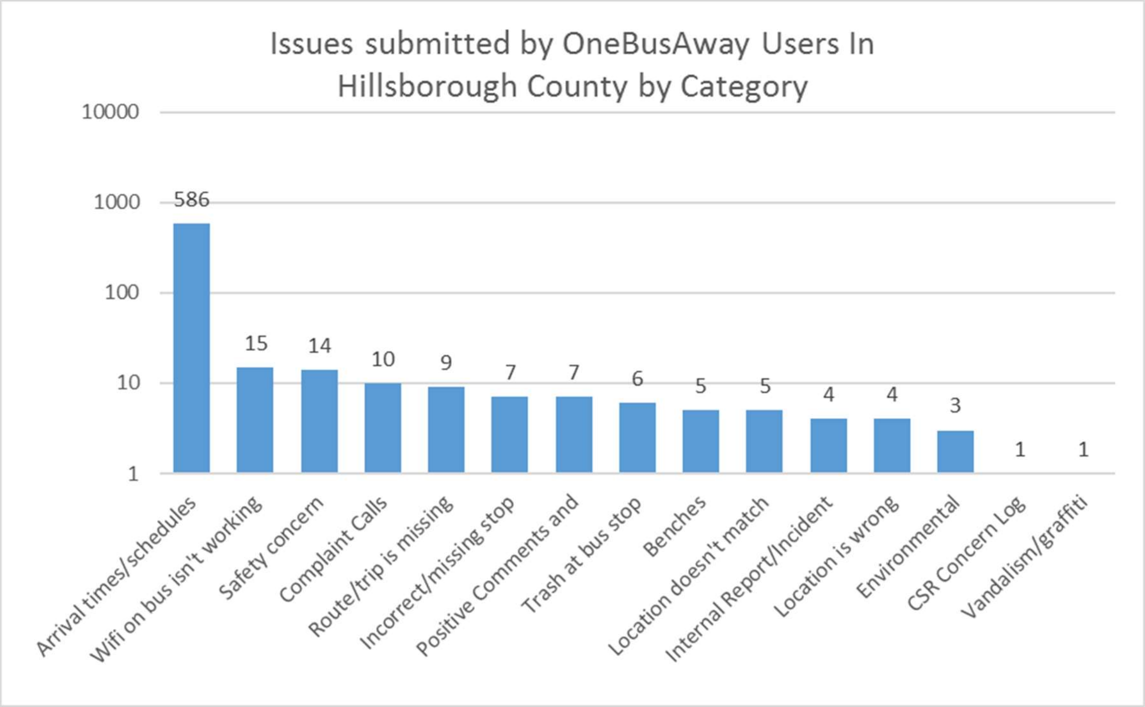


Figure 23 – Arrival times and schedules issues are bulk of reports, but other issue types are also captured

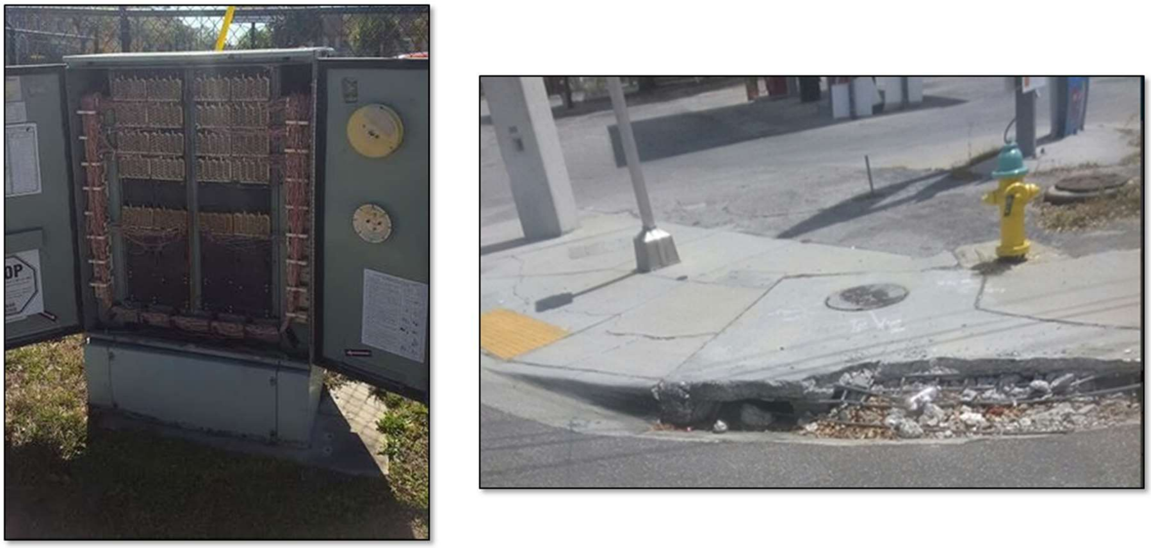


Figure 24 – Two cross-jurisdictional safety issues (left – electrical box open, right – broken storm drain) reported by OneBusAway users

Figure 24 shows photos from two different safety-related issues reported by OneBusAway users in Hillsborough County. On the left, an electrical utility box was left open near a bus stop [23]. On the right, a storm drain has been damaged with the debris extending into the pedestrian and bike travel paths [24]. These issues demonstrate the cross-jurisdictional nature of problems reported by the public. In the case of the electrical box, HART forwarded a link to this issue including the exact geographic latitude and longitude and photo to Frontier Communications, a telecommunications company, who was able to quickly resolve the problem. HART relayed the issue for the storm drain, again complete with exact latitude, longitude, and photo to the City of Tampa, who responded and repaired the drain the following week. The photos and exact geographic location help users communicate a first-person perspective to HART, who can in turn communicate it to other agencies, which facilities a rapid response for truly hazardous situations.

Users have also reported bus stops in dangerous locations via OneBusAway, again automatically tagging the location and bus stop ID for the stop for the agency's review [25]:

> *This stop is dangerous and should be moved. It is at MacDill and Granada serving Northbound Route 574 (South Tampa Flex). There is on-street parking directly in front of it. If someone is parked there bus patrons must step partially into the street to be seen by the driver, which puts them at risk of being hit by a passing motorist. Better locations for this stop would be either north of Barcelona, in front of Datz Deli, which has a parking lot, or MacDill and Santiago, under the Lee Roy Selmon Expressway and just before the right turn lane for traffic going onto northbound Bay To Bay starts. There is a bus bench and a sign stanchion at this intersection but no sign and according to the driver it is not a stop.*

In this case, the user didn't attach a picture to the report, but given the exact location a quick review of Google StreetView shows that there is indeed on-street parking in front of the bus stop (Figure 25).

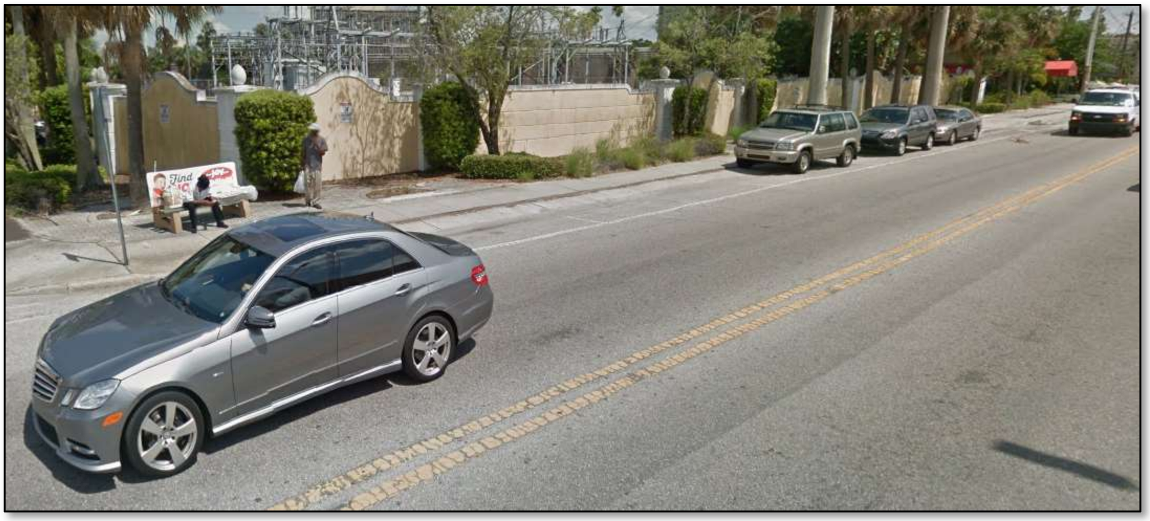


Figure 25 – Dangerous bus stop location reported by OneBusAway user

HART replied that they are scheduled an on-site review with their Service Planning and Safety and Security teams to determine if this stop needs to be moved.

HART has also seen positive feedback via the app. One user reported [26]:

> *Bus driver was kind enough to wait for everyone that got there by departure time. My husband and I are disabled so we can't run for buses anymore. This driver was patient while we walked to the stop. Just polite and kind. Bus 1510 not sure of operator number. Please thank him for his good attitude from 2 loyal patrons.*

HART has intentionally provided an issue category as "Positive comments" to encourage patrons to report good service when they receive it so they can pass positive feedback back to operators. And, even though the rider didn't know the operator number, HART could determine this from the trip metadata provided by the OneBusAway app.

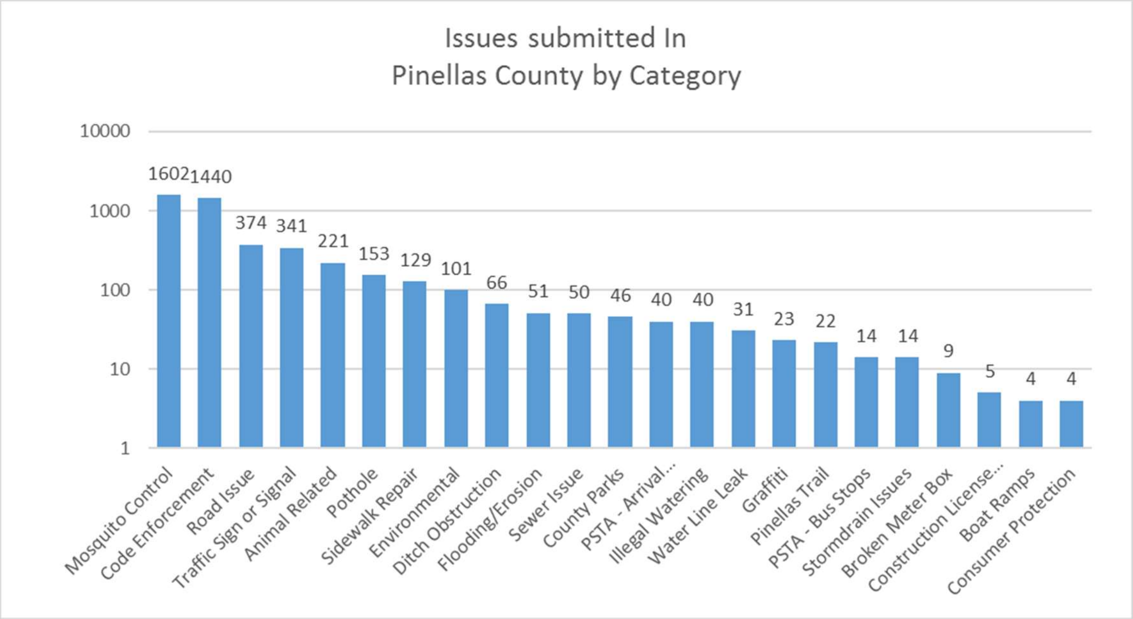


Figure 26 – Pinellas County has more issue reports, but fewer transit-specific issues

In Pinellas County, of the 4,780 issues reported only 54 (1.13%) were related to transit (Figure 26). This is because PSTA has not widely advertised the availability of their information in OneBusAway - PSTA stops and routes currently make up only 5% of total data traffic in OneBusAway Tampa Bay. Additionally, Pinellas County government has been using the SeeClickFix platform for several years for non-transit issues, which explains the magnitude of non-transit issue reports. Conversely, Hillsborough County government is not using the SeeClickFix platform for issue management so there is not an existing user base reporting non-transit issues. Of the transit-related reports, 74% (40) applied to arrival times and schedules, and 26% (14) involved bus stops.

One of the issue reports was related to a broken window at a bus shelter [27] (Figure 27):

> *Glass panel is broken & glass scattered around & inside bus stop.*

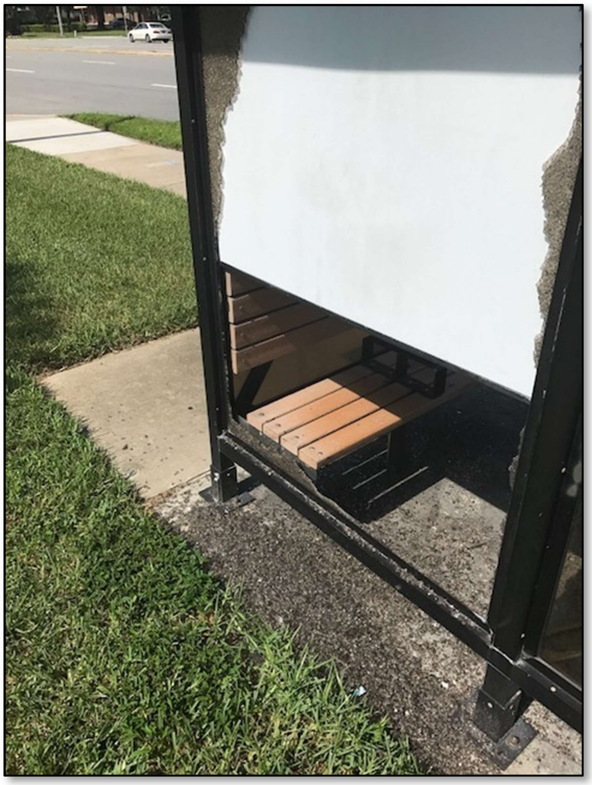


Figure 27 - Broken glass from a bus shelter window reported at a PSTA bus stop

4.2 Potential areas of improvement in the Open311 specification

During the design and deployment of the system, the research team learned that there are certain aspects of issue reporting that are not covered by the Open311 standard.

The discovery protocol of determining Open311 system coverage has emerged as an important feature for regional deployments. Initially it was thought that a single Open311 server would be used for an entire OneBusAway region. However, after further discussions, it is apparent that different agencies within the same large OneBusAway region may use different issue reporting vendors, and some areas may not be monitored by an Open311-compliant system at all. Figure 28 shows an example of a large geographic area in blue that is monitored by one vendor Open311 system (e.g., SeeClickFix), a second area monitored by a second vendor (e.g., PublicStuff), and a third area that is not monitored by any Open311 system.

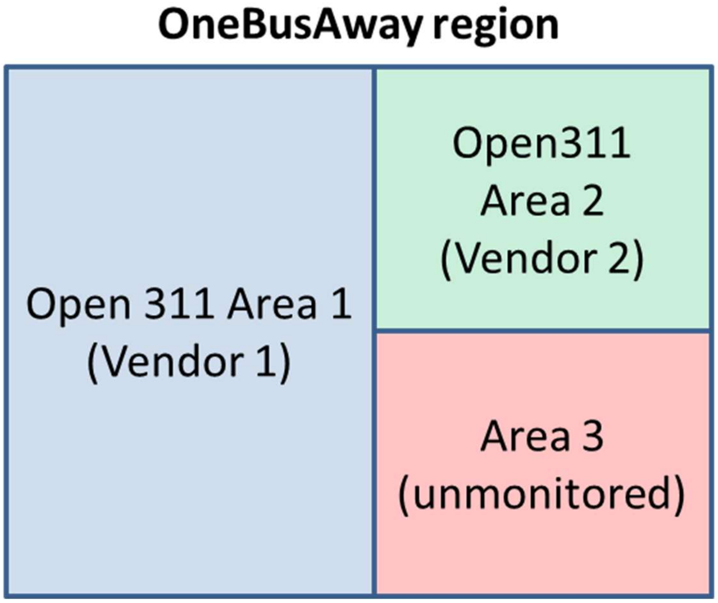


Figure 28 - A single OneBusAway region may have several geographic areas, some of which are monitored by different vendor/agency for Open311 issue reports, and others that are not

To ensure that issues for a geographic area are submitted to the correct issue reporting server, the project team designed and implemented a discovery protocol for Open311 servers within a OneBusAway region. As mentioned earlier, the OneBusAway Regions API now supports the definition of multiple Open311 server addresses. When a user reports an issue, each of the Open311 servers for the region are queried to determine which issue categories exist for that geographic location. If only one “Other” category is reported for a specific Open311 server, then that server does not monitor that area. If no Open311 servers are monitoring that area, then the issue is sent to the OneBusAway server. If an Open311 server returns more than one category, then the issue is sent to that Open311 server. If no Open311 servers exist for a region, then issues are sent to the OneBusAway server for that region (i.e., the same process that existed prior to the implementation of enhanced issue reporting capabilities).

The project team worked with the SeeClickFix engineering team to design, implement, and test software that executes this new Open311 server discovery protocol. It was successfully deployed during this project, with Pinellas County being monitored by the SeeClickFix server and Hillsborough County initially being unmonitored (and therefore, issues were still reported to the OneBusAway server for Hillsborough County). Hillsborough Area Regional Transit started monitoring Hillsborough County in SeeClickFix in January 2017, at which point, issues reported in Hillsborough County were instantly directed to SeeClickFix and successfully received by HART. Future work could examine proposing this implementation as part of the Open311 standard.

The project team also worked with SeeClickFix to enable the direct submission of images via their Open311 API, although this design currently falls outside the exact Open311 specification. Previously, a link to an image had to be supplied. Future work could propose the additional of images to the Open311 standard.

As mentioned earlier, Open311 allows local jurisdictions to define the issue reporting categories for their location. While this flexibility is needed, general guidance for the issue category types related to public transportation would help agencies standardize some category and enable easier cross-agency coordination and reporting. TCRP Report 179 “Use of Web-Based Rider Feedback to Improve Public Transit Services” came to the same conclusion that a standardization of issue reporting categories does not currently exist but should be the topic of future research [28].

Currently all vehicle and stop metadata are inserted into the body of the description, which makes it difficult to produce reports and analytics for particular trip_ids, route_ids, or schedule deviations (e.g., show me all problems for Route 5 when it was running more than 10 minutes behind schedule). Open311 does not support “hidden” fields – this means if these fields are added individually, they would be shown to the user when they submit the issue. Ideally, these fields would be hidden from the user’s view, but the OneBusAway app would still have a way to discover these field names and submit discrete data for these fields. Future work could examine expanding the Open311 specification to include this use case.

4.3 Potential areas of improvement for the OneBusAway mobile app

HART requested that when a user is preparing to submit a new issue report, the OneBusAway app show existing issues that are nearby the user’s location so the user can determine if the problem has already been reported. This could be implemented as part of future work.

HART was interested in capturing more information about the transit rider, including when they arrived at a bus stop (i.e., how long they waited for the bus), as well as travel path and origin/destination information. This feature would need to be balanced with privacy protections and transparency for the user.

4.4 Potential areas of improvement within HART’s issue management process

When reviewing issue reports, HART discovered that the block_id values shared in their GTFS data do not match the block_ids that are visible to customer service representatives in the internal AVL system interface. This is relevant for issues related to bus driver behavior and potential disciplinary issues, as the customer service representatives need to look up the driver’s ID based on block_id. Future work could examine transforming block_ids so customer service representatives do not have to go through a manual lookup process to map the GTFS block_ids to internal block_ids.

HART had an overall very positive experience with the new issue management platform. HART plans to eventually move their entire Customer Service department to using SeeClickFix instead of their legacy customer service software. However, to satisfy their customer service requirements HART would need be able to automatically generate mailers responses to people that request them, as well as generate monthly letters to the driver union monthly for any actionable items. HART is currently investigating to determine if SeeClickFix can meet these requirements as well.

4.5 Transit agency lessons learned

HART has provided the following suggestions to agencies considering moving to a dedicated issue management platform like SeeClickFix:

1. Have dedicated personnel ready and trained prior to making system available to the public – HART opened issue submissions to the public prior to having all departments trained on the tool, as the assumption was that customer service would train other departments (e.g., Operations, Safety and Security) on the fly as issues came in related to those departments. If they were to repeat the process again, they would train the other departments prior to opening issue submissions to the public.
2. Be prepared to manage two systems during a transition period – As mentioned above, there are a few remaining tasks that HART's legacy issue management system can perform that are required functions that HART cannot yet perform in the new software. HART advised new agencies to be prepared to manage more than one system during this transition period until the legacy system can be completely replaced.
3. Put thought into secondary questions for users – When defining issue categories, give a lot of thought to the secondary required fields that a user must fill out before submitting the issue report. Make sure you include all information you need to resolve the problem to help minimize unnecessary back-and-forth communication with the user. If all information is gathered in the initial submission, it makes it much simpler to use "canned" responses to quickly triage reports.
4. Be careful with private/public comments – Ensure that staff understand that some comments via the administrative portal are private among team members and others are publicly shown on the site. Be aware that even internal comments may be subject to public records requests depending on the agency and local regulations.

4.6 Improvement to the GTFS-realtime specification

As discussed in Appendix Section 6.4, the research team encountered several areas where the GTFS-realtime specification did not clearly indicate the expected behavior for producers and consumers. GTFS-realtime community voted to adopt several proposals by the research team so transit agencies, AVL vendors, and application developers have clarity when implementing real-time transit systems, which will lead to better quality data and reduced development costs:

1. Clarify behavior for dropping StopTimeUpdates for vehicles running ahead of schedule - https://github.com/google/transit/pull/16
2. Conditionally require stop_sequence in StopTimeUpdate - https://github.com/google/transit/pull/20
3. Clarify that stops upstream of predictions have unknown delay - https://github.com/google/transit/pull/18

Chapter 5 - Conclusions

The enhancements to the OneBusAway app to report issues via the Open311 protocol has been successfully deployed in OneBusAway Tampa Bay through collaboration with FDOT D7, HART, PSTA, SeeClickFix, Clever Devices, and Cambridge Systematics. The system has successfully captured issues for a variety of categories, including safety, from the public and facilitated the sharing and management of these issues across jurisdictional and agency boundaries. All software source code developed under this project has been shared with the public via Github and contributed to the OneBusAway open-source project. The results of this project are freely and instantly available to any transit agency or city that sets up a OneBusAway server [29] and uses an issue management system that supports the Open311 protocol [21].

HART and PSTA both reported that a key advantage to using a dedicated issue management platform was the additional transit metadata that OneBusAway was able to attach to the issues. This reduced the "back-and-forth" communication with riders and saved considerable time when triaging the issue, as staff no longer had to do extensive research to discover to which trip or stop an issue report was related. Additionally, with legacy issue reporting processes it was difficult to track team performance such as time-to-close issues. This information is captured in dedicated issue management tools and can now be used for data-driven performance goals for customer service, as well as to identify areas of improvement for the various departments (e.g., safety and security, scheduling, maintenance) that are responsible for closing out issues that belong to them.

The arrival time data collected via OneBusAway could serve as a valuable resource to identify locations where the bus typically runs early or late, as well as identify the areas where this most affects riders. These routes could then be prioritized for schedule adjustments. Future work could examine existing geospatial tools to determine if they can be used to produce sophisticated reports based on the above data. Similarly, the analytics data being collected by the OneBusAway mobile apps could serve as a rich resource for identifying how users interact with mobile transit apps, including information such as how far from the bus stop they are when they access real-time arrival information. This data could help agencies prioritize stop improvements (e.g., new shelter additions). Future work could also focus on collecting rider travel path (i.e., origin and destination) data, with their permission, although rider privacy concerns would need to be addressed.

Several improvements to the Open311 protocol are recommended in Section 4.2, including server discovery, direct submission of images, and support for hidden fields. The research team also suggests the creation of a basic taxonomy of transit issue categories to assist the industry in tracking common metrics across agencies.

One of the outcomes of this research project was the adoption of three proposals from the project team by the GTFS-realtime community to improve the GTFS-realtime specification. These changes will help improve the quality of GTFS-realtime feeds in the future as they are created and updated. Based on the project team's experience, the creation of an open-source GTFS-realtime validation tool is strongly recommended. Such a tool would save the industry significant transit agency, AVL vendor, and application developer time and resources that are currently being spent on identifying and debugging errors in feeds. By reducing the effort needed to troubleshoot problems, the quality of GTFS-realtime feeds would improve.

References

[1] National Center for Statistics and Analysis (NHTSA) (2017). *Bicyclists and other cyclists: 2015 data* (DOT HS 812 382). NHTSA, Washington, DC.

[2] National Center for Statistics and Analysis (NHTSA) (2017). *Pedestrians: 2015 data* (DOT HS 812 375). NHTSA, Washington, DC.

[3] Langston Taylor. (2017). "Vision Zero strategy for safer Hillsborough roads: repaint, narrow lanes, educate." *Tampa Bay Times,* August 22, 2017 http://www.tampabay.com/news/transportation/vision-zero-plan-to-make-hillsborough-roads-safer-to-be-unveiled/2334630

[4] Kari Edison Watkins, Brian Ferris, Alan Borning, G. Scott Rutherford, and David Layton (2011), "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, no. 8, pp. 839-848.

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

[6] Brian Ferris, Kari Watkins, and Alan Borning, "OneBusAway: results from providing real-time arrival information for public transit," Proceedings of CHI 2010: 28th ACM Conference on Human Factors in Computing Systems, Atlanta, Georgia, USA, April 10-15, 2010. Copyright 2010 ACM 978-1-60558-929-9/10/04.

[7] A. 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.

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

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

[10] C. Brakewood, S. Barbeau, and K. Watkins (2014), "An experiment evaluating the impacts of real-time transit information on bus riders in Tampa, Florida," *Transportation Research Part A: Policy and Practice,* Vol. 69 pp. 409-422.

[11] OneBusAway. "OneBusAway - The Open Source Platform for Real Time Transit Info." Accessed August 1, 2017 from http://onebusaway.org/

[12] S. Barbeau, A. Borning, and K. Watkins (2014), "OneBusAway Multi-Region – Rapidly Expanding Mobile Transit Apps to New Cities," *Journal of Public Transportation,* Vol. 17 pp. 14-34.

[13] Brian Ferris (2011), "OneBusAway: Improving the Usability of Public Transit," Doctor of Philosophy Thesis, Computer Science and Engineering, University of Washington.

[14] TransDev. "Seeing through the Eyes of our Passengers." Accessed August 9, 2017 from http://www.transdevna.com/Transit/Bus/Passenger-Care.aspx

[15] Connected Bits. "Where's Street Bump being used?". Accessed August 9, 2017 from http://www.streetbump.org/

[16] SeeClickFix. "SeeClickFix." Accessed August 1, 2017 from https://seeclickfix.com/

[17] PublicStuff. "PublicStuff." Accessed August 1, 2017 from http://www.publicstuff.com/

[18] Connected Bits. "Connected Bits." Accessed August 1, 2017 from http://www.spotreporters.com/

[19] FixMyTransport. "Looking for FixMyTransport?". Accessed August 9, 2017 from http://www.fixmytransport.com/

[20] Tiramisu Transit, LLC. "Tiramisu - The Real-time Bus Tracker." Accessed August 1, 2016 from http://www.tiramisutransit.com/

[21] Open311 Community. "Open311 - A collaborative model and open standard for civic issue tracking." Accessed August 1, 2017 from http://www.open311.org/

[22] Center for Urban Transportation Research at University of South Florida. "open311-client." Accessed August 1, 2017 from https://github.com/CUTR-at-USF/open311-client

[23] SeeClickFix. "Issue ID: 3200286 - Open Electrical Box." Accessed August 1, 2017 from https://seeclickfix.com/issues/3200286

[24] SeeClickFix. "Issue ID: 3339143 - Broken Storm Drain." Accessed August 1, 2017 from https://seeclickfix.com/issues/3339143

[25] SeeClickFix. "Issue ID: 3511434 - Dangerous Bus Stop Location." Accessed August 1, 2017 from https://seeclickfix.com/issues/3511434

[26] SeeClickFix. "Issue ID: 3128306 - Bus Driver Compliment." Accessed August 1, 2017 from https://seeclickfix.com/issues/3128306

[27] SeeClickFix. "Issue ID: 3589955 - Glass panel at shelter broken." Accessed August 21, 2017 from https://seeclickfix.com/issues/3589955

[28] K. Watkins, Y. Xu, S. Bregman, and K. Coffel (2015). "TCRP REPORT 179 - Use of Web-Based Rider Feedback to Improve Public Transit Services," 179.

[29] OneBusAway Organization. "OneBusAway Github Source Code Repository." Accessed July 31, 2017 from https://github.com/OneBusAway/onebusaway-application-modules

[30] S. Barbeau (2013), "Open Transit Data – A Developer's Perspective," in *APTA TransITech 2013*, Phoenix, Arizona, March 20th, 2013.

[31] Google, Inc. "Protocol Buffers." Accessed July 31, 2017 from https://developers.google.com/protocol-buffers/

[32] S. Barbeau. "PSTA Data Issues." Accessed July 31, 2017 from https://github.com/CUTR-at-USF/psta-data/issues?q=is%3Aissue

[33] Center for Urban Transportation Research at University of South Florida. "onebusaway-stop-consolidation-support." Accessed August 25, 2017 from https://github.com/CUTR-at-USF/onebusaway-stop-consolidation-support

Appendix A - Multiple agencies within OneBusAway

At the start of this project, Pinellas County Transit Authority (PSTA), which provides transit service to Pinellas County on the west side of Tampa Bay, was not included in the OneBusAway Tampa region. To demonstrate the ability of more than one agency to triage and manage issues reported via the Open311 interface, the research team worked with PSTA and Clever Devices to deploy and test a General Transit Feed Specification (GTFS) and GTFS-realtime feed for PSTA's automatic vehicle location (AVL) system.

The following sections present a short introduction to the GTFS and GTFS-realtime formats, as well as the various issues that were discovered and resolved related to the PSTA's GTFS and GTFS-realtime data when these data are integrated into the OneBusAway Tampa region.

A.1 Introduction to General Transit Feed Specification (GTFS) and GTFS-realtime data formats

GTFS forms the foundation for a GTFS-realtime feed – a GTFS-realtime feed cannot provide practical real-time prediction information without having a companion GTFS feed that defines the schedule. GTFS data is implemented as a set of comma-delimited text files added to a single zip file.

A subset of the full GTFS specification is required for a GTFS-realtime feed – the following are key for understanding real-time information:

- stops.txt – All bus stops included in a feed, with each record including a stop_id (identifier internal to agency), stop_code (rider-facing stop identifier), stop location, location_type (a single stop or station with multiple stops), etc. For some agencies, stop_id and stop_code may be the same.
- routes.txt – All routes defined for an agency, including a route_id and short and long name
- calendar.txt and calendar_dates.txt – Includes service days and times, each identified via a service_id, that the agency provides service
- trip.txt – All trips defined for an agency, including to which route_id each trip belongs. A route may have multiple trip patterns, depending on the day and/or time. The day/time that each trip is operational is defined by a service_id that relates to calendar.txt and/or calendar_dates.txt
- stop_times.txt – The core schedule file that defines, for each trip_id, the ordered list of bus stops that will be visited, along with a scheduled arrival and departure time, and whether or not each stop is a timepoint (optional).

A GTFS stop_times.txt file will look like the following:

trip_id	arrival_time	departure_time	stop_id	stop_sequence
2777	5:52:00	5:52:00	4301	1
2777	5:52:34	5:52:34	3471	2
2777	5:53:46	5:53:46	4456	3
2777	5:54:27	5:54:27	592	4
2777	5:55:11	5:55:11	593	5
2777	5:55:20	5:55:20	4457	6
2777	5:55:40	5:55:40	595	7
2777	5:56:34	5:56:34	596	8
2777	5:57:09	5:57:09	6898	9
2777	5:57:42	5:57:42	6899	10
2777	5:58:17	5:58:17	597	11
2777	5:58:56	5:58:56	599	12
2777	5:59:20	5:59:20	600	13
2777	5:59:50	5:59:50	601	14
2777	6:00:15	6:00:15	602	15

The GTFS-realtime specification can be broken down into three types of elements:

- **Trip Updates** – Real-time predictions for when vehicles arrive and depart. Predictions (stop_time_updates) are represented as an update to the time that the vehicle was scheduled to arrive or depart (defined in GTFS stop_times.txt), either as a relative "delay" or "time". stop_time_updates are identified using a trip ID from GTFS trips.txt.
- **Vehicle Positions** – Real-time vehicle location, trip assignment (defined using the trip ID from GTFS trips.txt), and occupancy information
- **Service Alerts** – Descriptions of events that affect transit service, along with the transit stops/routes that the event impacts. For example, "Route 5 is on detour due to flooding".

A GTFS-realtime Trip Update for trip_id 2777 that predicts a bus running 60 seconds late for stop_id 4456 (stop_sequence 3), running on time for stop_id 592 (stop_sequence 4), and 60 seconds early for stop_id 593 (stop_sequence 5), would look like the following:

```
trip_update {
  trip {
    trip_id: "2777"
  }
  stop_time_update {
    stop_sequence: 3
    arrival {
      delay: 60  // Schedule deviation of 60 seconds (running late)
    }
    stop_id: "4456"
  }
  stop_time_update {
    stop_sequence: 4
    arrival {
      delay: 0  // Schedule deviation of 0 seconds (on time)
    }
    stop_id: "592"
  }
  stop_time_update {
    stop_sequence: 5
    arrival {
      delay: -60  // Schedule deviation of -60 seconds (running early)

    }
   stop_id: "593"
  }
}
```

The architecture of a real-time transit information system can be divided up into two components [30]:

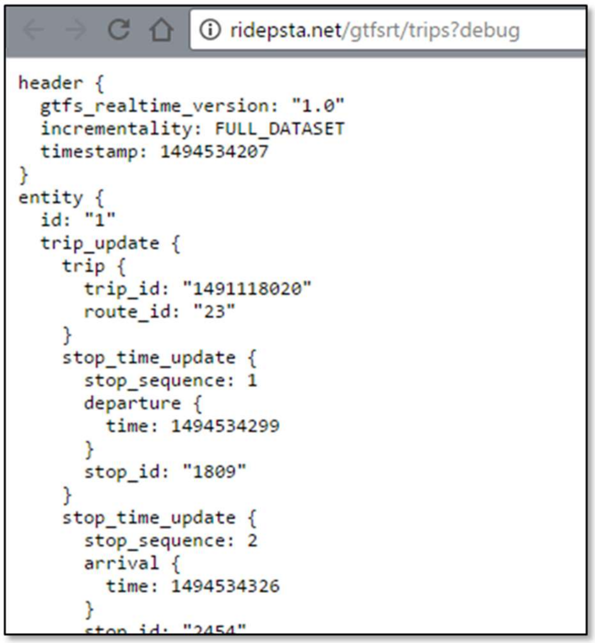
1. The Producer - The system generating the GTFS-realtime feed (typically the automatic vehicle location (AVL) system)
2. The Consumer – The system reading the GTFS-realtime feed (typically a server and mobile app displaying the information to a transit rider)

While GTFS datasets are typically updated 3-4 times per year (e.g., when new schedules are published), a GTFS-realtime Trip Updates and Vehicle Positions feed can be updated as often as every few seconds and are typically driven by an automatic vehicle location (AVL) system.

GTFS-realtime datasets are formatted in the Protocol Buffer format [31], which is a very efficient binary representation of the information in the feed. As a result, the actual GTFS-realtime messages produced and consumed by applications require special software to convert them to human-readable plain text.

The PSTA GTFS-realtime feeds used with OneBusAway were created by PSTA's AVL vendor, Clever Devices. A plain text version of these feeds is shown in Figure 29 and at the below URLs:

- http://ridepsta.net/gtfsrt/trips?debug
- http://ridepsta.net/gtfsrt/vehicles?debug
- http://ridepsta.net/gtfsrt/alerts?debug



```
header {
  gtfs_realtime_version: "1.0"
  incrementality: FULL_DATASET
  timestamp: 1494534207
}
entity {
  id: "1"
  trip_update {
    trip {
      trip_id: "1491118020"
      route_id: "23"
    }
    stop_time_update {
      stop_sequence: 1
      departure {
        time: 1494534299
      }
      stop_id: "1809"
    }
    stop_time_update {
      stop_sequence: 2
      arrival {
        time: 1494534326
      }
      stop_id: "2454"
```

Figure 29 – GTFS-realtime data feed from PSTA

The following sections discuss the various issues encountered during the deployment of OneBusAway with the new GTFS-realtime feed.

Erroneous GTFS-realtime arrival times were attributed to three sources, which are each discussed in subsequent sections:

1. **Producer Issues** - Bugs within the GTFS-realtime generation software and/or AVL system
2. **Consumer Issues** - Bugs or insufficient support of GTFS-realtime data within the OneBusAway software
3. **Different interpretations of the GTFS-realtime specification** – Some areas of the GTFS-realtime documentation have not been well-defined, and therefore consumers and producers may expect different output for these gray areas in the specification

A.2 GTFS-realtime producer issues

PSTA has been providing GTFS data to third party app developers since 2009 using the export feature of their scheduling software, GIRO's HASTUS. However, one key requirement for maintaining GTFS and GTFS-realtime data is that the identification numbers (IDs) within the GTFS data (trip_id, stop_id, etc.) must match the IDs in the GTFS-realtime data. To properly support matching IDs, PSTA transitioned from exporting their GTFS from HASTUS to exporting it from Clever Devices system, the same vendor being used for the AVL system. As a result, PSTA was creating a new version of their GTFS data in addition to the new GTFS-realtime feed.

The research team used the GTFS Feed Validator [12] to quickly identify and generate a report about issues in the new GTFS data. Identified issues include:

- **Incorrect route_long_names in routes.txt** – In PSTA's previous GTFS data, the route_long_name contained the descriptive name of the route like "Gateway Mall / Tyrone Square Mall", while route_short_name was "1". The new route_long_name contained the text "Route 1", which is an incorrect description of the route.
- **agency_url and timezone fields missing in agency.txt** – The agency.txt agency_url and timezone fields, which are both required by the GTFS specification to provide proper contact points and timezone information, were missing.
- **Stops have duplicate stop_codes in stops.txt** – The stop_code value should be the user-facing identifier displayed on a bus stop sign or shelter. However, for several stops the same stop_code was assigned to more than one stop. This resulted in duplicate stops being shown in the app for search results, one of which was missing a schedule (i.e., it showed "no arrivals or departures").
- **Duplicate times within trips in stops_times.txt** – arrival_time and departure_time must increase for each stop along the trip. Several trips showed the bus arriving at several stops in a row at the same exact time, which is incorrect.
- **"Too fast travel" warning for stop_times.txt** - This problem was a secondary issue resulting from the duplicate times within trips (above). Because the amount of time between sequential stops was very low (i.e., 0), the validator flagged the trips as traveling too fast for reality.
- **Bad shape data** - The shape data provided in GTFS shapes.txt to describe the actual travel path of the bus had some errors where a point would significantly deviate from the path of the vehicle. Because OneBusAway interpolates the vehicle position on the map based on the progress along the trip when no real-time information is available, this resulted in a strange display of information to the user where the vehicle is significantly off-route. This error was not flagged by the GTFS validation tool, but was found when manually testing the application.

The AVL vendor fixed these issues identified in the GTFS data and generated new GTFS data that did not have these problems. Some of these issues (incorrect route_long_name, missing agency_url and timezone fields, duplicate stop_code) were not software bugs, but were due to the way that PSTA staff had coded data within the data management tool. In these cases, the PSTA staff edited the data to correct the problem.

Troubleshooting the GTFS-realtime feed was significantly more challenging. The quality assurance process amounted to checking OneBusAway logs to determine if any errors were being identified, as well as physically visiting bus stops, checking arrival times shown in the app, and comparing them against when the bus actually arrived at the stop. However, the OneBusAway server software was built to be an application for the public, and not a validation tool. As a result, it often did not directly catch problems in the realtime feed or generate any errors. Instead, issues were identified when an abnormal arrival or departure time was manually identified within the OneBusAway mobile apps. Transit agency staff reported problems back to the research team, which then would attempt to identify the problem in logs and try to reproduce and/or manually catch the problem again in real-time. This was an extremely time-consuming process and involved significant communication between PSTA, the AVL vendor, and the research team.

The following issues were identified with the GTFS-realtime feed [32]:

- **stop_time_updates not sorted by stop_sequence** – To enable efficient processing by consumers, the GTFS-realtime specification requires that producers order predictions within a trip by stop_sequence. In other words, the predictions for stops within a real-time update should be in the same order as the stops occur within the trip, defined in GTFS stop_times.txt. The initial version of the PSTA TripUpdates feed did not include the optional stop_sequence field. The AVL vendor changed their software implementation to always sort stop_time_updates by stop_sequence, and eventually added the stop_sequence field to the GTFS-realtime feed so it was easier to confirm that each trip did indeed have updates sorted by stop_sequence.
- **Wrong stop_ids were included in trip_updates** – Occasionally stop_time_update estimates appeared in a trip with a stop_id that did not belong to that trip. This was caused by several problems, including more than one stop having the same stop_code in GTFS stops.txt and the handling of routes that contain a loop where a stop is visited more than once in the same trip (discussed in detail in a later section). The AVL vendor coordinated with PSTA to resolve this issue.
- **Stop_codes instead of stop_ids were included in alerts** – In the GTFS-realtime Alerts feed, alerts were published that related to particular stops. However, the stop_code, not the stop_id, appeared as the identifier in the alert. As a result, the alert could not be matched to the proper stop. The AVL vendor fixed this problem and published stop_ids to the alerts feed.
- **Invalid vehicle position data** – Occasionally a vehicle would have the latitude and longitude values of (0.0, 0.0) because of temporarily unavailable GPS data on-board the vehicle. The AVL vendor changed their feed to avoid publishing updates for vehicles with bad or unavailable GPS data.
- **Invalid vehicle route assignment data** – In the first version of the Vehicle Positions feed, vehicles that were not currently assigned to trips would appear in the feed with a route_id of "U" for "unassigned". Route_id should only be used for valid customer-facing routes that would appear in the GTFS routes.txt data, so these vehicles should not be included in the feed or should not have any route_id associated with then. The AVL vendor fixed the feed to remove this "unassigned" route information.
- **Unrealistic vehicle speeds** – In the initial version of the feed, very large vehicle speed values were observed (e.g., 129 miles per hour). This was because the speed values were being set in miles per hour, instead of the required units of

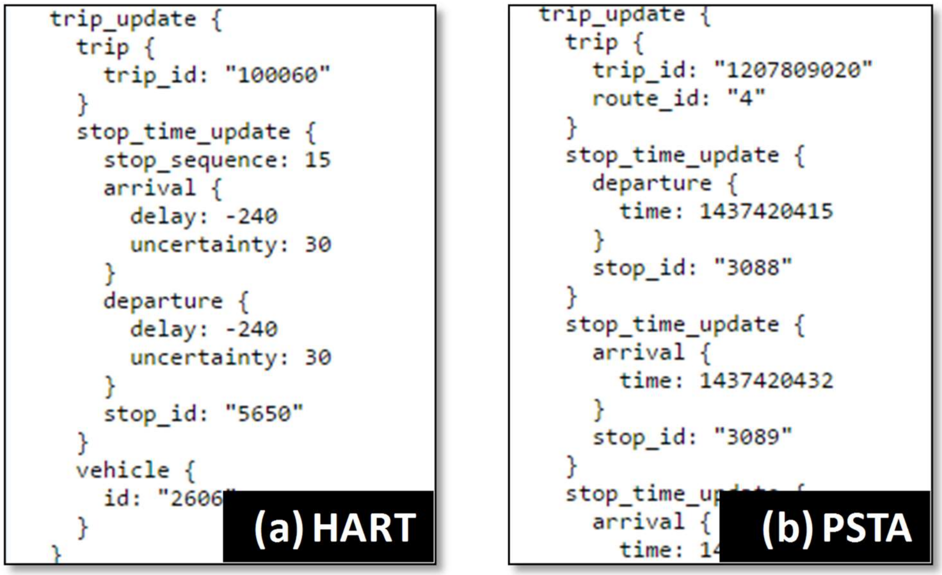
meters per second. The vendor resolved this issue by converting to the correct units before outputting the data to the feed. However, even after this was fixed, abnormally high-speed values were still observed. Apparently some vehicles were not calibrated to report speed accurately, so the AVL vendor worked on updating these vehicles to fix the reported speed.

- **Duplicate back-to-back stops in trip updates** – Some stops appeared more than once in sequence, each having a different predicted time of arrival in a stop_time_update. The AVL vendor fixed the problem to remove the duplicate stops and only have a single arrival time for each stop.

A detailed discussion of these issues, including sample data, is available on Github[8].

A.3 GTFS-realtime consumer issues

The research team discovered a few problems with the OneBusAway open-source software that negatively affected the predictions shown to riders. While OneBusAway already included basic support for GTFS-realtime feeds, the research team encountered several scenarios in PSTA's data that OneBusAway did not properly handle. These issues mostly stemmed from the fact that the PSTA GTFS-realtime feed provides many predictions (stop_time_updates) per trip – one for each stop (Figure 30).



```
trip_update {
  trip {
    trip_id: "100060"
  }
  stop_time_update {
    stop_sequence: 15
    arrival {
      delay: -240
      uncertainty: 30
    }
    departure {
      delay: -240
      uncertainty: 30
    }
    stop_id: "5650"
  }
  vehicle {
    id: "2606
  }
}
```

```
trip_update {
  trip {
    trip_id: "1207809020"
    route_id: "4"
  }
  stop_time_update {
    departure {
      time: 1437420415
    }
    stop_id: "3088"
  }
  stop_time_update {
    arrival {
      time: 1437420432
    }
    stop_id: "3089"
  }
  stop_time_up
    arrival {
      time: 14
```

Figure 30 - HART GTFS-realtime data (a) has only one estimated arrival (stop_time_update) per vehicle, while PSTA GTFS-realtime data (b) provides many arrival estimates (stop_time_updates) per vehicle

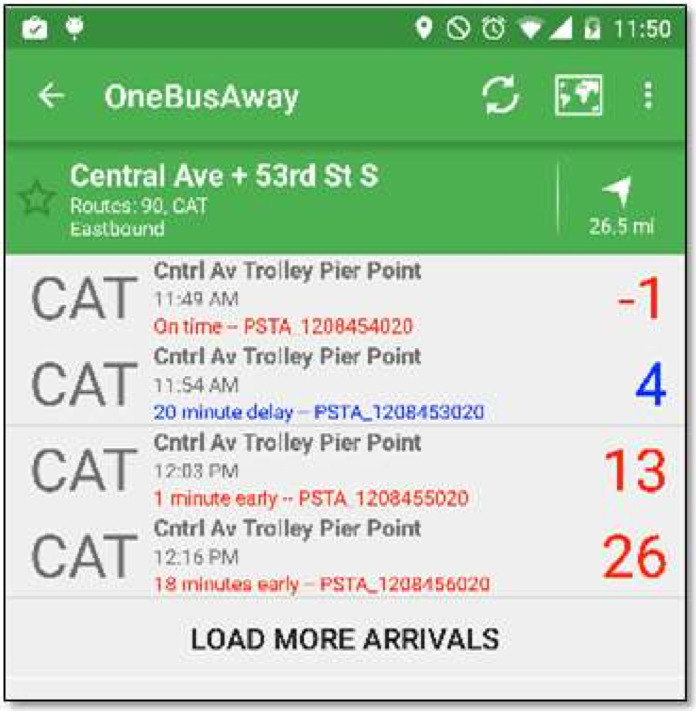
All previous GTFS-realtime feeds used in the various OneBusAway regions, including HART's GTFS-realtime feed, had only provided one prediction per vehicle. This single

[8] https://github.com/CUTR-at-USF/psta-data/issues?q=is%3Aissue

arrival estimate indicated whether a bus was running ahead, behind, or on schedule for a particular stop, and this same delay value was then applied to all stops for the rest of the trip (i.e., all stops “downstream” of the prediction). In contrast, PSTA’s GTFS-realtime feed provides an individual predicted time for *each* stop on the trip. Presumably, the additional arrival estimates for each stop in the trip have been calculated using an advanced prediction algorithm that takes other information (e.g., the route configuration, historical arrival information) into account when producing estimates. Therefore, it is in the best interested of transit riders to correctly consume each of these individual predictions, as it should result in more accurate estimates being shown to the transit rider. The research team developed improvements to OneBusAway to correctly handle multiple predictions per trip, including the specific issues discussed in the following subsections.

A.3.1 Per stop predictions resulted in large delays for stops that the bus has passed

When testing OneBusAway with the PSTA GTFS-realtime data, the research team saw large delays (e.g., 20 minutes) when viewing estimated arrival times in the mobile apps (Figure 31). OneBusAway was erroneously propagating predictions upstream of the stop for which the prediction was intended. This manifested in the app as a trip remaining in the upcoming arrivals list after the bus passes the stop, with a delay value that continues to grow until the bus has completed that trip.



Central Ave + 53rd St S
Routes: 90, CAT
Eastbound

CAT Cntrl Av Trolley Pier Point
11:49 AM
On time – PSTA_1208454020
-1

CAT Cntrl Av Trolley Pier Point
11:54 AM
20 minute delay – PSTA_1208453020
4

CAT Cntrl Av Trolley Pier Point
12:03 PM
1 minute early -- PSTA_1208455020
13

CAT Cntrl Av Trolley Pier Point
12:16 PM
18 minutes early – PSTA_1208456020
26

Figure 31 - When using per stop predictions, OneBusAway initially showed arrival information for vehicles that have already departed

The research team created a software patch to resolve this issue and stop propagating delays upstream of the intended stop[9]. Detailed documentation for the original problem is available on Github[10].

A.3.2 Departure predictions were not used by OneBusAway

The research team encountered a problem where the initial prediction for the first stop in a trip was not showing up in the OneBusAway app. For example, given the below input:

```
trip_update {
  trip {
   trip_id: "1208451020"
   route_id: "CAT"
  }
  stop_time_update {
   departure {
    time: 1436969397
   }
   stop_id: "4995"
  }
  stop_time_update {
   arrival {
    time: 1436969428
   }
   stop_id: "1605"
  }
  stop_time_update {
   arrival {
    time: 1436969533
   }
   stop_id: "1606"
}
```

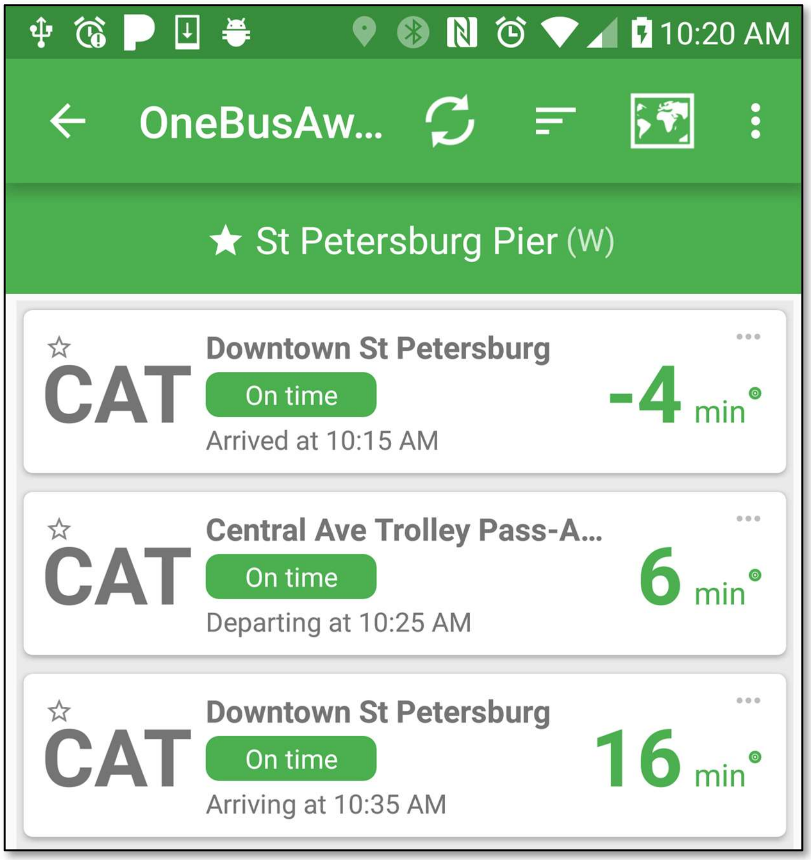
…OneBusAway would not show the estimated departure time for stop_id 4995. Upon further investigation, the research team found that OneBusAway was only designed to process per stop *arrival* times from GTFS-realtime feeds. The research team developed a software patch to process departure times as well as arrival times[11], and updated the mobile app user interface to better distinguish between arrivals and departures (Figure 32). The issue is documented in detail on Github[12].

[9] https://github.com/OneBusAway/onebusaway-application-modules/pull/142

[10] https://github.com/OneBusAway/onebusaway-application-modules/issues/127

[11] https://github.com/OneBusAway/onebusaway-application-modules/pull/142

[12] https://github.com/OneBusAway/onebusaway-application-modules/issues/138



★ St Petersburg Pier (W)

☆ CAT Downtown St Petersburg On time Arrived at 10:15 AM -4 min

☆ CAT Central Ave Trolley Pass-A... On time Departing at 10:25 AM 6 min

☆ CAT Downtown St Petersburg On time Arriving at 10:35 AM 16 min

Figure 32 – Arrival and departure predictions shown in the OneBusAway Android app

A.3.3 Interpolation of missing arrival times in trips

The research team encountered an issue with OneBusAway's interpretation of missing arrival predictions.

For example, if the following stop_ids exist:

- 1
- 2
- 3
- 4

…and the following deviations from stop_time_updates are in GTFS-realtime data:

- A
- --- (no data)
- B
- C

…when searching for the deviation for stop_id 2, OneBusAway attempted to interpolate the deviation value based on the A and C deviations. The interpolation software for OneBusAway was originally created prior to the development of GTFS-realtime, and as a result this behavior did not follow the GTFS-realtime specification. According to the GTFS-realtime specification, the deviation A provided for stop_id 1 should be propagated to stop_id 2, without any modifications. These portions of OneBusAway were created prior to the existence of the GTFS-realtime specification, and they needed to be updated to be compliant with the GTFS-realtime format. The research team developed a software patch to resolve this problem and correctly follow the GTFS-realtime propagation rules[13]. More detailed information on this issue can be found on Github[14].

A.3.4 Delay incorrectly prioritized over time for non-timepoints

The research team encountered a problem where OneBusAway was not showing real-time information for some stops in the middle of a trip with the following data:

```
stop_time_update {
  stop_sequence: 12
  arrival {
    time: 1436969397
    delay: 60
  }
  stop_id: "4995"
}
```

The PSTA GTFS schedule data did not provide scheduled arrival and departure times for this stop, as it was not a timepoint (the GTFS specification has since been updated to encourage provides to provide scheduled times for non-timepoints as well). In addition, OneBusAway was incorrectly prioritizing the "delay" value over the "time" value if both were provided in the feed. The result was the app failing to show a real-time prediction for this stop, because there was no schedule value to apply the "delay" to, which was needed to calculate the final predicted arrival time. The research team modified OneBusAway to follow the GTFS-realtime specification and use the provided "time", if both "delay" and "time" values are in the GTFS-realtime feed. This allowed OneBusAway to properly show the predicted arrival time to the user, even if the scheduled arrival time was not specified at that stop.

A.4 Different interpretations of the GTFS-realtime specification

Several scenarios were encountered where erroneous information was shown to transit riders, but the cause could not be attributed to a clear problem in the producer or the

[13] https://github.com/OneBusAway/onebusaway-application-modules/pull/142
[14] https://github.com/OneBusAway/onebusaway-application-modules/issues/139

consumer software given the current wording of the GTFS-realtime specification. Instead, these issues occurred because the producer and consumer interpreted certain portions of the GTFS-realtime specification differently. These “gray areas” of the spec resulted in a discussion among the members of the GTFS-realtime community, followed by a proposal by the research team to amend the specification and make the expected behavior of consumers and producers under these scenarios clear. The following subsections discuss each of the areas where the GTFS-realtime specification was improved.

A.4.1 Scheduled times are shown if a GTFS-realtime producer aggressively drops predictions

The research team encountered a problem when predictions were dropped from the GTFS-realtime feed for a stop just before or after a bus visited that stop. In these cases, if a vehicle was running early the user would see real-time information in the app until the bus arrived, and then the arrival time would jump back to the scheduled arrival time (even though the data indicated that the vehicle already left).

The research team worked with the GTFS-realtime community to clarify within the GTFS-realtime specification the GTFS-realtime feeds should not drop arrival predictions from a feed until after the scheduled arrival time for trips running early[15] and the AVL vendor updated their feed appropriately, and the research team developed a software patch[16] to handle this issue in OBA until the AVL vendor was able to update their GTFS-realtime feed. Additional documentation on this issue is available on Github[17].

A.4.2 Unmatched predictions for loop routes if stop_sequence was missing

The research team encountered a problem where large, incorrect delays were being shown for loop trips in OneBusAway. The problem was eventually traced to the GTFS-realtime feed providing ambiguous predictions for stops that appear twice in the trip – in other words, the GTFS-realtime feed was missing the stop_sequence for loop trips.

For example, with the following GTFS schedule data:

- stop_id = 1756, stop_sequence=1
- …
- stop_id = 1756, stop_sequence=30

…if the GTFS-realtime data includes an arrival prediction and only specifies that it should apply to stop_id = 1756, but not which “instance” or stop_sequence, OneBusAway does not have enough information to know which stop it should be matched to. In some cases,

[15] https://github.com/google/transit/pull/16
[16] https://github.com/OneBusAway/onebusaway-application-modules/pull/160
[17] https://github.com/OneBusAway/onebusaway-application-modules/issues/162

this resulted in arrival predictions for the later occurrence of the stop being applied to the earlier occurrence of the stop, which showed up in the app as large delays for each stops in the trip.

The research team worked with the GTFS-realtime community to require that GTFS-realtime feeds include the stop_sequence field if a stop is visited more than once in the same trip[18] and the AVL vendor updated their feed appropriately, and the research team also improved OneBusAway's handling of this situation[19]. Additional documentation for this issue is available on Github[20].

A.4.3 Stops upstream of predictions have unknown delay

In the process of attempting to clarify behavior for producers as to when they are allowed to drop per-stop predictions, it became apparent that the AVL vendor was assuming that when using per-stop predictions, consumers could either propagate predictions upstream or hold over predictions from a previous feed update and show these to end users.

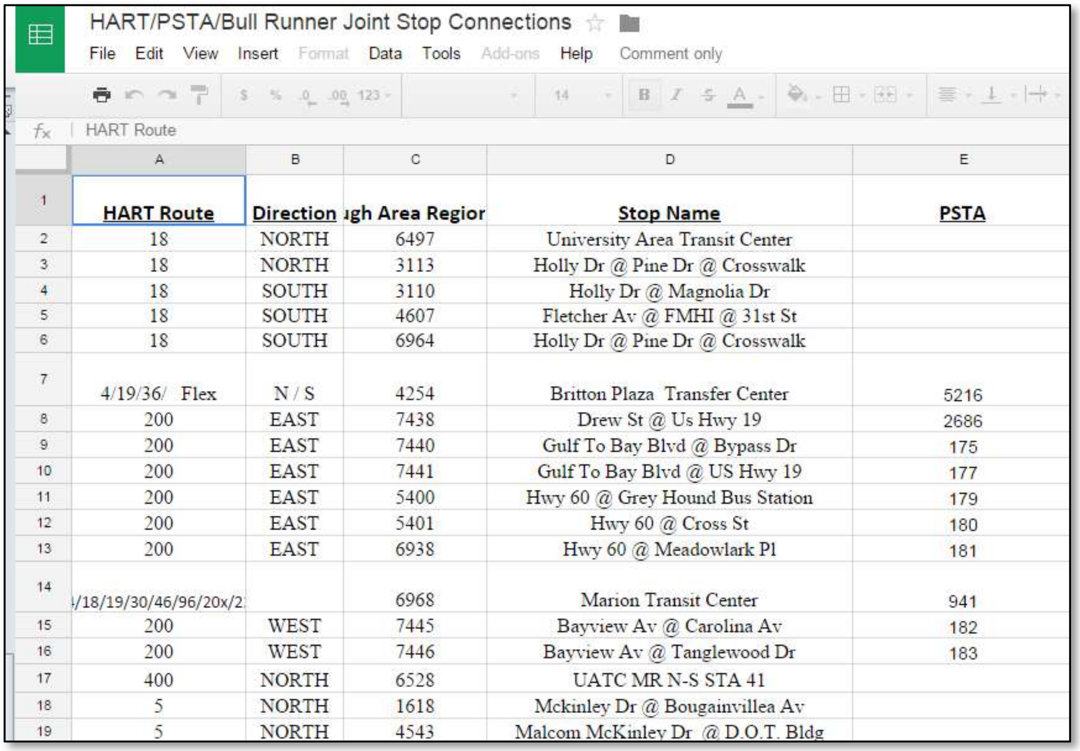
The research team proposed a clarification to the GTFS-realtime spec that that in the absence of any predictions upstream of a stop-specific prediction, it should be assumed that these upstream stops have an unknown delay[21]. The community accepted this proposal into the GTFS-realtime specification following a vote.

A.5 Merging co-located agency bus stops

Before deploying to the public, the research team initially configured and set up a demo instance of a OneBusAway server that includes PSTA and HART GTFS and GTFS-realtime data and used this server to evaluate issues with the PSTA data and how it interacted with the OneBusAway system and HART data.

To integrate HART and PSTA bus stop data in a regional deployment of OneBusAway, the research team created a software tool "onebusaway-stop-consolidation-support" [33] that compares the two sets of bus stops (one from HART's GTFS data, and another from PSTA's GTFS data) and presents them to transit riders as a unified stop. This allow riders to view both HART and PSTA arrival times at one stop, versus having to open one stop to view HART times and another stop to view PSTA times. The team gathered and processed stop datasets from PSTA and HART for all the HART and PSTA bus stops that are co-located, and placed it into a spreadsheet (Figure 33).

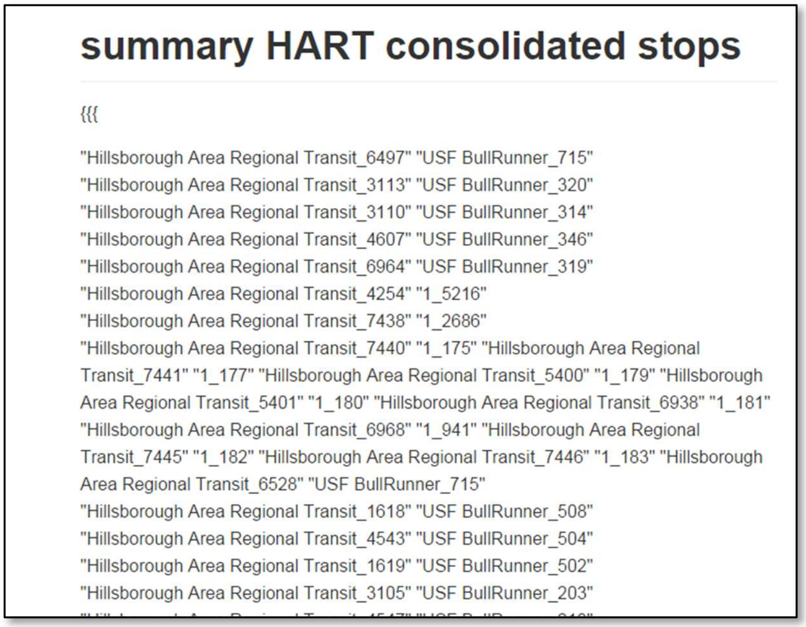
[18] https://github.com/google/transit/pull/20
[19] https://github.com/OneBusAway/onebusaway-application-modules/pull/166
[20] https://github.com/OneBusAway/onebusaway-application-modules/issues/163
[21] https://github.com/google/transit/pull/18



HART Route	Direction	ugh Area Regior	Stop Name	PSTA
18	NORTH	6497	University Area Transit Center	
18	NORTH	3113	Holly Dr @ Pine Dr @ Crosswalk	
18	SOUTH	3110	Holly Dr @ Magnolia Dr	
18	SOUTH	4607	Fletcher Av @ FMHI @ 31st St	
18	SOUTH	6964	Holly Dr @ Pine Dr @ Crosswalk	
4/19/36/ Flex	N / S	4254	Britton Plaza Transfer Center	5216
200	EAST	7438	Drew St @ Us Hwy 19	2686
200	EAST	7440	Gulf To Bay Blvd @ Bypass Dr	175
200	EAST	7441	Gulf To Bay Blvd @ US Hwy 19	177
200	EAST	5400	Hwy 60 @ Grey Hound Bus Station	179
200	EAST	5401	Hwy 60 @ Cross St	180
200	EAST	6938	Hwy 60 @ Meadowlark Pl	181
/18/19/30/46/96/20x/2		6968	Marion Transit Center	941
200	WEST	7445	Bayview Av @ Carolina Av	182
200	WEST	7446	Bayview Av @ Tanglewood Dr	183
400	NORTH	6528	UATC MR N-S STA 41	
5	NORTH	1618	Mckinley Dr @ Bougainvillea Av	
5	NORTH	4543	Malcom McKinley Dr @ D.O.T. Bldg	

Figure 33 - Spreadsheet used by the onebusaway-stop-consolidation-support tool to integrate multiple agencies co-located stops into a single logical stop

The onebusaway-stop-consolidation-support software tool then produces a configuration file that is used by OneBusAway to combine co-located stops. The configuration file that is output from the stop consolidation tool can be seen in Figure 34. This configuration data is then used by the OneBusAway server software to logically group stops from multiple agencies and present the unified information to riders (Figure 35). Detailed instructions for how to configure, compile, and execute this application can be found in the project README online [33].

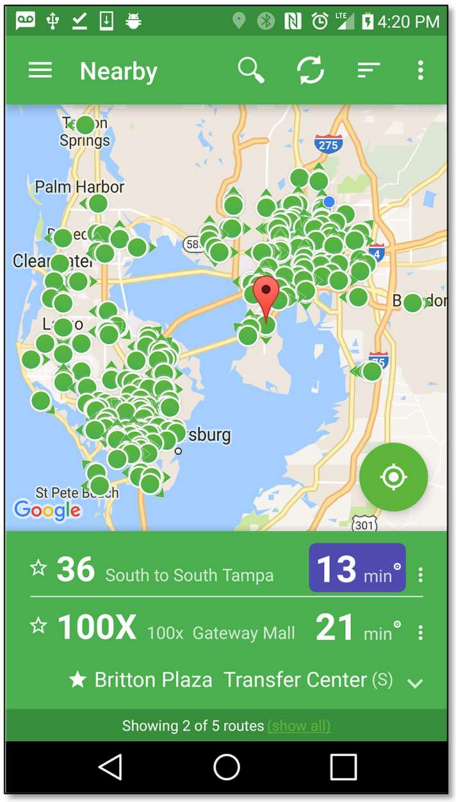


summary HART consolidated stops

```
{{{

"Hillsborough Area Regional Transit_6497" "USF BullRunner_715"
"Hillsborough Area Regional Transit_3113" "USF BullRunner_320"
"Hillsborough Area Regional Transit_3110" "USF BullRunner_314"
"Hillsborough Area Regional Transit_4607" "USF BullRunner_346"
"Hillsborough Area Regional Transit_6964" "USF BullRunner_319"
"Hillsborough Area Regional Transit_4254" "1_5216"
"Hillsborough Area Regional Transit_7438" "1_2686"
"Hillsborough Area Regional Transit_7440" "1_175" "Hillsborough Area Regional
Transit_7441" "1_177" "Hillsborough Area Regional Transit_5400" "1_179" "Hillsborough
Area Regional Transit_5401" "1_180" "Hillsborough Area Regional Transit_6938" "1_181"
"Hillsborough Area Regional Transit_6968" "1_941" "Hillsborough Area Regional
Transit_7445" "1_182" "Hillsborough Area Regional Transit_7446" "1_183" "Hillsborough
Area Regional Transit_6528" "USF BullRunner_715"
"Hillsborough Area Regional Transit_1618" "USF BullRunner_508"
"Hillsborough Area Regional Transit_4543" "USF BullRunner_504"
"Hillsborough Area Regional Transit_1619" "USF BullRunner_502"
"Hillsborough Area Regional Transit_3105" "USF BullRunner_203"
```

Figure 34 – The stop consolidation tool outputs a configuration file used by OneBusAway to combine co-located stops



☆ 36 South to South Tampa 13 min

☆ 100X 100x Gateway Mall 21 min

★ Britton Plaza Transfer Center (S)

Showing 2 of 5 routes (show all)

Figure 35 – A co-located HART/PSTA bus stop in OneBusAway Tampa Bay showing predictions for both agencies

The merging of this stop data allows more than one agency to host their schedule and real-time data in the same OneBusAway server instance – this enables agencies to share the support costs of the OneBusAway software, therefore reducing to total cost to each agency to provide real-time data to their riders. There is no limit to the number of regional agencies that can combine and share their data via OneBusAway. This open-source software will therefore serve as the basis for any new regional deployments of OneBusAway that include more than one agency, and can be re-used in other FDOT Districts.