ONEBUSAWAY MULTI-REGION – RAPIDLY EXPANDING MOBILE TRANSIT APPS TO NEW CITIES

Sean J. Barbeau, Ph.D. (Corresponding Author)
Center for Urban Transportation Research
University of South Florida
Tampa, FL 33620
813-974-7208
barbeau@cutr.usf.edu

Alan Borning, Ph.D.
Department of Computer Science & Engineering
University of Washington
Seattle, WA 98195
206-543-6678
borning@cs.washington.edu

Kari Watkins, Ph.D.
School of Civil and Environmental Engineering
Georgia Institute of Technology
790 Atlantic Drive, Atlanta, GA 30332
206-250-4415
kari.watkins@ce.gatech.edu

November 15, 2013

5,928 words + 5 figures (1,250) = 7,178 words

ABSTRACT

Real-time transit information offers many benefits to transit riders, including reduced wait times and increased customer satisfaction. However, offering real-time transit services has been challenging for many transit agencies. While mobile apps have emerged as a preferred dissemination method for real-time information, it is typically cost-prohibitive for transit agencies to fund custom development of native mobile apps for all popular smartphone platforms. Third-party developers can offer services if an agency openly shares real-time data, but these individuals are volunteers whose priorities and deadlines may not be the same as the agency's. As a result, few cities have full app portfolios that cover all smartphone platforms. This paper presents the OneBusAway multi-region project, a collaborative effort that is enabling the rapid expansion of native mobile transit apps to new cities. OneBusAway is an open-source transit information system that has provided real-time transit services to the Puget Sound, WA area since 2008. The new OneBusAway multi-region feature expands the coverage of the existing Android, iPhone, Windows Phone, and Windows 8 apps for OneBusAway to new cities including Tampa and Atlanta. The multi-region system architecture, collaborative design and development process, and lessons learned from this ground-breaking project are discussed. The fundamental shift from proprietary to open-source software in the transit industry that has made this type of project possible is also examined.

1. INTRODUCTION

Real-time transit information has many benefits for transit riders. Past research has shown that transit riders who have access to real-time information perceive their wait time to be around 30% shorter than riders who do not have access to real-time information [1]. Additionally, real-time information users save almost 2 minutes in waiting time, which has a very high disutility value and can be used to perform other tasks. Four Federal Transit Administration (FTA) workshops held in Seattle (WA), Salt Lake City (UT), Columbus (OH), and Providence/Kingston (RI) concluded that real-time information attracts new riders who are otherwise reluctant to start using transit [2]. Similarly, a study in Chicago found modest ridership gains from real-time information even prior to wide usage of smart phones [3]. Interviews with transit riders in San Francisco and Seattle in 2010 revealed that when the real-time information system was down, several riders elected not to ride the bus [4]. Riders can also use the information to adjust their own use of the transit system, e.g., by taking a different less-crowded bus, which can benefit other riders as well [5]. Other benefits found in surveys include increased walking (i.e., public health benefits), and for some riders, increased feelings of safety while waiting, particularly at night [6, 7].

However, offering real-time information services to transit riders has significant challenges. The cost for a transit agency to implement both Automatic Vehicle Location (AVL) technologies and information dissemination technologies (e.g., electronic signs, mobile phone apps) is not trivial ($70 million for a mid-sized city [5]), especially in the public sector where budgets are under pressure. This estimate does not include the cost of mobile apps, which is also significant. The development cost for a business app that includes real-time information can be upwards of $150,000 [8]. Understandably, agencies have cited development costs to be the primary barrier for offering "official" transit agency mobile apps [9]. Another issue is the multiplicity of smartphone platforms. Agencies are reluctant to support all major platforms due to costs, yet choosing which one or two platforms to support can also be difficult. Since riders have shown a preference for accessing real-time information via mobile apps (versus other methods such as text-messages or websites [1]), agencies must find another cost-effective solution for providing mobile apps to riders.

One strategy for increasing the number of mobile transit apps at a transit agency is for the agency to share static (i.e., schedule) and real-time transit information with the general public [9, 10]. Third-party developers (i.e., individuals not associated with the transit agency) can then independently develop and release mobile apps that use this data to the general public. This strategy has successfully produced a number of third-party transit apps at several agencies in the U.S., including Bay Area Regional Transit (BART) in San Francisco [11], TriMet in Portland [12], Metropolitan Transportation Authority (MTA) in New York [13], and Massachusetts Bay Transportation Authority (MBTA) [14].

However, these independent developers may not have the same priorities and deadlines as agencies. For example, if a developer doesn't fill the need for an app on a particular platform, or an app with particular features (e.g., an accessible interface for individuals with visual or other disabilities) then no such app will exist. Additionally, not all cities in the United States have robust high-tech transit populations and developer communities. In these cities, app growth is more modest [15]. And, since real-time transit data formats often differ between cities, apps for one city can't easily be shared with another.

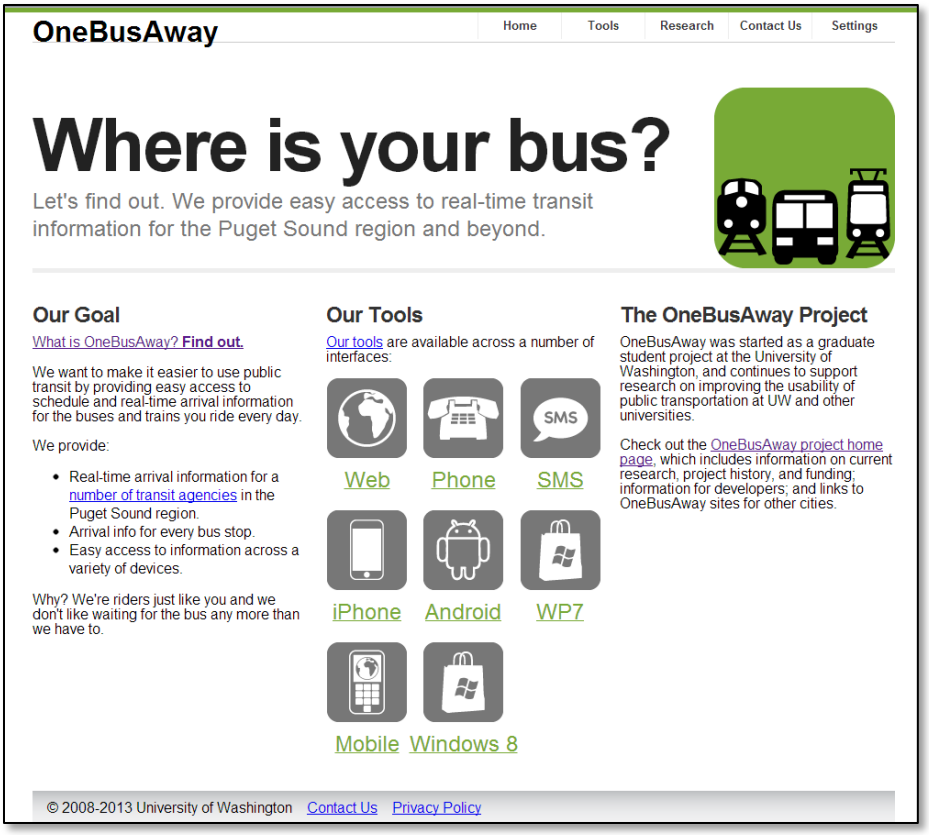

FIGURE 1 OneBusAway is the first open-source transit information system that includes iPhone, Android, Windows Phone, and Windows 8 native apps

OneBusAway, a real-time transit information system originally created by researchers at the University of Washington (UW) (FIGURE 1), takes a new approach to the problem of transit information dissemination [16].

Unlike traditional transit industry software, OneBusAway is open-source, meaning that the source code for the software is openly available for anyone to download, configure, alter, and deploy [17] . In addition to being open-source, OneBusAway supports popular bulk transit data formats such as General Transit Feed Specification (GTFS) [18], GTFS-realtime [19], and Service Interface for Real Time Information (SIRI) [20], which means that anyone with access to transit data in these formats can launch their own OneBusAway service for their city. Furthermore, OneBusAway includes open-source native mobile apps for iPhone, Android, Windows Phone, and Windows 8, which provide rich functionality and responsiveness beyond what is typically available in web applications. OneBusAway has been used to jump-start several pilot and production deployments of real-time transit information systems [21].

However, until recently there was a key limitation with the original OneBusAway project – the OneBusAway mobile apps on the respective app stores (i.e., Google Play, Apple App Store, Windows Phone Store, Windows Store) were only configured to work in Puget Sound, WA, where OneBusAway was originally developed.

Extending the reach of the OneBusAway apps for iPhone, Android, Windows Phone, and Windows 8 to new cities raised many questions:

- Should researchers or transit agencies launching new installations of OneBusAway in other regions also launch their own versions of each app in that region?
- If instead these researchers or transit agencies wanted to make use of project-wide OneBusAway apps, how could these apps be configured to work in new OneBusAway cities? Should OneBusAway app users be required to manually configure their apps to work in the correct city? Or, if a centralized server directory was provided, who would be responsible for implementing and supporting this directory? And who would make the required changes to the apps to use the directory?
- Would third-party developers be willing to support new versions of their apps in new cities?
- How should user feedback in multiple cities be directed to the right person (i.e., app developer or regional OneBusAway server administrator)?

This paper presents the OneBusAway multi-region project [22], which investigated these questions with the goal of producing a sustainable, low-maintenance, cost-effective system that would support the rapid expansion of mobile transit apps for iPhone, Android, Windows Phone, and Windows 8 to new cities around the world.

2. BACKGROUND

There are two primary developments in the transit industry over the last decade that made the OneBusAway multi-region project possible: the development of the original OneBusAway open-source project, and the emergence of open transit data.

OneBusAway started as a student project at UW in Seattle, motivated by the simple desire to have a truly usable interface for real-time transit information. It evolved into the Ph.D. dissertation work of Brian Ferris [23] and Kari Watkins [24], and at the same time spread virally to serve 100,000 unique weekly transit riders without official support from the transit agencies and with little outreach or publicity. Sound Transit, King County Metro, and Pierce Transit provided financial support for UW to continue operating OneBusAway from summer 2011 until summer 2013, at which point it was transitioned to Sound Transit.

The second factor that makes OneBusAway multi-region feasible is the growing availability of open transit data, and in particular the emergence of several de-facto transit data standards such as GTFS [18]. As of December 2012, over 500 agencies worldwide are sharing static (i.e., schedule) data in the GTFS format [25], which allows third-party developers to create transit apps based on this data. GTFS was originally created by Google and TriMet in 2005 as a lightweight and easily maintainable transit data format for the Google Transit trip planner [26]. While many agencies originally created GTFS data for Google Transit, many transit and multimodal applications based on GTFS data have emerged [27], including OneBusAway.

In addition to static data, OneBusAway also requires a real-time data source. Real-time transit data formats can be categorized into two magnitudes: fire hose and faucet [10]. Fire hose data formats contain a complete set of the entire state of the transit system, including all known estimated arrival times and all real-time vehicle locations for all routes and stops. In contrast, faucet data formats contain a precise subset of transit data, typically in response to a specific query (e.g., “The next bus on route 16 will arrive at stop ID 100 in five minutes”).

GTFS-realtime and SIRI have emerged as the two most popular fire hose open data formats [10]. The OneBusAway server software can import both GTFS-realtime and SIRI data frequently (e.g., every 30 seconds) to reflect real-time changes for the entire transit system. Other proprietary formats such as OrbCAD FTP and Nextbus are also supported. And, since OneBusAway is open-source, support for new formats can be added by any developer [28]. As a result of the above, the OneBusAway server software can be deployed with few modifications in any city that provides data in the above formats.

One of the primary functions of the OneBusAway server is to take fire-hose data as input, and provide "faucet" data as output, on demand, to thousands of apps. OneBusAway currently supports a custom-designed Representational State Transfer (REST) Application Programming Interface (API) for the faucet data, which allows the iPhone, Android, Windows Phone, and Windows 8 apps to retrieve real-time transit data specific to a device's location and/or user's request [17].

Comparison to other real-time transit applications

The open-source nature of OneBusAway is a key differentiator from commercially-available apps such as Moovit, Google Maps, Apple Maps, Microsoft Bing, Embark, RouteShout, Nokia Here, The Transit App, and Tiramisu. These "closed-source" applications are all operated by a single entity that has full control over what cities are supported. A city can request to be included, but they may not be added to the service. Business decisions, such as Apple's choice to remove Google Maps in mid-2012 which resulted in the loss of transit directions for iPhone users, can also instantly leave riders without any transit information.

OneBusAway provides a different model – the software source-code is openly provided to the general public. Therefore, each region can independently create and operate its own OneBusAway server, and one region's actions have no effect on another. Additionally, if a OneBusAway regional operator shuts down, another operator in the same region can resume the service.

While there are significant advantages to the independent nature of OneBusAway regions, this independent design also creates the need for some initial coordination when determining how the OneBusAway mobile apps will interact with these independently operated servers. A solution, the OneBusAway multi-region architecture, is discussed in the following section. This solution can be described as a "you bring the server, we bring the apps" approach, where the OneBusAway apps are centrally maintained and available to all regions, but each regional server is independently created and operated. This system design, enabled by the open-source nature of the project, is unique to OneBusAway. Additionally, OneBusAway provides native mobile apps on four different platforms (Android, iPhone, Windows Phone, and Windows 8), which is more than any of the above-mentioned commercially-available solutions.

3. MULTI-REGION ARCHITECTURE

Design decisions

There were several possible strategies for making the OneBusAway mobile apps available in other cities beyond Puget Sound, WA.

One potential strategy was to mirror the replication process of OneBusAway servers for new cities. When a new city wants to set up a new OneBusAway server, engineers will copy the OneBusAway server source code, configure it to access the new city's real-time transit data, and deploy the copy to a server in the new city. This new OneBusAway server would then provide real-time information via a website.

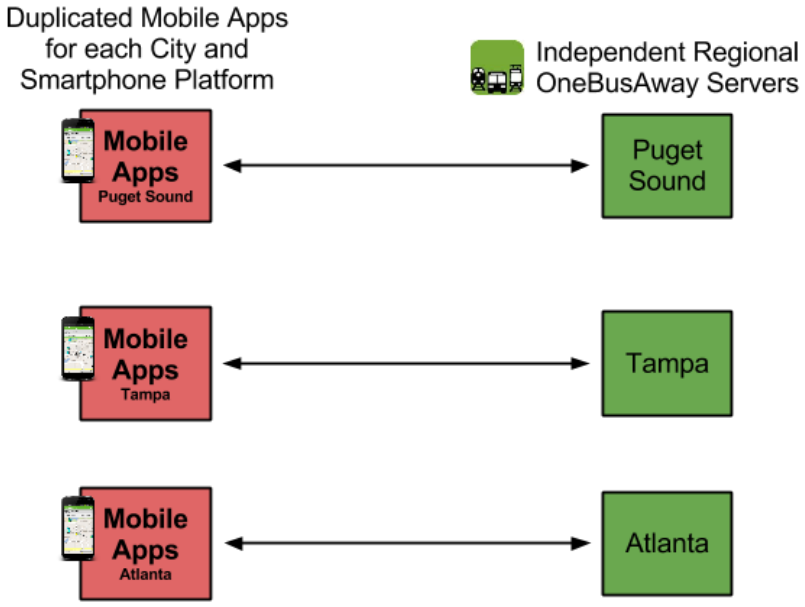

FIGURE 2 Duplicating mobile apps for each city leads to problems with sustainability, fragmentation, and scalability

To mirror this strategy for the mobile apps, engineers in the new city would copy the source code for the iPhone, Android, Windows Phone, and Windows 8 apps. Then, the source code would be changed use the local OneBusAway server (instead of the Puget Sound server) as shown in FIGURE 2. Finally, these modified apps would be deployed to the respective app markets with names such as "OneBusAway Tampa" or "OneBusAway Atlanta".

This strategy has the advantage of each city acting independently to deploy mobile apps, without requiring any coordination among cities. However, this approach has three major drawbacks:

1. *Sustainability* – Each city would need to find new developers to maintain and update the local Android, iPhone, Windows Phone, and Windows 8 apps. This is clearly undesirable, as it is already challenging for many cities to find developers interested in developing transit apps.
2. *Fragmentation* – There would be one copy of each mobile app source code for each city. Therefore, for every bug fix in each mobile app, developers in each city would all have to adapt that fix to their particular modified version of the app. This creates source code

that is difficult to maintain, limiting shared app improvements among cities. Additionally, when users try to download the app from the respective app store, they would be presented with a list of OneBusAway apps from all cities to choose from (e.g., "OneBusAway Tampa", "OneBusAway Atlanta") which places the burden on the user to find and install the correct app.

3. *Scalability* – The above two problems increase in complexity as OneBusAway is scaled up to include more and more cities.

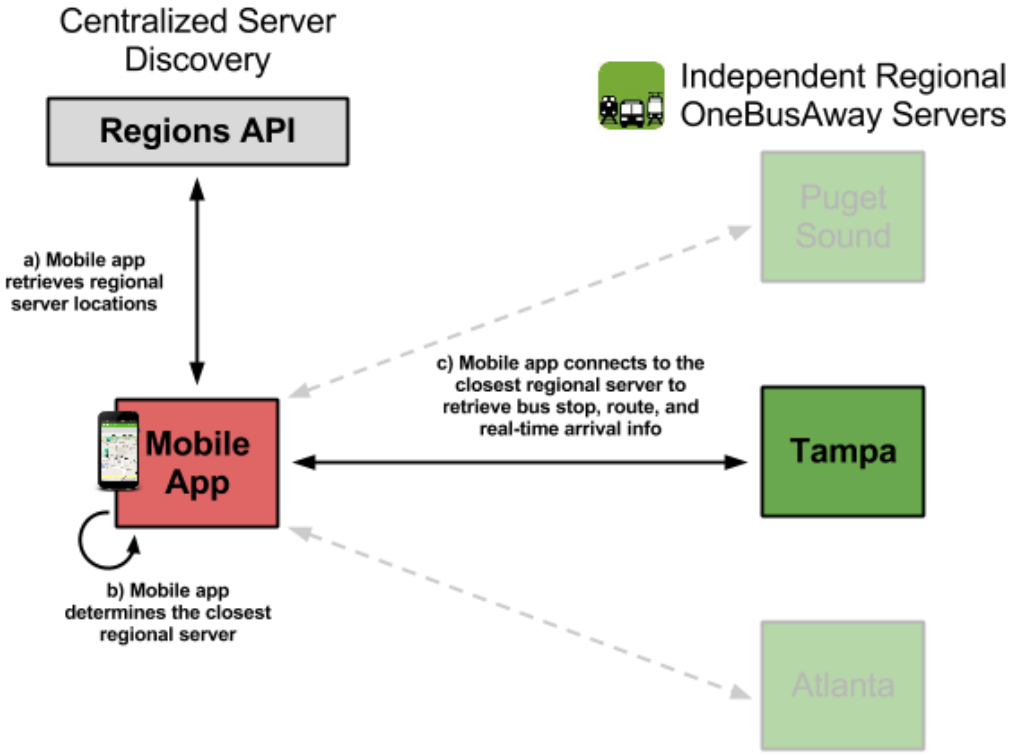

FIGURE 3 OneBusAway Multi-region Architecture efficiently extends mobile apps to new cities

An alternate approach is for a group of pilot cities to work together and create a coordinated OneBusAway multi-region system (FIGURE 3). Here, a centralized OneBusAway directory is created with a list of known OneBusAway servers in various cities. Then, the existing iPhone, Android, Windows Phone, and Windows 8 apps are modified so they discover available OneBusAway servers from the directory (i.e., "Regions API"), as shown in FIGURE 3a. Then, the app compares the user's real-time location to the list of server locations (FIGURE 3b), and then connects to the closest server to retrieve route, stop, and arrival information (FIGURE 3c).

Using this approach, the complexity of the OneBusAway multi-region system is hidden from the user, and users in all cities download the same app from the mobile app stores. Additionally, only a single copy of the source code for each app needs to be maintained, and users in all cities would immediately benefit from app improvements. This strategy does require more work and coordination up front for the pilot cities, including the original third-party app

developers. However, it drastically reduces sustainability and fragmentation problems for the future of the project, making the system scalable and reducing the overhead of adding more cities to the project. The overall OneBusAway project also benefits from this coordination through additional contributions and feedback from users and developers in multiple cities. Therefore, this strategy was chosen for the OneBusAway multi-region project.

Detailed Protocol

FIGURE 4 shows the detailed protocol used in the multi-region architecture, including interaction with both the Regions API as well as a regional OneBusAway server.

When the user first installs and starts the app, the app retrieves a list of region information from the project-wide Regions API and saves this list on the mobile device. Then, the device compares the real-time location of the user to the list of region locations, and automatically selects the closest region to the user. If there are any problems with device positioning, the user can also be presented with a list of available OneBusAway regions to choose from.

After the region has been selected, the app directly contacts the regional OneBusAway server to retrieve information about stops and routes that can then be shown to the user. For example, the app might show a set of nearby bus stops on a map. The user can then select a stop to see estimated arrival times for that location. The app then contacts the regional OneBusAway server again to get a list of estimated arrival times for the given stop ID, and show this information to the user. At this point, the user may close the app.

The next time the user starts the app, it compares the user's real-time location to the list of regions stored on the device (i.e., the most recently cached list from the Regions API) in the background to avoid interrupting the user experience. If the user is still in the same region, it continues using the previously identified server. In the less likely event that the user has moved into a different OneBusAway region (e.g., traveled between cities) since last app startup, the app will automatically switch to the currently closest OneBusAway region, fetch information from that regional server, and move the map to the user's new location. The implementation of different OneBusAway servers covering different geographic areas is thus completely transparent to the user.

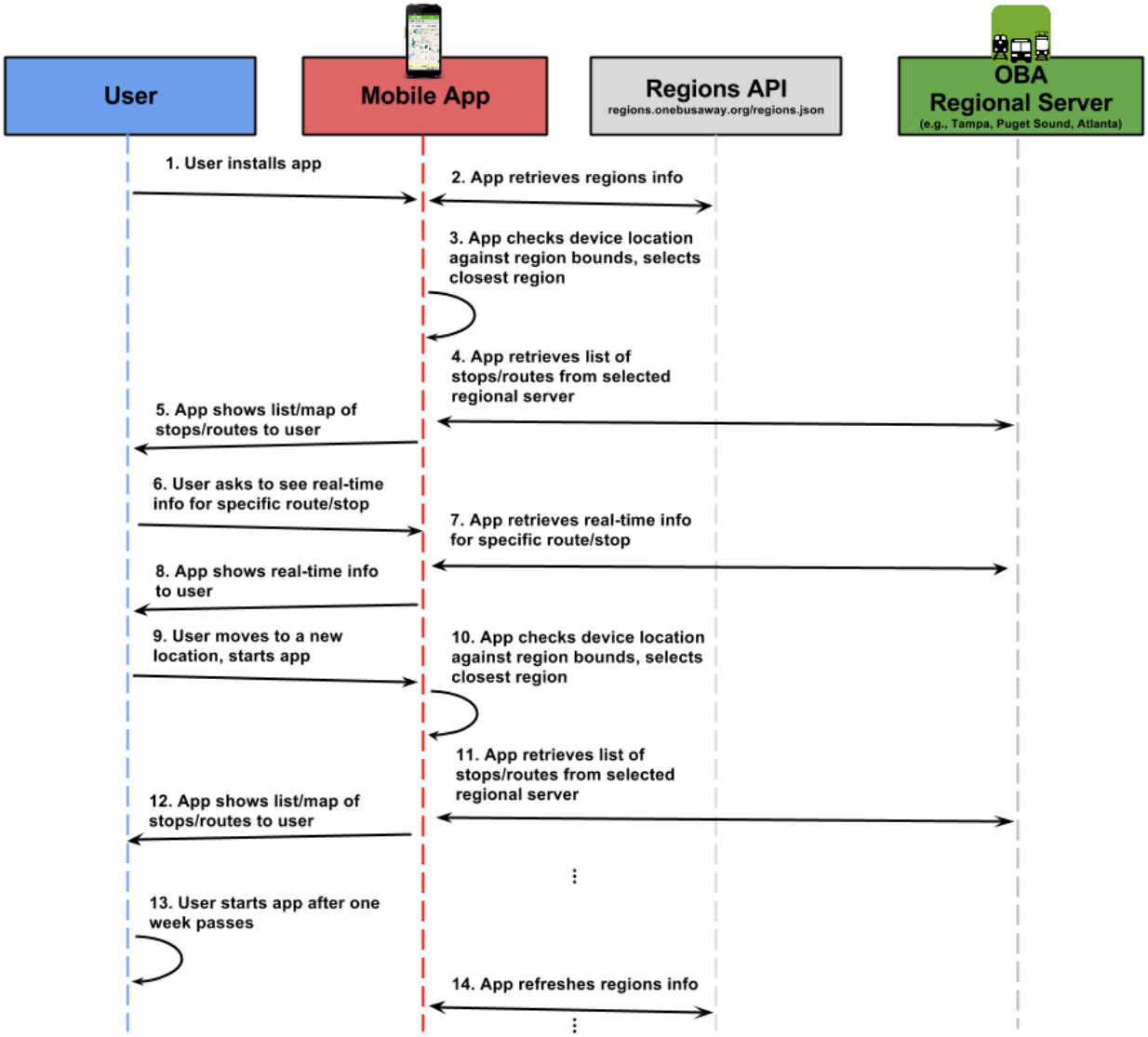

FIGURE 4 Protocol used by a mobile app to determine which OneBusAway regional server it should connect to, via the Regions API

There will occasionally be changes to the list of servers and configuration information, including the addition of new regions. Since this information isn't expected to change frequently, the mobile app only needs to occasionally refresh the local copy of region information from the Regions API – once per week in the current design. Thus, the mobile app operates mostly independently of the Regions API. This design also allows the system to scale easily, since as each new OneBusAway city is added, the vast majority of the new traffic will be handled by the regional OneBusAway server in that area, with only a small increase in traffic for the centralized Regions API. To protect against a potential Regions API failure, a copy of the regions list is also bundled with the app when it is installed on the device. This copy allows the app to continue to function immediately after initial installation without the Regions API, although with older region information.

Mobile App Modifications

For the multi-region project to be successful, two issues needed to be addressed for each of the iPhone, Android, Windows Phone, and Windows 8 apps:

1) A developer with skills specific to that mobile app platform would need to modify the app to support the multi-region architecture
2) The third-party developers who publish each of the OneBusAway apps to respective app stores (e.g., Google Play, Apple App Store, Windows Phone Store, Windows Store) would need to agree to publish a new multi-region version of their mobile apps

Since the apps are open-source, Issue #1 could also potentially be addressed by another developer, not necessarily the primary maintainer of the mobile app. A detailed discussion of the advantages of this open-source model, as well as various collaboration tools that facilitate this process, can be found in the following "Collaborative Process" section of this paper.

An important aspect of Issue #2 is the potential for a significant increase in user questions and feedback when the app is launched in a new city. For example, as of July 25th 2013, the OneBusAway Android app was actively installed on 141,817 devices, with a total of 234,281 downloads primarily for just the Puget Sound area. To avoid overwhelming the mobile app developers with a large amount of user feedback for new cities, the decision was made to have the "Contact Us" button in all the apps report information back to the regional OneBusAway administrator. This design scales well as new OneBusAway administrators and support teams for each new OneBusAway region are added. Further, the current OneBusAway app developers and OneBusAway server administrators indicate that the vast majority of user feedback pertains to issues specific to the region (e.g., errors in the schedule and real-time data), not to the mobile app. It is also often not clear to users where the source of the problem lies, and troubleshooting sometimes requires knowledge of the system operation. Therefore, the OneBusAway administrators will handle the majority of feedback, and can direct any application-specific feedback to the respective application developer as needed.

OneBusAway Server Administrators

In order for the mobile apps to have up-to-date information for each region, OneBusAway regional server administrators must have a way to update a centralized OneBusAway Server Directory. This process must require only low-levels of implementation and maintenance effort, both for the central server directory administrator as well as the individual regional OneBusAway server administrators.

A Google Doc spreadsheet was selected as the primary data entry tool for regional OneBusAway server administrators. Google Docs provides a reliable, ready-to-use platform for data entry into a spreadsheet that includes access control and data output in the Comma-Separated Values (CSV) file format. The Google Doc is configured to alert a set of administrators that oversee the entire OneBusAway open-source project, referred to as "Multi-region Administrators." Then, the multi-region admin will run a Python script to convert the CSV output of the Google Doc to regions.json and regions.xml files, which are then made available to mobile devices via a file server as the Regions API.

OneBusAway Regions

As of August 2013, the OneBusAway software suite is deployed to Puget Sound, WA, Tampa, FL, and Atlanta, GA. MTA in NY uses a modified version of OneBusAway for the MTA Bus

Time project [29]. Detroit has used the OneBusAway software to implement their "Text-My-Bus" text-messaging service for transit riders [30].

In Puget Sound, WA, real-time data from several regional transit agencies (King County Metro, Sound Transit, Pierce Transit, and Intercity Transit) is provided to a single OneBusAway instance hosted by Sound Transit. King County Metro's data is provided by a dedicated HTTP server that is made available to OneBusAway; Pierce Transit is provided via FTP from a secure file server; Intercity Transit via HTTP; Sound Transit data is provided via other agencies that operate the Sound Transit vehicles under contract. The system also has schedule-only data from a number of other agencies, including Community Transit, Washington State Ferries, City of Seattle, and the Seattle Children's Hospital Shuttle. Additional real-time data feeds are expected in the future.

In Tampa, the University of South Florida (USF) team created an open-source GTFS-realtime feed from Hillsborough Area Regional Transit (HART)'s OrbCAD AVL SQL Server database [31], and used the GTFS-realtime feed as input to the OneBusAway Tampa server.

In Atlanta, the Georgia Tech team created a GTFS-realtime feed from the Metropolitan Atlanta Rapid Transit Authority (MARTA) proprietary REST API real-time bus data feed, and used this as input to the OneBusAway Atlanta server.

The effort required to create a new OneBusAway deployment and participate in the OneBusAway multi-region project is moderate. An agency or researcher must:

1. Obtain access to static transit schedule data in GTFS format and to a real-time transit data source,
2. Install and configure a OneBusAway server,
3. Contact the OneBusAway group to include the new region in the central directory

4. COLLABORATIVE PROCESS

Creating the process and infrastructure to rapidly expand mobile transit apps to new cities required a large collaborative effort.

As mentioned earlier, individual OneBusAway server administrators were involved in the process to ensure that the process to add and maintain servers in the multi-region architecture was not effort-prohibitive. App developers were an integral part of the design process for the implementation and maintenance of the Regions API. The official formation of OneBusAway Board of Directors in January 2013 helped solidify the general OneBusAway project governance model, and members of the board served as key champions in Puget Sound, Atlanta, and Tampa to lead the multi-region process and coordinate the involved parties.

Since participants were geographically dispersed, modern technology played a large role in communication and coordination. The onebusaway-developers Google Group [32] served as the primary group email list. The OneBusAway Board of Directors also had a scheduled monthly phone call for progress updates.

Considering that the OneBusAway multi-region project involved a substantial software engineering effort, the most important enabler of the project was the open-source ecosystem surrounding OneBusAway. Recently, open-source projects such as OpenTripPlanner [33], a multimodal web-based trip planning solution, and OneBusAway have emerged as open-source alternative to proprietary vendor-based solutions. Open-source transit projects provide the opportunity for agencies to invest in a common set of tools for a common set of needs; in this case, trip planning and real-time customer information systems.

OneBusAway has flourished as an open-source system. Key tools enabling software development collaboration surrounding OneBusAway are the Git version control system [34] and Github.com, an online software project hosting site that uses Git for version control. The OneBusAway Github organizational account currently features 39 individual projects, or "code repositories", and 15 official members under this account. Among the open-source projects are the main OneBusAway server software, the Android app, the iPhone app, the Windows Phone app, the Windows 8 app, as well as various tools to produce and transform transit data.

An important benefit of Github is the ability of any Github user to easily "fork," or create a copy, of any OneBusAway project. These users can then edit and modify the project to meet their own needs. Major copies of the main OneBusAway server project include the modifications specific to OneBusAway Tampa, OneBusAway Atlanta, MTA Bus Time, and Detroit's TextMyBus.

However, perhaps the most important feature of Github is the ability to merge improvements back into the main project from any copies via "pull requests". In other words, a developer can create a copy of the project with little coordination with the original developer, learn about the project on their own timeline, implement a new feature or bug fix, and then submit this improvement back to the original project owner for review and possible inclusion in the main application. The Git version control system makes merging these contributions fairly straightforward.

The OneBusAway multi-region project benefitted heavily from contributions by developers who were not the original authors of the respective OneBusAway apps, indicating that this project would not have been successful in a traditional closed-source software environment where the only contributors are the official project owners. For example, the Android multi-region feature was started by the original author in Seattle, but then completed by a contributor from Tampa who was interested in accelerating the availability of the app in Tampa. Both the Windows 8 and Windows Phone multi-region updates were completed entirely by the author of the Windows 8 app. The iPhone app had the most contributors, with a total of five (four in Puget Sound and one in Atlanta) to bring the multi-region feature to full working order. Numerous developers and tech-savvy users from Puget Sound, Tampa, and Atlanta also helped in testing early versions of the applications.

5. RESULTS

In mid-August 2013, the multi-region apps were deployed in both Atlanta and Tampa, with no perceptible difference to users in Puget Sound. As a result, transit riders in Tampa and Atlanta now have access to real-time transit information via Android, iPhone, Windows Phone, and Windows 8 apps. To the knowledge of the authors, the simultaneous launch of real-time transit apps on four native app platforms in more than one city is unprecedented in the transit industry.

The most significant long-term result of the OneBusAway multi-region project is the ease of future expansion of the OneBusAway apps to new cities - adding a new city is as simple as that city setting up a new OneBusAway server, and adding that server information to the OneBusAway Server Directory. Other long-term benefits include an increased incentive for developers in the new cities (e.g., Tampa, Atlanta) to contribute to the OneBusAway project, as new features will now be visible in their own cities. This results in a larger OneBusAway developer community that will continue to grow as new cities are added. A larger development community also reduces the burden on a single entity (e.g., UW) to support the OneBusAway

project and instead spreads out demands for paid staff and volunteers amongst multiple agencies and universities.

In conjunction with the multi-region app launch, the home page for the OneBusAway project at http://onebusaway.org was converted from being specific to Puget Sound to encompassing all cities involved in the project (FIGURE 5).

FIGURE 5 The new OneBusAway multi-region website shows cities that are independently running OneBusAway server software, but leveraging the same OneBusAway mobile apps

This allows riders to conveniently access regional OneBusAway services. Information for transit agencies interested in their own OneBusAway deployments, developers who want to contribute to the project, and researchers interested in academic publications related to OneBusAway are also included. A straightforward naming scheme for region URLs (e.g., http://tampa.onebusaway.org, http://pugetsound.onebusaway.org) makes it easy to add new regions, while at the same time maintaining the identity of the project as a whole.

6. LESSONS LEARNED

As is the case with many pioneering efforts, the OneBusAway multi-region project yielded many lessons learned.

As discussed earlier, the open-source ecosystem of OneBusAway made this project possible. Without contributions from various developers outside of the initial app creators, it is

very likely that the effort would not have succeeded. Additionally, open-source software development tools (e.g., Github, Git) and collaboration tools (e.g., Google Groups) greatly facilitated collaboration.

Third-party developers can be extremely productive and responsive when they have time and are interested in a project. Various developers worked on the different mobile apps, many not having previously contributed to OneBusAway. However, third-party developers can also be unpredictable. During this effort several iPhone developers started and stopped work on the app. This is understandable, as often these developers are volunteers. However, managing this unpredictability can be difficult if a project is on a deadline, and in certain situations it may be necessary to use paid developers to finish time-critical work.

It is very useful to have project-wide funding that can pay for services that benefit all regions, including paid software engineers who can coordinate the work of many volunteers as needed, as well as hardware and license resources (e.g., website servers, domain name registration). The project is seeking federal support for OneBusAway as a research project, which can also pay for some infrastructure. However, this may not be sustainable, since research organizations (e.g., National Science Foundation) understandably want to fund new research, not operational support. In the future, an agency membership/subscription model surrounding an official non-profit organization may be necessary.

Open-source projects should have multiple administrators to prevent a single developer's lack of time to update or administer the source code from holding up the status of the entire project. During the multi-region effort, the Android, Windows Phone, and Windows 8 project were all transferred to the primary OneBusAway Github organizational account to enable additional project administrators. This relieves some of the administrative burden from the primary app developer and facilitates contributions from other developers. But, ultimately, the developer holding the account in the Google Play, Apple App Store, Windows Phone, and Windows Stores must be the one to publish new app updates. This can potentially be a bottleneck for development, depending on the smartphone platform. For example, until recently Apple prohibited transferring apps from an individual to an enterprise account, restricting the group's ability to build and sign applications for testing. Additionally, Apple has more complex requirements for distributing beta versions for testing. In contrast, Android, Windows Phone, and Windows 8 users could directly install beta versions on their device for testing.

An important consideration for testing is to ensure that the apps are tested on a range of mobile devices (i.e. different models of Android, iPhone, Windows Phone). Accessibility testing is also important, in particular for the OneBusAway iPhone app, which is the platform of choice for many blind riders who use it with "VoiceOver" mode.

7. CONCLUSIONS AND FUTURE WORK

The OneBusAway multi-region project [22] has succeeded in rapidly expanding mobile apps for Android, iPhone, Windows Phone, and Windows 8 to Tampa and Atlanta, beyond the original OneBusAway deployment in Puget Sound.

In the future, given that OneBusAway already supports a number of different transit data formats, one could envision OneBusAway components providing transit data translation services. Now that any OneBusAway server can provide instant access to mobile apps on four platforms, there is an even greater incentive for others to continue to add OneBusAway support for other data formats. For example, MTA chose to implement a REST API based on a mobile-friendly SIRI format instead of the original OneBusAway REST API. Future work could enable

the OneBusAway apps for New York City, either by adding SIRI support to the apps (thus requiring mobile app modification), or re-enabling support for the OneBusAway API in MTA's deployment.

As OneBusAway deployments are transferred from universities to transit agencies, it has become evident that procurement best practices should be established. Current recommendations include that agencies, when writing procurement contracts for OneBusAway installations, software extensions, or maintenance agreements, require that any customizations and extensions be open source, and written in a way that they can be contributed back to the project as a whole and benefit all regions, not just the requestor. Ideally any procurement requests will also include some funds to support shared resources, such as project-wide software engineers. The role of vendors in the open-source ecosystem should also be examined to ensure sufficient incentives for vendor support of OneBusAway deployments. Additional recommendations should also be established as new lessons are learned.

8. ACKNOWLEDGMENTS

The authors owe a tremendous amount of gratitude to the many people who have contributed to OneBusAway over the years. Brian Ferris created the original OneBusAway server software as well as the iPhone app, maintained the Puget Sound installation during its formative years, and continues to contribute code and expertise. Paul Watts (Android), Rob Smith (Windows Phone), and Michael Braude (Windows 8) created apps for their respective platforms that extended the reach of OneBusAway to many more users, and still contribute as well. Additional contributors to the multi-region features include Aaron Brethorst, Chaya Hiruncharoenvate, Caitlin Bonnar, Sebastian Kießling, and Ben Bodenmiller, as well as the first author. S. Morris Rose has had primary responsibility for maintaining OneBusAway in Puget Sound from summer 2011- 2013, and continues to contribute significantly. Landon Reed, Candace Brakewood, Aaron Gooze and Derek Edwards were instrumental in establishing OneBusAway Atlanta. Khoa Tran developed the GTFS-realtime feed in Tampa, FL. The authors would also like to acknowledge their funding sources at the National Center for Transit Research at the Center for Urban Transportation Research at USF; National Center for Transportation Productivity and Management, GVU Center and IPAT at Georgia Tech, and the National Science Foundation.

Finally, we thank our supporters at the transit agencies that have provided data and resources, including Sound Transit, King County Metro, Pierce Transit, Hillsborough Area Regional Transit, and Metropolitan Atlanta Rapid Transit Authority.

9. REFERENCES

[1] 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 pp. 839-848.

[2] C. Cluett, S. Bregman, and J. Richman (2003). "Customer Preferences for Transit ATIS," Federal Transit Administration. Available at http://ntl.bts.gov/lib/jpodocs/repts_te/13935/13935.pdf#sthash.jwn5Oltr.dpuf

[3] 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.

[4] Aaron Steinfeld and John Zimmerman, "Interviews with transit riders in San Francisco and Seattle," ed, 2010.
[5] John Zimmerman, Anthony Tomasic, Charles Garrod, Daisy Yoo, Chaya Hiruncharoenvate, Rafae Aziz, Nikhil Ravi Thiruvengadam, Yun Huang, and Aaron Steinfeld (2011), "Field trial of Tiramisu: crowd-sourcing bus arrival times to spur co-design," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Vancouver, BC, Canada, pp. 1677-1686,
[6] Brian Ferris, Kari Watkins, and Alan Borning (2010), "OneBusAway: results from providing real-time arrival information for public transit," in *Proceedings of the 28th International CHI Conference on Human Factors in Computing Systems*, Atlanta, Georgia, USA, pp. 1807-1816,
[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] Larry Lauvray "iPad App Development Cost - A Breakdown." *Propelics Blog*, Vol. Accessed at: http://www.propelics.com/ipad-app-development-cost-a-breakdown/
[9] James Wong, Landon Reed, Kari Watkins, and Regan Hammond, "Open Transit Data: State of the Practice and Experiences from Participating Agencies in the United States," presented at the Transportation Research Board 92nd Annual Meeting, Washington, D.C., USA, 2013.
[10] S. Barbeau (2013), "Open Transit Data – A Developer's Perspective," in *APTA TransITech 2013*, Phoenix, Arizona, March 20th, 2013. Accessed at: http://www.slideshare.net/sjbarbeau/apta-transitech-2013-open-transit-data-a-developers-perspective
[11] San Francisco Bay Area Rapid Transit District. "Third Party Apps." Accessed August 1, 2012 from http://www.bart.gov/schedules/appcenter/
[12] TriMet. "TriMet App Center." Accessed August 1, 2012 from http://trimet.org/apps/
[13] Metropolitan Transportation Authority. "App Center." Accessed November 19, 2012 from http://www.mta.info/apps/
[14] Massashusetts Bay Transportation Authority. "App Center." Accessed November 19, 2012 from http://www.mbta.com/rider_tools/apps/
[15] Hillsborough Area Regional Transit. "App Center." Accessed November 19, 2012 from http://www.gohart.org/developers/appcenter.html
[16] University of Washington. "OneBusAway - Where is your bus?". Accessed August 1, 2012 from http://onebusaway.org/
[17] OneBusAway Organization. "OneBusAway Github Source Code Repository." Accessed July 16, 2013 from https://github.com/OneBusAway/onebusaway-application-modules/wiki
[18] Google, Inc. "General Transit Feed Specification Reference." Accessed February 24, 2012 from http://developers.google.com/transit/gtfs/reference
[19] Google, Inc. "What is GTFS-realtime?". Accessed August 1, 2012 from https://developers.google.com/transit/gtfs-realtime/
[20] European Committee for Standardization (CEN). "CEN/TS 15531 (prCEN/TS-OO278181) SIRI Home Page." Accessed August 1, 2012 from http://www.kizoom.com/standards/siri/

[21] OneBusAway. "OneBusAway Deployments." Accessed August 1, 2012 from https://github.com/OneBusAway/onebusaway/wiki/OneBusAway-Deployments

[22] OneBusAway. "OneBusAway Multi-region." Accessed July 30, 2013 from https://github.com/OneBusAway/onebusaway/wiki/Multi-Region

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

[24] K. Watkins (2011), "Using Technology to Revolutionize Public Transportation," Doctor of Philosophy Thesis, Civil and Environmental Engineering, University of Washington.

[25] Front Seat Management, LLC. "City-Go-Round." Accessed March 1, 2012 from http://www.citygoround.org/agencies/

[26] Matthew Roth. (2012). "How Google and Portland's TriMet Set the Standard for Open Transit Data." *SF.STREETSBLOG.ORG*. January 5, 2010. Accessed: from http://sf.streetsblog.org/2010/01/05/how-google-and-portlands-trimet-set-the-standard-for-open-transit-data/

[27] Sean J. Barbeau and Aaron Antrim (2013), "The Many Uses of GTFS Data – Opening the Door to Transit and Multimodal Applications," in *ITS America 2013*, Nashville, Tennessee, April 22, 2013. Accessed at: http://prezi.com/-69luw8sfabp/the-many-uses-of-gtfs-data-its-america-april-2013/

[28] OneBusAway. "Real-Time Configuration Guide." Accessed August 1, 2012 from https://github.com/OneBusAway/onebusaway-application-modules/wiki/Real-Time-Data-Configuration-Guide

[29] Metropolitan Transportation Authority. "MTA Bus Time(R)." Accessed August 1, 2012 from http://bustime.mta.info/

[30] Code for America. "Text My Bus: Transit Info for Detroiters." Accessed July 26, 2013 from http://codeforamerica.org/2012/09/11/text-my-bus-transit-info-for-detroiters/

[31] University of South Florida. "HART-GTFS-realtimeGenerator Github wiki." Accessed July 26, 2013 from https://github.com/CUTR-at-USF/HART-GTFS-realtimeGenerator/wiki

[32] OneBusAway. "OneBusAway Developers Group." Accessed July 26, 2013 from https://groups.google.com/forum/#!forum/onebusaway-developers

[33] OpenPlans. "OpenTripPlanner." Accessed August 1, 2012 from http://opentripplanner.com/

[34] Software Freedom Conservancy. "Git." Accessed July 26, 2013 from http://git-scm.com/